AI VÀTÍCH HỢP AI VÀO MẠNG TRUYỀN THÔNG KHÔNG DÂY DI ĐỘNG 5G, 6G

BẢN GỐC

LỜI NÓI ĐẦU

AI đang thâm nhập vào mọi ngóc ngách của đời sống xã hội và mọi ngành nghề. Trong tương lai mọi thiết bị cũng như các hệ thống công nghệ sẽ trở thành AI bẩm sinh (AI Native) hay còn gọi là AI gốc. Điều này có nghĩa là giống như con người, ngay từ sinh ra đã có bộ não cùng với trí tuệ trong nó, các thiết bị và các hệ thống công nghệ sau khi xuất xưởng đều có sẵn AI bên trong. Xu thế này cũng sẽ xảy ra đối với các thiết bị và các hệ thống di động không dây. Vì là một bộ phận được nhúng trong các thiết bị và các hệ thống, nên các kỹ sư công nghệ để hiểu được hoạt động của công nghệ phải có các kiến thức cơ sở về AI.

Các hệ thống mạng không dây tương lai sẽ phát triển vào vào các hạ tầng không dây được tích hợp sâu "truyền thông", "cảm biến", "trí tuệ" và "lưu trữ". Các nền tảng này sẽ có khả năng cung cấp các dịch vụ tùy biến và cá nhân hóa theo yêu cầu, vượt trội các khả năng của các hệ thống truyền thông thông thường dựa trên các kiến trúc cố định và các quy tắc được định trước. Các hệ thống này cần sử dụng các công nghệ AI. Chúng sẽ sử dụng các cơ sở dữ liệu và hiểu biết lớn cho suy luận và đưa ra quyết định bằng cách khai thác các khả năng tổng quát hóa của các mô hình để thích ứng các môi trường và các kịch bản khác nhau và cung cấp ấn định tài nguyên tối ưu và các giải pháp quản lý. Hai động lực chính cho tích hợp truyền thông không dây và AI như sau.

1) **AI đẩy xa giới hạn của các hệ thống truyền thông không dây**. AI có các khả năng phi thường trong xử lý dữ liệu lớn. Dự báo đến năm 2030 dữ liệu được tạo ra hàng tháng toàn cầu đối với truyền thông không dây di động sẽ đạt đến 5016 EB (EB=10^{18} Bytes)). Dữ liệu này bao gồm dữ liệu thiết bị đầu cuối, giao diện vô tuyến, mạng và dịch vụ theo các định dạng khác nhau như văn bản, ngôn ngữ đánh dấu mở rộng (Extencible Marup Language), ngôn ngữ đánh dấu siêu văn bản (Hypertex Markup Language), đồ họa, thông tin Audio/Video. Các nhà mạng có thể sử dụng AI để đào tạo và đưa ra quyết định dựa trên suy luận trên các kiểu số liệu lớn khác nhau để nâng cao hiệu suất và tối ưu hóa mạng theo các mục tiêu và các kích thước khác nhau. AI thiếu các khả năng dự đoán trong các giải thuật thông thường. AI có thể học từ dữ liệu quá khứ thông qua các giải thuật ML và xây dựng các mô hình để dự báo các yêu cầu mạng và các vấn đề có thể xảy ra như nghẽn mạng, sự cố thiết bị hay sự thay đổi hành vi người dùng. Chẳng hạn AI có thể dự đoán trước các đỉnh lưu lượng tiềm năng trong các ngày nghỉ đặc thù hay các sự kiện diện rộng và đưa ra trước các biện pháp ấn định tài nguyên như tăng dung lượng của BS tạm thời hay tối ưu hóa các chiến lược, nhờ vậy ngăn chặn sự cố hệ thống và tăng cường trải nghiệm người dùng.
   Khả năng thích ứng là một đặc trưng nổi bật của AI. Các môi trường truyền thông không dây có tính biến động cao và chịu ảnh hưởng cao của các nhân tố khác nhau, như thời

tiết, địa hình và các thay đổi tại vị trí các thiết bị người dùng. Khả năng thích ứng của AI cho phép nó thực hiện điều chỉnh nhanh chóng theo các biến động môi trường và các thăng giáng trong các điều kiện mạng. Chẳng hạn khi, khi người dùng di chuyển từ trong nhà hay trong một xe ô tô tốc độ cao, cường độ tín hiệu và nhiễu sẽ thay đổi. AI có thể cảm nhận được các thay đổi này theo thời gian thực và tự động điều chỉnh các thông số truyền thông để đảm bảo sự ổn định của chất lượng truyền thông, thích ứng các kịch bản truyền thông khác nhau và cải thiện sự tinh cậy và linh họa của các hệ thống truyền thông không dây.

Khác với quản lý phân cấp của các mức khác nhau trong các hệ thống truyền thông thông thường, về mặt lý thuyết, AI có thể học các cấu trúc ẩn và các thông số khác nhau để phù hợp các chức năng phức tạp tùy ý. Điều này cung cấp một cách thức hiệu quả hơn để cảm nhận môi trường không dây thay đổi, phức tạp và mô tả đặc điểm không gian trạng thái mạng. Mặt khác, các thiết kế các mo-đun hệ thống truyền thông khác nhau có thể có các mục đích xung đột và có thể có các ràng buộc hiệu năng giữa các mo-đun. Tồn tại các đánh đổi giữa các số đo hiệu năng, như dung lượng kênh và nhiễu, sự tin cậy truyền dẫn và tiêu thụ năng lượng hệ thống. Quá trình tối ưu hóa từng mo-đun riêng lẻ thường không thể đạt được hiệu năng tối ưu tổng thể. Trong các trường hợp này, AI có thể tạo điều kiện thiết kết tối ưu hoá kết hợp giữa các mo-đun.

2) **AI để giải quyết các thách thức trong truyền thông không dây.**Các thách thức chính đối với mạng truyền thông không dây là phải đáp ứng các yêu cầu sau:
   - Sử dụng tài nguyên phổ tốt hơn,
   - Vùng phủ không dây carbon thấp,
   - O&M (Operation and Maintenace: khai thác và bảo dưỡng) hiệu qủa và chi phí hiệu quả hơn,
   - Khả năng cá nhân hóa và tùy biến theo yêu cầu khách hàng tốt hơn,
   - Truyền dẫn tin cậy và an ninh tốt hơn.

Tích hợp AI/ML vào các công nghệ truyền thông không dây đã trở thành một trong các xu thế công nghệ chính hướng đến 2030. Sự kết hợp này bao gồm sử dụng AI/ML để cải thiện hiệu năng truyền dẫn không dây và hỗ trợ triển khai AI/ML với các mạng không dây.

Tài liệu này này nhằm cung cấp cho các sinh viên các trường đại học, các kỹ sư, chuyên gia trong lĩnh vực truyền thông nắm được các kiến thức cơ bản của AI và quá trình cũng như viễn cảnh tích hợp AI vào các mạng truyền thông không dây di động 5G, 6G.

Tài liệu bao gồm 13 chương được chia thàm hai phần. Phần một bao gồm tám chương đầu dành cho các kiến thức cơ sở của AI. Các chương này không chỉ gúp cho người đọc hiểu được kiến thức cở sở về AI mà còn cho họ làm quen với rất nhiều thuật ngữ mới của AI. Phần hai bao gồm năm chương còn lại đề cập đến quá trình và viễn cảnh tích hợp AI và các mạng truyền thông di động 5G,6G. Có thể coi tài liệu này là 2 trong 1, vì nó vừa

cung cấp các kến thức cơ bản về AI rất mới đối với các các sinh viên các trường đại học, các kỹ sư, chuyên gia trong lĩnh vực truyền thông, vừa cung cấp các vấn đề tích hợp AI vào các mạng truyền thông không dây 5G, ,6G và đây cũng là một lĩnh vực rất rộng.

MỤC LỤC

Chương 1. Tổng quan trí tuệ nhân tạo và học máy
1.1. Tổng quan trí tuệ nhân tạo (AI)
1.2. Kết nối giữa học máy (ML), học sâu (DL) và AI
1.3. Cơ sở của học máy
1.4. Tổng quan quá trình học
1.5. Các đầu vào và các tính năng
1.6. Minh họa một số trường hợp đơn giản
1.7. Tổng kết
Chương 2. Các phương pháp học máy kinh điển
2.1. Mở đầu
2.2. Một số ký hiệu được sử dụng trong chương
2.3. Các mạng láng giềng gần nhất
2.4. Hồi quy tuyến tính
2.5. Hồi quy logistic
2.6. Quá khớp, Các mô hình phạt và chính quy hóa
2.7. Máy vectơ hỗ trợ (Support Vctor Machine)
2.8. Phân loại nhị phân (Binary Classification) bằng mộ bộ dựa đoán tuyến tính
2.9. Phân loại nhiều loại
2.10. Các hàm quyết định có phân phối chuẩn (Decision Function with Normal Distribution)
2.11. Các phương pháp dựa trên cây (Tree Based Meihod)
2.12. Phân cụm (Clustering)
2.13. Giảm chiiều (giảm kích thước không gian)
2.14. Các phương pháp hạt nhân (Kernel Method)
2.15. Tổng kết

BẢN GỐC

Chương 3. Mạng tế bào thần kinh sâu (Deep Neural Metwork)
3.1. Trí tuệ nhân tạo (AI) và mạng tế bào thần kinh sâu (DNN)
3.2. Các mạng sâu nối thẳng (Deep Feedforward networks)
3.3. Các hàm chính
3.4. Tối ưu hóa các mạng neuron sâu
3.5. Tổng quá hóa và quá khớp
3.6. Các bộ tối ưu hóa
3.7. Tổng kết

Chương 4. Phân loại hình ảnh
4.1. Tổng quan
4.2. Bộ phân loại láng giềng gần nhất (Nearest Neighbor Classifier)
4.3. Bộ phân loại k-láng giềng gần nhất (Nearest k-Neighbor Classifier)
4.4. Phân loại tuyến tính
4.5. Hàm mất mát và chính quy hóa
4.6. Các công thức SVM nhiều loại
4.7. Tối ưu hóa
4.8. Tổng kết

Chương 5. Các mạng neuron tích chập (Convolutional Neuron Network)
5.1. Phép toán chập
5.2. Các thuộc tính của phép toán tích chập
5.3. Các hàm và các biến thể
5.4. Tính toán trường tiếp nhận
5.5. Các kiến trúc mạng neuron tích chập kinh điển
5.6.Các lớp gộp và các lớp khác
5.7. Các kiến trúc cơ sở của CNN
5.7. Thừ thách Imagenet Challenger
5.9. LENET-5
5.10. AlExNET và Overfeat
5.11. VGG-16 và GoogLeNet
5.12. Mạng học dư sâu ResNET
513. Quá trình phát triển của DNN và các ứng dụng
5.14. Các tài nguyên để phát triển DNN
5.15. Tổng kết

Chương 6. Các mạng neuron hồi quy

BẢN GỐC

6.1. Mở đầu
6.2. Các kiến trúc RNN phổ biến
6.3. Bộ nhớ dài ngắn hạn (LSTM) và hồi quy có cổng (GRU)
6.4. RNN hai chiều (BRNN)
6.5. RNN sâu
6.6. Bộ mã hóa- bộ giải mã (Encoder-Decoder)
6.7. Các mô hình chú ý (Attention Models) hay các bộ biến đổi (Transformers)
6.8. Các mẹo và thủ thuật
6.9. Kết luận

Chương 7. Các mạng đối nghịch tạo sinh
7.1. Tổng quan
7.2. Các mô hình tạo sinh
7.3. Mạng đối nghịch tạo sinh (GAN)
7.4. Các kiến trúc GAN
7.5. Đào tạo GAN
7.6. Các trình bày khác
7.7. So sánh các phương án GAN
7.8. Khong gian tiềm ẩn (Laten Space)
7.9. Các ứng dụng của GAN

Chương 8. Các bộ biến đổi và các bộ biến đổi thị giác (Transformers an Vision Transformers)
Chương 8.1. Chú ý (Attention)
8.2. Bô biến đổi thị giác
8.3. Các cải tiến so với bộ biến đổi thị giác
8.4. Bộ biến đổi thị giác cho các nhiệm vụ khác với phân loại
8.5. Bộ biến đổi thị giác cho các nhiệm vụ khác (Vision Transformer for Other Tasks)
8.6. Kết luận

Chương 9. Ứng dụng AI/ML cho các dịch vụ trong các hệ thống truyền thông di động không dây
9.1. Tổng quan
9.2. Lựa chọn điểm phân chia cho hoạt động AI/ML giữa thiết bị di động và mạng lõi
9.3. Các chế độ hoạt động cho AI/ML phân chia giữa đầu cuối di động và mạng
9.4. Phân phát mô hình AI/ML trong các hệ thống truền thông di động không dây
9.5. Nguyên lý học phân tán/ liên kết trên các hệ thống truyền thông di động không dây

BẢN GỐC

9.6. Các kịch bản hoạt động của AI/ML được phân chia giữa các điểm cuối trong các hệ thống truyền thông di động không dây và các yêu cầu hiệu năng
9.7. Điều khiển được phân chia cho robot
9.8. Các hoạt động tính toán phân chia chuyển giao mô hình đặc thù phiên
9.9. Phân phối và chia sẻ mô hình/dữ liệu AI/ML trong hệ thống 5G
9.10. Học phân tán/ liên kết trên hệ thống 5G (Distributed/ Federated Learning Over 5G System)

Chương 10. Tích hợp AI/ML vào mạng truyền thông di động 5G
10.1. Tổng quan kiến trúc 5G dựa trên SBA (5G SBA)
10.2. Khung AI/ML chung
10.3. AI/ML trong nghiên cứu các khía cạnh hệ thống của 3GPP
10.4. Hoạt động AI/ML lớp ứng dụng trong hệ thống 5G
10.5. AI/ML cho các phương tiện truyền thông (Media) và quản lý AI/ML trong 5G
10.6. AI/ML cho mạng truy nhập vô tuyến 5G
10.7. Các trường hợp sử dụng AI/ML cho giao diện vô tuyến của 5G NR

Chương 11. Tích hợp AI/ML vào mạng truyền thông di động 6G
11.1. Lộ trình phát triển lên 6G
11.2. Tổng qua tích hợp AI và truyền thông trong 6G
11.3. AI cho mạng (AI4NET)
11.4. Mạng cho AI (NET4AI)
11.5. Các công nghệ hỗ trợ chủ chốt
11.6. AI as a Service (AIaaS)
11.7. Mô hình lớn của mạng không dây
11.8. Tổng kết

Chương 12. RAN mở (O-RAN) và hoạt động AI/ML trong O-RAN
12.1. Tổng quan
12.2. Các nguyên lý then chốt của O-RAN
12.3. RIC gần thời gian thực và điều khiển RAN
12.4. Khung RIC phi thời gian thực (Non RT-RIC) và điều phối
12.5. Các giao diện mở của O-RAN
12.6. Hoạt động AI/ML trong O-RAN
12.7. Các trường hợp sử dụng và nghiên cứu O-RAN
12.8. An ninh trong O-RAN
12.9. Tổng kết

Chương 13. Các mạng truy nhập vô tuyến dựa trên AI/ML
13.1. Tổng quan
13.2. KIến trúc tham chiếu
13.3. Các khái niệm và các khả năng then chốt của RAN AI gốc
13.4. Các thiết kế then chốt cho RAN AI gốc giai đoạn đầu
13.5. Các kiến trúc O-RAN AI gốc (AI-Native O-RAN) giai đoạn đầu
13.6. Các kiến trúc O-RAN AI gốc (AI-Native O-RAN) cho 6G
13.7. Tổng kết

Thuật ngữ và ký hiệu
Tham khảo

BẢN GỐC

PHẦN 1

MỤC LỤC

Chương 1. Tổng quan trí tuệ nhân tạo và học máy
1.8. Tổng quan trí tuệ nhân tạo (AI)
1.9. Kết nối giữa học máy (ML), học sâu (DL) và AI
1.10. Cơ sở của học máy

1.11. Tổng quan quá trình học
1.12. Các đầu vào và các tính năng
1.13. Minh họa một số trường hợp đơn giản
1.14. Tổng kết

Chương 2. Các phương pháp học máy kinh điển

2.1. Mở đầu
2.2. Một số ký hiệu được sử dụng trong chương
2.3. Các mạng láng giềng gần nhất
2.4. Hồi quy tuyến tính
2.5. Hồi quy logistic
2.6. Quá khớp, Các mô hình phạt và chính quy hóa
2.7. Máy vectơ hỗ trợ (Support Vctor Machine)
2.8. Phân loại nhị phân (Binary Classification) bằng mộ bộ dựa đoán tuyến tính
2.9. Phân loại nhiều loại
2.10. Các hàm quyết định có phân phối chuẩn (Decision Function with Normal Distribution)
2.11. Các phương pháp dựa trên cây (Tree Based Meihod)
2.12. Phân cụm (Clustering)
2.13. Giảm chiiều (giảm kích thước không gian)
2.14. Các phương pháp hạt nhân (Kernel Method)
2.15. Tổng kết

Chương 3. Mạng tế bào thần kinh sâu (Deep Neural Metwork)
3.1. Trí tuệ nhân tạo (AI) và mạng tế bào thần kinh sâu (DNN)
3.2. Các mạng sâu nối thẳng (Deep Feedforward networks)
3.3. Các hàm chính
3.4. Tối ưu hóa các mạng neuron sâu
3.5. Tổng quá hóa và quá khớp
3.6. Các bộ tối ưu hóa
3.7. Tổng kết

Chương 4. Phân loại hình ảnh
4.1. Tổng quan
4.2. Bộ phân loại láng giềng gần nhất (Nearest Neighbor Classifier)
4.3. Bộ phân loại k-láng giềng gần nhất (Nearest k-Neighbor Classifier)
4.4. Phân loại tuyến tính
4.5. Hàm mất mát và chính quy hóa

BẢN GỐC

4.6. Các công thức SVM nhiều loại
4.7. Tối ưu hóa
4.8. Tổng kết
Chương 5. Các mạng neuron tích chập (Convolutional Neuron Network)
5.1. Phép toán chập
5.2. Các thuộc tính của phép toán tích chập
5.3. Các hàm và các biến thể
5.4. Tính toán trường tiếp nhận
5.5. Các kiến trúc mạng neuron tích chập kinh điển
5.6.Các lớp gộp và các lớp khác
5.7. Các kiến trúc cơ sở của CNN
5.7. Thừ thách Imagenet Challenger
5.9. LENET-5
5.10. AlExNET và Overfeat
5.11. VGG-16 và GoogLeNet
5.12. Mạng học dư sâu ResNET
513. Quá trình phát triển của DNN và các ứng dụng
5.14. Các tài nguyên để phát triển DNN
5.15. Tổng kết

Chương 6. Các mạng neuron hồi quy
6.1. Mở đầu
6.2. Các kiến trúc RNN phổ biến
6.3. Bộ nhớ dài ngắn hạn (LSTM) và hồi quy có cổng (GRU)
6.4. RNN hai chiều (BRNN)
6.5. RNN sâu
6.6. Bộ mã hóa- bộ giải mã (Encoder-Decoder)
6.7. Các mô hình chú ý (Attention Models) hay các bộ biến đổi (Transformers)
6.8. Các mẹo và thủ thuật
6.9. Kết luận

Chương 7. Các mạng đối nghịch tạo sinh
7.1. Tổng quan
7.2. Các mô hình tạo sinh
7.3. Mạng đối nghịch tạo sinh (GAN)
7.4. Các kiến trúc GAN
7.5. Đào tạo GAN

BẢN GỐC

7.6. Các trình bày khác
7.7. So sánh các phương án GAN
7.8. Khong gian tiềm ẩn (Laten Space)
7.9. Các ứng dụng của GAN

Chưng 8. Các bộ biến đổi và các bộ biến đổi thị giác (Transformers an Vision Transformers)
Chương 8.1. Chú ý (Attention)
8.2. Bô biến đổi thị giác
8.3. Các cải tiến so với bộ biến đổi thị giác
8.4. Bộ biến đổi thị giác cho các nhiệm vụ khác với phân loại
8.5. Bộ biến đổi thị giác cho các nhiệm vụ khác (Vision Transformer for Other Tasks)
8.6. Kết luận

Chương 1

TỔNG QUAN TRÍ TUỆ NHÂN TẠO VÀ HỌC MÁY

1.1 TỔNG QUAN TRÍ TUỆ NHÂN TẠO (AI)

AI (Artificial Intellegence) là một lĩnh vực khoa học sử dụng các máy lập trình để bắt chiếc hành vi của con người nhằm giải quyết các nhiệm vụ mà con người thành thạo (chẳng hạn ngôn nghữ tự nhiên, giọng nói, nhận dạng hình ảnh …). AI liên qua đến giao điểm của nhiều lĩnh vực như khoa học máy tính và toán ứng dụng. Vị trí của trí tuệ nhân tạo là (đúng hơn là) xem xét điều mà chúng ta, như là con người, có một hiểu biết trực giác về trí tuệ là gì và vì thế ta có thể phán xét một máy có trí tuệ hay không. Định nghĩa hoạt động này của AI đã được Alan Turing đề xướng vào năng 1950 khi ông đưa ra "phép thử Turing" (Turing Test là một bài kiểm tra khả năng trí tuệ của máy tính) nổi tiếng. Phép thử Turing là một phép thử hoạt động, theo nó một máy được coi là có trí tuệ nếu nó có thể nói chuyện theo cách mà các người chất vấn (con người) không thể phân biệt nó với con người.

Các nỗ lực ban đầu với AI liên quan đến lập mô hình các neuron trong não (Neuron: tế bào thần kinh). Vào năm 1943, Mclulloch và Pitts đã lập mô hình cho neuron nhân tạo

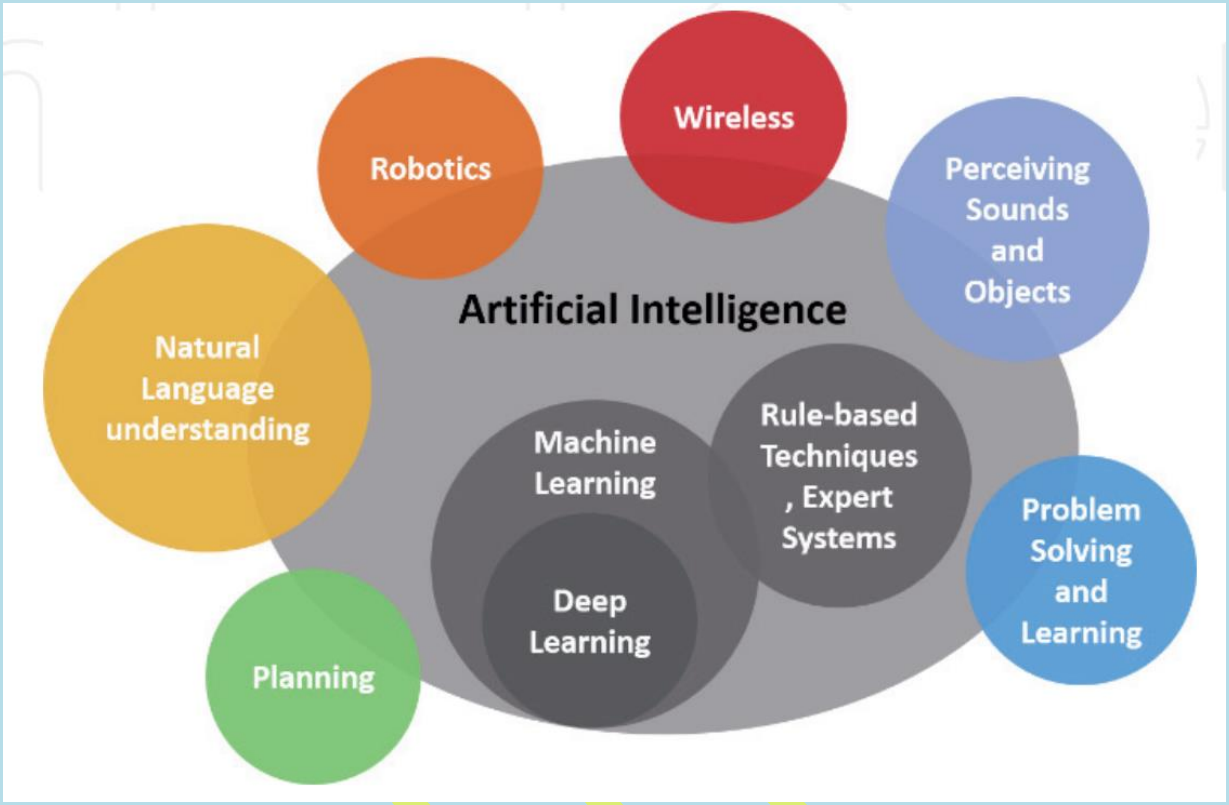
như là một biến có thể được bật hoặc tắt. Muộn hơn vào năm 1949, Donal Hebb đã phát triển một giải thuật cho các mạng thần kinh học. Vào năm 1951, Marvin Minsky và Dean Edmonths xây dựng máy tính gia cố tương tự thần kinh ngẫu nhiên (SNARC: Stochastic Neural Analog Reinforcement Calculator), máy tính thần kinh đầu tiên. Sau đó, một nhóm nhỏ các nhà khoa học quan tâm đến nghiên cứu trí tuệ đã gập nhau trong một hội thảo hai tháng tại đại học Dartmouth vào năm 1956. Theo niềm tin phổ biến, thuật ngữ AI lần đầu tiên được đưa ra và được định nghĩa bởi John Mcarthy trong cuộc hội thảo này, là: "AI liên quan đến các máy có thể thực hiện các nhiệm vụ mà các nhiệm vụ này là đặc trưng của trí tuệ con người".

Trong các thập kỷ gần đây, AI nhận được quan tâm ngày càng tăng giữa các nhà khoa học và trong công nghiệp. Điều này là do nhiều dạng ứng dụng khác nhau trong đó AI đã được sử dụng, chẳng hạn xử lý ngôn ngữ tự nhiên (ví du: phát quảng bản ghi chép tín tức (News Transcript), dịch giọng nói sang giọng nói (Speech to Speech Translation), chăm y tế (Health Care) (ví dụ: hỗ trợ các ca mổ, chẩn đoán được máy tính hỗ trợ),các ô tô và các drone thông minh (ví dụ: các ô tô tự lái, phát hiện vật cản) và cả các mạng di động (ví dụ: tối ưu hóa hiệu năng, dự báo lưu lượng).

1.2. KẾT NỐI GIỮA HỌC MÁY (ML), HỌC SÂU (DL) VÀ AI

Ngày nay AI là sự kết nối của các công nghệ khác nhau cùng làm việc để cho phép các máy cảm nhận, hiểu biết, hành động và học với các cấp độ trí tuệ như con người. Các kỹ thuật dựa trên nguyên tắc cũng như hệ thống chuyên gia là cách tiếp cận đầu tiên đến AI. Các công nghệ như ML (Machine Learning: học máy), DL (Deep Learning: học sâu) và dữ liệu lớn (Big Data) là tất cả các phần của bức tranh AI như minh họa trên hình 1.1. Vì hầu hết các tiến bộ gần đây trong AI là trong lĩnh vực học máy, mọi người thường nhầm kết hợp AI với ML. Cuối cùng, lĩnh vực ML đã được tạo ra từ mong muốn thiết kế AI với khả năng học và thu nhận kiến thức.

BẢN GỐC



Artificial Intellegence: trí tuệ nhân tạo; Machine Learning: học máy, Rule-based Techniques: các kỹ thuật dựa trên quy tắc; Expert System: hệ thống chuyên gia; Planning: quy hoạch; Natural Language Understanding: hiểu ngôn ngữ tự nhiên; Robotics: Robot; Wireless: không dây; Perceiving Sounds: cảm nhận âm thanh; Problem Solving and Learning: giải quyết vấ đề và học tập.

Hình 1.1. Kết nối và chồng lấn giữa học máy, học sâu và trí tuệ nhân tạo

ML là một tập con với mục đích trao cho một máy tính khả năng thực hiện các nhiệm vụ mà không có các chỉ dẫn tường minh cho trước là làm cách nào để giải quyết nó. Đây là một mô hình hướng đến xây dựng một máy tính có thể học giống như con người. Quá trình học bao gồm cung cấp một giải thuật ML với các thí dụ về nhiệm vụ mà ta muốn giai quyết (dữ liệu) và để máy tính tìm ra các mẫu và thực hiện các suy luận để tối ưu hóa thực hiện quyết định theo mục tiêu được người dùng định nghĩa. Nói chung, ML có thể được sử dụng để thực hiện kiểu các nhiệm vụ khác nhau bao gồm phân loại (Classification), phân cụm (Clustering) và đưa ra quyết định (Making Decision) về dữ liệu.

Các mạng neuron nhân tạo (ANN: Artificial Neural Network), còn được đề cập như là các mạng neuron (NN: Neural Network), là các mô hình học máy phổ biến được truyền cảm hứng bởi các quá trình sinh học. Giải thuật NN đầu tiên là Perceptron (là một neuron

nhân tạo cố gắng xấp xỉ một neuron sinh học) được phát triển bởi Rosenblatt vào năm 1958. Sự tìm thấy này được truyền cảm hứng bởi các mô hình toán học McCulloch của các neuron trong não người. Trong các thập kỷ tiếp sau, các kiểu kiến trúc của các mạng neuron khác nhau đã được đề xuất cùng với các giải thuật để đào tạo chúng hiệu quả. Tiếp theo các thành tựu này, thuật ngữ học sâu (Deep Learning) đã được đưa ra cho cộng đồng ML. Vào khoảng năm 2000, học sâu là một thể loại con của ML tập trung lên tham số hóa các mạng neuron đa lớp (Deep: sâu), các mạng học biểu diễn dữ liệu.

Trong những năm gần đây, học sâu dựa trên các phương pháp đã dành được mối quan tâm ngày càng tăng. Điều này là do hiệu năng cao của chúng trong các nhiệm vụ phân loại ảnh, nhận biết giọng nói và xử lý ngôn ngữ tự nhiên. Thực tế là, các kỹ thuật học sâu đã vượt trội hơn nhiều các kết quả hiện thời trong các nhiệm vụ này. Tuy nhiên học sâu và tất nhiên các kỹ thuật ML cũng có một số hạn chế. Hạn chế đầu tiên là khối lượng dữ liệu cần thiết trong quá trình đào tạo để đạt được các hiệu năng giống như con người. Một hạn chế khác là các khả năng tính toán cần thiết để đào tạo các mô hình học sâu trên các tập dữ liệu lớn.

1.3. CƠ SỞ CỦA HỌC MÁY (ML)

1.3.1. Các thể loại học máy

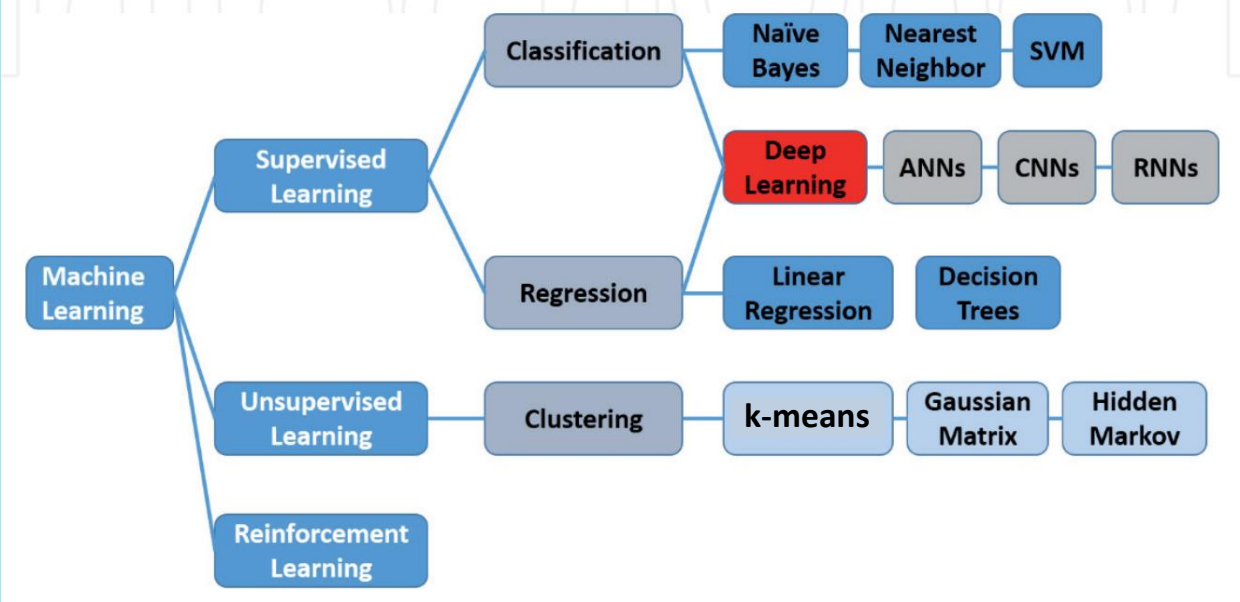
Như đã xét trong phần trước, ML là một lĩnh vực phức tạp. Tùy vào chiến lược dựa trên đào tạo, ML có thể được chia thành ba thể loại kinh điển. Các cách tiếp cận học khác nhau của các thể loại này được minh họa trên hình 1.2:

Học có giám sát (Supervised Learning): học với một tập đào tạo được gắn nhãn. Thể loại này bao gồm các nhiệm vụ phân loại và hồi quy (Regression).

Học không có giám sát (UnsSupervised Learning): Quá trình đào tạo một mô hình sử dụng dữ liệu đào tạo không gắn nhãn. Mô hình phải khám phá các mẫu trong dữ liệu không gắn nhãn. Nhiệm vụ được sử dụng rộng rãi trong học không có giám sát là phân cụm (Clustering).

Học tăng cường (Reinforcement Learning): Quá trình đào tạo một mô hình trên một chuỗi các hành động dẫn đến một kết cục phổ biến, trong đó hệ thống nhận được các phần thường cho thực hiện tốt và các trừng phạt cho thực hiện tồi trực tiếp từ môi trường của nó. Học tăng cường được sử dụng trong robot và các trò chơi.

BẢN GỐC



Machine Learning: học máy, Supervised Learning: học có giám sát, Unsupervised Learning: học không có giám sát, Reinforcement Learning: học tăng cường, Classification: ;phân loại, Regression: hồi quy, Clustering: phân cụm, Naïve Bayes: là một thuật toán đơn giản dựa trên định lý Bayes, Deep Learning: học sâu,Linear Repression: hồi quy tuyến tính, K-Means: là một giải thuật đơn giản thường được áp dụng cho các bài toán phân cụm, Nearest Neighbor: thuật toán láng giềng gần nhất, ANN (Artificial Neural Network: mạng tế bào thần kinh nhân tạo, Decision Trees: các cây quyết định, Gaussian Matrix: ma trận Gauss, SVM (Support Vector Machine): máy vectơ hỗ trợ, CNN (Colvolution Neural Network): Mạng neuron chập, RNN (Recurrent Neural Network): mạng neuron hồi quy, Hidden Markov: Markov ẩn.

Hình 1.2. Phân loại các tiếp cận ML khác nhau

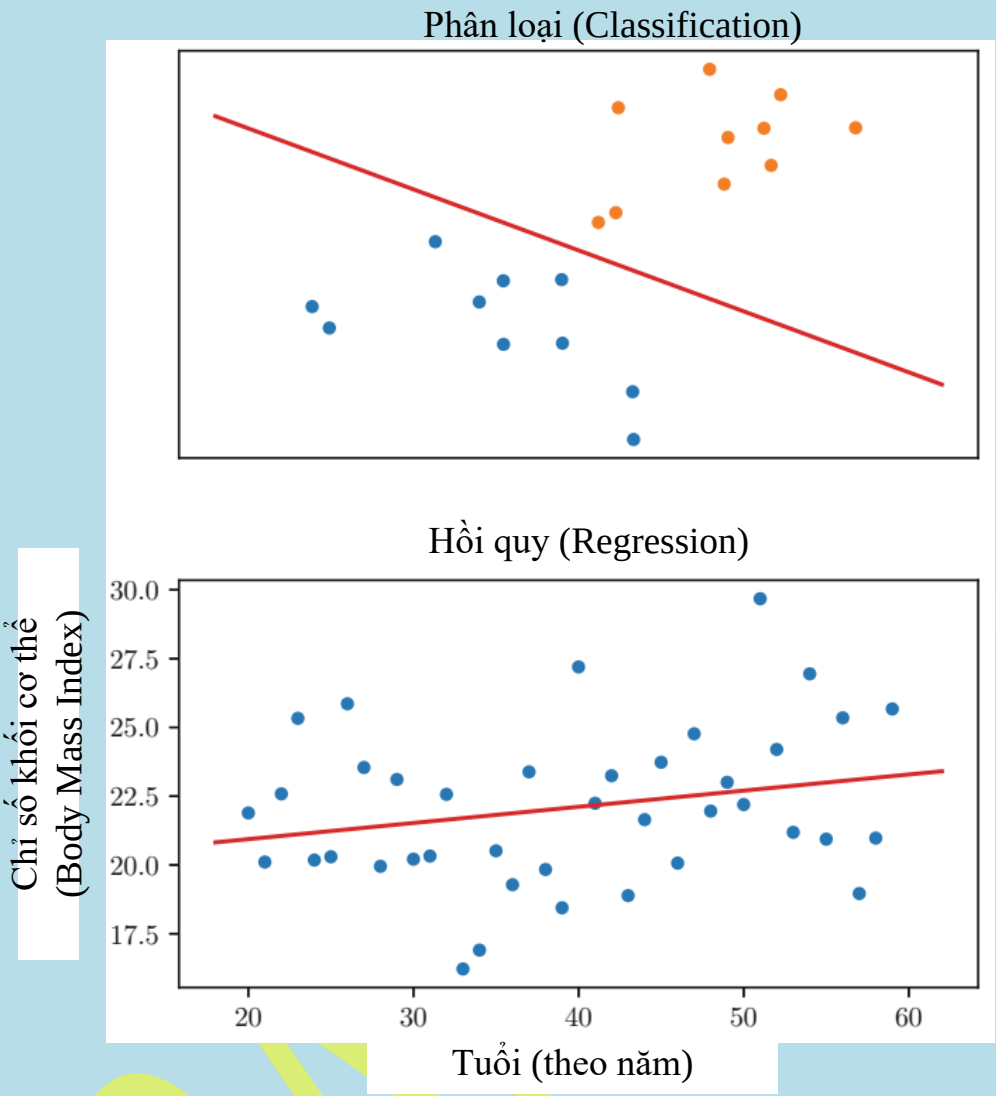
ML là một nhánh của AI, để dạy cho một hệ thống máy tính thực hiện dự báo dựa trên các dữ liệu. Nói chung, ML có thể được phân chia vào ba cách tiếp cận, bao gồm học có giám sát (SL: Supervised Learning), học không có giám sát (UL: Unsupervised Learning) và học tăng cường (RL: Reinforcement Learning) như thấy trên hình 1.2. Dữ liệu đào tạo trong học có giám sát bao gồm cả hai: các đầu vào và các đầu ra được gắn nhãn và mục đích của nó là ước tính mô hình chưa được biết và lập bản đồ (ánh xạ) các đầu vào đã biết vào các nhãn. Hai loại chính của học có giám sát là phân loại (Classification) và hồi quy (Regression). Loại thứ nhất học để dự đoán các nhãn loại (Class Label) rời rạc (ví dụ: phân loại đối tượng), trong khi đó loại thứ hai học để dự đoán số lượng liên tục (ví dụ: dự báo chi phí). Không giống như học có giám sát, trong học không có giám sát dữ liệu đào tạo không có nhãn và mục đích của học không có giám sát là học để để trình bày hiệu quả

hơn một tập các đầu vào chưa được biết. Một thí dụ nổi tiếng của học không có giám sát là phân cụm k-means (k-means clustering), có thể áp dụng cho nhiều vấn đề phân cụm trong các hệ thống IoT, như phân cụm người dùng và tối ưu hóa sắp xếp. Trong học tăng cường, tác nhân không nói ra các hành động cần thực hiện mà tương tác liên tục với môi trường học và cố gắng tìm ra chính sách /hành động tốt để tạo ra phần thưởng tích lũy tốt nhất.

Trong học có giám sát, máy học để thực hiện một nhiệm vụ bằng cách nghiên cứu một tập các ví dụ mà đầu ra của chúng đã biết (chẳng hạn: các ví dụ này đã được gắn nhãn). Hai nhiệm vụ phổ biến nhất trong học có giám sát là phân loại (Classification) và hồi quy (Regression) (hình 1.3). Phân loại có mục đích là gán một thể loại (Category) cho một thí dụ. Các thí dụ có thể là, chẳng hạn, các bệnh nhân khác nhau, và các thể loại là các chẩn đoán có thể có khác nhau. Như vậy các đầu ra là rời rạc. Thí dụ của các giải thuật phân loại phổ biến gồm cả hồi quy logistic (Logistic Regression) (mặc dù tên như vậy, nó là một phương pháp phân loại), phân tích phân biệt tuyến tính (Linear Discrimnant Analysis), máy vectơ hỗ trợ (Support Vector Machine), các bộ phân loại rừng ngẫu nhiên (Random Forest Classifier) và các mô hình học sâu (Deep Learning Model) cho phân loại. Trong hồi quy đầu ra là một số. Chẳng hạn, có thể là điểm số lâm sàng tương lai của một bệnh nhân mà ta tìm cách dự báo. Thí dụ về các phương pháp hồi quy phổ biến gồm cả hồi duy tuyến tính đơn biến (Simple Linear Regression) hay hồi quy tuyến tính đa biến (Multiple Linear Regression), hồi quy bị phạt (Penalized Regression) và hồi quy rừng ngẫu nhiên (Random Forest Regression).

Cuối cùng có nhiều nhiệm vụ khác có thể được đặt vào bộ khung như là một vấn đề học có giám sát, ví dụ: phân tích dữ liệu (Data Analysis), phân đoạn ảnh (Image Segmentation) và nhiều nhiệm vụ khác sẽ được trình bày trong các chương sau.

BẢN GỐC

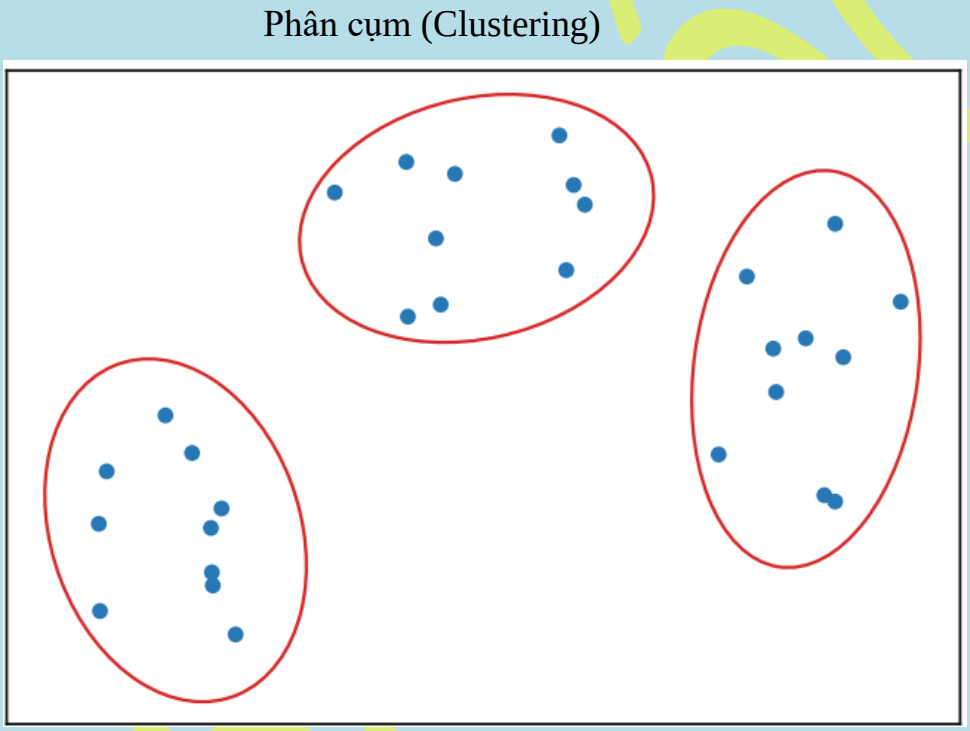


Hình 1.3. Hai trong số các nhiệm vụ chính của học có giám sát: phân loại và hồi quy. Hình vẽ trên trình bày một nhiệm vụ phân loại có mục đích phân tách tuyến tính loại mầu vàng và loại mầu xanh. Mỗi loại được trình bày bằng hai biến số. Hình dưới trình bày nhiệm vụ hồi quy tuyến tính có mục đích dự báo chỉ số khối cơ thể từ tuổi của một người.

Trong học máy không có giám sát, các thí dụ không được gắn nhãn. Hai nhiệm vụ phổ biến trong học không có giám sát là phân cụm (Clustering) và giảm kích thước (Dimensionality Reduction) (hình 1.4). Phân cụm hướng đến khám phá các nhóm trong một tập dữ liệu đào tạo, nhưng các nhóm này chưa được biết trước. Mục tiêu là tìm ra các nhóm sao cho các thành viên của cùng một nhóm giống nhau và thành viên của các nhóm khác nhau không giống nhau. Ví dụ, ta có thể cố gắng phát hiện các phân nhóm bệnh

chưa được biết trước. Chẳng hạn, một số phương pháp phân cụm kinh điển là tạo cụm k-means hay phân cụm phổ (Spectral Clustering).

Học giảm kích thước có mục đích tìm ra một không gian biến số (có kích thước thấp hơn kích thước của không gian đầu vào) và giải thích tốt hơn tính biến động của dữ liệu đầu và khi cho trước một tập các biến đầu vào lớn hơn. Điều này tạo ra một tập các biến mới, nói chung, không thuộc các biến đầu vào nhưng là các hết hợp của chúng. Các thí dụ của các phương pháp này bao gồm phân tích các thành phần chính (Pricipal Component Analisis), các bản đồ riêng Lamplace (Lamplacian Eigenmap) hay các bộ mã hóa tự động biến thiên (Variational Autoencoder).



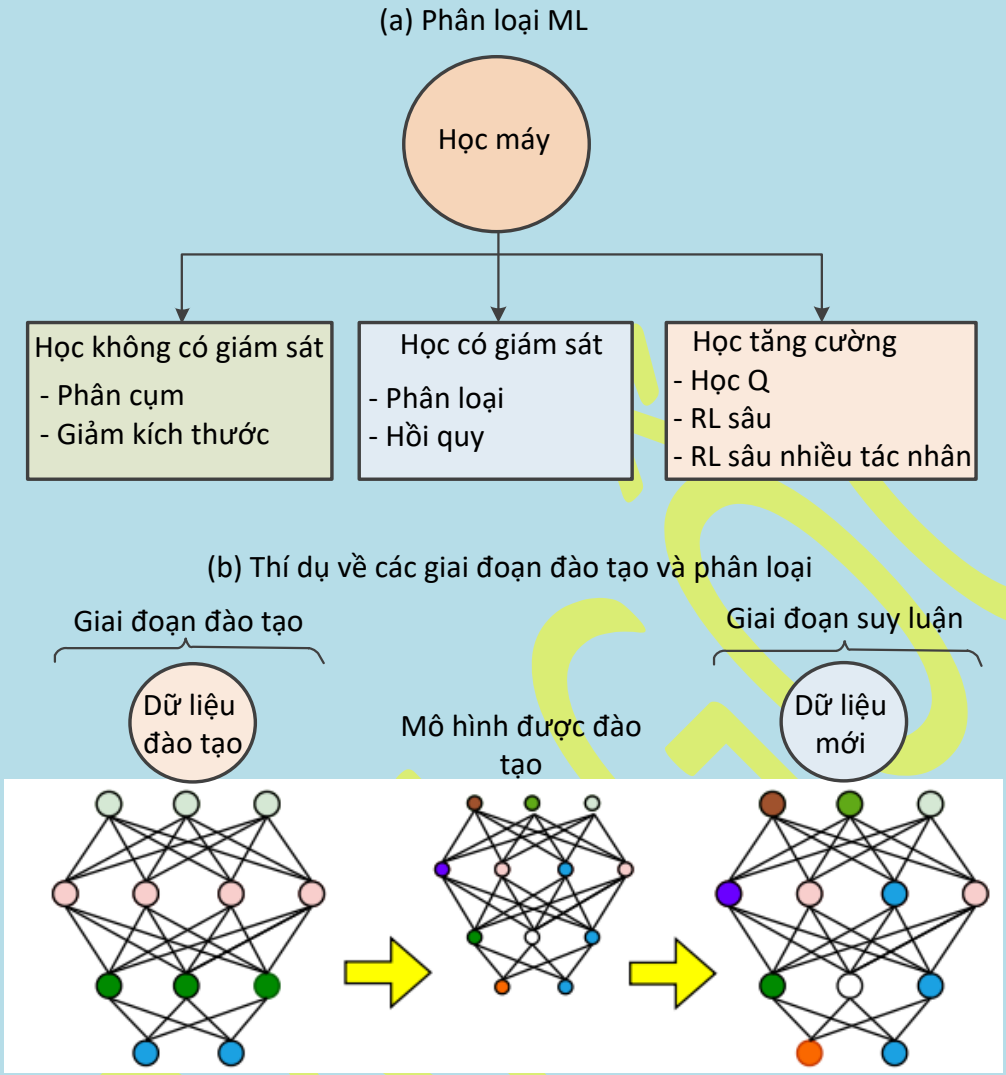
Hình 1.4. Nhiệm vụ phân cụm. Giải thuật tự động nhận dạng ba nhóm (tương ứng với các vòng mầu đỏ) từ các thí dụ không được đánh nhãn (các điểm mầu xanh). Các nhóm này không được biết trước

Trong học tăng cường, máy sẽ thực hiện một chuỗi các hành động để cực đại hóa một phần thường. Có thể là, thí dụ, trường hợp học máy để chơi cờ, nó sẽ đấu với chính mình để cực đại hóa các chiến thắng. Các phương pháp này được sử dụng rộng rãi cho học chơi game hay trong lĩnh vực robot.

1.3.2. Học máy trong các mạng tế bào thần kinh sâu (Deep Neural Network)

Hình 5 cho thấy học máy dựa trên các mạng thấy tế bào thần kinh mà ta sẽ xét trong các chương sau. Hình 1.5(a) cho thấy học máy được phân chia thành: học có giám sát (SL: Supervised Learning), học không có giám sát (UL: Unsupervised Learning) và học tăng cường (RL: Reinforcement Learning). hình 1.5(b) cho thấy học máy dựa trên các mạng tế bào thần kinh sâu (Deep Neural Netwwork) mà ta sẽ xét trong các chương sau. Trên hình 5(b) một vấn đề của ML có thể được chia thành hai giai đoạn: giai đoạn đào tạo và giai đoạn suy luận. Trong giai đoạn đào tạo, mô hình ML (dự báo hoặc phân loại) được xây dựng dựa trên dữ liệu đầu vào, các dữ liệu này có thể được đánh nhãn hoặc không đánh nhãn. Trong AI thông thường, dữ liệu được thu thập từ các người dùng cho đào tạo trung tâm trong đám mây tập trung có các tài nguyên lưu trữ và tính toán mạnh. Giai đoạn thứ hai của ML là suy luận, khi này dữ liệu trực tiếp (Live Data) được đưa vào mô hình được đào tạo để tạo ra các hành động đầu ra.

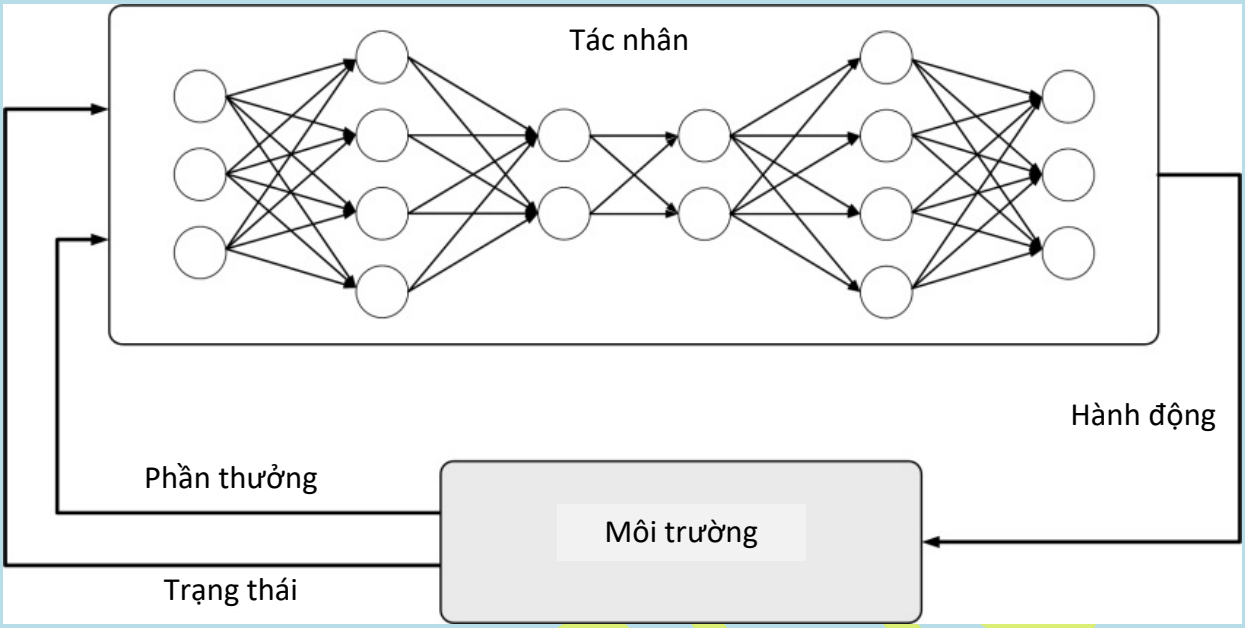
BẢN GỐC



Hình 1.5. Học máy: (a) phân loại chung của ML và (b) Xử lý ML nói chung bao gồm giai đoạn đào tạo và giai đoạn suy luận

Học tăng cường sâu (DRL: Deep reinforcement learning)

Học tăng cường sâu (DRL: Deep reinforcement learning) không phải là một mô hình DNN khác. Nó bao gồm DNN và học tăng cường. Như minh họa trong hình 1.6 mục tiêu của DRL là tạo ra một tác nhân thông minh có thể thực hiện các chính sách hiệu quả để tối đa hóa phần thưởng của các nhiệm vụ dài hạn với các hành động có thể kiểm soát được. Ứng dụng điển hình của DRL là giải quyết các vấn đề lập lịch khác nhau, chẳng hạn như các vấn đề quyết định trong trò chơi, lựa chọn tốc độ truyền video, v.v.

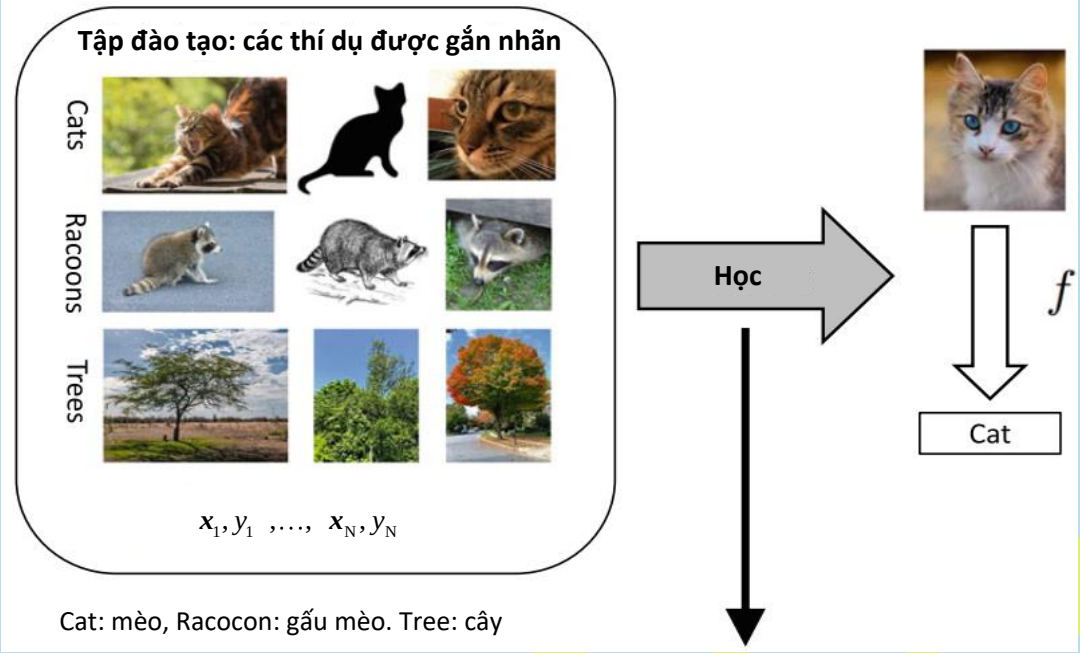


Hình 1.7. Học tăng cường

1.4. TỔNG QUAN QUÁ TRÌNH HỌC

Trong phần này ta sẽ hướng đến chính thức hóa các khái niệm chính là nền tảng của các phương pháp học máy. Một số trong số các khái niệm này, với một vài thay đổi, cũng mở rộng đến các trường hợp không có giám sát.

Nhiệm vụ mà ta sẽ xét sẽ là để cung cấp một đầu ra ký hiệu *y* từ một đầu vào cho trước của máy tính ký hiệu ***x***. Tại điểm này bản chất của ***x*** không quan trọng. Nó có thể, ví dụ, là một bức ảnh bất kỳ có thể có như trong thí dụ được trình bày trong hình 1.8. Nó cũng có thể là một số, một dãy số, một văn bản v.v. Tạm thời , bản chất của *y* cũng có thể thay đổi. Điển hình, trong trường hợp hồi quy, nó có thể là một số. Trong trường hợp phân loại nó tương ứng một nhãn (chẳng hạn, nhãn "cat' (con mèo) trong thí dụ của ta). Tạm thời, ta không cần bận tâm đến cách thức các dữ liệu này (các ảnh, các nhãn, v.v.) được trình bày trong máy tính.



$$\hat{f} = \arg\min_{f \in F} \frac{1}{N} \sum_{i=1}^{N} L_i \left(y_i, f\left(x_i \right) \right)$$

Hình 1.8. Các khái niệm chính là cơ sở cho học có giám sát, ở đây là trong trường hợp phân loại. Mục đích là có thể nhận dạng nội dung của bức ảnh (đầu vào *x*), để gán nó cho một nhãn (đầu ra *y*). Nói một cách khác, ta muốn có một hàm f để biến đổi *x* vào *y*, ta sẽ sử dụng một tập đào tạo x_1, y_1 ,…, x_N, y_N (trong trường hợp của ta là một tập các bức ảnh đã được gán nhãn)

Học máy nhắm đến tìm ra một hàm f có thể biến đổi $\boldsymbol{x}$ vào y, nghĩa là f=f($\boldsymbol{x}$). Tạm thời, f có thể là bất kỳ kiểu gì - ảnh chẳng hạn, nó là một phép toán liên kết một $\boldsymbol{x}$ cho trước với một *y* cho trước.Trong chương 3, các hàm f là các mạng tế bao thần kinh nhân tạo (để thuận tiện, trong tài liệu này ta sẽ dùng thuật ngữ mạng neuron nhân tạo). Học máy nhắm đến tìm ra một hàm f mà nó sẽ cung cấp đầu ra đúng đối với từng đầu vào cho trước. Ta gọi một hàm là hàm mất mát (Loss Function) và ký hiệu là L_i , là một hàm đo sai số gây ra bởi hàm f. Hàm f nhận hai đối số: đầu ra y_i đúng và đầu ra dự đoán f($\boldsymbol{x}_i$). Giá trị hàm mất mát càng nhỏ thì đầu ra dự báo càng gần với đầu ra đúng. Một thí dụ của hàm mất mát là mất mát bình phương tối thiểu: $L_i \left(y_i, f(\boldsymbol{x}_i) \right) = \left(y_i - f(\boldsymbol{x}_i) \right)^2$, nhưng cũng tồn tại nhiều hàm khác. Lý tưởng, hàm f phải là hàm tạo ra sai số tối thiểu đối với mọi đầu vào $\boldsymbol{x}$ có thể có và được liên kết với đầu ra *y*, không chỉ là các dữ liệu mà ta có thể sử dụng, mà

với mọi dữ liệu khác có thể có. Tất nhiên ta không thể trong tay bất cứ mọi dữ liệu. Vì thế ta sẽ sử dụng một tập dữ liệu được gọi là tập đào tạo. Trong học có giám sát, tập này được gắn nhãn, ví dụ: đối với từng thí dụ trong tập này, ta biết cả $\boldsymbol{x}$ lẫn y. Ta ký hiệu x_i, y_i ,…, $\boldsymbol{x}_N, y_N$ là N thí dụ của tập đào tạo, là N cặp của các đầu vào và các đầu ra. Bây giờ ta sẽ đi tìm hàm f tạo ra sai số cực tiểu trên N mẫu của tập đào tạo. Nói một cách khác ta đi tìm hàm cho phép giảm thiểu sai số trung bình trên tập đào tạo. Ta gọi sai số trung bình này là hàm chi phí (Cost Function):

$$L(f) = \frac{1}{N}\sum_{i=1}^{N} L_i \quad y_i, f(\boldsymbol{x}_i$$

Học máy sẽ nhắm đến tìm ra hàm $\hat{f}$ giảm thiểu hàm chi phí:

$$\hat{f} = \arg\min_{f \in F} \frac{1}{N}\sum_{i=1}^{N} L_i \quad y_i, f \quad \boldsymbol{x}_i$$

Trong phương trình trên, argmin chỉ ra rằng ta quan tâm hàm f mà nó giảm thiểu hàm chi phí L(f) và không phải giá trị của bản thân chi phí. F là không gian chứa tất cả các hàm có thể được chấp nhận. F có thể là, chẳng hạn, tập các hàm tuyến tính hay tập các mạng neuron với một kiến trúc cho trước.

Thủ tục nhằm tìm ra hàm f cho phép giảm thiểu chi phí được gọi là thủ tục tối ưu hóa. Đôi khi, có thể tìm cực tiểu bằng cách giải tích (ví dụ: bằng cách trực tiếp giải một phương trình cho f). Trong các trường hợp khác ta phải sử dụng thủ tục lặp (ví dụ: một giải thuật): hàm f được thay đổi lặp cho đến khi tìm được hàm giảm thiểu chi phí. Có các trường hợp trong đó ta có được giải thuật đảm bảo tìm được cực tiểu toàn cục (Global) và có các tường hợp khác trong đó ta chỉ tìm được cực tiểu cục bộ (Local).

Giảm thiểu sai số trên tập đào tạo không đảm bảo rằng máy tính được đào tạo sẽ thực hiện tốt trên các thí dụ mới khi chúng không phải là bộ phận của tập đào tạo. Lý do đầu tiên là tập đào tạo quá khác với dân số nói chung (tổng thể dân số) (chẳng hạn ta một mô hình trên một tập dữ liệu những người đàn ông trẻ tuổi và ta muốn áp dụng nó cho các bệnh nhân với giới tính và tuổi tác bất kỳ). Một lý do khác là ngay cả khi các đặc trưng của tập đào tạo tuân theo các đặc tính của tổng thể dân số, hàm f được học có thể quá đặc thù đối với tập đào tạo. Nói một các khác nó đã học tập đào tạo theo "trái tim" nhưng không khám phá được một quy tắc tổng quát hơn hiệu quả cho các thí dụ khác. HIện tượng này được gọi là quá khớp (Overfitting) và thường xuất hiện khi kích thước hay chiều (Dimentionality) quá cao (Có nhiều biến để trình bày một đầu vào), khi tập đào tạo

quá nhỏ hay khi hàm f quá linh hoạt. Một cách để ngăn chặn quá khớp là cải biến hàm chi phí sao cho nó không chỉ trình bày sai số trung bình trên tất cả các thí dụ mà còn ràng buộc hàm có một số thuộc tính đặc thù.

1.5. CÁC ĐẦU VÀO VÀ CÁC TÍNH NĂNG

Trong phần trước, ta đã không đặt giả thiết lên bản chất của đầu vào ***x***. Nó có thể là một ảnh, một số, một văn bản, v.v.

Dạng đơn giản nhất mà ta có thể xét là khi ***x*** là một con số đơn lẻ. Các thí dụ bao gồm độ tuổi, các điểm số lâm sàng, v,v, Tuy nhiên đối với hầu hết các vấn đề, mô tả đặc điểm của của một bệnh nhân đòi hỏi một tập các kết quả đo (bảng 1.1). Trong trường hợp này, đầu vào có thể là một dẫy các số x_1, …., x_K, mà chúng có thể được trình bày bằng một vectơ:

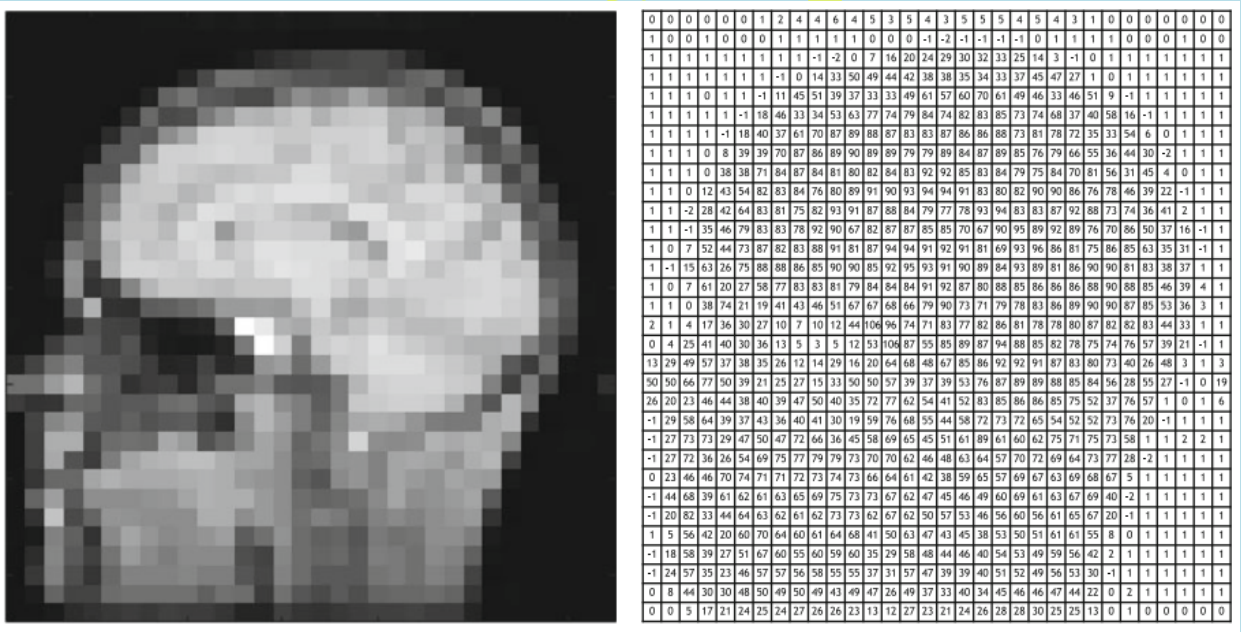
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_K \end{bmatrix}$$

Bảng 1.1. Thí dụ trong đó đầu vào là một dẫycác số. Trong trường hợp này, mỗi bệnh nhân được đặc trưng bởi một số biến số.

	Độ tuổi (năm)	**Chiều cao (cm)**	**Trọng lượng (kg)**
Bệnh nhân 1	52,5	172	58
Bệnh nhân 2	75,1	162	55
Bệnh nhân 3	32,7	160	54
Bệnh nhân 4	45	161	52

Tuy nhiên có các trường hợp trong đó đầu vào không phải là một vectơ. Đó là trường hợp, chẳng hạn, đầu vào là một ảnh một văn bản hay một chuỗi DNA. Tất nhiên trong máy tính mọi thứ đều được lưu trữ bằng số. Một ảnh là một mảng các giá trị cường độ theo thang độ xám của từng pixel (hình 1.9). Một văn bản là một chuỗi các ký tự, trong đó mỗi ký tự được mã hóa bằng một số. Tuy nhiên không giống như trong thí dụ được trình bày tại bảng 1.1 các số này bản thân không có nghĩa. Vì thế một cách tiếp cận chung là chiết xuất các tính năng (các đặc điểm), sẽ là dẫy các chữ số có nghĩa trình bày đầu vào. Thí dụ nếu đầu vào là ảnh cộng hưởng từ (MRI: Magnetic Resonance Image) của não bộ, thì các tính năng liên quan có thể là các thể tích của các vùng giải phẫu khác nhau của não bộ (quá trình đặc thù này được thực hiện bằng cách sử dụng một kỹ thuật được gọi là phân đoạn (Segmentation) sẽ được xét trong một chương khác). Điều này sẽ dẫn đến một dẫy các số tạo nên một vectơ đầu vào. Phát triển các phương pháp chiết xuất các

tính năng có nghĩa từ dữ liệu thô đóng một vai trò quan trọng trong học máy. Cách tiếp cận này thường được gọi là kỹ thuật tính năng (Feature Engineeering). Các phương pháp học sâu (Deep Learning) cho phép tránh được chiết xuất tính năng bằng cách cung cấp một cách tiếp cận đầu cuối-đầu cuối từ dữ liệu thô đến đầu ra. Trong một số lĩnh vực, điều này làm cho kỹ thuật tính năng ít quan trọng hơn, tuy nhiên vẫn còn các ứng dụng trong đó các tính năng được gọi là được xử lý thủ công vẫn cạnh tranh với các phương pháp học sâu.

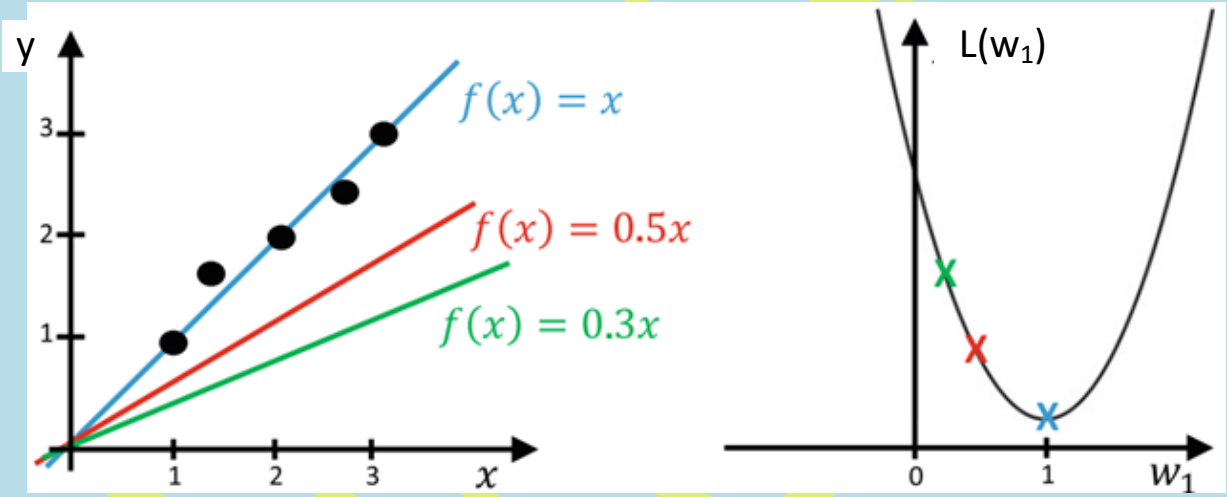


Hình 1.9. Trong máy tính, một ảnh được trình bày bằng một mảng các con số. Mỗi con số tương ứng với một mức xám của một pixel (điểm ảnh) cho trước. Trên hình này, thí dụ này là một cắt lớp của một ảnh chụp cộng hưởng từ giải phẫu (Anatomical MRI: Anatomical Magnetic Resonance Immage), đã được giảm mẫu đáng kể để có thể nhìn rõ các pixel khác nhau. Lưu ý rằng MRI giải phẫu thực chất là một ảnh 3D và phải được trình bày bằng một mảng 3D chứ không phải một mảng 2D.

1.6. MINH HỌA MỘT TRƯỜNG HỢP ĐƠN GIẢN

Bây giờ chúng ta sẽ minh họa từng bước một các khái niệm nói trên trong một trường hợp rất đơn giản: hồi quy tuyến tính một biến số. Một biến số có nghĩa là đầu vào là một số đơn như minh hoạ trên hình 1.10. Tuyến tính có nghĩa là mô hình f là một đường thẳng đơn giản. Đầu vào là một số *x* và đầu ra là một số *y*. Hàm mất mát sẽ là mất mát bình

phương cực tiểu: $L\ y, f(x) = y - f(x)^2$. Mô hình f sẽ là một hàm tuyến tính của x, nghĩa là $f(x) = w_1 x + w_0$ tương ứng với một đường thẳng, w_1 sẽ là độ dốc của đường thẳng này và w_0 là điểm cắt. Để đơn giản hơn nữa, ta sẽ xét trường hợp không có điểm cắt, nghĩa là đường thẳng đi qua gốc tọa độ. Các giá trị khác nhau của w_1 tương ứng với các đường thẳng khác nhau (tương ứng với các hàm f khác nhau) và tương ứng với hàm chi phí L(f) và trong trường hợp này ta có thể viết lại hàm chi phí là $L(w_1)$ vì f chỉ phụ thuộc vào w_1 (hình 1.10). Mô hình tốt nhất là mô hình mà đối với nó $L(w_1)$ cực tiểu.

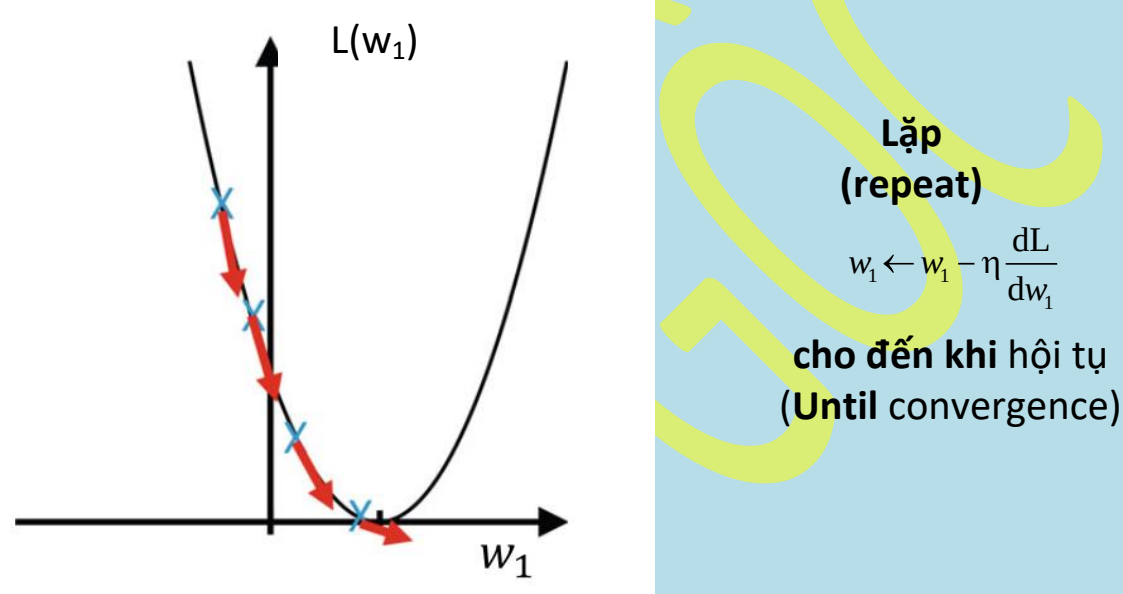


Hình 1.10. Minh họa các khái niệm của học có giám sát bằng một trường hợp rất đơn giản: hồi quy tuyến tính một biến số không có điểm cắt. Các mẫu đào tạo tương ứng với các hình tròn đen. Các mô hình khác nhau: $f(x)=w_1x$ **tương ứng với các đường thẳng khác nhau. Mỗi mô hình (tương ứng với mỗi giá trị của thông số** w_1**) tương ứng với một giá trị trị của hàm chi phí** $L(w_1)$**. Mô hình tốt nhất (đường xanh chàm) là mô hình giảm thiểu** $L(w_1)$**, tương ứng với độ dốc= 1 trong trường hợp này.**

Làm thế nào tìm được w_1 sao cho $L(w_1)$ cực tiểu?. Ta sẽ sử dụng đạo hàm của L: $\frac{dL}{dw_1}$.

Để được cực tiểu L, cần $\frac{dL}{dw_1} = 0$ (trong trường hợp của ta đảo lại cũng đúng). Trong trường hợp của ta, có thể tực tiếp giải $\frac{dL}{dw_1} = 0$. Tuy nhiên đây không phải là tường hợp tông quát. Rất thường xuyên ta không thể giải nó bằng giải tích. Vì thế, ta sẽ phải sử dụng một giải thuật lặp. Một phương pháp lặp kinh điển là giảm gradient hay giảm độ dốc (Gradient Descent). Trong trường hợp tổng quát, f không chỉ phụ thuộc vào một thông

số w_1 mà còn vào một tập các thông số (w_1, …., w_K) mà ta có thể sắp xếp chúng vào một vectơ $\mathbf{w}=[w_1, \ldots, w_K]^T$ (T ký hiệu cho đảo vị). Vì thế, thay vì làm việc với đạo hàm $\frac{dL}{d\mathbf{w}_1}$, ta sẽ là việc với gradient $\Delta_{\mathbf{w}} L = \left[\frac{dL}{dw_1} \quad \cdots \quad \frac{dL}{dw_K} \right]^T$. Gradient này là một vectơ chỉ ra phương mà theo đó ta phải đi theo để leo (Climb) dọc L. Vì thế, ta sẽ leo theo hướng ngược với gradient này, nên có tên là giảm gradient (Gradient Descent). Quá trình này được ninh họa trên hình 1.11. cùng với giải thuật tương ứng.



Hình 1.11. Phía phải: minh họa khái niệm giảm độ dốc (Gradient Descent) trong một thí dụ đơn giản trong đó mô hình f được định nghĩa bằng cách sử dụng chỉ một thông số w_1. Giá trị của w_1 được cập nhật lặp bằng cách đi theo phương ngược của gradient. Phía phải: giải thuật giảm gradient trong đó η là tốc độ học (Learning Rate), nghĩa là tốc độ mà tại đó w_1 sẽ được cập nhật.

1.7. TỔNG KẾT

Các khái niệm chính đã của quá trình đào tạo trong học máy:

- Đầu vào $\boldsymbol{x}$
- Đầu ra y
- Các mẫu đào tạo $\boldsymbol{x}_1, y_1 \ , \ldots, \ \boldsymbol{x}_N, y_N$
- Mô hình: biến đổi đầu vào thành đầu ra f sao cho $y=f(\boldsymbol{x})$
- Tập các mô hình có thể có F
- Mất mát: đo sai số giữa đầu ra được dự báo và đầu ra đúng đối với một mẫu cho trước: $L_i \ \ y_i, f(\boldsymbol{x}_i)$
- Hàm chi phí (Loss Function): đo sai số trung bình trên tất cả các mẫu đào tạo

$$L(f) = \frac{1}{N}\sum_{i=1}^{N} L_i \ \ y_i, f(\boldsymbol{x}_i$$

- Quá trình đào tạo: $\hat{f} = \arg\min_{f \in F} \frac{1}{N}\sum_{i=1}^{N} L_i \ \ y_i, f \ \ x_i$

Lưu ý: Thông thường đầu vào là một vectơ nên ta ký hiệu nó và chữ tô đậm $\boldsymbol{x}$, trường hợp chỉ là một số ta ky hiệu là chữ bình thường x. Tương ứng với đầu vào tham số của hàm f có thể là một vectơ được ký hiệu là chữ tô đậm $\boldsymbol{w}$ hoặc chỉ là một chữ thường nếu đầu và chỉ là một số. Nhãn đầu ra thường là một số nên được ký hiệu là chữ thường y.

Chương 2
CÁC PHƯƠNG PHÁP HỌC MÁY KINH ĐIỂN

2.1. MỞ ĐẦU

Chương này trình bày các phương pháp học máy kinh điển chính với tập trung lên các phương pháp học máy có giám sát cho phân loại và hồi quy, bao gồm các phương pháp láng giềng gần nhất (Nearest Neighbor), hồi quy tuyến tính và logistic, hỗ trợ các máy vectơ và các giải thuật dựa trên cây (Tree Based Algorithm). Chương cũng trình bày vấn đề quá khớp (Overfitting) cùng với các chiến lược khắc phục nó. Ngoài ra chương cung cấp tổng quan ngắn gọn các phương pháp học máy không có giám sát cho phân vùng và giảm kích thước.

Tổng kết các phương pháp ML chính được trình bày trong chương này.

- **Học có giám sát**

- **Phân loại**: Các láng giềng gần (Nearest Neighbor), hồi quy logistic (Logistic Regression), máy vectơ hỗ trợ (SVM: Support Vector Machine), phân tích phân biệt

tuyến tính (LDA: Linear Discriminant Analysis), mô hình dựa trên cây: cây quyết định (Decision Tree), rừng ngẫu nhiên (Random Forest), các cây được ngẫu nhiên hóa cực đoan (Extremely Randomized Trees)

- **Hồi quy**:Các láng giềng gần (Nearest Neighbor), hồi quy logistic (Logistic Regression), hồi quy máy vectơ hỗ trợ (Support Vector Machine), mô hình dựa trên cây: cây quyết định (Decision Tree), rừng ngẫu nhiên (Random Forest), các cây được ngẫu nhiên hóa cực đoan (Extremely Randomized Trees), hồi quy kenel ridge

- **Học máy không có giám sát**
- **Phân cụm (Clustering)**: k-means, mô hình trộn Gauss (Gausian Mixture Model)
- **Giảm kích thước (Dimensionality Reduce):** phân tích phần tử chính (PCA: Principal Component Analysis)**,** phân tích phân biệt tuyến tính (LDA: Linear Discriminant Analysis), phân tích phần tử chính hạt nhân (Kenel Principal Component Analysis)

2.2. MỘT SỐ KÝ HIỆU ĐƯỢC SỬ DỤNG TRONG CHƯƠNG

Giả sử N là số lượng các mẫu và K là số lượng các tính năng. Khi này, một mẫu đầu vào là một vectơ kích thước $K\times 1$:

$$\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_K \end{bmatrix}^T$$

Mẫu đầu ra được ký hiệu bởi *y*. Vì thế, một mẫu là $(\boldsymbol{x},y)$. Khi này tập dữ liệu N mẫu có thể tổng kết là một ma trận **X** kích thước $N\times K$ (trong đó mỗi cột chứa tính năng của một mẫu) trình bày dữ liệu đầu vào và một vectơ kích thước $N\times 1$ trình bày dữ liệu mục tiêu:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{K,1} & \cdots & x_{K,N} \end{bmatrix}, \mathbf{X}^T = \begin{bmatrix} x_{1,1} & \cdots & x_{K,1} \\ \vdots & \ddots & \vdots \\ x_{1,N} & \cdots & x_{K,N} \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} y_i & \dots & y_i & \dots & y_N \end{bmatrix}^T$$

trong đó

$\mathbf{x}_i = \begin{bmatrix} x_{1,i} & \dots & x_{j,i} & \dots & x_{K,i} \end{bmatrix}^T$ là các vectơ cột và $j=1,\dots,K$; $i=1,\dots,N$

Không gian đầu vào được ký kiệu là *I* và tập các mẫu đào tạo được ký hiệu là *X*.
Trong trường hợp hồi quy, *y* là một số thực. Trong trường hợp phân loai, *y* là một nhãn đơn. Nói môt cách chính xác hơn, y chỉ nhận một tập hữu hạn các giá trị được gọi là các

nhãn (Label). Tập các loại có thể có (nghĩa là các nhãn) được ký hiệu là $\mathcal{C}=\{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$ với M là số lượng các loại. Các giá trị của các loại không có nghĩa khi chỉ có hai loại, các loại này thường được gọi là các loại dương và âm. Trong trường hợp này và cũng vì lý do toán học, không mất tính tổng quát, ta giả thiết là các giá trị của các loại này đối với phân loại nhị phân là -1 và +1 hoặc 0 và 1.

2.3. CÁC MẠNG LÁNG GIỀNG GẦN NHẤT

2.3.1. Tổng quan

Một trong số các cách tiếp cận bản năng nhất đến học máy là các láng giềng gần nhất hay các hàng xóm gần nhất (Nearest Neighbor). Nó dựa trên bản năng sau: đối với đầu vào cho trước, đầu ra tương ứng của nó có khả năng gần giống các đầu ra của các đầu vào gần giống. Giải thuật láng giềng gần nhất là phương pháp học máy phân loại một điểm dữ liệu mới dựa trên các láng giềng gần nhất trong tập dữ liệu.

Một cách hình thức hơn, các phương pháp láng giềng gần nhất hay lân cận gần nhất sử dụng các mẫu từ vùng láng giềng (Neighborhood) của một điểm $\boldsymbol{x}$ cho trước ký hiệu bởi $N(\boldsymbol{x})$ để tiến hành dự báo.

Đối với các nhiệm vụ hồi quy, dự báo được tính toán là trung bình được đánh trọng số (Weighted Mean) của các giá trị mục tiêu trong $N(x)$:

$$\hat{y} = \sum_{\boldsymbol{x}_i \in N(x)} w_i^{(x)} y_i$$

Trong đó $w_i^{(x)}$ là trọng số liên kết với $\boldsymbol{x}_i$ để dự báo đầu ra của $\boldsymbol{x}$, với $w_i^{(x)} \geq 0 \forall i$ và $\sum_i w_i^{(x)} = 1.$

Đối với các nhiệm vụ phân loại (Classification), nhãn được dự báo tương ứng với nhãn có tổng được đánh trọng số (Weighted Sum) của các sự xuất hiện của từng nhãn:

$$\hat{y} = \underset{C}{\operatorname{argmax}} \sum_{\boldsymbol{x}_i \in N(x)} w_i^{(x)} \mathbf{1}_{y_i = C_m}$$

Trong đó $1_{y_i = C_m} = \begin{cases} 1, \text{if } y_i = C_m \\ 0, \text{ otherwise} \end{cases}$

Thông số quan trọng của các phương pháp láng giềng gần nhất là số đo (metrics) được ký hiệu là d, là một hàm định nghĩa sự khác biệt. Metrics được sử dụng để định nghĩa vùng láng giềng của mọi điểm và cũng có thể được sử dụng để tính toán các trọng số.

2.3.2. Các số đo (Metrics)

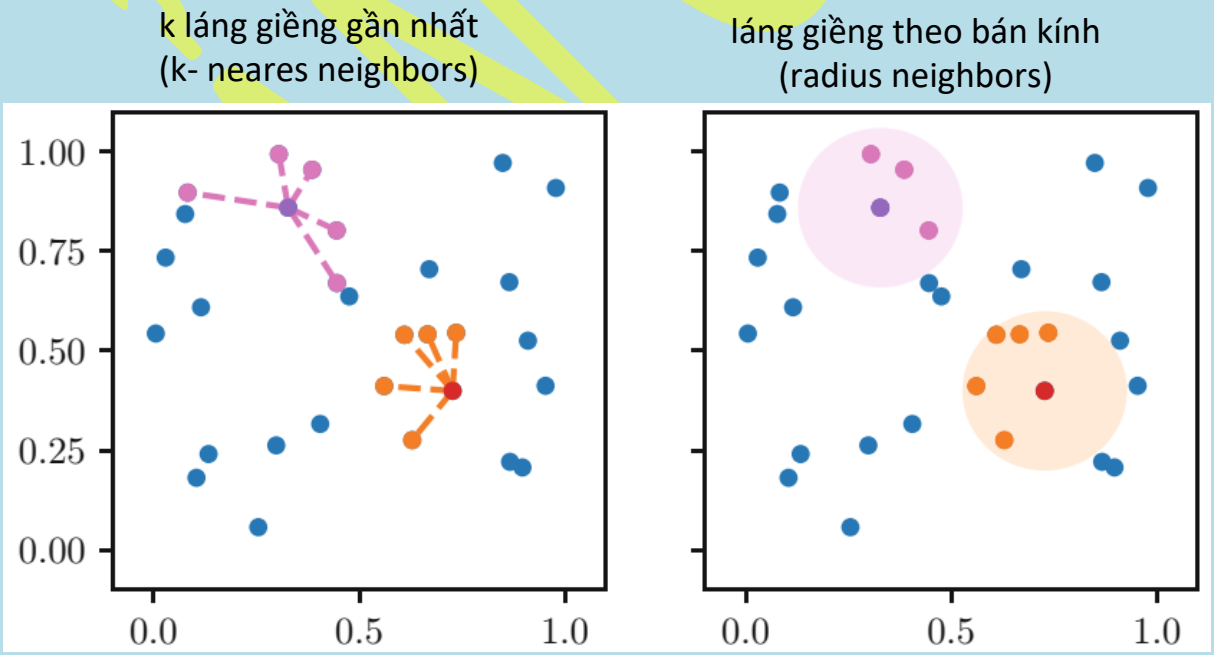
Rất nhiều các số đo đã được định nghĩa cho các kiểu dữ liệu đầu vào khác nhau, như các vectơ các số thực, các số nguyên hay đại số Boole. Trong số các kiểu khác nhau này, các vectơ các số thực là một trong số các kiểu dữ liệu đầu vào phổ biến nhất, đối với nó số đo được sử dụng phổ biến nhất là khoảng cách Euclid (Euclidean Distance), được định nghĩa là:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in I, \left\| \boldsymbol{x} - x' \right\|_2 = \sqrt{\sum_{j=1}^{K} x_j - x_j^{'}}^{2}$$

Khoảng cách Euclid đôi khi được gọi là khoảng cách "gốc", vì nó dựa trên định lý Pytagor mà mọi người thường sử dụng nó hàng ngày.

2.3.3. Láng giềng (Neighborhood)

Hai định nghĩa láng giềng phổ biến nhất dựa trên số lượng láng giềng hoặc bán kính xung quanh một điểm cho trước. Hình 2.1 minh họa các khác nhau giữa hai định nghĩa nói trên.



Hình 2.1. Các định nghĩa khác nhau của vùng láng giềng. Bên trái, vùng láng giềng của một điểm cho trước là tập các láng giềng gần nhất. Bên phải, vùng láng giềng của một điểm cho trước là tập các điểm có sự khác nhau nhỏ hơn bán kính. Đối với

một với đầu vào cho trước, vùng lân cận có thể khác nhau tùy thuộc vào định nghĩa. Khoảng cách Euclid được sử dụng là số đo cho cả hai thí dụ.

Phương pháp k-neares neighbor định nghĩa láng giềng của một điểm cho trước $\boldsymbol{x}$ là tập k điểm gần $\boldsymbol{x}$ nhất:

$$N(\mathrm{x}) = \left\{\boldsymbol{x}_i\right\}_{i=1}^{k} \text{với } \mathrm{d}\left(\boldsymbol{x}, \boldsymbol{x}_1\right) \leq \ldots \leq \mathrm{d}\left(\boldsymbol{x}, \boldsymbol{x}_k\right)$$

Phương pháp láng giềng theo bán kính (Radius Neighbor) định nghĩa làng giềng của một điểm cho trước $\boldsymbol{x}$ là tập các điểm có khác nhau với $\boldsymbol{x}$ nhỏ hơn một bán kính cho trước được ký hiệu là r:

$$N(\boldsymbol{x}) = \left\{\boldsymbol{x}_i \in X \middle| \mathrm{d}\left(\boldsymbol{x}, x_i\right) < \mathrm{r}\right.$$

2.3.4. Trọng số (Weight)

Hai cách tiếp cận phổ biến nhất để tính toán các trọng số là sử dụng:

- Các trọng số đồng nhất (tất cả các trọng số đều bằng nhau):

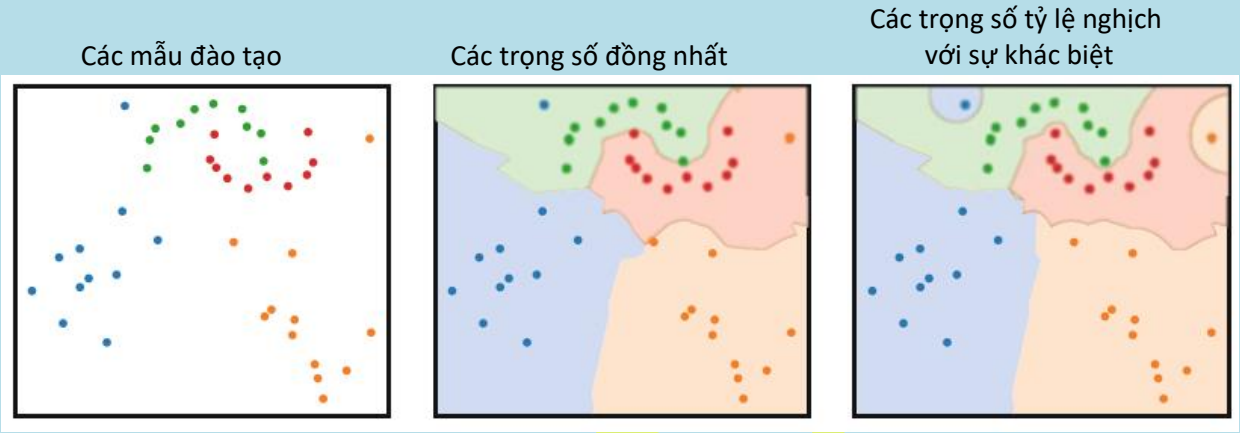
$$\forall \mathrm{i}, w_i^{(\mathrm{x})} = \frac{1}{\left|N(\boldsymbol{x})\right|}$$

- Các trọng số tỷ lệ nghịch với sự khác biệt:

$$\forall \mathrm{i}, w_i^{(\mathrm{x})} = \frac{\frac{1}{\mathrm{d}\left(\boldsymbol{x}_i, x\right)}}{\sum_j \frac{1}{\mathrm{d}\left(\boldsymbol{x}_j, \boldsymbol{x}\right)}} = \frac{1}{\mathrm{d}\left(\boldsymbol{x}_i, \boldsymbol{x}\right) \sum_j \frac{1}{\mathrm{d}\left(\boldsymbol{x}_j, \boldsymbol{x}\right)}}$$

Với các trọng số đồng nhất, mỗi điểm trong vùng lân cận đóng góp như nhau vào dự báo. Với các trọng số tỷ lệ nghịch với sự khác biệt, các điểm gần hơn đóng góp nhiều hơn vào dự báo so với các điểm xa hơn. Hình 2.2 minh họa các hàm quyết định khác nhau nhận được từ các trọng số đồng nhất và các trong số tỷ lệ nghịch với sự khác biệtđối với mô hình phân loại 3 láng giềng gần nhất.

BẢN GỐC



Hình 2.2. Ảnh hưởng của định nghĩa các trọng số lên hàm dự báo của mô hình phân loại 3 láng giềng gần nhất. Khi các trọng số là nghịch đảo của sự khác biệt, bộ phân loại phụ thuộc nhiều hơn vào dữ liệu ngoại lai vì các dự báo láng giềng gần của bất kỳ đầu vào hầu như được dành riêng cho đầu vào này, độc lập với số lượng láng giềng. Với phân phố đồng nhất, hàm dự báo có xu thế mịn màng hơn.

2.3.5. Tìm láng giềng

Phương pháp vét cạn hay toàn diện (Brute Force Method) tính toán vùng lân cận cho N điểm với K tính năng là phương pháp tính toán số đo (metric) cho từng cặp đầu vào,có độ phức tạp tuật toán $O(N^2K)$ (giả thiết là đánh giá số đo cho một cặp đầu vào có độ phức tạp là $O(K)$, đây là trường hợp cho hầu hết các số đo). Tuy nhiên có thể giảm độ phức tạp giải thuật nếu số đo là khoảng cách, nghĩa là, nếu số đo thỏa mãn các thuộc tính sau:

1. Không âm: $\forall a, b, d(a, b) \geq 0$
2. Đồng nhất: $\forall a, b, d(a, b) = 0$ nếu và chỉ nếu a=b

2.4 HỒI QUY TUYẾN TÍNH

Hồi quy tuyến tính là một phương pháp hồi quy kết hợp tuyến tính các tính năng. Mỗi tính năng liên kết với một hệ số thể hiện một trọng số tương đối của tính năng này so với các tính năng khác. Một ẩn dụ đời thực (Real Life Metaphor) sẽ xem các hệ số này như là các thành phần của một công thức bánh: chìa khóa để tìm được cân bằng tốt nhất (ví dụ các tỷ lệ) giữa các thành phần để có thể là được chiếc bánh tốt nhất.

Về mặt toán học, mô hình tuyến tính là một mô hình kết hợp tuyến tính các tính năng được biểu diễn như sau:

$$f(x) = w_0 + \sum_{j=1}^{K} \boldsymbol{w}_j x_j$$

Trong đó w_0 là độ lệch (Bias), vectơ trọng số $\boldsymbol{w}$ và vectơ tính năng đầu vào $\boldsymbol{x}$ được định nghĩa như sau:

$$\boldsymbol{w} = \begin{bmatrix} w_1 & \cdots & w_j & \cdots & w_K \end{bmatrix}^T, \boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_j & \cdots & x_K \end{bmatrix}^T; \; j = 1, \ldots, K$$

Để tiện lợi độ liệch w_0 được ghép vào đầu (hoặc cuối) vectơ trọng số $\boldsymbol{w}$ và 1 được thêm vào phần tử đầu (hoặc cuối) của vectơ tính năng đầu vào $\boldsymbol{x}$:

$$\boldsymbol{w} = \begin{bmatrix} w_0 & w_1 & \ldots & w_j & \ldots & w_K \end{bmatrix}^T, \boldsymbol{x} = \begin{bmatrix} 1 & x_1 & \ldots & x_j & \ldots & x_K \end{bmatrix}^T; \; j = 0.1, \ldots, K$$

hay

$$\boldsymbol{w} = \begin{bmatrix} w_1 & \ldots & w_j & \ldots & w_K & w_0 \end{bmatrix}^T, \boldsymbol{x} = \begin{bmatrix} x_1 & \ldots & x_j & \ldots & x_K & 1 \end{bmatrix}^T; \; j = 0.1, \ldots, K$$

Khi này ta có thể viết lại $f(\boldsymbol{x})$ là tích vô hướng giữa vectơ $\boldsymbol{x}$ và vectơ $\boldsymbol{w}$:

$$f(\boldsymbol{x}) = w_0 \times 1 + \sum_{j=1}^{K} w_j x_j = \boldsymbol{x}^T \boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{x}$$

Trong đó vectơ $\boldsymbol{w}$ bao gồm:

- Điểm cắt, còn được gọi là độ lệch, w_0
- Các hệ số (w_1,…,w_K) trong đó mỗi hệ số w_j liên kết với một tính năng x_j

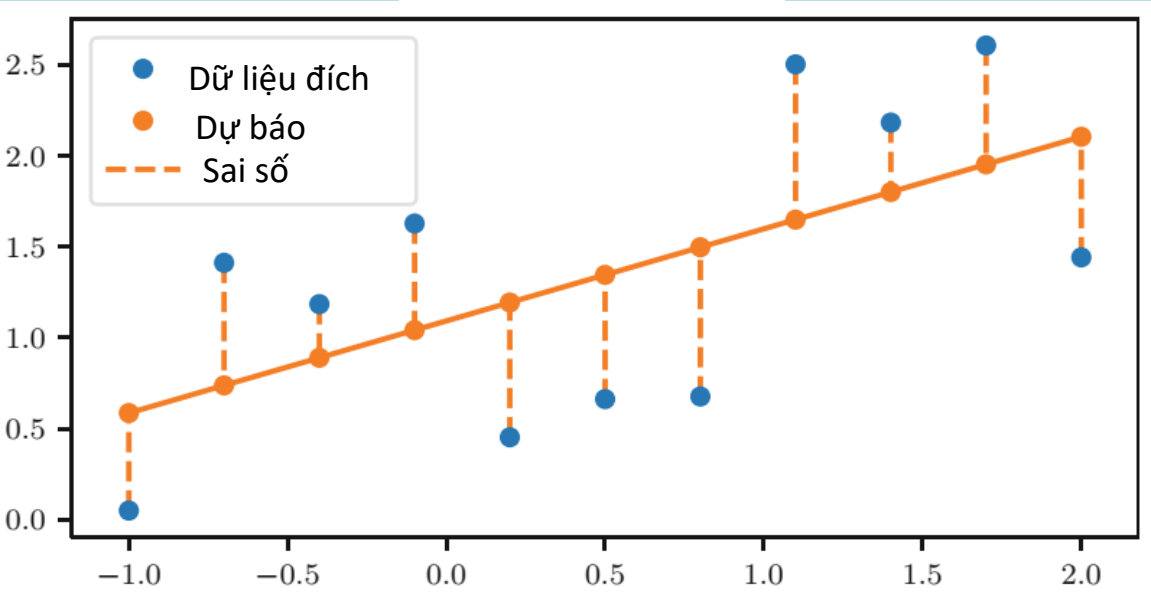
Trong trường hợp hồi quy tuyến tính, $f(\boldsymbol{x})$ là đầu ra được dự đoán đối với mẫu i của N mẫu sẽ là:

$$\hat{y} = f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{x}$$

Trong đó

d báo đầu ra.

BANGOC



Hình 2.3. Hồi quy bình phương cực tiểu gốc. Các hệ số (nghĩa là, điểm cắt và độ dốc với một bộ dự đoán đơn được đánh giá bằng cách cực tiểu hóa tổng sai số bình phương

Có một số phương pháp đánh giá các hệ số ***w***. Trong phần này ta sẽ xét một phương pháp lâu đời nhất được biết đến là hồi quy bình phương cực tiểu gốc (Ordinary Least Squares Regression). Trong trường hợp hồi quy bình phương cực tiểu gốc, hàm chi phí L là tổng các sai số bình phương trên tập dữ liệu đào tạo (xem hình 2.3) :

$$L(\boldsymbol{w}) = \sum_{i=1}^{N} y_i - \hat{y}_i^{\,2} = \sum_{i=1}^{N} y_i - \boldsymbol{x}_i^T \boldsymbol{w}^{\,2} = \left\| \boldsymbol{y} - \mathbf{X}^T \boldsymbol{w} \right\|_2^2 = \left\| \boldsymbol{y} - \boldsymbol{w}^T \mathbf{X} \right\|_2^2$$

trong đó:

$y_i, \hat{y}_i$ là đầu ra đúng và đầu ra dự báo của mẫu i

$\boldsymbol{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$, $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 & \dots & \hat{y}_N \end{bmatrix}^T = \mathbf{X}^T \boldsymbol{w} = \boldsymbol{w}^T \mathbf{X}$ là vectơ đúng và vectơ dự báo,

$\mathbf{X} = \begin{matrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_i & \dots & \boldsymbol{x}_N \end{matrix} = \begin{bmatrix} 1 & \cdots & 1 \\ x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{K,1} & \cdots & x_{K,N} \end{bmatrix}$, $\boldsymbol{x}_i = \begin{bmatrix} 1 \\ x_{i,1} \\ \vdots \\ x_{i,K} \end{bmatrix}$ là ma trận và vect[ơ cột tính năng đầu vào và $\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}$ là vectơ trọng số.

Ta muốn tìm ra các thông số tối ưu w* cho phép giảm thiểu hàm chi phí :

$$\boldsymbol{w}^* = \underbrace{\arg\min}_{\boldsymbol{w}} \mathrm{L}(\boldsymbol{w})$$

Vấn đề tối ưu là lồi, nghĩa là mọi cực tiểu cục bộ là một cực tiểu toàn cục và có thể lấy đạo hàm, nghĩa là mọi cực tiểu cục bộ đều có gradient bằng không. Vì thế ta nhắm đến tìm các gradient không của hàm chi phí:

$$\Delta_{\boldsymbol{w}^*} \mathrm{L} = 0$$

$$\Rightarrow 2\mathbf{X}\mathbf{X}^{\mathrm{T}}\boldsymbol{w}^* - 2\mathbf{X}\boldsymbol{y} = 0$$

$$\Rightarrow \mathbf{X}\mathbf{X}^{\mathrm{T}}\boldsymbol{w}* = \mathbf{X}\boldsymbol{y}$$

$$\Rightarrow \boldsymbol{w}^* = \mathbf{X}\mathbf{X}^{\mathrm{T}}{}^{-1}\mathbf{X}\boldsymbol{y}$$

Trong đó $\Delta_{\mathbf{w}*}\mathrm{L} = \left(\frac{\partial \mathrm{L}}{w_1^*}, \ldots, \frac{\partial \mathrm{L}}{w_K^*}\right)^{\mathrm{T}}$ là một vectơ cột mà thành phần là các đạo hàm riêng phần theo các biến w của f.

Hồi quy bình phương cực tiểu gốc là một trong số ít các vấn đề tối ưu học máy mà đối với nó tồn tại công thức dạng đóng, nghĩa là, có thể tính được giải pháp tối ưu bằng cách sử dụng một số lượng hữu hạn các phép toán tiêu chuẩn như cộng, nhân, và đánh giá chi phí bằng các hàm biết rõ.

Ta có thể tổng kết hồi quy tuyến tính như sau:

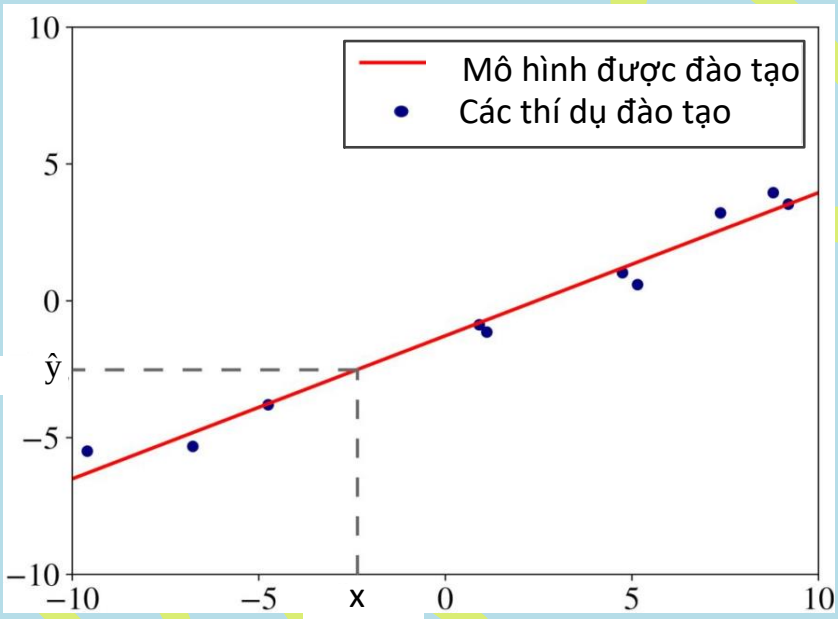
BAMBOO

- **Mục đích:** tìm siêu mặt phẳng tốt nhất (nghĩa là đường thẳng khi K=1, mặt phẳng khi K=2) ánh xạ các vào vào các đầu ra.
- **Công thức toán:**quan hệ tuyến tính giữa đầu ra được dự báo ŷ và đầu vào để gảm thiểu tổng các sai số bình phương:

  $$\hat{y} = w_0^* + \sum_{j=1}^{K} w_j^* x_j \quad \textbf{với} \quad \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^T \boldsymbol{w} \right)^2$$

  Đối với K=1, $\hat{y} = w_0^* + w_1^* x$ mô hình là một đường thẳng như trên hình 2.4.
- **Chính quy hóa (Regularization)**: có thể bị phạt để tránh quá khớp (ridge), để thực hiện chọn tính năng hay cả hai (Elastic-Net), (xem 2.4.2).

Hình 2.4 cho thấy hồi quy tuyến tính cho thí dụ một chiều.



Ước tính đầu ra mô hình được đào tạo

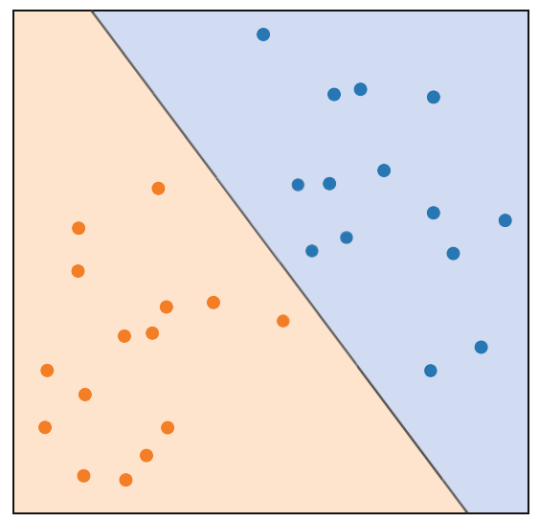
$\hat{y} = w_0^* + w_1^* x$

Hình 2.4. Hồi quy tuyến tính cho các thí dụ một chiều (K=1)

2.5 HỒI QUY LOGISTIC

Một cách trực quan, hồi quy tuyến tính bao gồm tìm một đường thẳng khớp nhất với dữ liệu: đầu ra đúng phải càng gần với đường thẳng này các tốt. Đối với phân loại nhị phân, ta muốn đường thẳng này phân tách cả hai loại tốt nhất có thể: các mẫu từ một loại phải tất cả nằm trong một không gian con và các mẫu từ lớp khác phải nằm trong không gian con khác với các đầu vào càng xa càng tốt với đường này.

Về mặt toán học, đối với các nhiệm vụ phân loại nhị phân, mô hình tuyến tính được định nghĩa bởi một siêu măt phẳng phân chia không gian đầu vào thành hai không gian con sao cho mỗi không gian con đặc trưng cho một loại. Chẳng hạn, một đường thẳng phân chia một mặt phẳng thành hai không gian con trong trường hợp hai chiều. Một siêu mặt phẳng được định nghĩa bởi một vectơ $\boldsymbol{w} = \begin{bmatrix} w_0 & w_1 & \dots & w_K \end{bmatrix}^T$ và một hàm $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{x}$, trong đó $\boldsymbol{x} = \begin{bmatrix} 1 & w_1 & \dots & w_K \end{bmatrix}^T$ tương ứng với khoảng cách có dấu (Signed Distance) giữa đầu vào $\boldsymbol{x}$ và siêu mặt phẳng $\boldsymbol{w}$; trong một không gian con, khoảng cách với một đầu vào bất kỳ luôn dương, trong khi đó trong gian con khác, khoảng các với một đầu vào bất kỳ luôn âm. Hình 2.5 minh họa hàm quyết định trong trường hợp hai chiều trong đó cả lai loại đều có thể phân tách tuyến tính được.



Hình 2.5. Hàm quyết định của một mô hình hồi quy logistic. Một hồi quy logistic là một mô hình tuyến tính, nghĩa là, hàm quyết định là tuyến tính. Trong trường hợp hai chiều, nó phân tách một mặt phẳng bằng một đường thẳng

Dấu của khoảng cách có dấu tương ứng với hàm quyết định của một mô hình phân loại nhị phân tuyến tính:

$$\hat{y} = \text{sign}\ f(\boldsymbol{x}) = \begin{matrix} +1 \text{ if } f(\boldsymbol{x}) > 0 \\ -1 \text{ if } f(\boldsymbol{x}) < 0 \end{matrix}$$

Mô hình hồi quy logistic là một một hình tuyến tính xác suất, nó biến đổi khoảng cách có dấu đến siêu mặt phẳng vào một xác suất sử dụng hàm sigmoid, được ký hiệu là:

$$\sigma(u) = \frac{1}{1+\exp(-u)}$$

Ta xét một mô hình tuyến tính:

$$f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{x} = w_0 + \sum_{j=1}^{K} w_j x_j \quad , j=1,2,\ldots,K$$

trong đó $\boldsymbol{x} = \left[1, x_1, \ldots, x_j, \ldots, x_K\right]^T$; $\boldsymbol{w} = \left[w_0, w_1, \ldots, w_j, \ldots, w_K\right]^T$.

Xá suất thuộc về loại dương là:

$$P(y=+1 \mid \mathbf{x}=\boldsymbol{x}) = \sigma\ f(\boldsymbol{x}) = \frac{1}{1+\exp\ -f(\boldsymbol{x})}$$

Và xác suất thuộc về loại âm là:

$$P(y=-1 \mid \mathbf{x}=\boldsymbol{x}) = 1 - P(y=+1 \mid \mathbf{x}=\boldsymbol{x})$$

$$= \frac{\exp\ -f(\boldsymbol{x})}{1+\exp\ -f(\boldsymbol{x})}$$

$$= \frac{1}{1+\exp\ f(\boldsymbol{x})}$$

$$p\ \ y=-1 \mid \mathbf{x}=\boldsymbol{x}\ \ = \sigma\ -f(\boldsymbol{x})$$

Bằng các sử dụng đảo hàm sigmoid, được biết đến là hàm logarit, ta có thể tấy rằng logarit của tỷ lệ chênh lệch được mô hình như là một kết hợp của các tính năng:

$$\log\left(\frac{P(y=+1 \mid \mathbf{x}=\boldsymbol{x})}{P(y=-1 \mid \mathbf{x}=\boldsymbol{x})}\right) = \log\left(\frac{P(y=+1 \mid \mathbf{x}=\boldsymbol{x})}{1-P(y=+1 \mid \mathbf{x}=\boldsymbol{x})}\right) = f(\boldsymbol{x})$$

Các hệ số của $\boldsymbol{w}$ được ước tính bằng cách cực đại hóa hàm khả năng (Likelihood Function), là hàm đo khả năng khớp của mô hình với dữ liệu đào tạo:

$$L_{likelihood}(\boldsymbol{w}) = \prod_{i=1}^{N} P\ \ y=y_i \mid \mathbf{x}=\boldsymbol{x}_i, \boldsymbol{w}$$

Vì lý do tính toán, sẽ dễ hơn khi cực đại hóa log khả khả năng (Log-likeligood), chỉ đơn giản là logarit của hàm khả năngg để biến tích các thành phàn thành tổng các thành phần:

$$\log\ L_{\text{likelihood}}(\boldsymbol{w}) = \sum_{i=1}^{N} \log\ P(y = y_i \mid \mathbf{x} = \boldsymbol{x}_i; \boldsymbol{w}$$

$$= \sum_{i=1}^{N} \log\ \sigma(y_i\, f(\boldsymbol{x}_i; \boldsymbol{w}))$$

$$= \sum_{i=1}^{N} -\log\ 1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{w})$$

$$\log\ L_{\text{likelihood}}(\boldsymbol{w}) = -\sum_{i=1}^{N} \log\ 1 + \exp(y_i \boldsymbol{x}_i^T \boldsymbol{w})$$

Cuối cùng, ta có thể viết lại bài toán cực đại như là bài toán cực tiểu với lưu ý rằng

$$\max_{\mathbf{w}} \log L_{\text{likelihood}}\ \ \boldsymbol{w} = -\min_{\mathbf{w}}\ -\log(L_{\text{likelihood}}(\boldsymbol{w}))\ :$$

$$\max_{\mathbf{w}} \log(L_{\text{likelihood}}(\mathbf{w})) = -\min \sum_{i=1}^{N} \log\ 1 + \mathbf{e}\exp(y_i \boldsymbol{x}_i^T \boldsymbol{w})$$

Ta có thể thấy các hệ số $\boldsymbol{w}$ cực đại hóa hàm khả năng cũng là các hệ số cực tiểu hóa tổng các giá trị mất mát logistic (Logistic Loss) với mất mát logistic được định nghĩa như sau:

$$L_{\text{logistic}}\ \ y, f(\boldsymbol{x})\ = \log_2\ 1 + \exp(y f(\boldsymbol{x}))\ = \log\ 1 + \exp(y f(\boldsymbol{x}))\ / \log(2)$$

Không giống như hồi quy tuyến tính, không có công thức dạng đóng cho cực tiểu hóa này. Như vậy ta phải sử dụng một phương páp tối ưu hóa như đã trình bày trong chương 1. Trong thực tế, các phương pháp phức tạp hơn được sử dụng, chẳng hạn các phương pháp tựa Newton (Quasi-Newton) hay các biến thể của giảm gradient ngẫu nhiên (Stochastic Gradient Descent) thường được sử dụng. Khái niệm chính nền tảng cho hồi quy logistic được được tổng kết dưới đây.

- **Bài toán:** siêu mặt phẳng tốt nhất (ví dụ: đường thẳng) phân chia hai loại.
- **Công thức toán:** khoảng cách có dấu đến siêu mặt phẳng được ánh xạ vào xác suất thuộc lớp dương bằng cách sử dụng hàm sigmoid:

  $$f(\boldsymbol{x}_i) = \boldsymbol{x}_i^T \boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{x} = w_0 + \sum_{j=1}^{K} w_j x_j \quad , j=1,2,\ldots,K$$

  trong đó: $\boldsymbol{x}_i = \begin{bmatrix} 1 & x_1 & \ldots & x_K \end{bmatrix}^T$, $\boldsymbol{w} = \begin{bmatrix} w_0 & w_1 & \ldots & w_K \end{bmatrix}^T$

$$P(y=+1|\mathbf{x}=\boldsymbol{x}_i=\sigma\ f(\boldsymbol{x}_i)\ =\frac{1}{1+\exp\ -f(\boldsymbol{x}_i)}$$

- **Ước tính**: cực đại hóa hàm khả năng.
- **Chính quy hóa**: có thể bị phạt để tránh quá khớp (phạt ℓ_2), để thực hiện chọn tính năng (phạt ℓ_1) (elastic-net penalty: phạt lưới đàn hồi).

2.6. QÚA KHỚP, CÁC MÔ HÌNH PHẠT VÀ CHÍNH QUY HÓA

2.6.1. Quá khớp

Các trình bày gốc của hồi quy bình phương tối thiểu và hồi quy logistic là các mô hình không được chính quy hóa, nghĩa là, mô hình được đào tạo để khớp với dữ liệu càng nhiều càng tốt. Ta xét một thí dụ đời thực rất giống với học tập của con người. Nếu một người học thuộc lòng một cuốn sách, anh ta có thể giải các bài tập trong cuốn sách, nhưng không thể áp dụng các khái niệm lý thuyết cho các bài tập mới hay các tình huống đời thực. Nếu một người chỉ đọc lướt một cuốn sách, người này sẽ không thể giải các bài tập trong cuốn sách cũng như các bài tập mới.

Các khái niệm tương ứng được biết đến như là quá khớp (Overfitting) hay chưa khớp (underfiting) trong học máy. Quá khớp xẩy ra khi một mô hình khớp quá tốt dữ liệu đào tạo và mở rộng kém sang dữ liệu mới. Trái lại, chưa khớp xẩy ra khi một mô hình không nắm bắt đủ tốt các đặc trưng của dữ liệu đào tạo và vì thế cũng mở rộng kém sang dữ liệu mới.

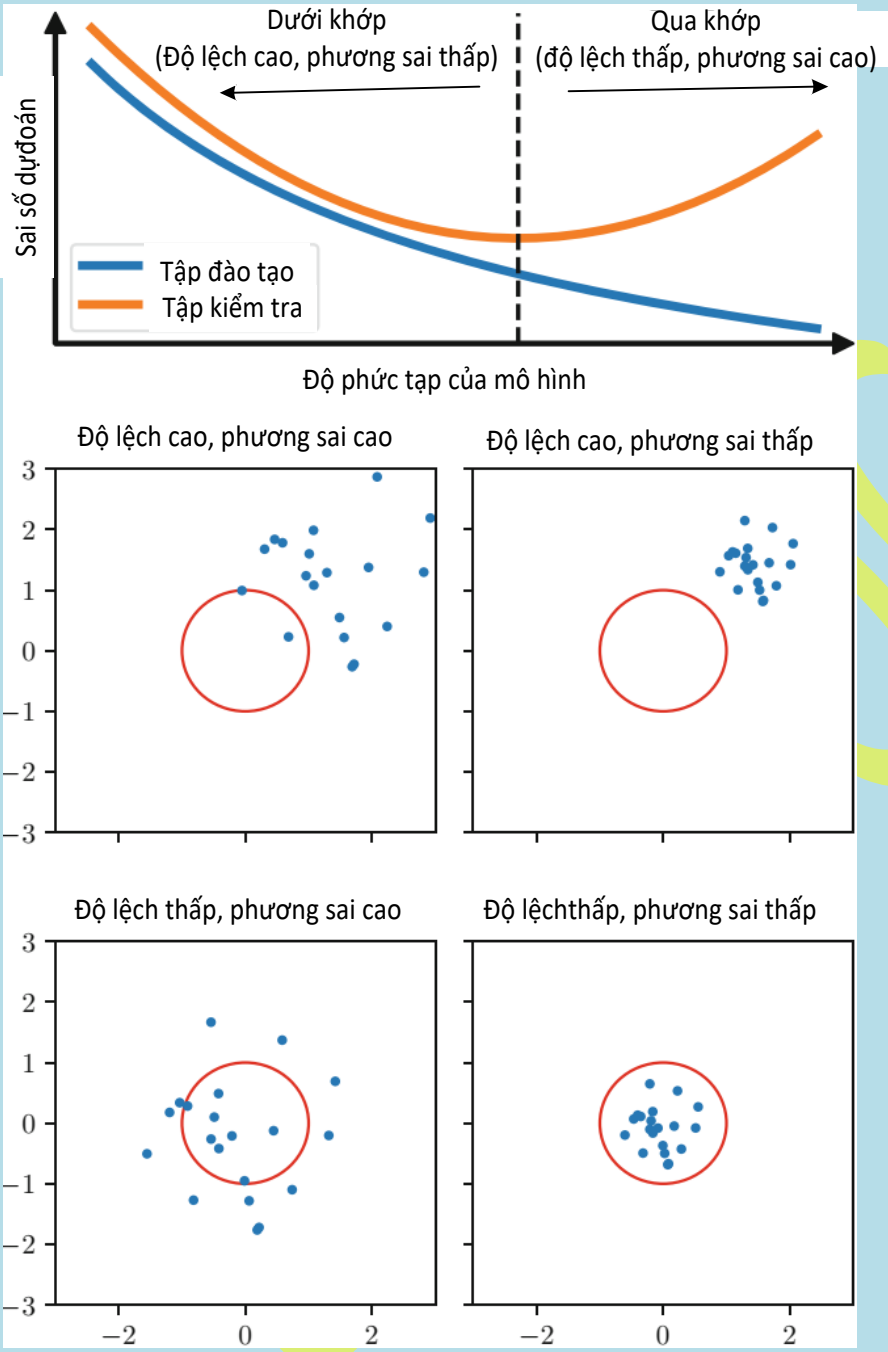
Qua khớp và chưa khớp liên quan đến các thuật ngữ thường xuyên được dùng trong học máy: độ lệch (Bias) và phương sai (Variance). Độ lệch được định nghĩa như là sự khác nhau được kỳ vọng (ví dụ: trung bình) giữa đầu ra thực và đầu ra được dự báo. Phương sai được định nghĩa như là sự biến đổi (Variability) của đầu ra được dự báo. Chẳng hạn, ta xét một mô hình dự báo tuổi của một người từ một bức ảnh. Nêu mô hình luôn luôn đánh giá thấp hoặc đánh giá cao tuổi, khi này mô hình bị lệch. Nếu mô hình mắc lỗi hoặc lớn mắc hoặc lỗi nhỏ, khi này mô hình có phương sai cao. Lý tưởng, ta muốn có mô hình có độ lệch và phương sai nhỏ. Tuy nhiên, độ lệch của mô hình có khuynh hướng tăng khi giảm độ lệch của nó và phương sai của mô hình có khuynh hương tăng khi giảm độ lệch của nó. Hiện tượng này được gọi là sự đánh đổi giữa độ lệch và phương sai (Bias-Variance Trade-off). Hình 2.6 minh họa trường hợp này. Ta cũng có thế nhận thấy nó bằng cách tính sai số bình phương (Squared Error) giưã đầu ra (cố định) y và đầu vào được dự báo $\hat{y}$ (một biến ngẫu nhiên):

$$E\left[(y-\hat{y})^2\right] = E\left[y^2 - 2y\hat{y} + \hat{y}^2\right]$$
$$= y^2 - 2yE[\hat{y}] + E[\hat{y}^2]$$
$$= y^2 - 2yE[\hat{y}]^2 + E[\hat{y}^2] + E[\hat{y}]^2 - E[\hat{y}]^2$$
$$= E[\hat{y}^2] - y^2 + E[\hat{y}^2] - E[\hat{y}]^2$$
$$= E[\hat{y}^2] - y^2 + E[\hat{y}^2 - E[\hat{y}]^2]$$
$$= E[\hat{y}] - y^2 + E[\hat{y}^2 - 2\hat{y}E[\hat{y}] + E[\hat{y}]^2]$$
$$= E[\hat{y}] - y^2 + E\left[\hat{y} - E[\hat{y}]\right]^2$$
$$E\left[(y-\hat{y})^2\right] = \underbrace{E[\hat{y}] - y^2}_{\text{bias}^2} + \underbrace{\text{Var}[\hat{y}]}_{\text{variance}}$$

Trong đó bias là độ lệch.

Hình 2.6 cho thấy trục ngang là một vòng lặp học máy và trục đứng là sai số, đường xanh là sai số đào tạo và đường đỏ là sai số chấp thuận. Ít có khả năng mô hình tiếp tục học sau điểm mà tại đó sai số chấp thuận sẽ tăng trong khi sai số đào tạo vẫn giảm. ít có khả năng ta sẽ bỏ học trước sự kiện chưa khớp. Vì thế mục đích là định vị điểm nổi bật đặc thù để rời bỏ học. Như vậy ta nhận được khớp lý tưởng sau điểm ta nhận được quá khớp. Đối với mạng tế bào thần kinh, vòng lặp sẽ phải định vị một sắp xếp lý tưởng giữa độ lệch và phương sai.

BAN SOC



Hình 2.6. Minh họa chưa khớp và quá khớp. Chưa khớp xẩy ra khi một mô hình quá đơn giản và không nắm bắt đủ tốt các đặc trưng của dữ liệu đào tạo, dẫn đến độ lệch cao và phương sai thấp. Ngược lại, quá khớp xẩy ra khi một mô hình quá phức tạp và học tập tạp âm (noise) trong dữ liệu đào tạo dẫn đến độ lệch thấp và phương sai cao

2.6.2. e)

2.6.2.1. Tổng quan

Phụ thuộc vào loại của các phương pháp, tồn tại các chiến lược xử lý quá khớp khác nhau. Đối với các phương pháp láng giềng, số lượng các láng giềng được định nghĩa cho vùng lân cận của mọi đầu vào và chiến lược tính toán các trọng số là các siêu tham số để điều khiển sự đánh đổi giữa độ lệch và phương sai (Bias-Variance Trade-off). Đối với các mô hình được trình bày trong chương này, ta nhắc lại các chiến lược xử lý sự đánh đổi giữa độ lệch và phương sai trong các phần tương ứng. Trong phần này, ta xét các chiến lược được sử dụng phổ biến nhất cho các mô hình mà các thông số của chúng được tối ưu hóa bằng cách cực tiểu hóa một hàm chi phí được định nghĩa như là các giá trị mất mát trung bình trên toàn bộ các mẫu đào tạo:

$$\min_{\mathbf{w}} \mathrm{L}(\mathbf{w}) \text{ với } \mathrm{L}(\boldsymbol{w}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{L}_{\mathrm{i}} \quad y_{\mathrm{i}}, \mathrm{f} \quad \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{w}$$

Đây là, chẳng hạn, trường hợp của các phương pháp hồi quy tuyến tính và logistics mà ta đã xét trong các phần trước.

Ý tưởng chính của việc đưa ra thành phần phạt Pen(***w***) là nó sẽ ràng buộc các tham số ***w*** để có được một số thuộc tính mong muốn. Các hình thức phạt phổ biến là phạt ℓ_2, phạt ℓ_1 và phạt lưới đàn hồi (elastic-net).

2.6.2.2. Phạt ℓ_2

Phạt ℓ_2 được định nghĩa như là chuẩn bình phương các hệ số của ***w***:

$$\ell_2(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2 = \sum_{\mathrm{j}=1}^{\mathrm{K}} w_{\mathrm{j}}^2$$

Phạt ℓ_2 buộc từng hệ số w_j không được quá lớn và làm cho các hệ số này bền vững hơn đối với cộng tuyến (Collinearity) (chẳng hạn khi một số tính năng gần như là tổ hợp tuyến tính của các tính năng khác).

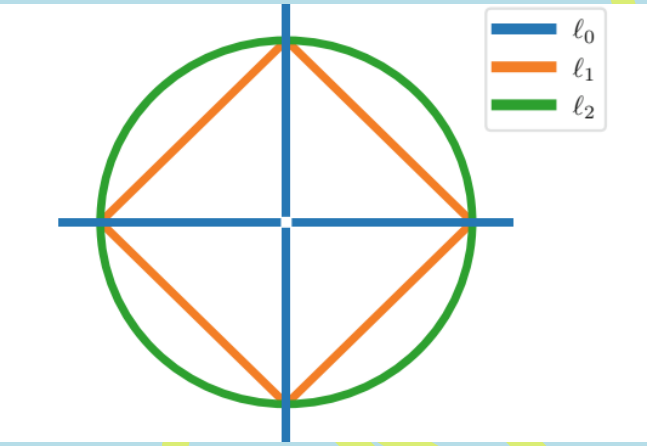
2.6.2.3. Phạt ℓ_1

Phạt ℓ_2 buộc các giá trị của các tham số không được quá lớn, nhưng không thúc đẩy việc làm cho các giá trị nhỏ tiến đến không. Tất nhiên, bình phương các giá trị nhỏ thậm chí là một số nhỏ hơn. Khi số lượng các tính năng lớn hoặc khi khả năng diễn giải quan trọng, tạo lập mô hình lựa chọn các tính năng quan trọng nhất có thể hữu ích. Số đo tương ứng là "chuẩn" (Norm) ℓ_0, được định nghĩa như là số lượng của các phần tử khác không:

$$\ell_0(\boldsymbol{w}) = \|\boldsymbol{w}\|_0 = \sum_{j=1}^{K} \mathbf{1}_{w_j \neq 0}$$

Tuy nhiên ℓ_0 không thể lấy đạo hàm cũng như không lồi (Convex) (đây là các thuộc tính cần thiết để giải một bài toán tối ưu, (ta sẽ không xét chi tiết điều này thêm vì mục đích ngắn gọn). Xấp xỉ hóa khả đạo hàm lồi tốt nhất của "chuẩn" ℓ_0 là "chuẩn" ℓ_1 (xem hình 2.7), được định nghĩa như là tổng các giá trị tuyết đối của từng phần tử:

$$\ell_1(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 = \sum_{j=1}^{K} |w_j|$$



Hình 2.7. Các quả bóng đơn vị của các chuẩn ℓ_0, ℓ_1 và ℓ_2. Đối với từng chuẩn, tập các điểm trong $\mathbb{R}^2$ có chuẩn bằng 1 được vẽ. chuẩn ℓ_1 là xấp xỉ cho hóa ℓ_0 lồi tốt nhất. Lưu ý các đường thẳng cho chuẩn ℓ_0 kéo dài đến $-\infty$ và $+\infty$ nhưng không bị cắt vì lý do vẽ đồ thị

2.6.2.4. Phạt lưới đàn hồi (Elastic-Net Penalty)

Cả hai phạt ℓ_2 và ℓ_1 đều có mặt tích cực và tiêu cực. Để cố gắng nhận được tốt nhất trong số các phạt, ta có thể bổ sung cả hai phạt vào hàm mục tiêu. Sự kết hợp của hai phạt này được gọi là phạt lưới đàn hồi (Elastic-Net):

$$\mathrm{EN}(\boldsymbol{w}, \alpha) = \alpha \|\boldsymbol{w}\|_1 + (1-\alpha)\|\boldsymbol{w}\|_2^2$$

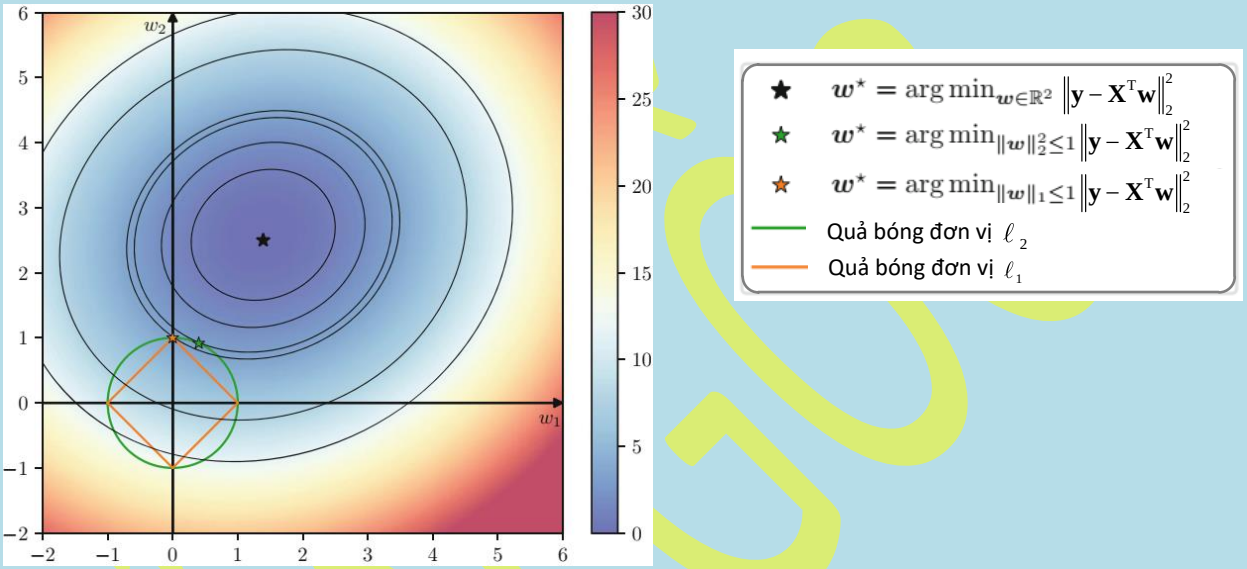
trong đó $\alpha \in [0,1]$ là một siêu thông số trình bày tỷ lệ phạt ℓ_1 so với phạt ℓ_2.

2.6.3. Các vấn đề tối ưu hóa mới và chính quy hóa

Một cách tiếp cận tự nhiên sẽ là bổ sung một ràng buộc đến bài toán cực tiểu hóa:

$$\min_{\boldsymbol{w}} \mathrm{L}(\boldsymbol{w}) \text{ phải tuân thủ } \mathrm{Pen}(\mathrm{w}) < c \qquad (2.1)$$

Có thể phát biểu là "Tìm các thông số cực tiểu hóa hàm chi phí trong số tất cả các thông số w mà chúng thỏa mãn $\mathrm{Pen}(\boldsymbol{w}) < c$" đối với một số thực dương *c*. Hình 2.8 minh họa lời giải tối ưu một nhiệm vụ hồi quy tuyến tính với các ràng buộc khác nhau.



Hình 2.8. Minh họa bài toán giảm thiểu với ràng buộc lên thành phần phạt. Đồ thị trình bày giá trị của hàm mất mát đối với các giá trị khác nhau của hai hệ số cho một nhiệm vụ hồi quy tuyến tính. Ngôi sao đen chỉ ra lời giải tối ưu khi không bị ràng buộc. Các ngôi sao xanh lá cây và vàng chi ra các lời giải tối ưu khi áp đặt ràng buộc lên các chuẩn ℓ_2 và ℓ_1 của các thông số *w* tương ứng.

Hình này cho thấy thuộc tính thưa của phạt ℓ_1(thông số tối ưu cho trục ngang được đặt bằng không) mà phạt ℓ_2 không có (thông số tối ưu cho trụng ngang nhỏ nhưng khác không) .

Mặc dù cách tiếp cận này hấp dẫn vì tính trực giácvà khả năng của nó thiết lập phạt cực đại có thể lên các thông số ***w*** dẫn đến một bài toán cực tiểu hóa khó giải quyết. Một cách tiếp cận tương tự bao gồm bổ sung thành phần chính quy hóa vào hàm chi phí:

$$\min_{\boldsymbol{w}} \mathrm{L}(\boldsymbol{w}) + \lambda \times \mathrm{Pen}(\boldsymbol{w}) \qquad (2.2)$$

Trong đó λ>0 là một siêu thông số điều khiển các trọng số của thành phần phạt so với các giá trị mất mát trung bình trên tất cả các mẫu đào tạo. Sự diễn đạt này liên quan đến hàm Lagriange của bài toán cực tiểu hóa với ràng buộc phạt.

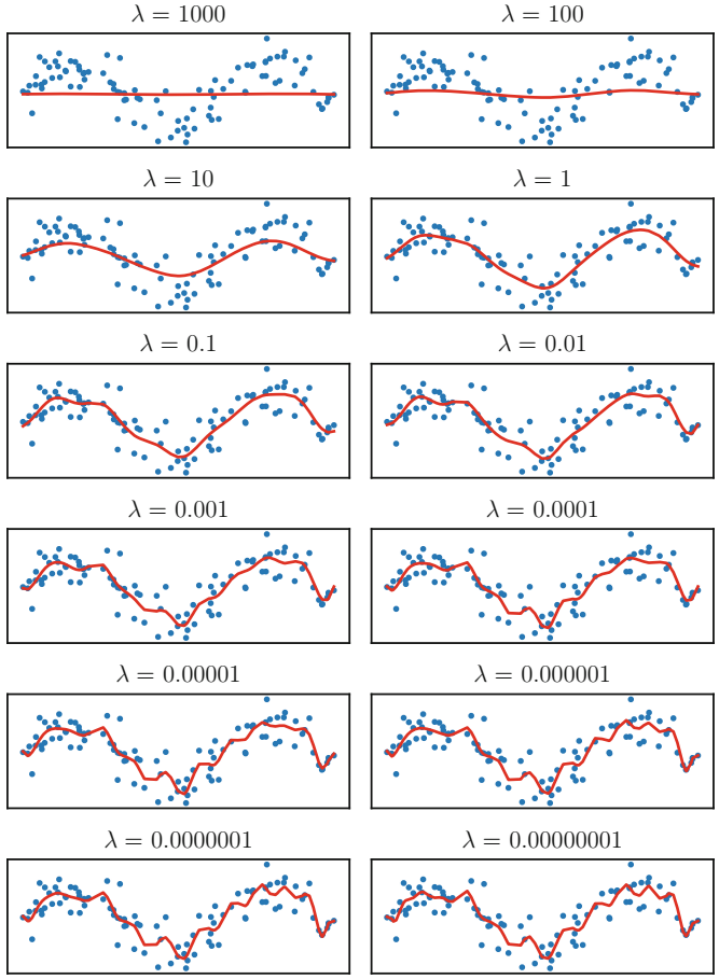
Sự diễn đạt này dẫn đến một bài toán cực tiểu hóa không bị ràng buộc dễ giải hơn nhiều. Thực sự , ta có thể chỉ ra rằng phương trình (2.1) và phương trình (2.2) liên quan với nhau: giải phương trình (2.2) với λ cho trước mà lời giải tối ưu của nó được ký hiệu là $\boldsymbol{w}_\lambda^*$, tương đương với giải phương trình (2.1) đối với $c = \text{Pen}\ \boldsymbol{w}_\lambda^*$. Nói một các khác, giải phương trình (2.2) với λ cho trước tương đương với giải phương trình (4.1) đối với c mà giá trị của nó chỉ biết được sau khi tìm được lời giải tối ưu của phương trình (4.2).

Hình 2.9 minh họa ảnh hưởng của thành phần chính quy hóa $\lambda \times \text{Pen}(\boldsymbol{w})$ lên hàm dự báo của một giải thuật hồi quy kenel ridge (đường gờ hat nhân) (xem phần 2.11 cho chi tiết hơn) đối với các giá trị λ khác nhau. Đối với các giá trị λ cao, thành phần chính quy hóa chiếm ưu thế giá trị mất mát trung bình dẫn đến hàm dự báo không khớp đủ tốt với dữ liệu đào tạo (Unfitting). Đối với các giá trị λ thấp, giá trị mất mát trung bình chiếm ưu thế thành phần chính quy hóa dẫn đế hàm dự báo khớp quá tốt với dữ liệu đào tạo (Overfitting) . Để học được hàm này tốt nhất, yêu cầu phải có được cân bằng tốt giữa giá trị mất mát trung bình và thành phần chính quy hóa.

Vì hồi quy tuyến tính là một mô hình lâu đời nhất và được biết tốt nhất, các phạt nói trên ban đầu được đưa ra cho hồi quy tuyến tính:

- Hồi quy tuyến tính với phạt ℓ_2 cũng được biết như là ridge (đường gờ):

$$\min_{\boldsymbol{w}} \|\mathrm{y} - \mathrm{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2$$



Hình 2.9. Minh họa chính quy hóa. Một giải thuật hồi quy kenel ridge được khớp với dữ liệu đào tạo (các điểm xanh) bằng các giá trị λ khác nhau, các giá trị này là các trọng số của chính quy hóa trong hàm chi phí. Các giá trị λ cảng nhỏ thì trọng số của chính quy hóa ℓ_2 càng nhỏ. Giải thuật không khớp (hoặc quá khớp) với dữ liệu đào tạo khi giá trị λ quá lớn (tương ứng, quá nhỏ).

2.7. MÁY VECTƠ HỖ TRỢ (SUPPORT VECTOR MACHINE)

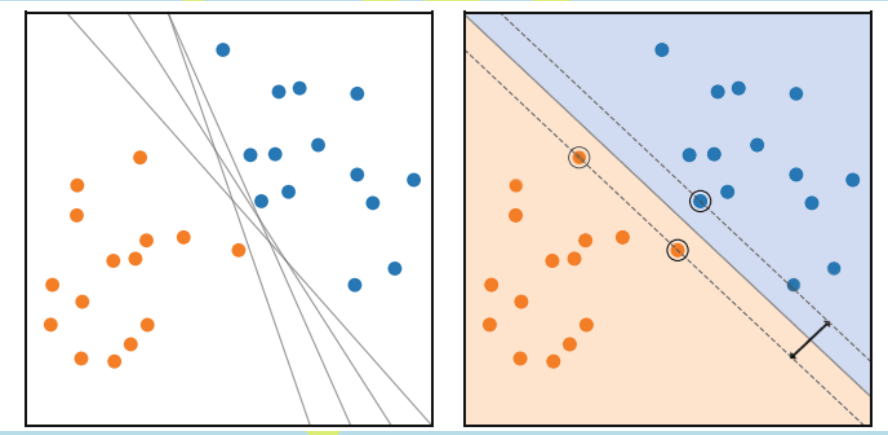
Hồi quy tuyến tính và logistic xét đến mọi mẫu đào tạo để tìm ra đường thẳng tốt nhất, điều này là do các hàm mất mát tương ứng của chúng: sai số bình phương bằng không chỉ khi các đầu ra đúng và dự báo bằng nhau và mất mát logistic luôn dương. Ta có thể tranh luận là các mẫu đào tạo có các đầu ra "dễ " dự báo tốt không có liên quan, chỉ các đầu ra không "dễ" dự báo tốt là có liên quan hay dự báo sai là cần xét đến. Máy vectơ hỗ trợ

(SVM: Support Vector Machine) dựa trên nguyên tắc này. Có thể tóm tắt nguyên lý SVM như sau:

Mục đích chính: Siêu mặt phẳng (chẳng hạn đường thẳng) cực đại hóa bên độ hay khoảng cách (Margin) (khỏang cách giữa siêu mặt phẳng và các đầu vào gần nhất tới siêu mặt phẳng).
Các vectơ hỗ trợ (Support Vector): Chỉ xét đến các đầu vào bị phân loại nhầm và các đầu vào được phân loại tốt nhưng với độ chắc chắn thấp.
Không tuyến tính: Hàm quyết định có thể không tuyến tính với sử dụng các kenel không tuyến tính.
Chính quy hóa: phạt ℓ_2**.**

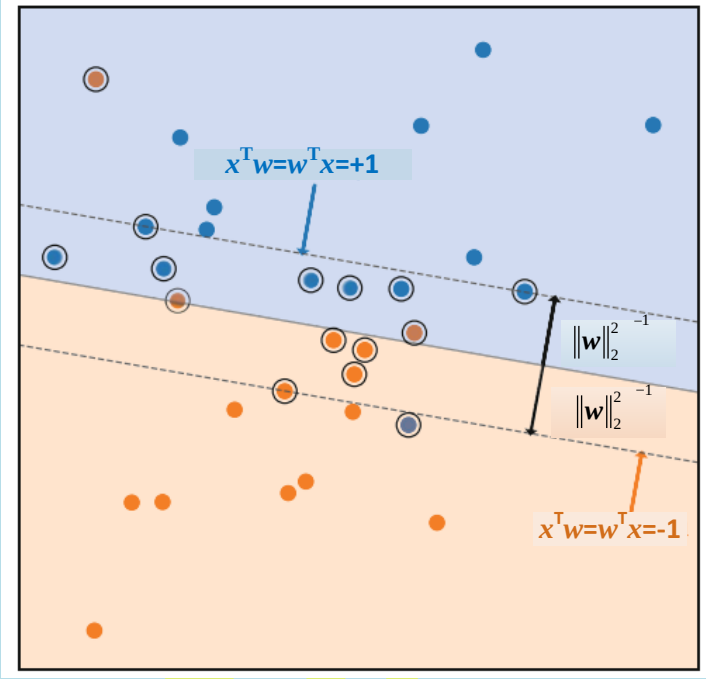
2.7.1. Trình bày gốc

Máy vectơ hỗ trợ gốc được phát minh năm 1963 và là một phương pháp phân loại nhị phân tuyến tính. Hình 2.10 minh họa khái niệm chính của phiên bản gốc này. Khi cả hai loại có thể phân tách tuyến tính, tồn tại môt số lượng vô hạn các siêu mặt phẳng phân tách cả hai loại này. SVM tìm ra siêu mặt phẳng cực đại hóa biên độ (Margin), nghĩa là, khoảng cách giữa siêu mặt phẳng này và các điểm gần nhất của cả hai loại đến siêu mặt phẳng này.



Hình 2.10.Bộ phân loại máy vectơ hỗ trợ (Support Vector Machine Classifier) với hai loại khả phân tách. Khi cả hại loại khả phân tách tuyến tính, tồn tại một số lượng vô hạn các siêu mặt phẳng phân tách chúng (hình trái). Hàm quyết định của bộ phân loại máy vectơ hỗ trợ là siêu mặt phẳng cực đại hóa khoảng cách (Margin), nghĩa là, khoảng cách giữa siêu mặt phẳng và các điểm gần nhất đến siêu mặt phẳng này (hình phải). Các vectơ hỗ trợ được làm nổi bật bằng các đường tròn đen bao quanh chúng

Sau đó SVM được cập nhật đến các loại không khả phân tách (non-Saparable Class). Hình 2.11 minh họa vai trò của biên độ (Margin) trong trường hợp này. Các đường đứt nét tương ứng với các siêu mặt phẳng được định nghĩa bởi các phương trình $\boldsymbol{x}^T\boldsymbol{w}= \boldsymbol{w}^T\boldsymbol{x}= +1$ và $\boldsymbol{x}^T\boldsymbol{w}= \boldsymbol{w}^T\boldsymbol{x}= =-1$. Biên độ (Margin) là khoảng cách giữa cả hai siêu mặt phẳng và bằng $2/\|\boldsymbol{w}\|_2^2$. Nó định nghĩa các mẫu được bao gồm trong hàm quyết định của mô hình: một mẫu được bao hàm nếu và chỉ nếu nó nằm trong biên độ hay ngoài biên độ và bị phân loại nhầm. Các mẫu này được gọi là các vectơ hỗ trợ (Support Vector) và được minh họa trên hình 2.11 với các đường tròn đen bao quanh chúng. Trong trường hợp này có thể coi biên độ là một thành phần chính quy hóa (Regularization Term): biên độ càng lớn thì vectơ hỗ trợ càng được bao hàm trong hàm quyết định, mô hình càng được chính quy hóa hơn.

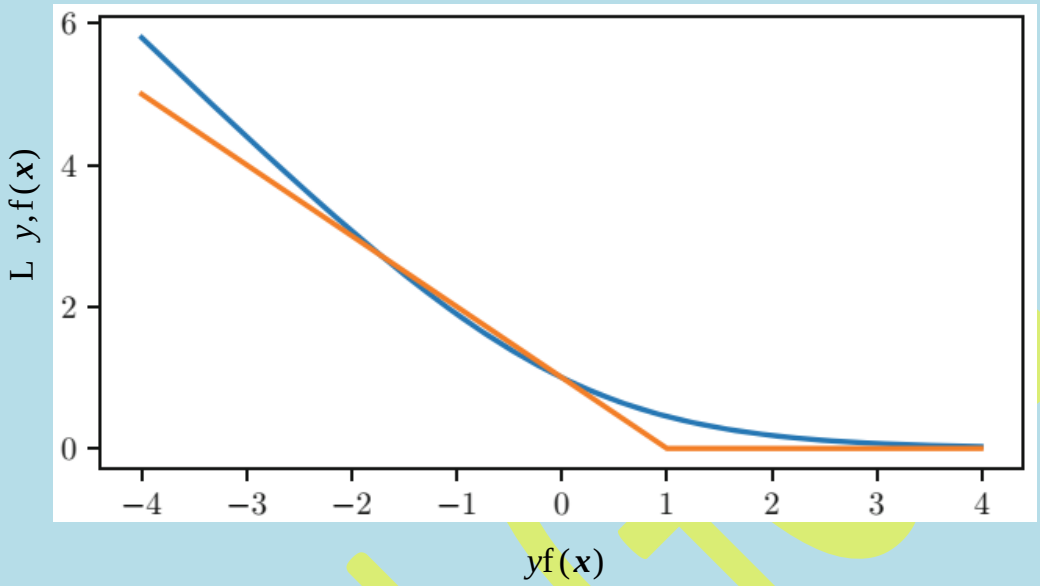


Hình 2.11. Hàm quyết định của bộ phân loại máy vectơ hỗ trợ với một kenel tuyến tính khi cả hai loại không khả phân tách tuyến tính chặt chẽ. Các vectơ hỗ trợ là các điểm đào tạo nằm bên trong biên độ (khoảng cách) của hàm quyết định và là các điểm đào tạo bị phân loại lỗi. Các vectơ hỗ trợ được làm nổi bật bằng các đường tròn đen bao quanh chúng

Hàm mất mát cho SVM được gọi là mất mát bản lề (Hinge Loss) và được định nghĩa như sau:

$$L_{hinge}\ y, f(x)\ = \max\ 0, 1 - f(\boldsymbol{x})$$

Hình 2.12 minh họa các đường cong của các mất mát logistic và hinge. Mất mát logistic luôn luôn dương ngay cả khi điểm được phân loại chính xác với độ chắc chắn cao (nghĩa là, khi $yf(\boldsymbol{x})>>0$). Ta có thể thấy rằng một mẫu $(\boldsymbol{x},y)$ là một vectơ hỗ trợ khi và chỉ khi $yf(\boldsymbol{x}) \geq 1$, nghĩa là, khi và chỉ khi $L_{\text{hinge}}(y, f(\boldsymbol{x})) = 0$.



Hình 2.12. Các mất mát nhị phân. Mất mát logistic luôn luôn dương ngay cả khi điểm được phân loại chính xác với độ chắc chắn cao (nghĩa là, khi $yf(\boldsymbol{x})>>0$), trong khi đó mất mát hinge bằng không khi điểm được phân loại chính xác với độ chắc chắn cao (nghĩa là, khi $yf(\boldsymbol{x})>>1$)

Các hệ số $\boldsymbol{w}$ tối ưu cho phiên bản gốc đầu tiên được ước tính bằng cách cực tiểu hóa hàm đối tượng bao gồm tổng của các giá trị mất mát hige và một thành phần phạt ℓ_2 (là nghịch đảo của lề):

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} \max\left(0, 1 - y_i \boldsymbol{x}_i^T \boldsymbol{w}\right) + \frac{1}{2C}\|\boldsymbol{w}\|_2^2$$

2.7.2. Phát biểu tổng quát với các kenel

SVM sau đó đã được cập nhật cho các hàm quyết định không tuyến tính với việc sử dụng các kenel.

Để có một hàm không tuyến tính, ta phải ánh xạ không gian đầu vào *I* vào một không gian khác (thường được gọi là không gian tính năng (Feature Space)), được ký hiệu là *G* với sử dụng một hàm được ký hiệu là ϕ:

$$\begin{aligned}\phi &: I \to G \\ \boldsymbol{x} &\to \phi(\boldsymbol{x})\end{aligned}$$

Hàm quyết định sẽ phải vẫn tuyến tính (với tích vô hướng), nhưng trong không gian tính năng:

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$$

Rất tiếc, giải bài toán cực tiểu hóa tương ứng không dễ ràng:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \max\left[0, 1 - \boldsymbol{y}_i \phi(\boldsymbol{x}_i)^T \boldsymbol{w}\right] + \frac{1}{2C}\|\boldsymbol{w}\|_2^2 \quad (2.3)$$

Tuy nhiên, hai thuộc tính toán học giúp cho việc sử dụng các chuyển đổi tphi tuyến tính trong không gian tính năng khả thi: kỹ thuật kenel (Kenel Trick) và định lý biểu diễn (Representer Theorem).

Kỹ thuật kenel khẳng định rằng có thể tính được tích vô hướng trong không gian tính năng với sử dụng chỉ các điểm từ không gian đầu vào và một hàm kenel (Kenel Function), được ký hiệu là K:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in I, \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}') = K(\boldsymbol{x}, \boldsymbol{x}')$$

Định lý biểu diễn khẳng định rằng trong một số điều kiện nhất định của kenel *K* và không gian *G* được liên kết với hàm ϕ, mọi bộ giảm thiểu của phương trình (2.3) cho phép chấp nhận dạng sau:

$$f = \sum_{i=1}^{N} \alpha_i K(\cdot, \boldsymbol{x}_i)$$

Trong đó $\boldsymbol{\alpha}$ được giải từ:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \max\left[0, 1 - y_i (\boldsymbol{K}\boldsymbol{\alpha})_i\right] + \frac{1}{2C}\boldsymbol{\alpha}^T \boldsymbol{K}\boldsymbol{\alpha}$$

Trong đó ***K*** là một ma trận kích thước $N \times N$ bao gồm các ước tính kenel trên tất cả các cặp của các mẫu đào tạo: $\forall i, j \in 1,\ldots,N \ , K_{ij} = K\ x_i, x_j$.

Do mất mát hinhge bằng không nếu và chỉ nếu $yf(\boldsymbol{x})$ lớn hơn hoặc bằng 1, chỉ khi các mẫu đào tạo $\boldsymbol{x}_i, y_i$ có một giá trị hệ số α_i khác không. Các điểm này được gọi là các vectơ hỗ trợ và đây là lý do chúng chỉ là các mẫu đóng góp cho hàm quyết định của mô hình:

$$SV = \{ i \in 1,\cdots,N \mid \alpha_i \neq 0 \}$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K\ \boldsymbol{x}, x_i = \sum_{i \in SV} \alpha_i K\ \boldsymbol{x}, \boldsymbol{x}_i$$

Kỹ thuật kenel và lý thuyết trình diễn cho thấy rằng làm việc với kenel K thay vì ánh xạ hàm ϕ thực tiễn hơn . Các hàm kenel phổ biến bao gồm:

- Kenel tuyến tính (Linear Kenel):
  $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
- Kenel đa thức (Polymomial Kenel):
  $K\ \boldsymbol{x}, \boldsymbol{x}' = \gamma \boldsymbol{x}^T \boldsymbol{x}' + c_0^{\ d}$ với $\gamma > 0, c_0 \geq 0, d \in \mathbb{N}^*$
- Kenel sigmoid (Sigmoid Kenel):
  $K\ \boldsymbol{x}, \boldsymbol{x}' = \tanh\ \gamma \boldsymbol{x}^T \boldsymbol{x}' + c_0$ với $\gamma > 0, c_0 \geq 0$
- Hàm cơ sở xuyên tâm (RBF: Radial Basis Function):
  $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\ -\gamma \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|_2^2$ với $\gamma > 0$

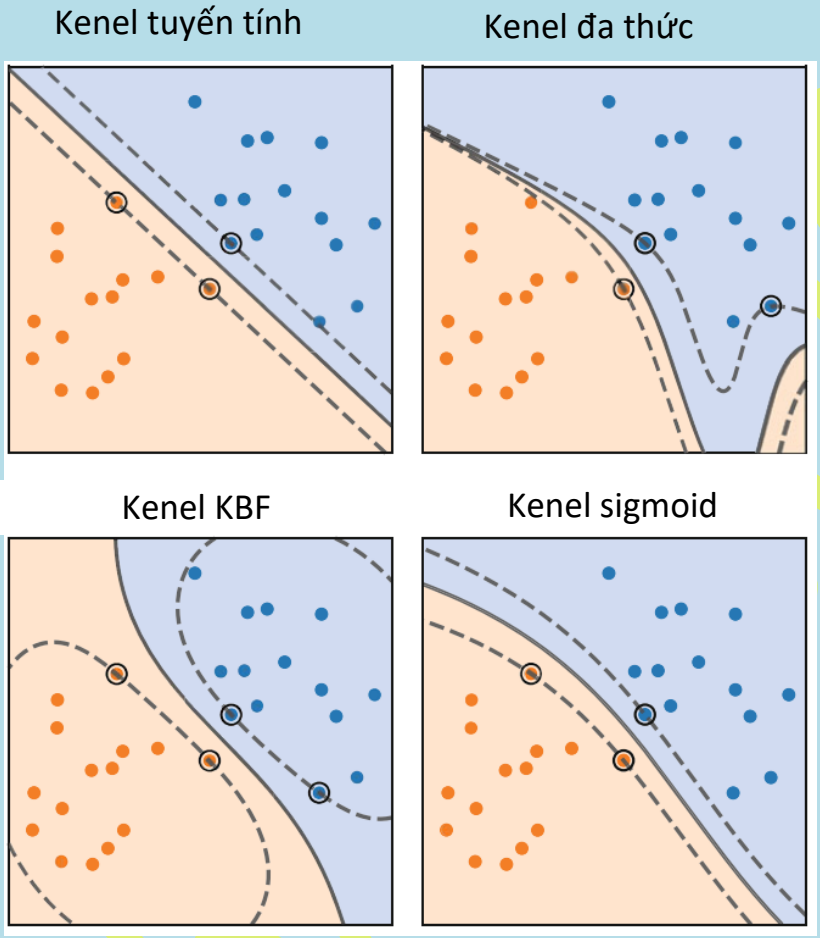
Kenel tuyến tính tạo ra một hàm quyết định tuyến tính và thực chất giống như trình bày gốc của SVM (ta có thể chỉ ra rằng có ánh xạ giữa α và các hệ số của *w*). Điều này đặc biệt hữu ích khi dữ liệu không khả phân tách tuyến tính, đây là trường hợp phổ biến nhất. Hình 2.13 minh họa hàm quyết định và biên độ (Margin) của mô hình phân loại đối với bốn kenel khác nhau.

Cũng có thể mở rộng các nhiệm vụ hồi quy bằng việc sử dụng mất mát không nhạy cảm ε (ε-insensitive Loss). Giống như mất mát hinge bằng không đối với các điểm được phân

BẢN GỐC

loại đúng và nằm ngoài lề, mất mát ε-insensttive bằng không khi sai số giữa giá trị đích và giá trị dự báo không lớn hơn ε:

$$L_{\varepsilon-\text{insensitive}}(\boldsymbol{y}, f(\boldsymbol{x})) = \max\ 0, |\boldsymbol{y} - f(\boldsymbol{x})| - \varepsilon$$

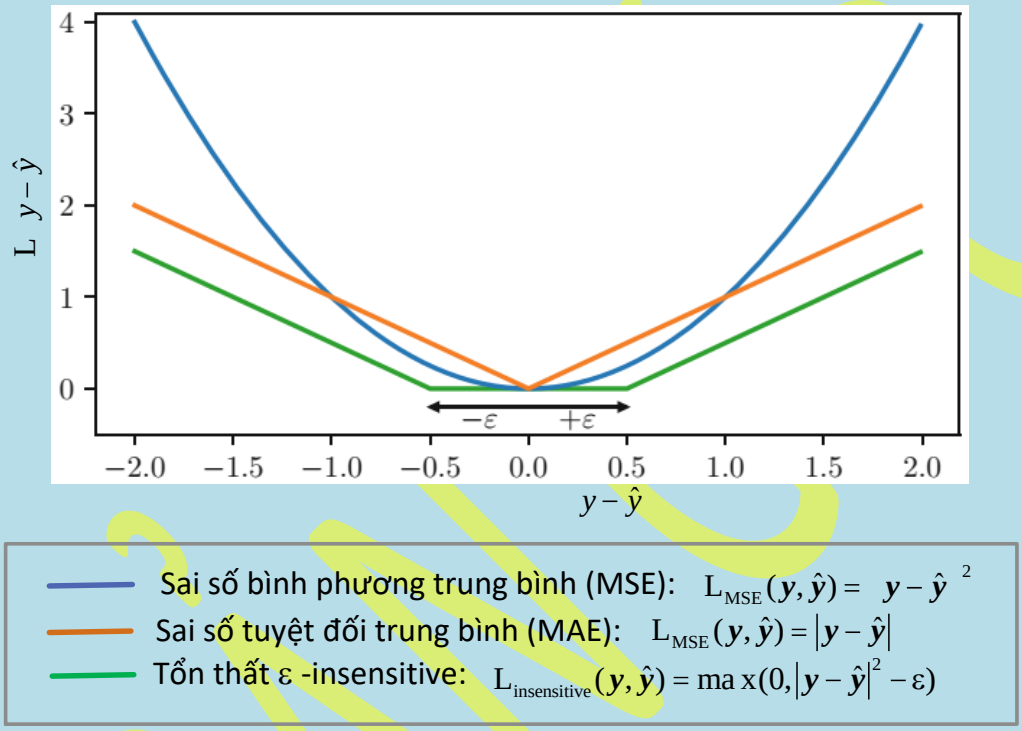


Hình 2.13. Ảnh hưởng của kenel lên hàm quyết định của bộ phân loại máy vecơ hỗ trợ. Kenel không tuyến tính cho phép hàm quyết định không tuyến tính

Hàm mục tiêu cho các phương pháp hồi quy SVM kết hợp các giá trị của mất mát ε-insenssitive của các điểm đào tạo và phạt ℓ_2:

$$\max_{w} \sum_{i=1}^{n} \max\ 0, \left| y_i - \phi(\boldsymbol{x}_i)^T - \varepsilon \right| + \frac{1}{2C} \|\boldsymbol{w}\|_2^2$$

Hình 2.14 minh họa các đường cong của các mất mát hồi quy. Mất mát sai số bình phương nhận các giá trị rất nhỏ đối với các sai số thấp và các giá trị rất cao đối với các sai số cao, trong khi đó mất mát sai số tuyệt đối nhận các giá trị thấp nhưng khác không khi sai số nhỏ. Cả hai mất mát đều nhận các giá trị thấp khác không khi sai số thấp. Trái lại, mất mát ε-insenssitive bằng không khi sai số thấp và nếu không thì bằng mất mát sai số thuyệt đối trừ ε.



Hình 2.14.Các mất mát hồi quy. Mất mát sai số bình phương nhận các giá trị rất nhỏ đối với các sai số nhỏ và các giá trị rất lớn đối với các sai số lớn, trong khi đó mất mát tuyệt đối nhận các giá trị rất nhỏ khi các sai số nhỏ và các giá trị lớn khi các sai số lớn. Cả hai mất mát đều nhận các các giá trị nhỏ nhưng khác không khi sai số nhỏ. Trái lại, mất mát ε-insensitive bằng không khi sai số nhỏ và nếu không thì bằng mất mát tuyệt đối trừ ε. Khi tính toán trên một số mẫu , các mất mát sai số bình phương và tuyệt đối thường được đề cập như là sai số bình phương trung bình (MSE: Mean Squared Error) và sai số tuyệt đối trung bình (Mean Absolute Error)

2.8. PHÂN LOẠI NHỊ PHÂN (BINARY CLASSFICATION) BẰNG BỘ DỰ ĐOÁN TUYẾN TÍNH

2.8.1 Tổng quan

Công thức chung cho các bộ phân loại nhi phân tuyến tính được thể hiện như sau:

$$\hat{y} = \underbrace{\text{argmax}}_{y} \text{f} \quad y; \boldsymbol{w}, \boldsymbol{x}$$

Trong đó: $\text{x} \in \mathbb{R}^{\text{K}}$, $\boldsymbol{w} \in \mathbb{R}^{\text{K}}$, $\text{y} = +1, -1$ và $\hat{y}$ là ước tính đầu ra củabộ phân loại.

Hai bài toán được thực hiện với bộ phân loại:

- Hoc/đào tạo (Learning/ Training): tìm $\boldsymbol{w}$ tốt nhất để thỏa mãn phương trình trên khi cho trước $\boldsymbol{x}$ và y;
- Suy luận/kiểm tra (Inference/ Test): cho trước $\boldsymbol{w}$ và $\boldsymbol{x}$, giải argmax

Ta nhắm đến tìm một mô hình học **H** sao cho $\mathbf{H}(\boldsymbol{x}_i)=y_i$ đối với các mẫu chưa được biết. Vấn đề này được biểu diễn đơn giản vào hai trường hợp loại, trong đó các nhãn y_i chỉ là chỉ có hai kết cục chẳng hạn +1 hoặc -1 cho hai trường hợp liên quan.

Các bộ phân loại nhị phân tuyến tính bao gồm:

- Bộ phân loại dựa trên mô hình perceptron
- Bộ phân loại dựa trên máy vectơ hỗ trợ
- Bộ phân loại dựa trên hồi quy logistic

Để hiểu được nguyên lý của các bộ phân loại nhị phân, trong phần dưới đây ta sẽ xét chi tiết mô hình phân loại nhị phân dựa trên mô hình perceptron (cụ thể về perceptron xem chương 3).

2.8.2. Mô hình phân loại nhị phân cho perceptron

Ta xét mô hình phân loại nhị phân như sau:

- Mẫu đầu vào $\boldsymbol{x}_i$ có kích thước $\text{K} \times 1$
- Mẫu đầu ra nhị phân $y_i = +1, -1$ hay $y_i = 1, 0$
- $\text{i} = \{1, \ldots, \text{N}\}$
- Loại giả định $\text{f} = \boldsymbol{x}_i \rightarrow \text{sign} \ \boldsymbol{w}^{\text{T}} \boldsymbol{x}_i + \text{b}$,

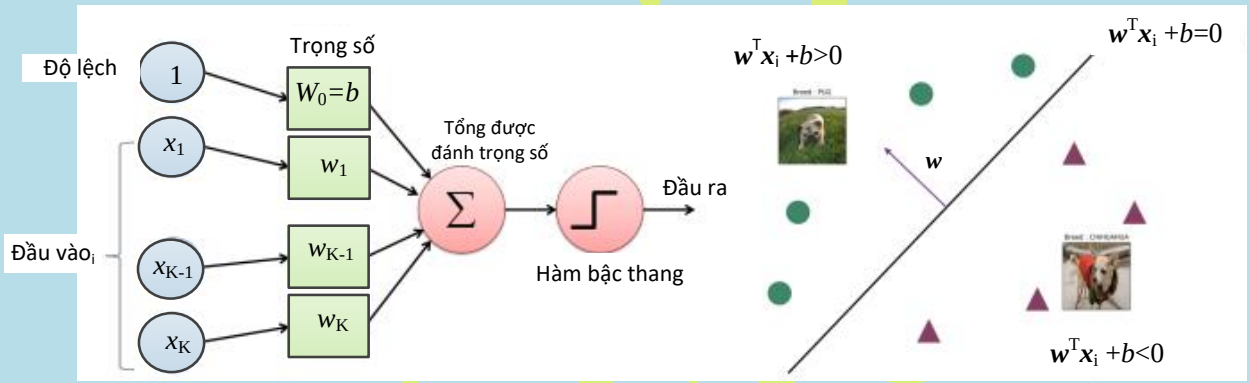
  trong đó $\boldsymbol{w}$ và $\boldsymbol{x}_i$ có kích thước $\text{K} \times 1$ và $\text{sign(a)} = \begin{matrix} +1 & \text{khi a} \geq 0 \\ -1 & \text{khi a} < 0 \end{matrix}$

BẢN GỐC

- Hàm mất mát: $L_i = \begin{cases} 0 & \text{khi } \hat{y} = y_i \\ 1 & \text{khi } \hat{y} \neq y_i \end{cases}$

  Lưu ý: điều này có nghĩa là mất mát 1 sẽ cho ta dự báo sai và mất mát 0 sẽ cho ta dự báo đúng

Hình 2.15 cho thấy bộ phân loại nhị phân chia không gian thành hai bán không gian cho mô hình perceptron (xem chương sau).



Hình 2.15. Bộ phân loại nhị phân chia không gian thành hai bán không gian. (trái) mô hình perceptron. (phải) Trực quan hóa nửa không gian 2D. Lưu ý rằng *w* vuông góc với siêu mặt phẳng. Các điểm phía trên đường thẳng được gán nhãn dương còn các điểm phía dưới đường thẳng được gán nhãn âm

Để đơn giản biểu diễn công thức ta có thể kết hợp độ lệch b và đầu (hoặc cuối) vectơ trọng số $\boldsymbol{w}$ và chèn 1 vào đầu (hoặc cuối) vectơ tính năng đầu vào $\boldsymbol{x}_i$:

$$\boldsymbol{x}_i \rightarrow \begin{bmatrix} 1 & x_1 & \dots & x_{K-1} & x_K \end{bmatrix}^T, \boldsymbol{w} = \begin{bmatrix} b & w_1 & \dots & w_{K-1} & w_K \end{bmatrix}^T \quad (2.4)$$

Hay

$$\boldsymbol{x}_i \rightarrow \begin{bmatrix} x_1 & \dots & x_{K-1} & x_K & 1 \end{bmatrix}^T, \boldsymbol{w} = \begin{bmatrix} w_1 & \dots & w_{K-1} & w_K & b \end{bmatrix}^T \quad (2.5)$$

Khi này:

$$\boldsymbol{w}^T \boldsymbol{x}_i + b \rightarrow \boldsymbol{w}^T \boldsymbol{x}_i = \boldsymbol{x}_i^T \boldsymbol{w}$$

2.8.3. Giải thuật cho bài toán tối ưu cho trọng số *w*

Gia sử được cho một tập đào tạo $S= \{(\boldsymbol{x}_1,y_1), \ldots, (\boldsymbol{x}_N,y_N)\}$, Các hệ số $\boldsymbol{w}$ tối ưu cho trọng số $\boldsymbol{w}$ được tìm bằng các tối thiểu hóa như sau:

$$\boldsymbol{w}* = \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{i=1}^{N} 1\left[\text{sign}\ \ \boldsymbol{w}^T\boldsymbol{x}_i \ \ \neq y_i\right]$$

Dưới đây ta sẽ xét giải thuật cho bài toán nói trên.

Giải thuật
Initialize $\boldsymbol{w}_1 = 0 \in \mathbb{R}^K$ For t=1,2….. do if $i \in 1,\ldots,N$, $y_i \neq \text{sign}\ \ \boldsymbol{w}^T\boldsymbol{x}_i$ then update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \lambda y_i \boldsymbol{x}_i$ else ouput $\boldsymbol{w}_t$ end

Giả sử $\boldsymbol{x}_i$ bị phân loại nhầm và giả thiết $\|\boldsymbol{x}_i\|^2 = 1$, từ giải thuật ta có cập nhập như sau:

- Nếu nhãn thực là dương, thì $\boldsymbol{w}_{t+1}^T\boldsymbol{x}_i = \boldsymbol{w}_t^T\boldsymbol{x}_i + \lambda y_i \|\boldsymbol{x}_i\|^2 = \boldsymbol{w}_t^T\boldsymbol{x}_i + \lambda.1$
- Nếu nhãn thực là âm, thì $\boldsymbol{w}_{t+1}^T\boldsymbol{x}_i = \boldsymbol{w}_t^T\boldsymbol{x}_i + y_i \|\boldsymbol{x}_i\|^2 = \boldsymbol{w}_t^T\boldsymbol{x}_i - \lambda 1$

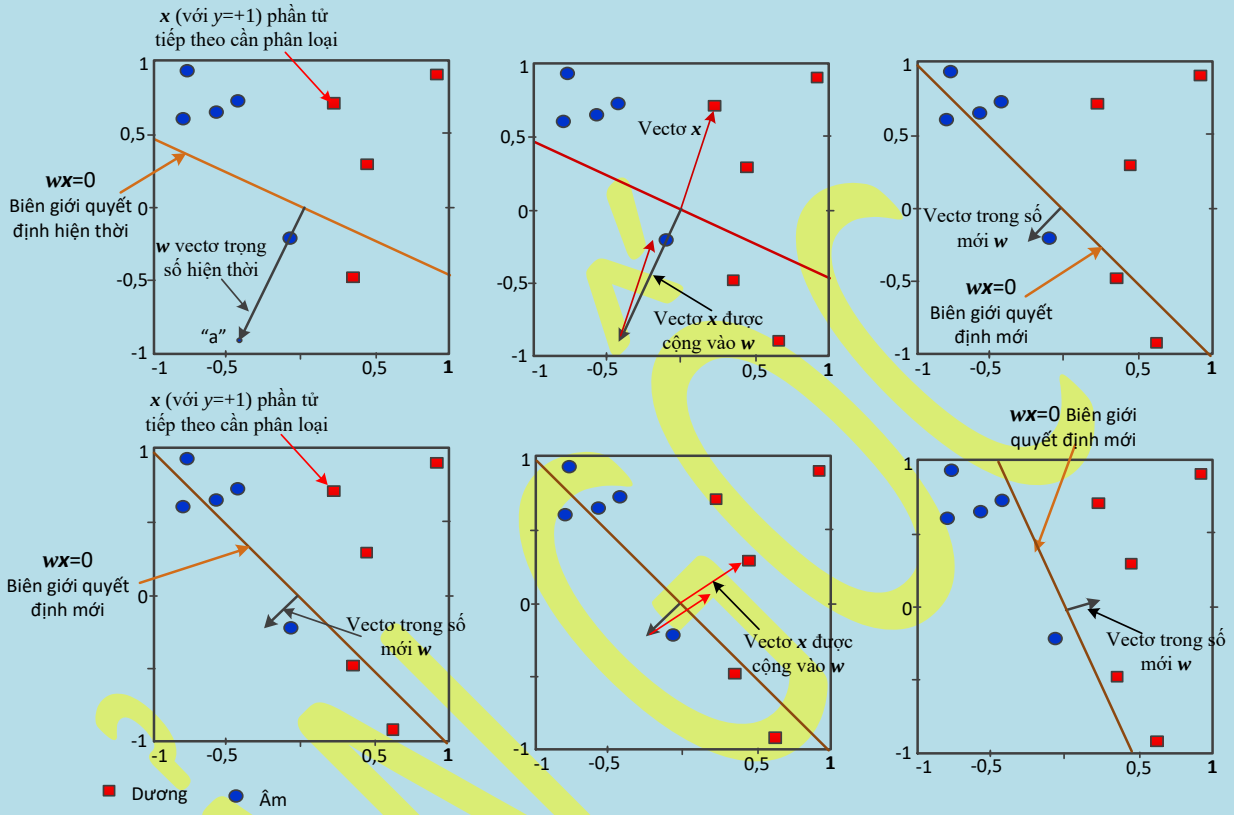
2.8.4. Không gian dữ liệu

Không gian đầu tiên ta cần làm quen là không gian dữ liệu hay không gian đầu vào. Mỗi điểm trong không gian này tương ứng với một vectơ có thể có (ta sẽ sử dụng thuật ngữ "điểm" và "vectơ" đổi lẫn). Ta sẽ trình bày các thí dụ âm và dương bằng các ký hiệu "-" và "+".

Sau khi chọn song các trọng số $\boldsymbol{w}$ và độ lệch b ta có thể chia các điểm được phân loại thành các vùng dương và âm. Biên giới giữa các vùng này là tập tại đó $\boldsymbol{w}^T\boldsymbol{x}+b=0$, được gọi là biên giới quyết định và tập được xác định bởi phương trình này được gọi là siêu mặt phẳng (Hyperplane). Tập các điểm tại một bên của biên giới này được gọi là bán không gian (Halfspace). Điều này không có nghĩa rằng cập nhật này tốt cho các thí dụ khác. Dựa trên trực giác ta hãy thử phân tích điều này. Để đơn gian ta giả thiết dữ liệu nằm trong một quả bóng đơn vị, sao cho $\|\boldsymbol{x}\|^2 \leq 1$. Ta cũng giả thiết là $\|\boldsymbol{w}*\|^2 = 1$.

Trong cả hai trường hợp, cập nhật di chuyển trong lượng theo đúng hướng và λ định nghĩa tốc độ cập nhật. Hình 2.16 diễn giải trực quan về điều này và để đơn giản ta giả

thiết dữ liệu nằm trong một quả bóng đơn vị, sao cho $\|\boldsymbol{x}\|^2 \leq 1$. Ta cũng giả thiết là $\|\boldsymbol{w}*\|^2 = 1$ và λ=1.



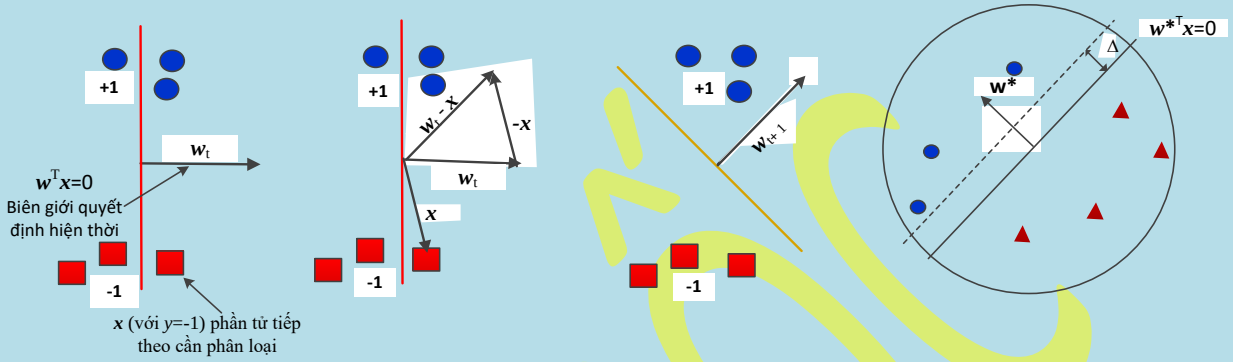
Hình 2.16. Minh họa quá trình cập nhật vectơ trọng số

Giải thích hình 4.16 như sau.Tại hình đầu tiên ngoài cùng bên trái, vecttơ trọng số ***w*** hiện thời tương ứng với điểm "a" trong không gian dữ liệu và siêu mặt phẳng tương ứng là mặt phẳng vuông góc với mà theo đó tích vô hướng của mọi vectơ ***x*** tương ứng với mọi điểm trong trong mặt phẳng với vectơ trọng số ***w*** này đều bằng không: $\boldsymbol{w}^T\boldsymbol{x}=0$. Tại hình liền kề bên phải, do nhãn của điểm phân loại y=+1, nên cập nhật thực hiện trừ vectơ trọng số ***w*** hiện thời với vectơ ***x*** của điểm cần phân loại để được vectơ trọng số ***w*** mới với kết quả được thể hiện trong hình thứ ba ngoài cùng bên phải. Tương ứng với vectơ trọng số mới ***w*** mới là siêu mặt phẳng vuông góc với nó như được thể hiện trong hình đầu tiên của hàng thứ hai. Liền kề với hình này là hình thể hiện cập nhật tiếp theo thực hiện trừ vectơ ***x*** của điểm cần phân loại với vectơ trọng số ***w*** mới và kết quả được thể hiện tại hình ngoài cùng bên phải của hàng thứ hai.

Cuối cùng ta định nghĩa một đại lượng biên độ (Margin) (hình 2.17). Biên độ Δ là khoảng cách nhỏ nất từ một điểm bất kỳ đến siêu mặt phẳng. Theo cách tính toán trước

đây, ta tính được khoảng các từ một điểm đến siêu mặt phẳng phân cách $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{i}} = 0$. Do siêu mặt phẳng phân cách $\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}_{\mathrm{i}} = 0$, nên:

$$\Delta = \min_{\mathrm{i}\in 1,2,\ldots,\mathrm{N}} \left|\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}_{\mathrm{i}}\right|$$



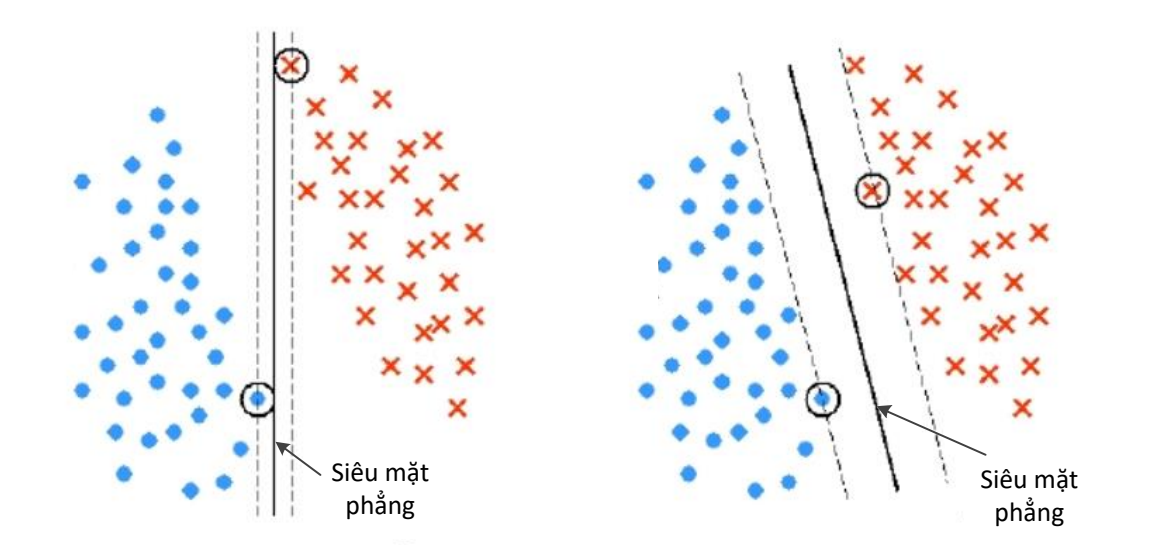
Hình 2.17. (trái) cập nhật của giải thuật sử dụng một thí dụ bị đánh nhãn nhầm. (phải) biên độ (Margin) được định nghĩa như là khoảng cách từ siêu mặt phẳng đến thí dụ gần nhất.

Giải thích hình 2.17 như sau. Ba hình bên phải minh họa quá trình cập nhật vectơ trọng số tại thời điểm $\boldsymbol{w}_t$ cho trường hợp phần tử cần được phân loại có nhãn y=-1, trong trường hợp này vectơ trọng số tại thời điểm t+1 được trừ với vectơ $\boldsymbol{x}$ của phần tử cần phân loại. Hình ngoài cùng bên phải minh họa sử dụng biên độ Δ cho phân loại.

Giải thuật cập nhật $\boldsymbol{w}$ cho trường hợp sử dụng biên độ Δ cho phân loại như sau.

Giải thuật
Initialize $\boldsymbol{w}_1 = 0 \in \mathbb{R}^{\mathrm{K}}$
For t=1,2….. do
if $\mathrm{i} \in 1,\ldots,\mathrm{N}$, $y_{\mathrm{i}} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{i}} < \Delta$ then update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \lambda y_{\mathrm{i}}\boldsymbol{x}_{\mathrm{i}}$
else ouput $\boldsymbol{w}_t$
end

Hình 2.18 minh họa cập nhật vectơ trọng số $\boldsymbol{w}$ với sử dụng biên độ Δ cho hai trường hợp có siêu mặt phẳng khác nhau.



Hình 2.18. Minh họa sử dụng biên độ Δ cho hai trường hợp có siêu mặt phẳng phân tách khác nhau

2.9. PHÂN LOẠI NHIỀU LOẠI

2.9.1. Tổng quan

Giải thuật phân loại nhiều loại (Multiclasss Classification) có giám sát nhắm đến gán một nhãn cho từng thí dụ đầu vào. Khi cho trước một tập dữ liệu đào tạo (x_i,y_i) trong đó $x_i \in \mathbb{R}^K$ là mẫu thứ i là một vectơ kích thước là $K\times 1$ và $y_i \in \{C_1,\ldots, C_M\}$với $C_m=1$, i=1,….,N và M là các loại khác nhau. Đối phân loại nhị phân $y_i\in\{-1.+1\}$ va đối với phân loại ảnh $y_i \in\{1,\ldots, M\}$. Đối với ảnh đầu vào ba mẫu RGB với chiều cao H, chiều rộng W ta có kích thước ảnh là :$H\times W\times 3$. Chẳng hạn trong CIFAR-10 ta có kích thước ảnh đầu vào $32\times 32\times 3$ sau khi làm phẳng (Flatten) mẫu đầu vào $\mathbf{x}_i$ có kích thước $32\times 32\times 3$ =3720 và tập ảnh mẫu cho huấn luyện là N 50000 và M=10.

Tóm lại:

- Phân loại nhị phân:
  $y_i \in -1,1$
- Phân loại nhiều loại:

Các bộ phân loại nhị phân tuyến tính bao gồm:

- Bộ phân loại dựa trên mô hình perceptron
- Bộ phân loại dựa trên máy vectơ hỗ trợ
- Bộ phân loại dựa trên hồi quy logistic

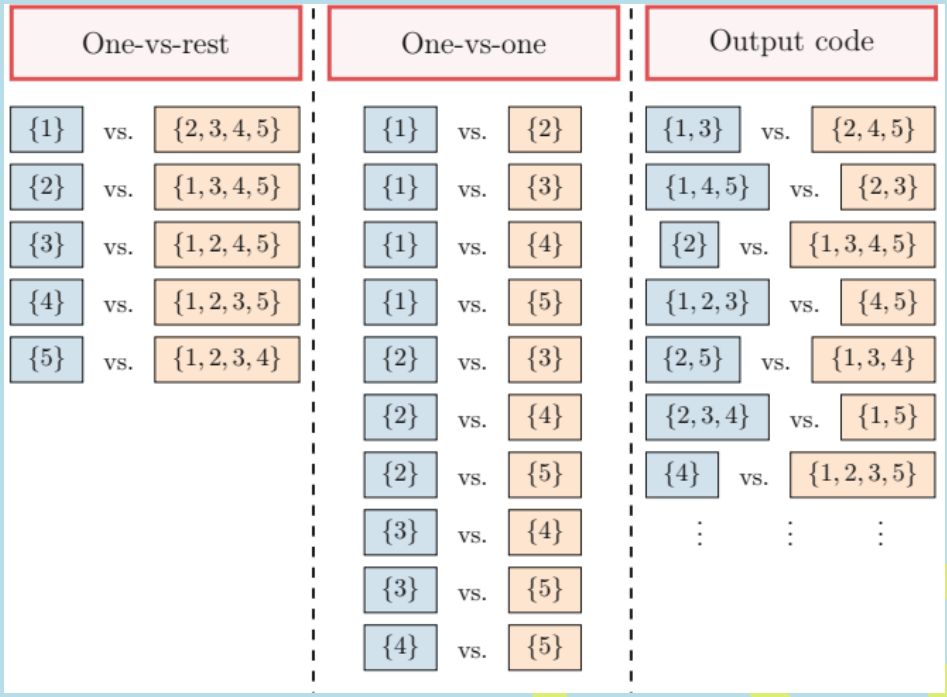
$y_i \in \{C_1, \ldots, C_M\}$ hay

$y_i = 1, 2, 3, \ldots, M$ với $C_m=1$

Các bộ phân loại đã được xét trong các phần trước bao gồm: perceptron, SVM (Support Vector Machine: máy veectơ hỗ trợ), và hồi qui logistics.

Các phương pháp phân loại được trình bày trước đây, perceptron, hồi quy logistic và các máy vectơ hỗ trợ, là các bộ phân loại nhị phân: chúng chỉ có thể sử dụng được khi chỉ có thể xẩy ra hai kết cục. Tuy nhiên trong thực tế, thường có thể xẩy ra nhiều hơn hai kết cục. Môt số chiến lược đã được đề xuất để mở rộng phương pháp phân loại nhị phân đến các nhiệm vụ phân loại nhiều loại. Trong phần này ta sẽ xét các chiến lược được sử dụng phổ biến nhất: một đối chọi phần còn lại (One-vs-Rest) một đối một (One vs One) và mã đầu ra sửa lỗi (Error Correcting Output Code). Hình 2.19 minh họa các ý tưởng chính của các cách tiếp cận này. Nhưng trước tiên, ta sẽ xét một mở rộng tự nhiên của hồi quy logistic đến các nhiệm vụ phân loại nhiều loại, thường được đề cập như là hồi quy logistic đa thức (Multinomial Logistic Regression).

BẢN NHÁP



One-vs-Rest: một đối chọi phần còn lại, One vs One: một đối một, Output Code: mã đầu ra.

Hình 2.19. Một số cách tiếp cận chính để chuyển một nhiệm vụ phân loại nhiều loại vào một số nhiệm vụ phân loại nhị phân. Trong các tiếp cận một đối với phần còn lại, mỗi loại được liên kết với một mô hình phân loại nhị phân được đào tạo để phân tách loại này với các loại khác. Trong các tiếp cận một đối một, một bộ phân loại nhị phân được đào tạo trên từng cặp của các loại. Trong cách tiếp cận mã sửa lỗi đầu ra, các loại được phân chia (ngẫu nhiên) vào hai nhóm và một bộ phân loại nhị phân được đào tạo cho từng phân chia

2.9.2. Hồi quy logistic đa thức (Multinominal Logistic Regression)

Đối với phân loại nhị phân, hồi quy logistic được đặc trưng bởi một siêu mặt phẳng: khoảng cách có dấu (Signed Distance) đến siêu mặt phẳng này được ánh xạ vào xác suất thuộc về loại dương với sử dụng hàm sigmoid. Tuy nhiên đối với phân loại nhiều loại, một siêu mặt phẳng đơn không đủ để đặc trưng tất cả các lớp. Thay vào đó, mỗi loại C_m được đặc trưng bởi một siêu mặt phẳng $\boldsymbol{w}_m$ và, đối với một đầu vào $\boldsymbol{x}$ bất kỳ, ta có thể tính được một khoảng cách có dấu $\boldsymbol{x}^T\boldsymbol{w}_m$ giữa đầu vào $\boldsymbol{x}$ và siêu mặt phẳng $\boldsymbol{w}_m$. Các khoảng cách có dấu này được ánh xạ vào các xác suất bằng cách sử dụng hàm softmax được định nghĩa như sau:

$$\text{sofmax}\ \ x_1,\ldots,x_K\ \ =\left(\frac{\exp(x_1)}{\sum_{j=1}^{K}\exp(x_j)},\ldots,\frac{\exp(x_K)}{\sum_{j=1}^{K}\exp(x_j)}\right)$$

Trong đó K là số phần tử đầu vào và cũng là số trọng số của $\boldsymbol{w}$.

Ánh xạ cho ta các xác suất sau:

$$\forall \mathrm{m}=\ 1,\ldots,\mathrm{M}\ \ ,\mathrm{P}\ \ \mathrm{y}=C_{\mathrm{m}}\mid \mathbf{x}=\boldsymbol{x}\ \ =\frac{\exp(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{m}})}{\sum_{j=1}^{K}\exp(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{j}})}$$

Trong đó $\boldsymbol{x}=\left[\begin{array}{cc}1 & x_1,\ldots,x_K\end{array}\right]^{\mathrm{T}}$, $\boldsymbol{w}_{\mathrm{m}}=\left[\begin{array}{cc}w_{0,\mathrm{m}} & \mathrm{w}_{1,\mathrm{m}},\ldots,\mathrm{w}_{K,\mathrm{m}}\end{array}\right]$, K là kích thước của vectơ đầu vào và M là số lượng loại.

Các trọng số của $(\mathbf{w}_{\mathrm{m}})_{1\le \mathrm{m}\le \mathrm{M}}$ vẫn được ước tính bằng cách cực đại hóa hàm khả năng (Likelihoood Function):

$$\mathrm{L}_{\text{likeligood}}\ \ \boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\mathrm{M}}\ \ =\prod_{\mathrm{i}=1}^{\mathrm{N}}\prod_{\mathrm{m}=1}^{\mathrm{M}}\mathrm{P}\ \ \mathrm{P}\ \ y=C_{\mathrm{m}}\mid \mathbf{x}=\boldsymbol{x}_{\mathrm{i}}\ \ ^{\mathbf{1}_{y_{\mathrm{i}}=C_{\mathrm{m}}}}$$

Trong đó M là số lượng loại, N là số lượng mẫu, C_{m}=m là nhãn của loại và $\mathbf{1}_{y_{\mathrm{i}}=C_{\mathrm{m}}}=\begin{array}{l}1,\ \text{if}\ y_{\mathrm{i}}=C_{\mathrm{m}}\\ 0,\ \ \text{else}\end{array}$.

Tương ứng với cực tiểu hóa log khả giống (log likelihood) âm:

$$-\log\ \ \mathrm{L}_{\text{likelihood}}(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\mathrm{M}})\ \ =-\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{m}=1}^{\mathrm{M}}\mathbf{l}_{y_{\mathrm{i}}=C_{\mathrm{m}}}\log\ \ \mathrm{P}\ \ y=C_{\mathrm{m}}\mid \mathbf{x}=\boldsymbol{x}_{\mathrm{i}}$$

$$=-\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{m}=1}^{\mathrm{M}}\mathbf{l}_{y_{\mathrm{i}}=C_{\mathrm{m}}}\log\left(\frac{\exp\ \ \boldsymbol{x}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{m}}}{\sum_{j=1}^{\mathrm{M}}\exp\ \ \boldsymbol{x}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{j}}}\right)$$

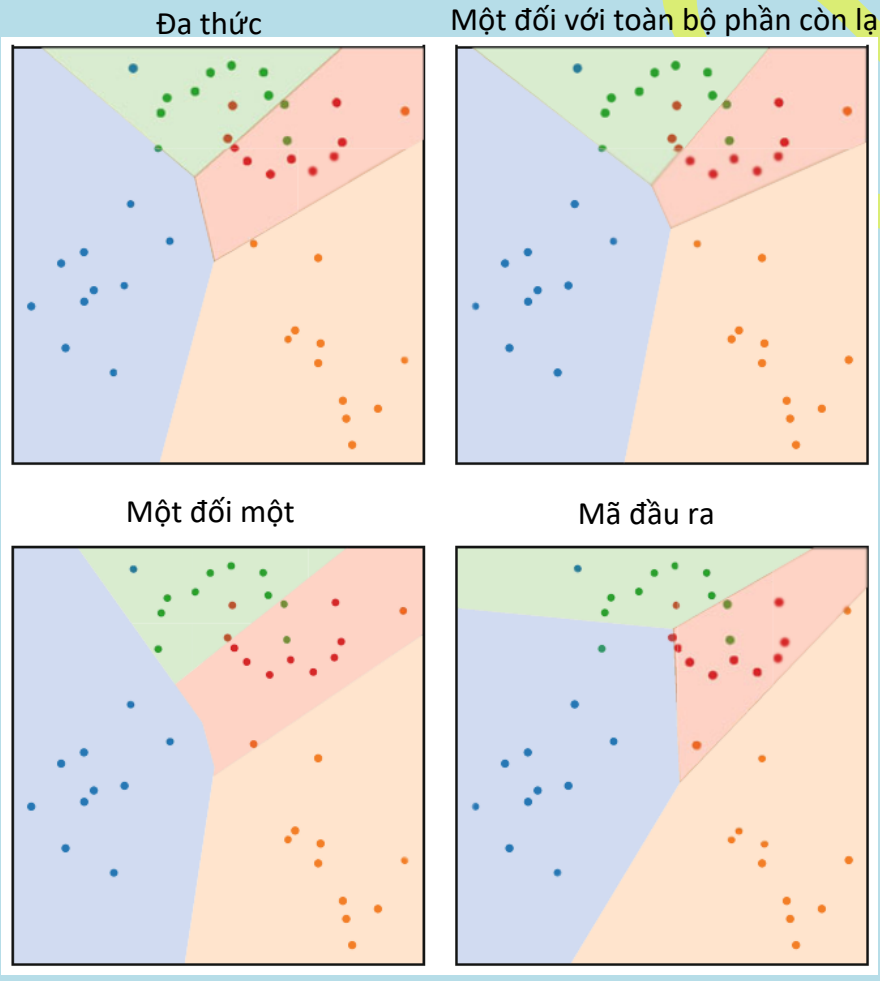
$$=\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{L}_{\text{cross-entropy}}\ \ y_{\mathrm{i}},\text{sofmax}\ \ \boldsymbol{x}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{w}_1,\ldots,\boldsymbol{x}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{M}}$$

Trong đó $L_{cross-entropy}$ được biết đến như là mất mát entropy chéo và được định nghĩa cho mọi nhãn y và vectơ xác suất $(\pi_1,\ldots,\pi_p)$ như sau:

$$L_{cross-entropy}\ \ y,(\pi_1,\ldots,\pi_M\ \ = -\sum_{m=1}^{M} \mathbf{l}_{y=C_m} \log \pi_m$$

Mất mát này thường được sử dụng để đào tạo các mạng tế bào thần kinh nhân tạo dựa trên các nhiệm vụ phân loại và tương đương với mất mát logistic trong trường hợp nhị phân.

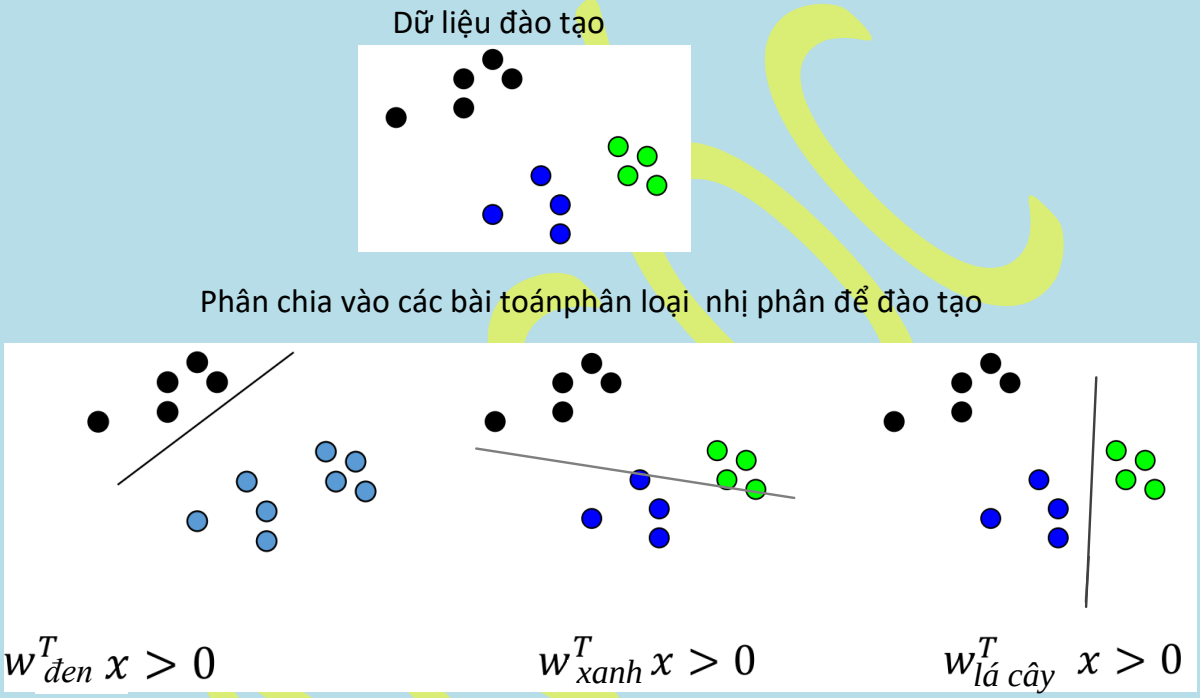
Hình 2.20 minh họa ảnh hưởng của chiến lược được sử dụng để xử lý các nhiệm vụ phân loại nhiều loại lên hàm quyết định.



Hình 2.20. Minh họa ảnh hưởng của chiến lược được sử dụng để xử lý một nhiệm vụ phân loại nhiều loại lên hàm quyết định của một mô hình hồi quy logistic

2.9.3. Hồi quy một đối chọi phần còn lại (One- vs- Rest)

Một chiến lược biến đổi một phân loại nhiều lớp thành các nhiệm vụ phân loại nhị phân là khớp nối một bộ phân biệt nhị phân cho từng loại: phân loại dương cho loại được cho trước và phân loại âm cho tất cả các loại khác được hòa nhập vào một loại đơn. Chiến lược này được gọi là một đối chọi phần còn lại (One-vs-Rest) hay một đối chọi tất cả (One Against All). Hình 2.21 minh họa chiến lược One-vs-Rest.



Hình 2.21. minh họa chiến lược One-vs-Rest

Trường hợp lý tưởng chỉ một nhãn đúng có diểm số dương.

Giải thuật học một đối chọi phần còn lại như sau:

- Bộ phân loại:
  - Hàm f: $\mathbb{R}^N \rightarrow \{1,2,3,\ldots,M\}$
- Phân chia thành các bài toán nhị phân
- Học máy: Cho trước tập dữ liệu D = $\boldsymbol{x}_i, y_i$
  - $\boldsymbol{x}_i \in \mathbb{R}^N, y_i \in 1,2,...,M$
- Phân chia thành M nhiệm vụ phân loại nhị phân
  - Học M mô hình: $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \ldots, \boldsymbol{w}_M$

- Đối với loại m, cấu trúc bộ phân loại nhi phân như sau:
  - Các thí dụ dương: các phần tử của D có nhãn m
  - Các thí dụ âm: tất cả các phần tử khác của D
- Phân loại nhị phân có thể được giải quyết bằng mọi giải thuật đã biết

Suy luận một đối chọi phần còn lại như sau:

- Học: Cho trước một tập dữ liệu D = $\boldsymbol{x}_i, y_i$
- Phân chia thành M nhiệm vụ phân loại nhị phân
  - Học M mô hình: $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \ldots, \boldsymbol{w}_M$
- Suy luận "người thắng lấy hết" (Winner Take All)
  - $\hat{y} = \underset{y \in 1,2,..,M}{\text{agmax}} \boldsymbol{w}_y^T \mathbf{x}$

Thí dụ: $y = \text{argmax} \ \boldsymbol{w}_{\text{black}}^T \boldsymbol{x}, \boldsymbol{w}_{\text{blue}}^T \boldsymbol{x}, \boldsymbol{w}_{\text{green}}^T \boldsymbol{x}$

- Dạng tổng quát:

$$\hat{y} = \underset{y \in Y}{\text{agmax}} f(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{x})$$

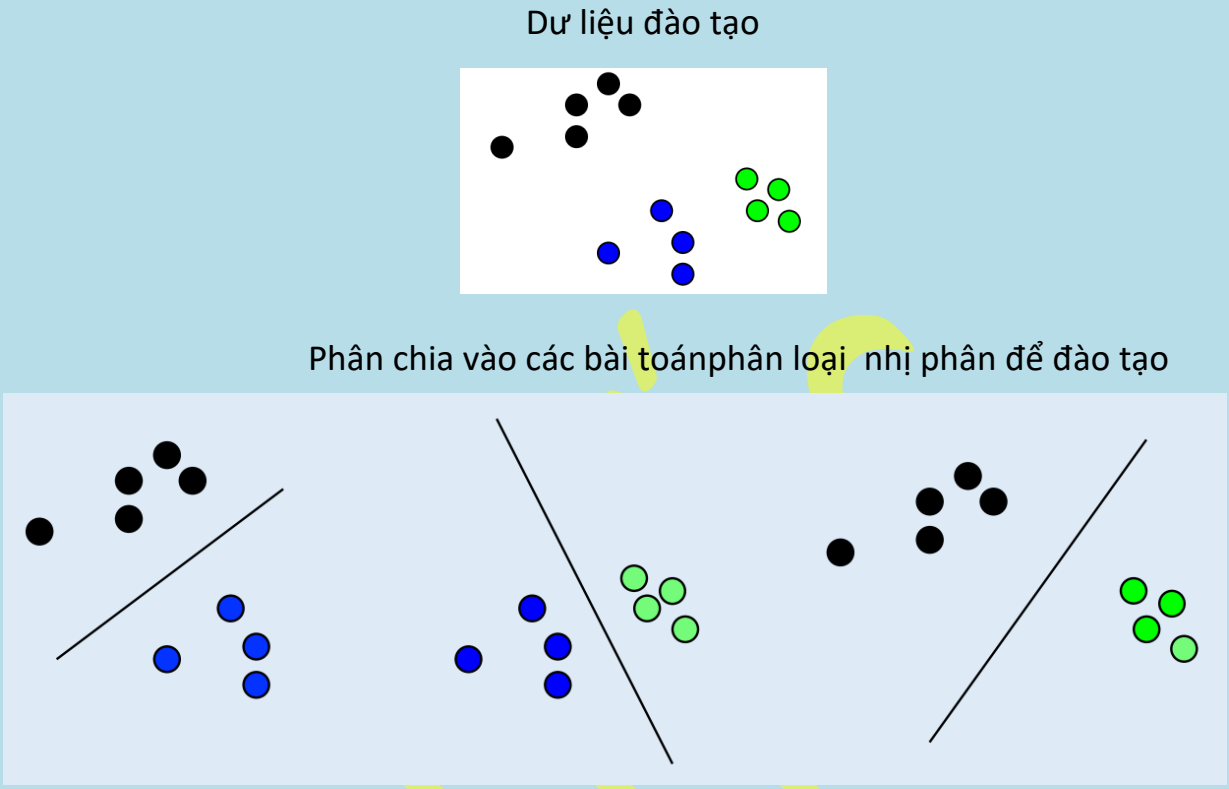
$$\boldsymbol{w} = \boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \ldots, \boldsymbol{w}_M \ , f \ y, \boldsymbol{w}, \boldsymbol{x} = \boldsymbol{w}_y^T \boldsymbol{x}$$

Ưu điểm của chiến lược này là mỗi loại được đặc trưng bởi một mô hình duy nhất, vì thế có thể hiểu biết sâu hơn về lọa này bằng cách xem xét kỹ lưỡng mô hình tương ứng của nó. Vì thế các dự báo cho các mẫu mới tính đến cả độ chắc chắn của mô hình: loại được dự báo cho một đầu vào mới là loại mà đối với nó mô hình tương ứng là mô hình chắc chắn nhất mà đầu vào này thuộc về loại của nó. Chiến lược một đối chọi phần còn lại là chiến lược được sử dụng phổ biến nất và thường là lựa chọn mặc định tốt.

2.9.3. Một đối một (One – vs-One)

Một chiến lược khác khớp nối một bộ phân loại nhị phân cho từng cặp loại: chiến lược này được gọi là một đối một (One-vs-One) hay tất cả đối chọi tất cả (All Against All). Hình 4.22 minh họa chiến lược đào tạo một đối một.

BÁN GỐC



Hình 4.22. Minh họa chiến lược đào tạo một đối một

Ta có thể tóm lược chiến lược này như sau:

- Bộ phân loại:
  - Hàm f: $\mathbb{R}^N \rightarrow \{1,2,3,\ldots,M\}$
- Phân chia thành các bài toán nhị phân
- Cho trước một tập dữ liệu $D=\{\boldsymbol{x}_i,y_i^{)}\}, x_i^{)} \in \mathbb{R}^N, y_i=\{1, 2, 3, \ldots, M\}$
- Phân chia vào C(M,2) các nhiệm vụ phân loại nhị phân:
  - Học đào tạo các mô hình: $\boldsymbol{w}_1.\boldsymbol{w}_2,\boldsymbol{w}_3, \ldots,\boldsymbol{w}_{M\times(M-1)/2}$
- Đối với cặp loại (i,j), xây dựng một phân loại nhị phân sao cho:
  - Các thí dụ dương: các phần tử D với nhãn i
  - Các thí dụ âm: các phần tử D với nhãn j
  - Phân loại nhị phân có thể được giải quyết bằng bất kỳ giải thuật được biết

Giả thuật suy luận một đối một:

- Các tùy chọn quyết định:

- Phức tạp hơn; mỗi nhãn nhận k-1 phiếu chọn
- Đầu ra của bộ phân loại nhị phân có thể không gắn kết
- Đa số: phân loại thí dụ ***x*** nhận mẫu i nếu i thắng trên ***x*** nhiều hơn j (j=1,…, k)
- Cuộc thi đấu: bắt đầu với N/2 cặp; tiếp tục với các người chiến thắng

Ưu điểm của chiến lược này là các loại trong từng nhiệm vụ phân loại nhị phân là loại "nguyên chất", theo nghĩa là các loại khác nhâu không hòa nhập vào một loại đơn. Tuy nhiên, số lượng các bộ phân loại nhị phân cần cho đào tạo đối với một đối một $\left[\frac{1}{2}M(M-1)\right]$ lớn hơn so với một đối chọi phần còn lại (M). Tuy nhiên, đối với chiến lược một đối một, số lượng các mẫu đào tạo trong từng nhiệm vụ nhị phân nhỏ hơn tổng số các mẫu vì thế đào tạo từng bộ nhị phân nhanh hơn. Một nhược điểm khác là chiến lược này khó giải thích hơn so với chiến lược một đối chọi phần còn lại, vì loại được dự báo tương ứng với loại nhận được bình chọn nhiều nhất (nghĩa là, thắng được các trùng khớp một đối một nhiều nhất) và nó không tính đến độ chắc chắn trong mỗi chiến thắng trùng khớp. Chẳng hạn, chiến thắng một trùng khớp với xác suất 0.99 cho ta cùng kết quả với chiên thắng cùng trùng khớp với các suất 0,51, nghĩa là một bình chọn.

2.9.4. Mã đầu ra sửa lỗi (Error Corecting Output Code)

Một chiến lược khác một cách đáng kể được lấy cảm hứng từ lý thuyết mã sửa lỗi, bao gồm hòa nhập một tập con của các loại vào một loại và tập con khác vào loại khác cho từng nhiệm vụ phân loại nhị phân. Dữ liệu này thường được gọi là sổ mã (Code Book) và có thể được trình bày như là một ma trận có các hàng tương ứng các loại và các cột tương ứng các nhiệm vụ phân loại nhị phân. Ma trận chỉ chứa các giá trị -1 và +1 trình bày nhãn tương ứng cho từng loại và cho từng nhiệm vụ nhị phân. Đối với từng đầu vào, từng bộ phân loại cho ra điểm số (hay xác suất) liên kết với loại dương (loại mục tiêu cần dự báo). Loại được dự đoán cho đầu này là loại có vectơ tương ứng giống nhất với vectơ của các điểm số với sự giống nhau được được đánh giá bằng khoảng cách Euclid (càng nhỏ, càng giống nhau). Tồn tại các chiến lược tiên tiến để định nghĩa sổ mã, nhưng có ý kiến cho rằng sổ mã ngẫu nhiên thường cho các kết quả tốt như là một kết quả phức tạp.

2.10. CÁC HÀM QUYẾT ĐỊNH CÓ PHÂN PHỐI CHUẨN (DECISION FUNCTION WITH NORMAL DISTRIBUTION)

2.10.1 Tổng quan

Phân phối chuẩn là các phân phối phổ biến vì chúng thường được tìm thấy trong tự nhiên. Chẳng hạn, chẳng hạn phân phối chiều cao và trọng lượng trẻ sơ sinh có thể xấp xỉ hóa bằng cách sử dụng cách phân phối chuẩn. Hơn nữa, từ quan điểm toán học, các phân phối chuẩn đặc biệt dễ sử dụng. Vì thế, một mô hình chung bao gồm giả thiết rằng các vectơ đầu vào đào tạo được lấy mẫu độc lập từ các phân phối chuẩn.

Các mô hình phân loại có thể có bao gồm giả thiết rằng đối với từng loại, tất cả các đầu vào tương ứng được lấy mẫu từ một phân phối chuẩn có vectơ trung bình $\boldsymbol{\mu}_m$ và ma trận hiệp phương sai $\boldsymbol{\Sigma}_m$:

$$\forall \text{i sao cho } y_i = C_m, \boldsymbol{x}_i \sim N\ \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$$

Bằng cách sử dụng hàm mật độ xác suất cho một phân phối chuẩn, ta có thể tính mật độ xác suất của bất cứ đầu vào $\boldsymbol{x}$ liên kết với phân phối $N\ \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m$ của loại (Class) C_m:

$$p_{\boldsymbol{x}|y=C_m}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}_m|}} \exp\left(-\frac{1}{2} |\boldsymbol{x} - \boldsymbol{\mu}_m|^T \sum_m^{-1} \boldsymbol{x} - \boldsymbol{\mu}_m \right)$$

Với mô hình xác suất như vậy, dễ ràng tính xác suất mà một mẫu thuộc về lớp C_m bằng cách sử dụng quy tắc Bayes:

$$P\ y = C_m | \mathbf{x} = \boldsymbol{x}\ = \frac{p_{\boldsymbol{x}|y=C_m}(\boldsymbol{x}) P(y = C_m)}{P_{\mathbf{x}}(\boldsymbol{x})}$$

Với phân phối chuẩn, về mặt toán học, sử dụng log các xác suất dễ hơn:

$$\log P\ y = C_m | \mathbf{x} = \boldsymbol{x}$$

$$= \log p_{\mathbf{x}|y=C_m}(\boldsymbol{x}) + \log P(y = C_m) - \log P_{\mathbf{x}}(\boldsymbol{x})$$

$$= -\frac{1}{2}\ \boldsymbol{x} - \boldsymbol{\mu}_m\ ^T \boldsymbol{\Sigma}_m^{-1} |\ \boldsymbol{x} - \boldsymbol{\mu}_m\ | - \frac{1}{2} |\boldsymbol{\Sigma}_m| + \log P\ \ y = C_m$$

$$-\frac{K}{2} \log(2\pi) - \log p_{\mathbf{x}}(\boldsymbol{x})$$

$$= -\frac{1}{2} \boldsymbol{x}^T \boldsymbol{\Sigma}_m^{-1} \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m$$

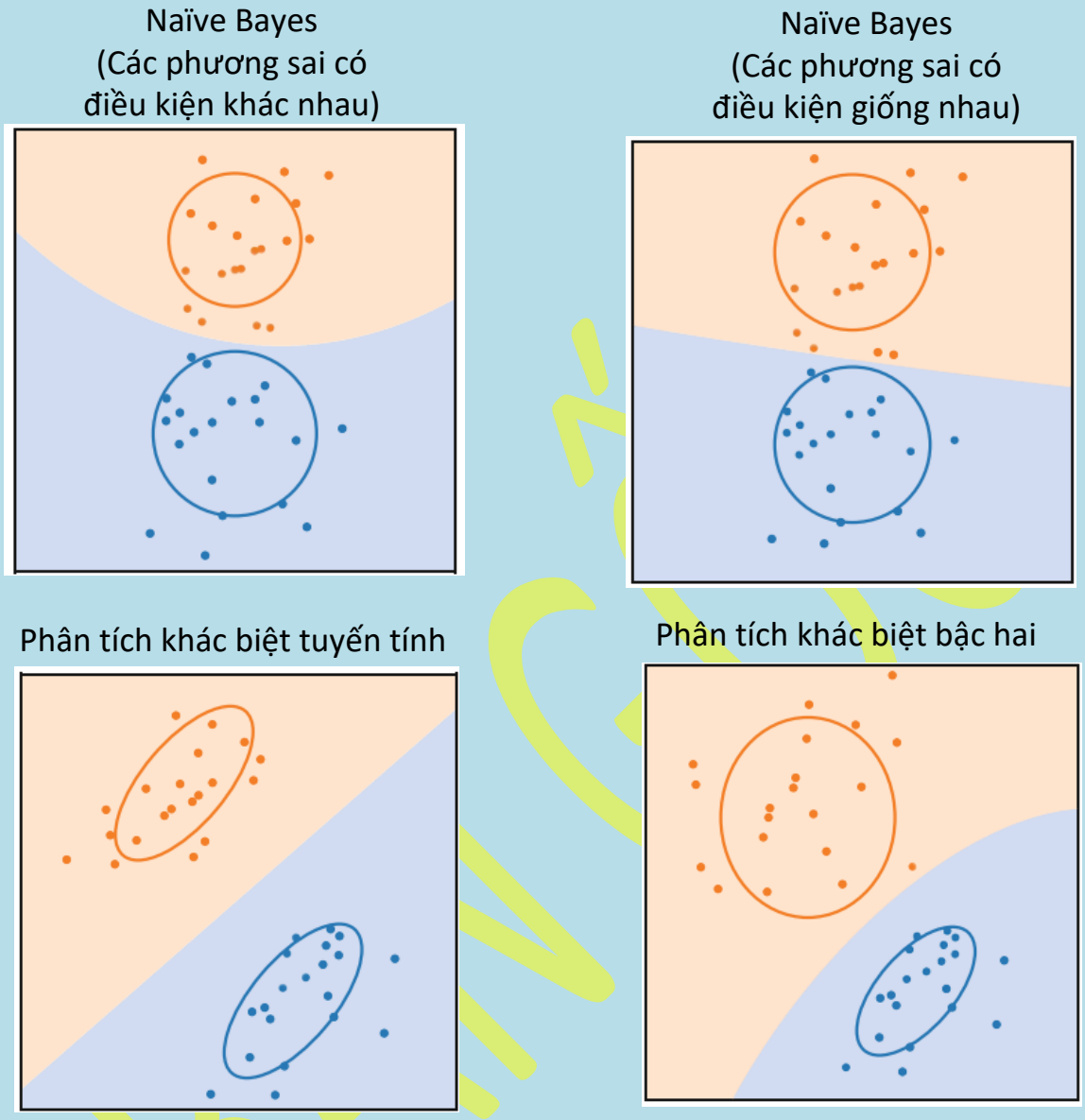
BẢN GỐC

$$-\frac{1}{2}\boldsymbol{\mu}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathrm{m}}^{-1}\boldsymbol{\mu}_{\mathrm{m}}-\frac{1}{2}\log\left|\boldsymbol{\Sigma}_{\mathrm{m}}\right|+\log\mathrm{P}\quad y=C_{\mathrm{m}}$$

$$-\frac{\mathrm{K}}{2}\log(2\pi)-\log\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) \qquad (2.4)$$

Ta có thể đưa ra thêm các giả định về các ma trân hiệp phương sai để được các mô hình khác nhau. Phần này trình bày các mô hình được sử dụng phổ biến nhất Bayes ngây thơ (Naïve Bayes), phân tích phân biệt tuyến tính (Linear Discriminant Analysis), phân tích phân biệt bậc hai (Quadratic Discriminant Analysis). Hình 2.23 minh họa các ma trận hiệp phương sai và các hàm quyết định cho các mô hình này trong các trường hợp hai chiều (Two-Dimentional).

BẢN GỐC



Hình 2.23.Minh họa các hàm quyết địnhbằng các phân phối chuẩn. Một ma trận hiệp phương sai hai chiều có thể được trình bày như là một ellipse. Trong mô hình Naïve Bayes, các tính năng (Feature) được giả thiết là độc lập và có cùng phương sai có điều kiện đối với loại (Class), dẫn đến các ma trận hiệp phương sai được trình bày như là các ellipse. Khi giả hiết là các ma trận hiệp phương sai giống nhau, các hàm quyết định sẽ tuyến tính thay vì bậc hai

2.10.2. Naive Bayes

Naïve Bayes giả thiết rằng, có điều kiện đối với từng loại C_m, các tính năng độc lập và có chung phương sai σ_m^2:

$$\forall m,\ \boldsymbol{\Sigma}_m = \sigma_m^2 \boldsymbol{I}_K$$

Từ phương trình (2.4) ta có thể viết:

$$\log P\ \ y = C_m \mid \mathbf{x} = \boldsymbol{x}$$

$$= -\frac{1}{2\sigma_m^2}\boldsymbol{x}^T\boldsymbol{x} + \frac{1}{\sigma_m^2}\boldsymbol{x}^T\boldsymbol{\mu}_m - \frac{1}{2\sigma_m^2}\boldsymbol{\mu}_m^T\boldsymbol{\mu}_m - \log\sigma_m + \log P(y = C_m)$$

$$-\frac{K}{2}\log(2\pi) - \log p_{\mathbf{x}}(\boldsymbol{x})$$

$$= \boldsymbol{x}^T\mathbf{W}_m\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{w}_m + w_{0m} + s$$

trong đó:

$\mathbf{W}_m = -\frac{1}{\sigma_m^2}\mathbf{I}_K$ là ma trận của thành phần bậc hai đối với loại C_m

$\boldsymbol{w}_m = \frac{1}{\sigma_k^2}\boldsymbol{\mu}_m$ là vectơ của thành phần tuyến tính đối với loại C_m

$w_{0m} = -\frac{1}{2\sigma_m^2}\boldsymbol{\mu}_m^T\boldsymbol{\mu}_m - \log\sigma_m + \log P(y = C_m)$ là điểm cắt đối với loại C_k

$s = -\frac{K}{2}\log(2\pi) - \log p_{\mathbf{x}}\ \ \boldsymbol{x}$ là thành phần không phụ thuộc vào loại C_k

Vì thế mô hình naïve Bayes này là bậc hai. Ta có thể dễ ràng tính toán các xác suất cho đầu vào $\boldsymbol{x}$ thuộc từng loại C_m:

$$P(y = C_m \mid \mathbf{x} = \boldsymbol{x}) = \frac{\exp\ \ \boldsymbol{x}^T\mathbf{W}_m\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{w}_m + w_{0k}}{\sum_{j=1}^{M}\exp\ \ \boldsymbol{x}^T\mathbf{W}_j\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{w}_j + \boldsymbol{w}_{0j}}$$

BẢN GỐC

Với naïve Bayes, ta thường có các phương sai có điều kiện σ_m^2 đều bằng nhau:

$$\forall m,\ \Sigma_m = \sigma_m^2 \boldsymbol{I}_K = \sigma^2 \boldsymbol{I}_K$$

Từ phương trình (2.4) ta có thể đơn giản hơn nữa:

$$\log P\ \ y = C_m \mid \mathbf{x} = \boldsymbol{x}$$

$$= -\frac{1}{2\sigma^2}\boldsymbol{x}^T\boldsymbol{x} + \frac{1}{\sigma^2}\boldsymbol{x}^T\boldsymbol{\mu}_m - \frac{1}{2\sigma^2}\boldsymbol{\mu}_k^T\boldsymbol{\mu}_k - \log\sigma\ + \log P(y = C_m)$$

$$-\frac{K}{2}\log(2\pi) - \log p_{\mathbf{x}}(\boldsymbol{x})$$

$$= \boldsymbol{x}^T\boldsymbol{w}_k + w_{0k} + s$$

trong đó:

$\boldsymbol{w}_m = \frac{1}{\sigma^2}\boldsymbol{\mu}_m$ là vectơ của thành phần tuyến tính đối với loại C_m

$w_{0m} = -\frac{1}{2\sigma^2}\boldsymbol{\mu}_m^T\boldsymbol{\mu}_m + \log P(y = C_m)$ là điểm cắt đối với loại C_m

$s = -\frac{1}{2\sigma^2}\boldsymbol{x}^T\boldsymbol{x} - \log\sigma\ - \frac{K}{2}\log(2\pi) - \log p_{\boldsymbol{x}}\ \ \boldsymbol{x}$ là thành phần không phụ thuộc vào loại C_m

Trong trường hợp này naive Bayes trở thành mô hình tuyến tính.

2.10.3. Phân tích phân biệt tuyến tính (Linear Discriminant Analysis)

Phân tích phân biệt tuyến tính (LDA: Linear Discriminant Analysis) đưa ra giả thiết là tất cả các ma trân hiệp phương sai là như nhau nếu không là tùy ý khác:

$$\forall m,\ \boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}$$

Vì thế ta có thể đơn giản hóa phương trình (2.4) hơn nữa:

BAN GOC

$$\log \mathrm{P}\ \ y = C_{\mathrm{m}} \mid \mathbf{x} = \boldsymbol{x}$$

$$= \log \mathrm{p}_{\mathbf{x}|\mathrm{y}=C_{\mathrm{m}}}(\boldsymbol{x}) + \log \mathrm{P}(\mathrm{y} = C_{\mathrm{m}}) - \log \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})$$

$$= -\frac{1}{2}\ \boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{k}}\ ^{\mathrm{T}}\ \Sigma^{-1}\ \ \boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{m}}\ \ - \frac{1}{2}\log\left|\Sigma\right| + \log \mathrm{P}\ \ y = C_{\mathrm{m}}$$

$$-\frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})$$

$$= -\frac{1}{2}\ \ \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\mathrm{m}} - \boldsymbol{\mu}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\mathrm{m}}\ \ - \frac{1}{2}\log\left|\Sigma\right| + \log \mathrm{P}\ \ y = C_{\mathrm{m}}$$

$$-\frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})$$

$$= -\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\mathrm{m}} - \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_{\mathrm{m}}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\mathrm{m}} + \log \mathrm{P}\ \ y = C_{\mathrm{m}}\ \ - \frac{1}{2}\log\ \ \left|\Sigma\right|$$

$$-\frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{P}_{\mathbf{x}}(\boldsymbol{x})$$

$$= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_{\mathrm{k}} + w_{0\mathrm{m}} + \mathrm{s}$$

trong đó:

$\boldsymbol{w}_{\mathrm{k}} = \Sigma^{-1}\ \boldsymbol{\mu}_{\mathrm{m}}$ là vectơ của của các hệ số đối với loại C_{m}

$w_{0\mathrm{m}} = -\frac{1}{2}\boldsymbol{\mu}_{\mathrm{m}}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_{\mathrm{m}} + \log \mathrm{P}\ \ \mathrm{y} = C_{\mathrm{m}}$ là điểm cắt đối với loại C_{m}

$\mathrm{s} = -\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\log\left|\Sigma\right| - \frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{P}_{\mathbf{x}}(\boldsymbol{x})$ là thành phần không phụ thuộc vào loại C_{m}

Vì thế phân tích phân biệt tuyến tính là một mô hình tuyến tính. Khi ∑ là ma trận đường chéo, phân tích phân biệt tuyến tính giống như naitive Bayes có cáphương sai gống nhau.

Khi này có thể dễ ràng tính toán các xác suất cho đầu vào $\boldsymbol{x}$ thuộc từng loại C_{m}:

BẢN GỐC

$$\mathrm{P}(y = \mathrm{C}_\mathrm{m} \mid \mathbf{x} = \boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^\mathrm{T}\boldsymbol{w}_\mathrm{m} + w_{0\mathrm{m}})}{\sum_{\mathrm{j}=1}^{\mathrm{M}} \exp(\boldsymbol{x}^\mathrm{T}\boldsymbol{w}_\mathrm{j} + w_{0\mathrm{j}})}$$

2.10.4. Phân tích phân biệt bậc hai (Quadratic Discriminant Analysis)

Phân tích phân biệt bậc hai không đặt giả thiết cho các ma trận hiệp phương sai $\sum_\mathrm{k}$, cho rằng các ma trận này có thể bất kỳ. Có thể viết phương trình (2.4) như sau:

$$\log \mathrm{P}\ \ y = C_\mathrm{m} \mid \mathbf{x} = \boldsymbol{x}$$

$$= -\frac{1}{2}\boldsymbol{x}^\mathrm{T}\boldsymbol{\Sigma}_\mathrm{m}^{-1}\boldsymbol{x} + \boldsymbol{x}^\mathrm{T}\boldsymbol{\Sigma}_\mathrm{m}^{-1}\boldsymbol{\mu}_\mathrm{m} - \frac{1}{2}\boldsymbol{\mu}^\mathrm{T}\boldsymbol{\Sigma}_\mathrm{m}^{-1}\boldsymbol{\mu}_\mathrm{m} - \frac{1}{2}\log|\boldsymbol{\Sigma}_\mathrm{m}|$$

$$+ \log \mathrm{P}(y = C_\mathrm{m}) - \frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{p}_\mathbf{x}\ \ \boldsymbol{x}$$

$$= \boldsymbol{x}^\mathrm{T}\mathbf{W}_\mathrm{m}\boldsymbol{x} + \mathbf{x}^\mathrm{T}\boldsymbol{w}_\mathrm{m} + w_{0\mathrm{m}} + \mathrm{s}$$

Trong đó:

$\mathbf{W}_\mathrm{m} = -\frac{1}{2}\boldsymbol{\Sigma}_\mathrm{m}^{-1}$ là ma trận có thành phần bậc hai đối với loại C_m.

$\boldsymbol{w}_\mathrm{m} = \boldsymbol{\Sigma}_\mathrm{m}^{-1}\boldsymbol{\mu}_\mathrm{m}$ là vectơ có thành phần tuyến tính đối với loại C_m.

$w_{0\mathrm{m}} = -\frac{1}{2}\boldsymbol{\mu}^\mathrm{T}\boldsymbol{\Sigma}_\mathrm{m}^{-1}\boldsymbol{\mu} - \frac{1}{2}\log|\boldsymbol{\Sigma}_\mathrm{m}| + \log \mathrm{P}\ \ y = C_\mathrm{m}$ là điểm cắt đối với loại C_m.

$\mathrm{s} = -\frac{\mathrm{K}}{2}\log(2\pi) - \log \mathrm{p}_\mathrm{x}\ \ \boldsymbol{x}$ không phụ thuộc vào loại C_m.

Vì thế phân tích phân biệt bậc hai là một mô ình bậc hai.

Khi này, ta có thể dễ dàng tính toán các xác suất đầu vào $\boldsymbol{x}$ thuộc từng loại C_k:

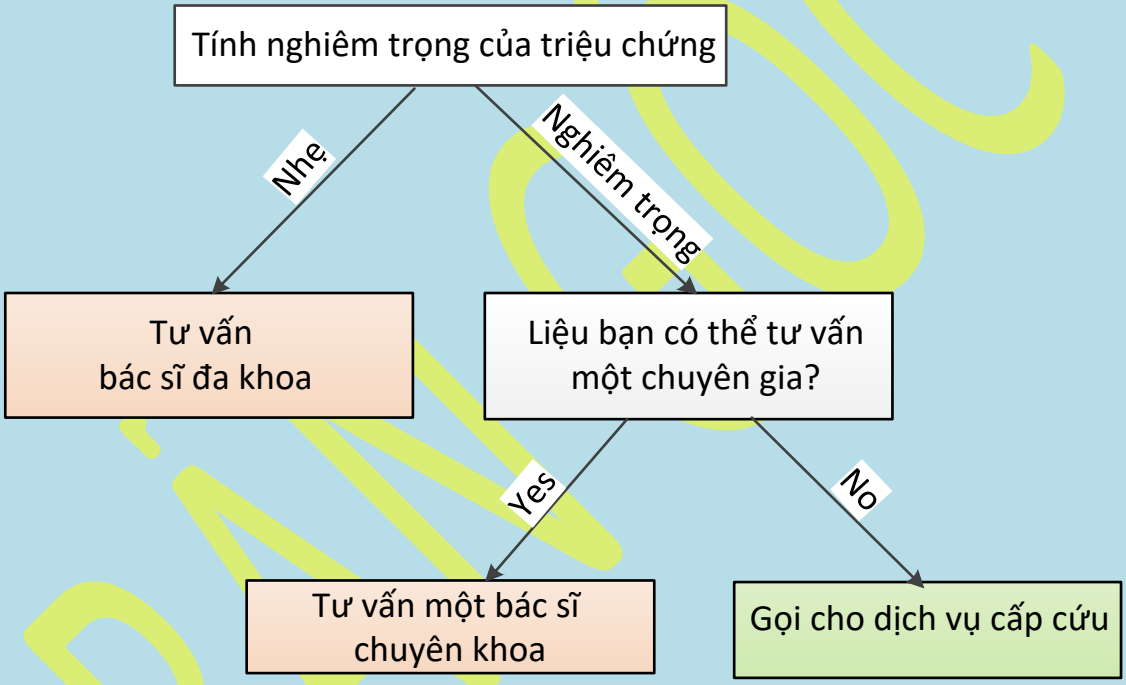
$$\mathrm{P}\ \ y = C_\mathrm{m} \mid \mathbf{x} = \boldsymbol{x}\ \ = \frac{\exp\ \ \boldsymbol{x}^\mathrm{T}\mathbf{W}_\mathrm{m}\boldsymbol{x} + \boldsymbol{x}^\mathrm{T}\boldsymbol{w}_\mathrm{k} + w_{0\mathrm{k}}}{\sum_{\mathrm{j}=1}^{\mathrm{M}} \exp\ \ \boldsymbol{x}^\mathrm{T}\mathbf{W}_\mathrm{j}\boldsymbol{x} + \boldsymbol{x}^\mathrm{T}\boldsymbol{w}_\mathrm{j} + w_{0\mathrm{j}}}$$

2.11.CÁC PHƯƠNG PHÁP DỰA TRÊN CÂY (TREE BASED METHOD)

BẢN GỐC

2.11.1. Định nghĩa cây

Các báo cáo có điều kiện dưa trên các quyết định nhị phân thường được sử dụng trong cuộc sống hàng ngày vì chúng trực giác và dễ hiểu. Hình 2.24 minh họa một cách tiếp cận chung khi một người nào đó bị ốm. Phụ thuộc và các báo cáo có điều kiện (tính nghiêm trọng của các triệu chứng, khả năng nhanh chóng tham vấn một chuyên gia), quyết định (tham vấn một bác sĩ đa khoa hoặc một bác sĩ chuyên khoa hay gọi dịch vụ cấp cứu) sẽ khác nhau. Các mô hình với một kiến trúc như vậy thường được sử dụng trong học máy và được gọi là các cây quyết định (Decision Trees).



Hình 2.24. Quá trình suy nghĩ chung khi bị bệnh. Phụ thuộc vào báo cáo có điều kiện (mức độ nghiêm trọng của triệu chứng, khả năng nhanh chóng tư vấn một chuyên gia), quyết định (tư vấn bác sĩ đa khoa của bạn hay một bác sĩ chuyên khoa hay gọi cho dịch vụ cấp cứu)

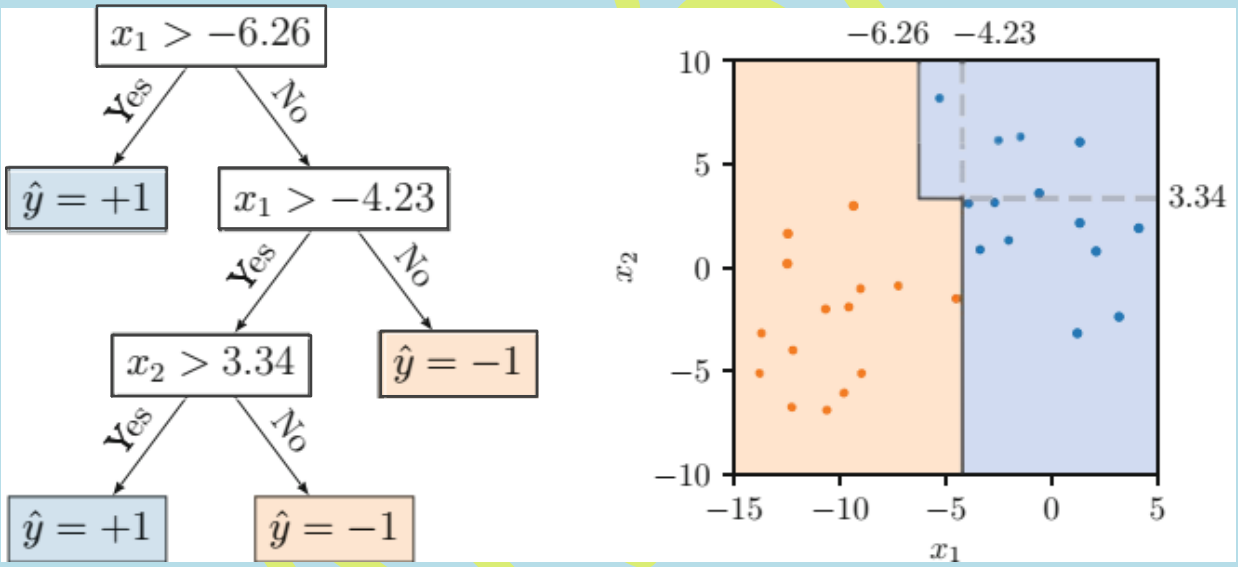
Cây quyết định là mộtgiải thuật chỉ chứa các báo cáo có điều kiện và có thể được trình bày bởi một cây. Đồ thị này bao gồm:

- Các nút quyết định cho báo cáo có điều kiện
- Các nhánh của các kết cục tiềm năng của từng nút quyết định

- Các nút lá cho quyết định cuối cùng

Hình 2.25 minh họa một cây quyết định và hàm quyết định tương ứng của nó. Đối với một mẫu cho trước, quyết định cuối cùng nhận được bằng cách đi theo đường dẫn tương ứng của nó với khởi đầu từ nút gốc.

Cây quyết định phân chia hồi quy không gian tính năng để nhóm các mẫu có cùng các nhãn hay các giá trị đích khác nhau. Tại mỗi nút , mục tiêu là tìm cặp tốt nhất (tính năng hoặc ngưỡng) sao cho cả hai tập con nhận được bằng sự phân chia này là nguyên chất (Pure), nghĩa là, thuần nhất. Làm như vậy, cặp tốt nhất (tính năng, ngưỡng) được định nghĩa như là cặp giảm thiểu tiêu chuẩn tạp chất.



Hình 2.25. Một cây quyết định: (trái) các quy tắc được học bởi cây quyết định và (phải) hàm quyết định tương ứng

Giả sử $S \subseteq X$ là một tập con của các mẫu đào tạo. Đối với các nhiệm vụ phân loại, phân phối của các loại, nghĩa là tỷ lệ của từng loại, được sử dụng để đo độ nguyên chất của tập con. Giả sử p_m là tỷ lệ của các mẫu từ lớp C_m trong một phân chia cho trước:

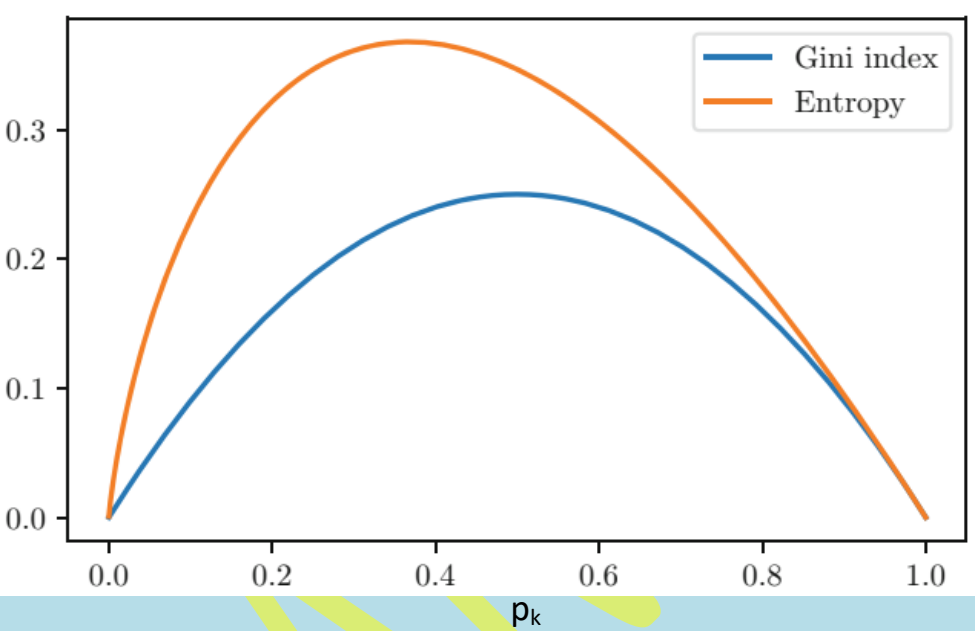
$$p = \frac{1}{|S|}\sum_{y \in S} 1_{y=C_m}$$

Các tiêu chuẩn tạp chất phổ biến bao gồm:

- Chỉ số Gini (Gini Index): $\sum m p_m \ 1 - p_m$
- Entropy: $-\sum p_m \log(p_m)$

- Phân loại sai (Misclassification): $1 - \max_m p_m$

Hình 2.26 minh họa các giá trị của chỉ số Gini và entropy cho một loại C_m duy nhất đối với các các tỷ lệ khác nhau của p_m. Ta có thể nhận thấy rằng hàm entropy nhận các giá trị lớn hơn chỉ số Gini, đặc biệt đối với p_m <0,8. Do tổng các tỷ lệ bằng 1, hầu hết các lớp chỉ trình bày một tỷ lệ nhỏ của các mẫu. Vì thế , một giải thích đơn giản là entropy có tính phân biệt cao hơn đối với các tập con đồng nhất so với chỉ số Gini. Phân loại nhầm chỉ xét đến tỷ lệ của lớp chung nhất và có khuynh hướng trở nên thậm chí phân biệt đối với các tập con đồng nhất kém hơncả entropy lẫn Gini.



Hình 2.26. Minh họa chỉ số Gini và entopy. Hàm entropy nhận các giá trị lớn hơn chỉ số Gini, đặc biệt khi p_m <0,8, vì thế nó có tính phân biệt đối với các tập con đồng nhất cao hơn chỉ số Gini (khi hầu hết các loại chỉ trình bày một tỷ lệ nhỏ của các mẫu)

Đối với các nhiệm vụ hồi quy, sai số chính từ một giá trị suy luận (chẳng hạn giá trị trung bình của trung vị (Mean of Median)) thường được sử dụng như là một tiêu chuẩn tạp chất:

- Sai số bình phương trung bình $\frac{1}{|S|}\sum_{y\in S} (y - \bar{y})^2$ với $\bar{y} = \frac{1}{|S|}\sum_{y\in S} y$
- Sai số tuyệt đối trung bình: $\frac{1}{|S|}\sum_{y\in S} |y - \text{median}_s(y)|$

Về mặt lý thuyết, một cây có thể phát triển cho đến khi mọi nút là nguyên chất hoàn hảo. Tuy nhiên cây như vậy sẽ có rất nhiều nhánh và sẽ rất phức tạp là cho nó dễ bị quá khớp. Một số chiến lược thường được sử dụng để hạn chế kích thước của cây. Một cách tiếp cận là phát triển cây không hạn chế sau đó cắt tỉa cây, nghĩa là, thay thế các cây con với các nút. Các chiến lược phổ biến khác để hạn chế sự phức tạp của cây thường được áp dụng khi cây đã phát triển và bao gồm thiết lập:

- Độ sâu cực đại cho cây
- Số lượng mẫu yêu cầu tại một nút bên trong cực tiểu
- Số lượng mẫu đòi hỏi để phân chia một phân vùng cho trước cực tiểu
- Số lượng các nút lá cực đại
- Số lượng các tính năng được xét (thay vì tất cả các tất cả các tính năng) để tìm được phân chia tốt nhất
- Tạp chất tối thiểu giảm để phân chia một nút bên trong

2.11.2. Rừng ngẫu nhiên (Random Forest)

Một hạn chế của các cây quyết định là sự đơn giản của chúng. Các cây quyết định có khuynh hướng sử dụng một phần nhỏ các tính năng trong hàm quyết định của chúng. Để sử dụng nhiều hơn các tính năng trong cây quyết định, đòi hỏi phát triển cây lớn hơn, nhưng cây lớn hơn có khuynh hướng bị quá khớp, nghĩa là, độ lệch thấp nhưng phương sai cao. Một giải pháp giảm phương sai nhưng không tăng nhiều độ lệch là xây dựng một tập hợp các cây mang tính ngẫu nhiên, từ đó có cái tên rừng cây ngẫu nhiên. Tổng quan rừng cây ngẫu nhiên được tổng kết dưới đây.

Rừng ngẫu nhiên

- **Rừng ngẫu nhiên**: Tập hợp các cây quyết định mang tính ngẫu niên được đưa ra để xây dựng các cây khác nhau
- **Cây quyết định**: Giải thuật chỉ chứa các báo cáo có điều kiện và được trình bày bằng một cây
- **Chính quy hóa**: độ sâu cực đại cho một cây, số các mẫu đòi hỏi để phân chia một phân vùng cho trước, …

2.11.3. Các cây cực kỳ ngẫu nhiên (Extremelly Randomized Trees)

Mặc dù các rừng cây ngẫu nhiên bao gồm tính ngẫu nhiên trong lấy mẫu cả các mẫu và các tính năng , các cây bên trong một rừng cây ngẫu nhiên vẫn có xu thế tương quan với nhau, vì hế hạn chế giảm phương sai. Để giảm thậm chí nhiều hơn phương sai của mô

hình (trong khi có thể tăng độ lệch) bằng cách phát triển các cây ít tương quan hơn, rừng cây cực kỳ ngẫu nhiên (Extremely Randommixed Tree) được đưa ra với tính ngẫu nhiên lớn hơn. Thay vì tìm kếm phân chia tốt nhất trong số tất cả các cặp ứng cử (tính năng, ngưỡng), một ngưỡng được rút ra ngẫu nhiên cho từng tính năng ưng cử và tốt nhất trong số các ngưỡng được tạo ra nhẫu nhiên này được chọn như là quy tắc phân chia.

2.12. PHÂN CỤM (CLUSTERING)

Cho đến nay, ta đã trình bày các phương pháp học máy kinh điển cho phân loại và hồi quy, đây là các phần tử chính của học máy có giám sát. Một đầu vào $\boldsymbol{x}_i$ có một đầu ra liên kết y_i. Trong phần này ta sẽ trình bày phân cụm, đây là một nhiệm vụ của học máy không có giám sát. Trong học máy không có giám sát, chỉ có đầu vào là có sẵn, không có đầu ra liên kết. Vì không có sẵn sự thật cơ bản (Ground Truth), mục tiêu sẽ là chiết suất thông tin từ dữ liệu đầu vào mà không giám sát quá trình học bằng dữ liệu đầu ra.

Phân cụm bao gồm việc tìm ra các nhóm mẫu sao cho:

- Các mẫu từ một nhóm đều giống nhau
- Các mẫu từ các nhóm khác nhau khác nhau

Chẳng hạn, phân cụm có thể được sử dụng để nhận dạng bệnh của một nhóm bệnh cụ thể như bệnh Alzheimer và bệnh Parkinson.

Trong phần này ta sẽ xét hai phương pháp phân vùng phổ biến nhất: giải thuật phân vùng k-means (k-means clustreing) và mô hình hỗn hợp Gauss (Gaussian Mixture Model)

2.12.1. k-means (k –trung bình)

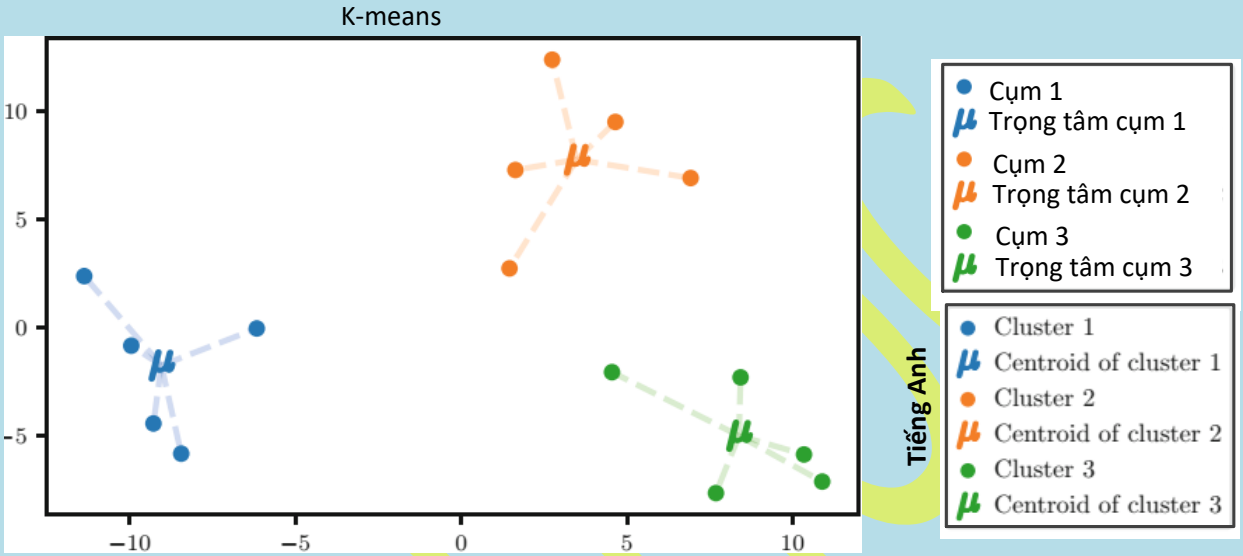
Giải thuật k-means chia một tập hợp N mẫu, được ký hiệu là X, vào một tập hợp k cụm không liên kết (Disjoint Clusters), mỗi cụm được ký hiệu là X_j, sao cho $X = X_1,\ldots,X_k$ Hình 2.27 minh họa khái niệm của giải thuật này. Mỗi cụm X_j được đặc trưng bởi trọng tâm (Centroid) của mình, được ký hiệu là $\boldsymbol{\mu}_j$, nghĩa là, trung bình của các mẫu này trong cụm:

$$\boldsymbol{\mu}_j = \frac{1}{|X_j|}\sum_{\boldsymbol{x}_i \in X_j} \boldsymbol{x}_i$$

Các trọng tâm hoàn toàn định nghĩa tập hợp các cụm vì mỗi mẫu được ấn định cho cụm có trọng tâm gần nhất.

Giải thuật k-means (k trung bình) nhắm đến tìm các trọng tâm cho phép giảm thiểu quán tính (Inertia), còn được biết là tiêu chuẩn tổng bình phương trong cụm:

$$\min_{\mu_1,\ldots,\mu_k} = \sum_{j=1}^{k} \sum_{x_i \in X_j} \left\| x_i - \mu_j \right\|_2^2$$



Hình 2.27. Minh họa giải thuật k-means. Mục tiêu của giải thuật là tìm được các trọng tâm (Centroid) để giảm thiểu tiêu chuẩn tổng các bình phươnbg trọng cụm. Thí dụ,, quán tính vào khoảng 184,80 và là quán tính thấp nhất có thể, nghĩa là các trọng tâm được trình bày là tối ưu

Giải thuật gốc được sử dụng để tìm các trong tâm thường đực đề cập là giải thuật LLoud và được trình bày trong giải thuật 1. Sau khi khởi đầu các trọng tâm, vòng hai bước được lặp cho đến khi hội tụ (khi các trong tâm giống nhau đối với hai lần lặp liên tiếp) bao gồm:

- Bước ấn định (Asignment Sep), trong đó các cụm được cập nhật dựa trên các trọng tâm hiện thời
- Bước cập nhật (Update Step) , trong đó các trọng tâm được cập nhật dựa trên các cụm hiện thời

Khi đã định nghĩa rõ ràng các cụm, một điểm từ một cụm cho trước rất có thể ở trong cụm này. Vì thế bước ấn định có thể được tăng tốc nhờ bất đẳng thức tam giác (Triangle Inequality) bằng cách theo dõi các giới hạn dưới và giới hạn trên đối với các khoảng cách giữa các điểm và các trung tâm, với trả giá phải sử dụng bộ nhớ lớn.

Giải thuật 1: giải thuật Lloyd (hay giải thuật naïve k-means)

Result: centroids $\{\mu_1, \ldots, \mu_k\}$; "kết quả tìm được các trọng tâm"

Initialize centroids $\{\mu_1, \ldots, \mu_k\}$; "khởi đầu các trọng tâm"

While not converged **do,** **"**nếu không hội tụ thì thực hiện**"**

Asignment Step: Compute the clusters (nghĩa là, ấn định từng mẫu đến trọng tâm gần nó nhất):

$$\forall j \in 1,\ldots,k \ , X_j = \left\{ \boldsymbol{x}_i \in X \mid \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|_2^2 = \min_j \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|_2^2 \right\}$$

Update Step: tính toán các trọng tâm của các cụm được cập nhật

$$\forall j \in 1,\ldots,k \ , \boldsymbol{\mu}_j = \frac{1}{\left| X_j \right|} \sum_{\boldsymbol{x}_i \in X_j} \boldsymbol{x}_i$$

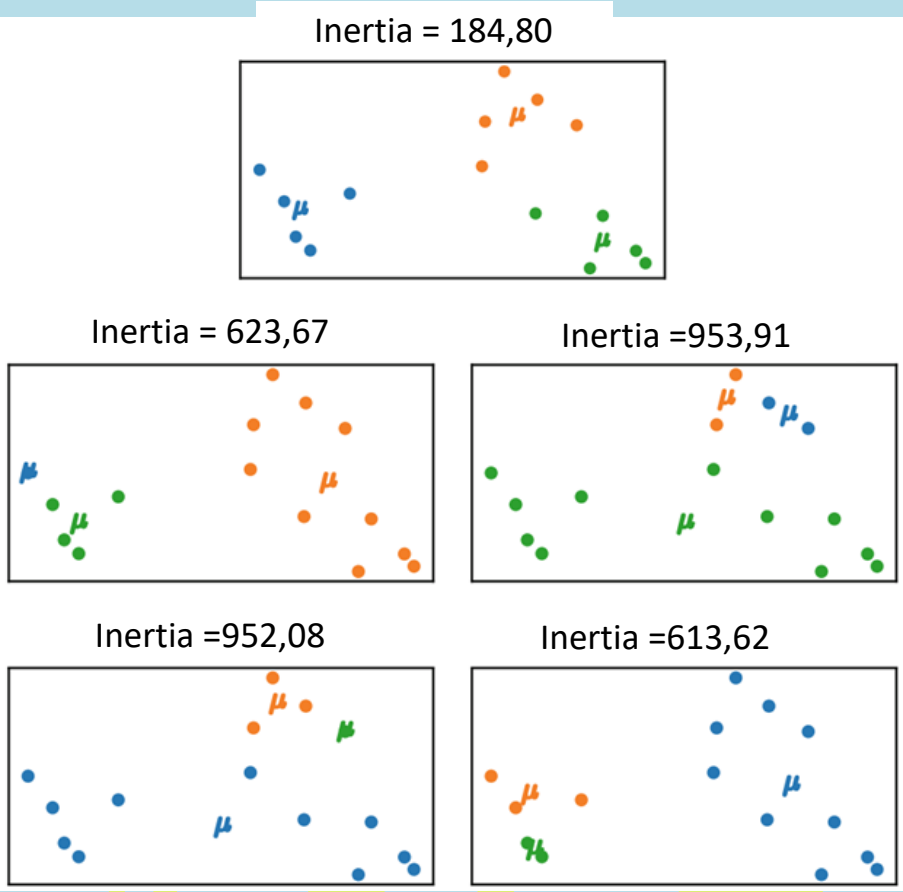
Mặc dù giải thuật k-means là một trong số các giải thuật đơn giản nhất và là một trong các phương pháp phân cụm được sử dụng phổ biến nhất, nó có một số nhược điểm mà ta cần lưu ý.

Trước hết, số lượng các cụm k là một siêu tham số. Đặt một giá trị khác nhiều với số lượng thực tế của các cụm có thể cho ra các cụm tồi.

Thứ hai, quán tính (Inertia) không phải là một hàm lồi. Mặc dù giải thuật Lloyd đảm bảo hội tụ, nó thể thể hội tụ đến một cực tiểu cục bộ mà không phải cực tiểu toàn cục. Hình 2.28 minh họa hội tụ đến các trọng tâm như vậy. Một số chiến lược thường được áp dụng để giải quyết vấn đề này bao gồm cả khởi đầu trọng tâm phức tạp, chạy giải thuật nhiều lần và duy trì chạy giải thuật tốt nhất (nghĩa là, giải thuật cho ra quán tính thấp nhất).

Thứ ba, quán tính đặt ra giả thiết rằng các cụm là lồi và đẳng hướng. Giải thuật k-means có thể cho các kết quả tồi khi giải thiết này không đảm bảo. chẳng hạn các cụm bị kéo dài hay đa tạp có các hình dáng không bình thường.

Thứ tư, khoảng cách Euclid có khuynh hướng bị thổi phồng (nghĩa là, tỷ số các khoảng cách của láng giếng gần nhất và xa nhất đối với một đích cho trước gần bằng 1) trong không gian đa chiều. Một cách để gảm nhẹ vấn đề này là vận hành một phương pháp giảm chiều (Dimationality Reduction) chẳng hạn phân tích thành phần chính trước giải thuật k-means.



Hình 2.28. Minh họa hội tụ của giải thuật k-means đến các cực tiểu cục bộ tồi. Trong hình trên, giải thuật hội tụ đến một cực tiểu toàn cục, vì giá trị cực tiểu có thể (184,80), vì thế, các cụm nhận được là tối ưu. Trong bốn hình khác, giải thuật hội tụ đến các cực tiểu cục bộ không phải toàn cục vì các quán tính này cao hơn giá trị cực tiểu có thể, như vậy, các cụm nhận được là dưới tối ưu (Suboptimal)

2.12.2. Mô hình hỗn hợp Gauss (Gaussian Mixture Model)

Mô hình hỗn hợp đặt giả thiết là mỗi mẫu được tạo ra từ một hỗn hợp của một số phân phối độc lập.

Giả sử k là số lượng các phân phối. Mỗi phân phối F_j được đặc trưng bởi xác suất mà nó được chọn, được ký hiệu là π_j, và mật độ của nó p_j với các tham số θ_j, được ký hiệu là $p_j(.;\theta_j)$. Giả sử $\Delta = \Delta_1,\ldots,\Delta_k$ là một biến ngẫu nhiên có giá trị vectơ sao cho:

$$\sum_{j=1}^{k}\Delta_j = 1 \text{ và } \forall i \in 1,\ldots,k \ , P\ \Delta_j = 1\ = 1 - P\ \Delta_j = 0\ = \pi_j$$

và $(x_1, \ldots, x_k)$ là các biến ngẫu nhiên sao cho $x_j \sim F_j$. Giả thiết các mẫu được tạo ra từ một biến ngẫu nhiên x có mật độ p_x là:

$$x = \sum_{j=1}^{k}\Delta_j x_j$$

$$\forall x \in X,\ p_x(x;\theta_j) = \sum_{j=1}^{k}\pi_j p_j(x;\theta_j)$$

Mô hình hỗn hợp Gauss là một trường hợp cụ thể của mô hình hỗn hợp trong đó mỗi phân phối F_j là một phân phối Gauss có vectơ trung bình μ_j và ma trận hiệp phương sai Σ_j:

$$\forall j \in 1,\ldots,k \ , F_j = N(\mu_j;\Sigma_j)$$

Hình 2.29 minh họa các phân phối được học từ một mô hình hỗn hợp Gauss.

Mục tiêu là tìm được các tham số θ cho phép cực đại hóa hàm khả năng, với $\theta = \mu_{j=1}^{k}, \Sigma_{j=1}^{k}, \pi_{j=1}^{k}$ ta có hàm khả năng:

$$L_{likelihood}\ \theta\ = \prod_{i=1}^{N} p_X\ \ x_i;\theta$$

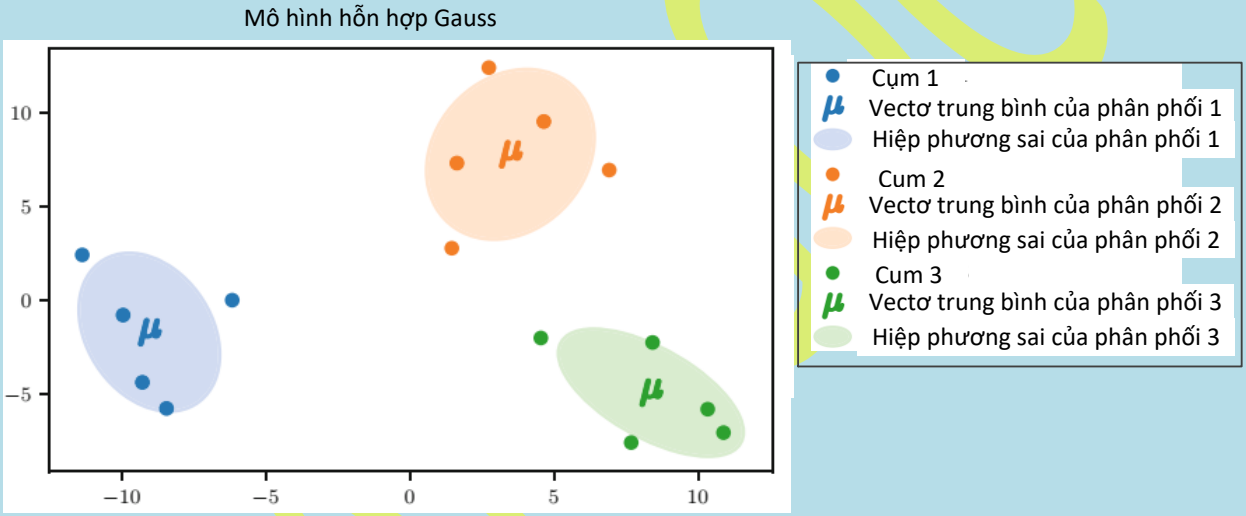
Vì lý do tính toán, sẽ dễ hơn khi cực đại hóa hàm lg khả năng (log likelihod):

$$\log\ L_{likelihood}(\theta)\ = \sum_{i=1}^{N}\log(p_X(x_i;\theta)) = \sum_{i=}^{N}\log\left(\sum_{j=1}^{k}\pi_j p_j(x_i;\theta_j)\right)$$

Do mật độ $p_X(.,\theta)$ là tổng được đánh trọng số của các mật độ Gauss, ta không thể đơn giản nó hơn nữa. Để giải bài toán cực đại, một giải thuật được gọi là cực đại hóa kỳ vọng (EM: Expectation Maximization) thường được áp dụng. Giải thuật 2 mô tả các khái niệm

chính của giải thuật này. Sau khởi đầu các tham số của từng phân phối, một vòng hai bước được lặp cho đến khi hội tụ (khi các tham số ổn định sau nhiều vòng liên tiếp):

- Bước kỳ vọng (E-Step: Expectation Step): trong đó xác suất của từng mẫu $\boldsymbol{x}_i$ được tạo ra từ phân phối F_j được tính
- Bước cực đại hóa (M-Step: Maximization Step): trong đó xác suất và các tham số của từng phân phối được cập nhật
- Vì không thể biết các mẫu nào đã được tạo bởi từng phân phối, nên cũng không thể trực tiếp cực đại hóa log-likelihood, vì thế ta tính giá trị được kỳ vọng của nó bằng cách sử dụng các xác suất hậu nghiệm, từ đó có cái tên bước cực đại hóa.



Hình 2.29. Mô hình hỗn hợp Gauss. Đối với từng phân phối được ước tính, vectơ trung bình cà các ellipse bao gồm tất cả các điểm trong độ lệch tiêu chuẩn (Standarrd Variation) của trung bình này được vẽ

Giải thuật 2: EM: Expectation-Maximization (kỳ vọng-tối đa hóa) cho các mô hình hỗn hợp Gauss

Kết quả (Result): Các vectơ trung bình $\boldsymbol{\mu}_{j=1}^{k}$, các ma trận hiệp phương sai $\boldsymbol{\Sigma}_{j=1}^{k}$ và các xác suất $\boldsymbol{\pi}_{j=1}^{k}$

Khởi đầu các vectơ trung bình $\boldsymbol{\mu}_{j=1}^{k}$, các ma trận hiệp phương sai $\boldsymbol{\Sigma}_{j=1}^{k}$ và các xác suất $\boldsymbol{\pi}_{j=1}^{k}$

While not converged **do** "nếu không hội tụ thì thực hiện)

E-step: tính xác suất hậu nghiệm $\gamma_i(j)$ cho từng mẫu $\boldsymbol{x}_i$ được tạo ra từ phân phối F_j:

$$\forall i = 1,\cdots,N \ , \forall j = 1,\cdots,k \ , \gamma_i(j) = \frac{\pi_j p_j \left(\boldsymbol{x}_i;\boldsymbol{\theta}_j;\boldsymbol{\Sigma}_j\right)}{\sum_{\ell=1}^{k} \pi_\ell p_\ell \left(\boldsymbol{x}_i;\boldsymbol{\theta}_\ell;\boldsymbol{\Sigma}_\ell\right)}$$

M-step: Cập nhật các tham số của từng phân phối F_j:

$$\forall j \in 1,\ldots,k \ , \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{N} \gamma_i(j)\boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_i(j)}$$

$$\forall j \in 1,\ldots,k \ , \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{N} \gamma_i(j)\left[\boldsymbol{x}_i - \boldsymbol{\mu}_j\right]\left[\boldsymbol{x}_i - \boldsymbol{\mu}_j\right]^T}{\sum_{i=1}^{N} \gamma_i(j)}$$

$$\forall j \in 1,\ldots,k \ , \boldsymbol{\pi}_j = \frac{1}{N}\sum_{i=1}^{N} \gamma_i(j)$$

Các giải thuật Lloyd và EM có nhiều điểm giống nhau. Trong bước thứ nhất, bước ấn định ấn định từng mẫu đến cụm gần nhất của nó, trong khi bước kỳ vọng tính toán xác suất cho từng mẫu được tạo ra từ từng phân phối. Trong bước thứ hai, bước cập nhật tính toán trọng tâm của từng cụm như là trung bình của các mẫu trong một cụm cho trước, trong khi bước cực đại hóa cập nhật xác suất và các tham số của từng phân phối như là một trung bình được đánh trọng số trên tất cả các mẫu. Vì các lý do này, giải thuật k-means thường được đề cập như là phương pháp phân cụm bỏ phiếu cứng (Hard-Voting), đối lập với mô hình hỗn hợp chung được đề cập như là phương pháp phân cụm bỏ phiếu mềm (Soft-Voting).

Mô hình hỗn hợp chung có một số lợi điểm so với giải thuật k-means.

Thứ nhất, việc sử dụng các mật độ phân phối chuẩn thay vì các khoảng cách Euclid giảm dần vấn đề lạm phát (Inflation) trong các không gian đa chiều cao (High- Dimentional Space). Thứ hai, mô hình hỗn hợp chung có các ma trận hiệp phương sai, cho phép các cụm có dạng ellipse, trong khi giải thuật k-means chỉ có các trọng tâm buộc các cụm này có dạng hình tròn.

Tuy nhiên, mô hình hỗn hợp chung cũng có một số nhược điểm chia sẻ với một vài nhược điểm của giải thuật k-means.
Thứ nhất, số lượng k phân phối là một siêu tham số. Việc thiết lập một tham số khác nhiều so với số lượng các cụm thực tế có thể cho ra các cụm tồi. Thứ hai, log khả năng không phải là một hàm lồi. Giống như giải thuật Lloyd, giải thuật EM được đảm bảo hội tụ, nhưng nó có thể hội tụ đến một cực đai cục bộ chứ không phải một cực đại toàn cục. Một số chiến lược thường được áp dụng để xử lý vấn đề này, bao gồm khởi đầu trọng tâm phức tạp, chạy giải thuật tốt nhất và duy trì chạy tốt nhất (nghĩa là giải thuật mang lại log khả năng cao nhất). Thứ ba, mô hình hỗn hợp chung không có nhiều tham số so với giải thuật k-means. Vì thế, nó thường đòi hỏi nhiều mẫu hơn để ước tính chính xác các tham số của mình(chẳng hạn các ma trận hiệp phương sai) so với giải thuật k-means.

2.13. GIẢM CHIỀU (GIẢM KÍCH THƯỚC KHÔNG GIAN)

Giảm chiều bao gồm tìm kiếm một ánh xạ tốt từ một không gian đầu vào vào một không gian có kích thước nhỏ hơn (có ít chiều hơn). Giảm chiều có thể là học có giám sát hoặc không có giám sát.

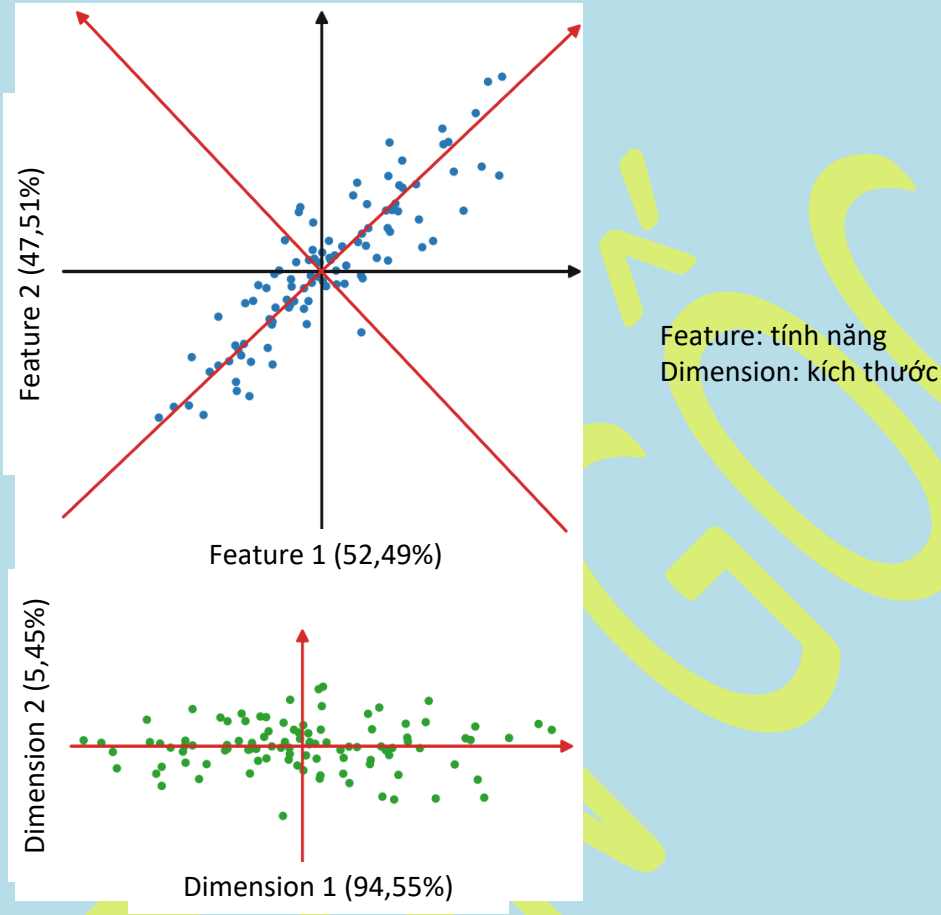
2.13.1. Phân tích thành phần chính (Principal Component Analysis)

Để phân tích dữ liệu khám phá, ta có thể quan tâm đến nghiên cứu các phương sai của K tính năng và các hiệp phương sai $\frac{1}{2}$K 1− K hay các tương quan. Tuy nhiên do giá trị K tăng, quá trình này trở nên phát triển quá lớn. Hơn nữa, môt tính năng có thể giải thích một phần nhỏ của phương sai tổng thể. Nên ta có thể mong muốn hơn có một trình bày khác của dữ liệu, trong đó một số lượng nhỏ các tính năng giải thích hầu hết phương sai tổng thể. Nói một cách khác, có được một hệ tọa độ thích ứng với dữ liệu đầu vào.

Phân tích thành phần chính (PCA: Principal Component Analysis) bao gồm tìm ra một trình bày dữ liệu thông qua các thành phần chính. Các thành phần chính là một chuỗi các vectơ đơn vị sao cho vectơ i là là xấp xỉ tốt nhất của dữ liệu (tức là, cực đại hóa phương sai) trong khi tực giao với i-1 các vectơ đầu tiên.

Hình 2.30 minh họa phân tích thành phần chính khi không gian đầu vào hai kích thước. Tại hình trên, dữ liệu đào tạo trong không gian gốc được vẽ. Cả hai tính năng giải thích cùng một lượng như nhau của phương sai tổng thể, mặc dù ta có thể thấy rõ cả hai tính năng tương quan mạnh với nhau. Phân tích thành phần chính xác định một hệ tọa độ Decarrteses mới dựa trên dữ liệu đầu vào. Tại hình dưới, dữ liệu trong hệ tọa độ mới

được vẽ. Kích thước (chiều) thứ nhất giải thích phương sai nhiều hơn kích thước (chiều) thứ hai.



Hình 2.30. Minh họa phân tích thành phần chính. Tại hình trên, dữ liệu đào tạo trong không gian gốc (các điểm xanh cùng với các trục đen) được vẽ. Cả hai tính năng giải thích cùng một khối lượng như nhau của phương sai tổng thể, mặc dù ta có thể thấy rõ một mẫu tuyến tính. Phân tích thành phần chính học một hệ tọa độ Decarrteses mới dựa trên dữ liệu đầu vào (các trục đỏ). Tại hình dưới, dữ liệu trong hệ tọa độ mới được vẽ (các điểm xanh với các trục đỏ). Kích thước 1 (chiều 1) giải thích phương sai nhiều hơn kích thước 2 (chiều 2)

2.13.1.1 Phân tích hoàn toàn (Full Decomposition)

Về mặt toán học, cho trước một đầu vào $\boldsymbol{X} \in \mathbb{R}^{N \times K}$ được đặt ở chính giữa (tức là, giá trị trung bình của từng cột $\boldsymbol{X}_{i,j}$ bằng không), mục tiêu là tìm một ma trận $\mathbf{W} \in \mathbb{R}^{K \times K}$ sao cho:

- **W** là một ma trận trực giao, tức là, các cột của nó là các vectơ đơn vị và trực giao với nhau.
- Trình bày mới của dữ liệu đầu vào, được ký hiệu là ***T***, bao gồm các tọa độ trong hệ tọa độ Decarteses được suy ra từ **W** (có các cột tạo thành một cơ sở trực giao $\mathbb{R}^K$ với tích vô hướng Euclid):

  $$\boldsymbol{T} = \boldsymbol{X}\mathbf{W}$$

- Mỗi cột của **W** cực đại hóa phương sai được giải thích (Explained Variance).

Mỗi cột $\boldsymbol{w}_i = \mathbf{W}_{:,i}$ là một phần tử chính. Từng vectơ đầu vào ***x*** được biến đổi vào một vectơ ***t*** khác bằng cách kết hợp tuyến tính từng tính năng có các trọng số từ ma trận **W**:

$$\boldsymbol{t} = \boldsymbol{x}^T\mathbf{W}$$

Thành phần chính thứ nhất $\boldsymbol{w}_1$ là vectơ đơn vị mà nó cực đại hóa phương sai được giải thích (Explained Vảriance):

$$\sigma^2_{\boldsymbol{w}_1} = \arg\max_{\|\boldsymbol{w}\|=1} \sum_{i=1}^{N} \boldsymbol{x}_i^T\boldsymbol{w}$$

$$= \arg\max_{\|\boldsymbol{w}\|=1} \|\boldsymbol{X}\boldsymbol{w}\|$$

$$= \arg\max_{\|\boldsymbol{w}\|=1} \|\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}\|$$

$$= \arg\max_{\boldsymbol{w}\in\mathbb{R}^K} \left\{\frac{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}\right\}$$

Vì $\mathbf{X}^T\mathbf{X}$ là một ma trận bán xác định dương (có tất cả các giá trị riêng (eigen) không âm), từ đại số tuyến tính ta biết rằng $\mathbf{w}_1$ là vectơ riêng (Eigenvector) liên kết với giá trị eigen lớn nhất của $\mathbf{X}^T\mathbf{X}$.

Ta có thể tìm được thành phần k bằng cách trừ đi k-1 các thành phần đầu tiên từ ***X***:

$$\hat{\boldsymbol{X}}_k = \boldsymbol{X} - \sum_{s=1}^{k-1} \boldsymbol{X}\mathbf{w}_s\mathbf{w}_s^T$$

BẢN GỐC

Và sau đó tìm vectơ đơn vị, vectơ này giải thích phương sai cực đại từ ma trận dữ liệu mới này:

$$\boldsymbol{w}_{\mathrm{k}} = \arg\max_{\|\boldsymbol{w}\|=1} \left\| \hat{\boldsymbol{X}}_{\mathrm{k}} \boldsymbol{w} \right\| = \arg\max_{\boldsymbol{w}\in\mathbb{R}^{\mathrm{K}}} \left\{ \frac{\boldsymbol{w}^{\mathrm{T}} \hat{\boldsymbol{X}}^{\mathrm{T}} \hat{\boldsymbol{X}} \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}} \right\}$$

Ta có thể chỉ ra rằng vectơ riêng (Eigenvector) liên kết với giá trị riêng (Eigenvalue) cực đại của ma trận $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ sẽ cực đại hóa đại lượng cần được cực đại hóa. Vì thế, ma trận **W** là ma trận có các cột là các vectơ riêng của ma trận $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ được phân loại bằng cách giảm dần thứ tự các giá trị liên kết của chúng.

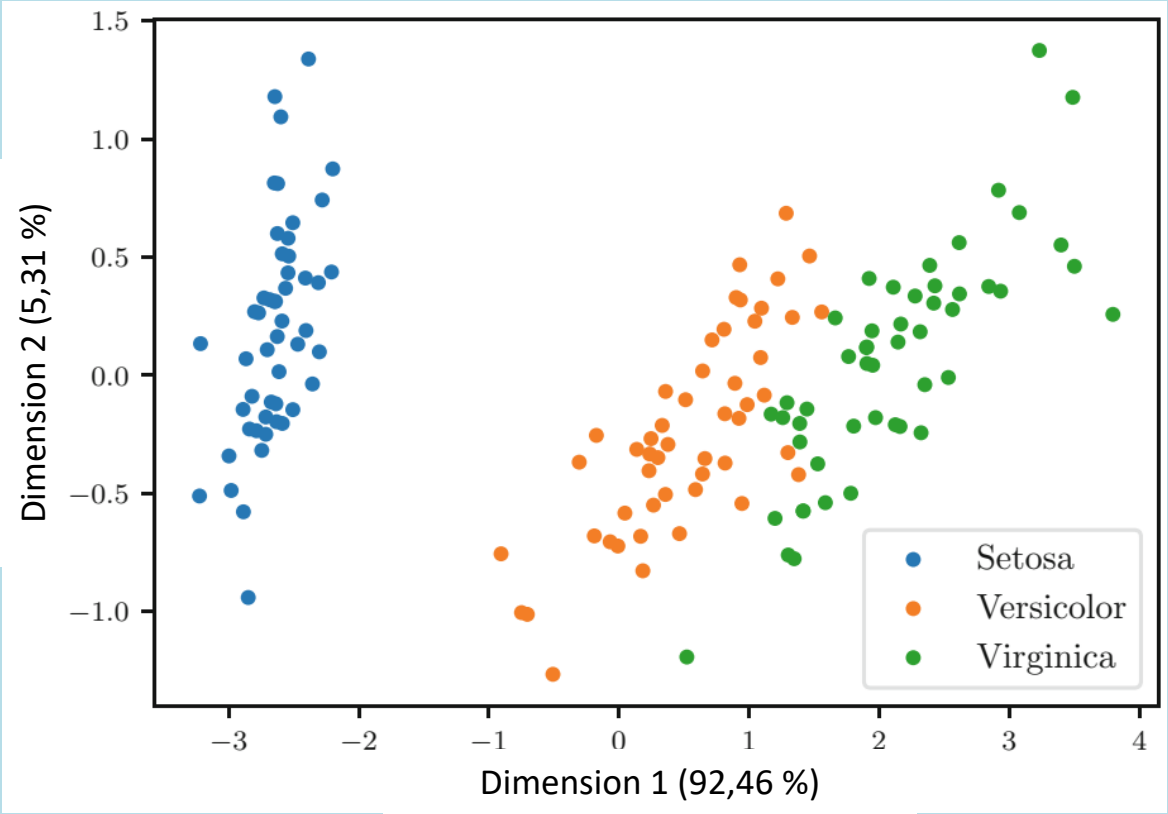
2.13.1.2. Phân tích được cắt ngắn

Vì từng thành phần chính giảm phương sai còn lại bằng cách lặp, thành phần chính đầu tiên giải thích hầu hết tổng phương sai, trong khi các thành phần còn lại chỉ giải thích một phần nhỏ tổng phương sai. Vì thế, chỉ giữ lại một tập con của các thành phần chính được sắp xếp theo thứ tự thường cho ta một trình bày tốt của dữ liệu đầu vào.

Về mặt toán học, giả sử cho trước một số lượng các kích thước (hay các chiều) ℓ, trình bày mới nhận được bằng cách cắt ngắn ma trận các thành phần chính **W** để chỉ giữ lại ℓ cột đầu tiên, dẫn đến ma trận $\mathbf{W}_{:,:\ell}$ sau:

$$\boldsymbol{T} = \boldsymbol{X}\mathbf{W}_{:,:\ell}$$

Hình 2.31 minh họa việc sử dụng phân tích thành phần chính ngư là giảm kích thước (giảm chiều). Tập dữ liệu hoa Iris (diên vĩ) bao gồm 50 mẫu cho từng ba loài Iris (setossa, versicolor và virginica), đối với chúng 4 tính năng được đo, độ dài và độ rộng của đài hoa và cánh hoa theo centimet. Hình chiếu của từng mẫu lên hai thành phần chính thứ nhất được cho thấy trên hình vẽ.



Dimention: chiều (kích thước)

Hình 2.31. Minh họa phân tích thành phần chính như là giảm kích thước (giảm chiều). Tập dữ liệu hoa Iris gồm 50 mẫu đối với mỗi mẫu gồm ba loài Iris setossa, versicolor và virginica), đối với chúng 4 tính năng được đo, độ dài và độ rộng của đài hoa và cánh hoa theo centimet. Hình chiếu của từng mẫu trên hai thành phần chính thứ nhất được cho thấy trên hình vẽ. Chiều thứ nhất (Dimention 1) giải thích hầu hết phương sai (92,46%)

2.13.2. Phân tích phân biệt tuyến tính (Linear Discriminant Analysis)

Trong mục 2.8 ta đã đưa ra phân tích phân biệt tuyến tính như là một phương pháp phân loại. Tuy nhiên, ta cũng có thể sử dụng nó như là một phương pháp giảm kích thước có giám sát. LDA phù hợp với một phân phối chuẩn đa biến (Multivariate Normal Distribution) cho từng loại C_m, sao cho mỗi loại được đặc trưng bởi vectơ trung bình của nó $\mu_m \in \mathbb{R}^K$ và có cùng ma trận hiệp phương sai $\Sigma \in \mathbb{R}^{K\times K}$. Tuy nhiên , một tập m điểm nằm trong một không gian kích thước có chiều (kích thước tối) đa m-1. Chẳng hạn, một tập hai điểm nằm trên một đường thẳng, trong khi tập 3 điểm nằm trên một mặt phẳng. Vì

thế không gian con được tạo ra bởi các vectơ trung bình m $\boldsymbol{\mu}_{\mathrm{m}}$ có thể được sử dụng như là giảm kích thước. Tồn tại một trình bày khác của phân tích phân biệt tuyến tính tương đương và trực giác hơn đối với giảm kích thước. Phân tích phân biệt tuyến tính nhắm đến tìm được một phép chiếu tuyến tính sao cho các loại (Class) được phân tách nhiều nhất có thể (tức là, hình chiếu của các mẫu từ cùng một loại sẽ gần nhau, còn hình chiếu của các mẫu từ loại khác nhau thì xa nhau).

Về mặt toán học, mục tiêu là tìm được ma trận $\boldsymbol{W} \in \mathbb{R}^{\mathrm{K}\times\ell}$ (với $\ell \leq$m-1) để cực đại hóa phân tán giữa các lớp (Between Class Scatter) trong khi cũng giảm thiểu phân tán trong cùng lớp (Within-Class Scatter):

$$\max_{\boldsymbol{W}} \mathrm{tr} \quad \boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{w}\boldsymbol{W} \quad ^{-1} \quad \boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{b}\mathbf{W}$$

Ma trận phân tán trong cùng một loại $\boldsymbol{S}_w$ tổng kết khuếch tán giữa các vectơ trung bình $\boldsymbol{\mu}_{\mathrm{m}}$ của loại $\mathrm{C_m}$ và tất cả các đầu vào $\boldsymbol{x}_{\mathrm{i}}$ thuộc loại C_{m} trên tất cả các loại:

$$\boldsymbol{S}_{w} = \sum_{\mathrm{m}=1}^{\mathrm{M}} \sum_{y_{\mathrm{i}} \in \mathrm{C_m}} \quad \boldsymbol{x}_{\mathrm{i}} - \boldsymbol{\mu}_{\mathrm{m}} \quad \boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{m}} \quad ^{\mathrm{T}}$$

Ma trận phân tán giữa các loại $\boldsymbol{S}_b$ tổng kết khuyếch tán giữa tất cả các vectơ:

$$\boldsymbol{S}_{b} = \sum_{\mathrm{m}=1}^{\mathrm{M}} n_{\mathrm{m}} \quad \boldsymbol{\mu}_{\mathrm{m}} - \boldsymbol{\mu} \quad \boldsymbol{\mu}_{\mathrm{k}} - \boldsymbol{\mu} \quad ^{\mathrm{T}}$$

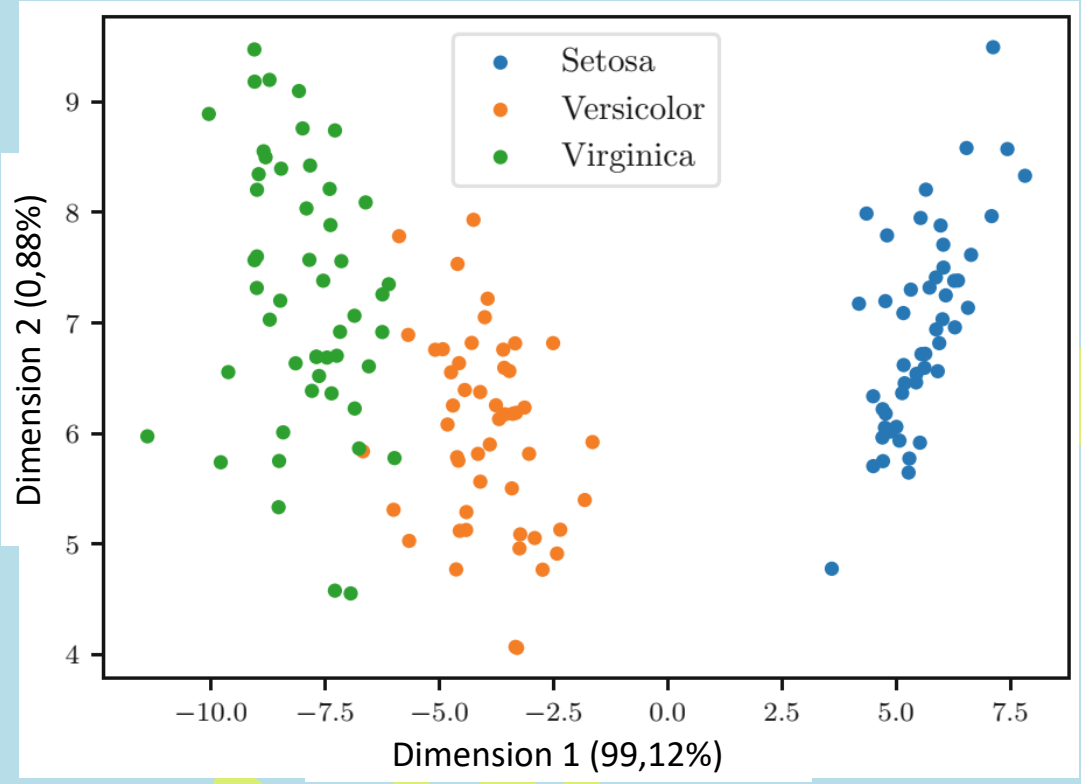
Trong đó n_{m} là tỷ phần của các mẫu thuộc loại $\mathrm{C_m}$ và $\boldsymbol{\mu} = \sum_{\mathrm{m}=1}^{\mathrm{M}} n_{\mathrm{m}}\boldsymbol{\mu}_{\mathrm{m}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \boldsymbol{x}_{\mathrm{i}}$ là vectơ trung bình trên tất cả các vectơ đầu vào.

Có thể chỉ ra rằng ma trận $\boldsymbol{W}$ **b**ao gồm ℓ vectơ riêng đầu tiên của ma trận $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ với các vectơ riêng (Eigenvector) được sắp xết theo thứ tự giảm dần. Cũng như trong phân tích thành phần chính, các giá trị riêng (Eigenvalue) có thể được sử dụng để xác định đóng góp của từng kích thước (chiều). Tuy nhiên, tiêu chuẩn của phân tích phân biệt tuyến tính khác với tiêu chuẩn của phân tích thành phần chính: cực đại hóa sự phân tách các loại thay vì cực đại hóa phương sai được giải thích.

Hình 2.32 minh họa việc sử dụng phân tích phân biệt tuyến tính như là một kỹ thuật giảm kích thước. Ta sử dụng cùng một tập dữ liệu hoa Iris như trên hình 2.31 để minh họa phân tích phần tử chính. Hình chiếu của từng mẫu trên không gian hai chều được học

BẢN GỐC

được trình bày và ta có thể thấy rằng trục thứ nhất (trục ngang) có tính phân biệt ba loại cao so với phân tích thành phần chính.



Hình 2.32. Minh họa phân tích phân biệt tuyến tính như là một kỹ thuật giảm chiều (kích thước). Tập dữ liệu hoa Iris gồm 50 mẫu với mỗi mẫu gồm ba loài (sentosa, versicolor và virgina), đối với chúng 4 tính năng được đo, độ đài và độ rộng của đài hoa và cánh hoa theo centimet. Hinh chiếu của từng mẫu trên không gian hai chiều được học được cho thấy trên hình vẽ

2.14. CÁC PHƯƠNG PHÁP HẠT NHÂN (KENEL METHOD)

Các phương pháp kenel cho phép tổng quát hóa các mô hình tuyến tính thành các mô hình phi tuyến tính bằng cách sử dụng các hàm hạt nhân (kenel function).

Như đã đề cập trước đây, ý tưởng chính của các phương pháp hạt nhân là trước hết ánh xạ dữ liệu đầu vào từ không gian đầu vào gốc vào một không gian tính năng và sau đó thực hiện các tích vô hướng trong không gian này.

Dựa trên một số giả thiết, một lời giải tối ưu của bài toán cực đại hóa hàm chi phí có dạng sau:

$$\mathrm{f} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}} K \ .; \boldsymbol{x}_{\mathrm{i}}$$

Trong đó K là hàm hạt nhân bằng tích vô hướng trong không gian tính năng:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in I, K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^{\mathrm{T}} \phi(\boldsymbol{x}')$$

Trong đó I ký hiệu cho không gian đầu vào.

Vì thành phần này thường xuyên xuất hiện, nên ta sẽ ký hiệu nó là ma trận đối xứng K kích thước $\mathrm{N} \times \mathrm{N}$ chưa tất cả các ước tính hạt nhân trên tất cả các cặp mẫu đào tạo:

$$\forall \mathrm{i}, \mathrm{j} \in \ 1, \ldots, \mathrm{N} \ , K_{\mathrm{i},\mathrm{j}} = K \ \ \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}}$$

Trong phần này ta sẽ trình bày mở rộng của hai mô hình đã được đưa ra trước đây trong chương này: hồi quy ridge (ridge regression) và phân tích phần tử chính (Pricipal Component Analysis) bằng các hàm hạt nhân.

2.14.1. Hồi quy ridge hạt nhân

Hồi quy ridge hạt nhân kết hợp hồi quy ridge với kỹ thuật hạt nhân và vì thế học một hàm tuyến tính trong không gian được tạo bởi hạt nhân tương ứng và dữ liệu đào tạo.Đối với các hạt nhân không tuyến tính, điều này tương ứng với một hàm không tuyến tính trong không gian gốc.

Về mặt toán học, mục tiêu là tìm được một hàm f có dạng sau:

$$\mathrm{f} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}} K \ ., \boldsymbol{x}_{\mathrm{i}}$$

Để giảm thiểu tổng các sai số bình phương với một thành phần phạt ℓ_2:

$$\min_{f} \sum_{i=1}^{N} \left(y_i - f(\boldsymbol{x}_i) \right)^2 + \lambda \|f\|^2$$

Ta có thể đơn giản hóa hàm chi phí bằng cách sử dụng dạng đặc thù của các hàm có thể có:

$$\sum_{i=1}^{N} \left(y_i - f(\boldsymbol{x}_i) \right)^2 + \lambda \|f\|^2$$

$$= \sum_{i=1}^{N} \left(y_i - \sum_{i=1}^{N} \alpha_j K\left(\boldsymbol{x}_j, \boldsymbol{x}_i \right) \right)^2 + \lambda \left\| \sum_{i=1}^{N} \alpha_i K\left(., \boldsymbol{x}_i \right) \right\|^2$$

$$= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\alpha}^T \boldsymbol{K}_{:,i} \right)^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$$

$$= \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$$

Vì thế bài toán cực tiểu hóa sẽ là:

$$\min_{\alpha} \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$$

Và lời giải cho bài toán này như sau:

$$\boldsymbol{\alpha}^* = \left(\boldsymbol{K} + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{y}$$

Hình 2.9 minh họa hàm dự đoán của phương pháp hồi quy ridge bằng một hàm cơ sở xuyên tâm. Hàm dự đoán là không tuyến tính vì hàm kenel không tuyến tính.

2.14.2. Phân tích thành phần chính kenel

Như đã đã được nhắc đến trong phần 2.10, phân tích thành phần chính bao gồm việc tìm được không gian con trực giao tuyến tính trong không gian đầu vào gốc sao cho mỗi phần tử chính giải thích phần lớn phương sai. Các vectơ riêng $\boldsymbol{X}^T \boldsymbol{X}$ cùng với các giá trị riêng tương ứng được lưu trữ theo thứ tự giảm dần sẽ cho ta lời giải tối ưu.

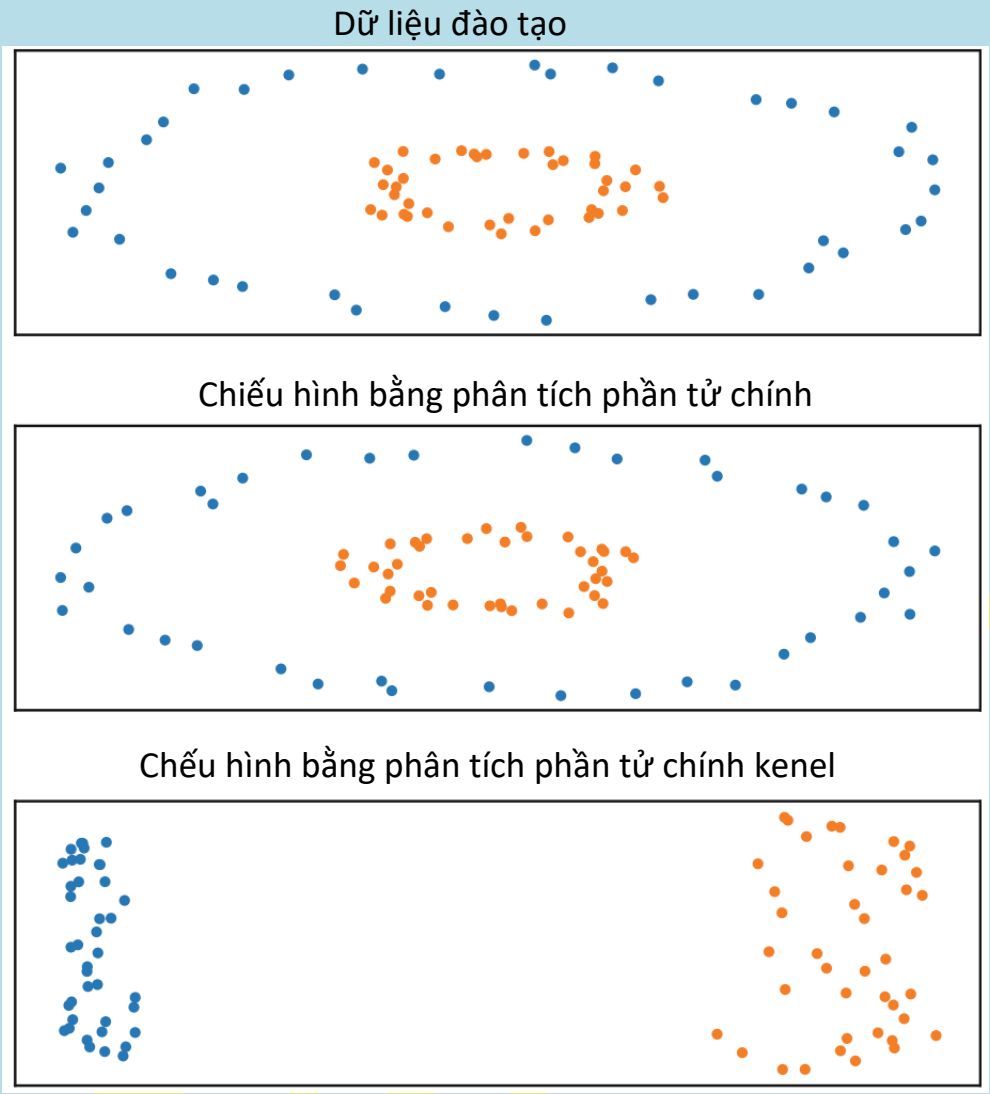
Với phân tích thành phần chính kenel, mục tiêu là tìm được một không gian con trực giao tuyến tính trong không gian con tính năng sao cho mỗi phần tử chính trong không gian tính năng sẽ giải thích hầu hết phương sai. ℓ vectơ đầu tiên $\boldsymbol{\alpha}_{k \; 1 \le k \le \ell}$ của ma trận K cùng với các giá trị riêng được lưu theo thứ tự giảm dần sẽ cho ta lời giải . Các vectơ riêng này được chuẩn hóa vào các vectơ đơn vị trong không gian tính năng.

Cuối cùng, hình chiếu của mọi phần tử $\boldsymbol{x}$ trong không gian gốc lên phần tử k có thể được tính như sau:

$$\phi(\boldsymbol{x})^{T}\boldsymbol{\alpha}_{k} = \sum_{i=1}^{N}\alpha_{ki}\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$$

Hình 2.33 minh họa chiếu hình của một số dữ liệu phân loại khả phân tách không tuyến tính bằng cách phân tích thành phần chính và bằng cách phân tích thành phần chính với một kenel không tuyến tính. Các dữ liệu đầu vào được chiếu hình hình bằng cách sử dụng phân tích phần tử chính kenel trở thành khả phân tách tuyến tính còn dữ liệu đầu vào được chiếu hình bằng cách sử dụng phân tích thành phần chính (tuyến tính) vẫn giữ nguyên khả phân tách không tuyến tính.

BẢN GỐC



Hình 2.30. Minh họa phân tích phần tử chính kenel. Hình trên cho thấy dữ liệu đào tạo khả phân tách không tuyến tính. Hình giữa cho thấy chiếu hình dữ liệu đào tạo sử dụng phân tích phần tử chính và dữ liệu này vẫn giữ nguyên khả phân tách không tuyến tính. Hình cuối cùng cho thấy chiếu hình chiếu dữ liệu đào tạo sử dụng phân tích phần tử chính kenel (bằng một kenel không tuyến tính) và dữ liệu trở nên khả phân tách tuyến tính

2.15. TỔNG KẾT

Chương này trình bày các phương pháp học máy kinh điển. Do khuôn khổ có hạn, các phương pháp này chỉ được trình bày tóm tắt. Người đọc có thể tìm được thông tin chi tiết hơn trong các tài liệu tham khảo. Điểm chung của các tiếp cận được trình bày trong chương này là chúng sử dụng đầu vào như là một tập của các tính năng được cho trước

hoặc được chiết suất trước. Trái lại, các phương pháp học máy thường cung cấp một thiết lập học từ đầu đến cuối trong đó các tính năng được học. Các kỹ thuật này sẽ được trình bày trong các chương sau.

Chương 3

MẠNG TẾ BÀO THẦN KINH SÂU (DEEP NEURAL NETWORK)

Chương này sẽ tập trung lên trình bày và bàn luận về các phần tử chính của các giải thuật học sâu nhằm cung cấp cho người đọc hiểu biết tốt hơn về các nô hình mạnh mẽ này. Các mạng tế bào thần kinh sâu (DNN: Deep Neural Netwwork) hiên nay là nền tảng của nhiều ứng dụng trí tuệ nhân tạo (AI). Vì các ứng dụng đột phá của các DNN cho nhận dạng giọng nói và nhận dạng ảnh, số lượng các sử dụng các DNN đăc biệt bùng nổ. Các DNN ngày nay được sử dụng trong vô vàn các ứng dụng AI bao gồm thị giác máy tính, nhận dạng ảnh, xe tự lái, phát hiện ưng thư, các trò chơi phức tạp và robot. Trong một số ứng dụng, DNN hiện nay có thể vượt trội độ chính xác của con người. Hiệu suất vượt trội của các DNN bắt nguồn từ khả năng của chúng chiết xuất các tính năng mức độ cao từ dữ liệu cảm biến thô sau khi sử dụng học thống kế trên một khối lượng dữ liệu lớn nhận được từ một trình bày hiệu quả một không gian đầu vào. Điều này khác hẳn với các cách tiếp cận trước đây sử dụng các tính năng nhân công hay các quy tắc được thiết kế bới các chuyên gia.

Trong khi các máy tính đa năng, đặc biệt là các đơn vị xử lý đồ họa (GPU: Graphics Processing Unit), đã trở thành trụ cột cho nhiều xử lý DNN, thì người ta ngày càng quan tâm cung cấp tăng tốc được chuyên môn hóa hơn của tính toán DNN.

Trong khi DNN đem lại độ chính xác vượt trội trên nhiều nhiệm vụ AI, thì cái chi phí phải trả là sự phức tạp tính toán. Vì thế các kỹ thuật cho phép xử lý hiệu suất của DNN để cải thiện năng lượng và thông lượng mà không giảm độ chính xác ứng dụng hoặc tăng chi phí thành phần cứng là vấn đề quan trọng để triển khai rộng rãi các DNN trong các hệ thống AI.

Chương này sẽ tập trung lên trình bày và bàn luận về các phần tử chính của các DNN nhằm cung cấp cho người đọc hiểu biết tốt hơn về các mô hình mạng mạnh mẽ này.

3.1. TRÍ TUỆ NHÂN TẠO (AI) VÀ MẠNG THẦN KINH SÂU (DNN)

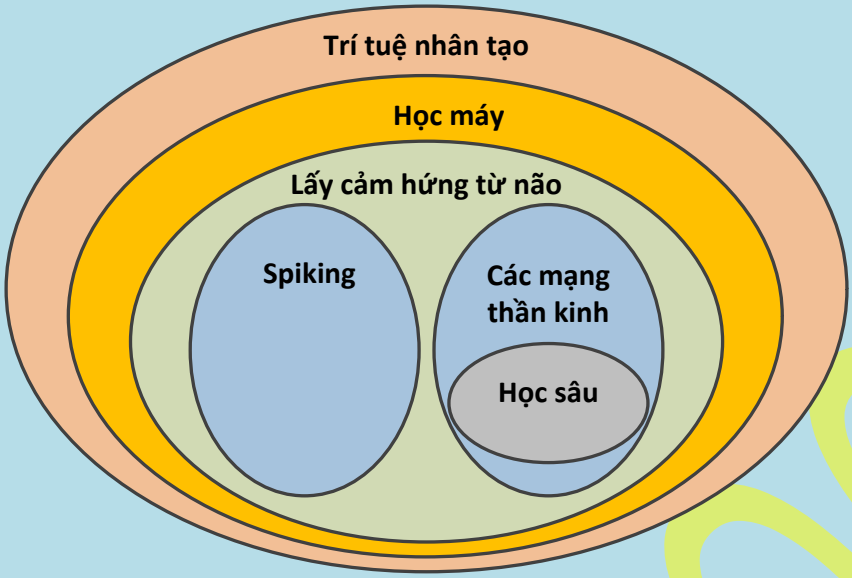
3.1.1. DNNN và trí tuệ nhân tạo

DNN còn được gọi là học máy sâu (DL: Deep Learning) là một phân của lĩnh vực rộng của AI. Trí tuệ nhân tạo (AI) là khoa học và kỹ thuật để chế tạo những cỗ máy thông minh có khả năng thực hiện các nhiệm vụ như con người, được xác định bởi John McCarthy vào năm 1956. Quan hệ gữa học máy sâu và toàn bộ trí tuệ nhân tạo được minh họa trên hình 3.1.

Trong trí tuệ nhân tạo, có một lĩnh vực con lớn được gọi là học máy được Arthur Samuel định nghĩa vào năm 1959 như là một lĩnh vực nghiên cứu cho phép các máy tính khả năng học mà không cần được lập trình tường minh. Nghĩa là một chương trình duy nhất , được tạo lập một lần, có khả năng học cách thực hiện một số hoạt động thông minh bên ngoài khái niệm lập trình. Điều này trái ngược với các chương trình được xây dựng theo mục đích có hành vi được định nghiã theo kinh nghiệm thủ công định nghĩa tường minh và tĩnh tại hành vi của chúng.

Ưu điểm của một giải thuật học máy hiệu quả hoàn toàn rõ ràng. Thay vì cách tiếp cận tốn nhiều công sức và may rủi là tạo ra một chương trình tùy chỉnh, riêng biệt để giải quyết từng vấn đề riêng lẻ trong một lĩnh vực, một thuật toán ML duy nhất chỉ cần học, thông qua một quá trình gọi là đào tạo, để xử lý từng vấn đề mới.

Trong lĩnh vực học máy, có một lĩnh vực thường được đề cập đến như là tính toán lấy cảm hứng từ não Vì não hiện nay là ‘máy’ tốt nhất mà ta biết để học và giải quyết các vấn đề, nên đây là một vị trí tự nhiên để tìm kiếm phương pháp học máy. Vì thế tính toán lấy cảm ứng từ não là một chương trình hay một giải thuật lấy một số khía cạnh của dạng cơ sở của não hay chức năng từ cách thức mà não làm việc. Điều này trái ngược với chương trình nhằm mô phỏng một số nét theo cách mà ta hiểu hoạt động của não.



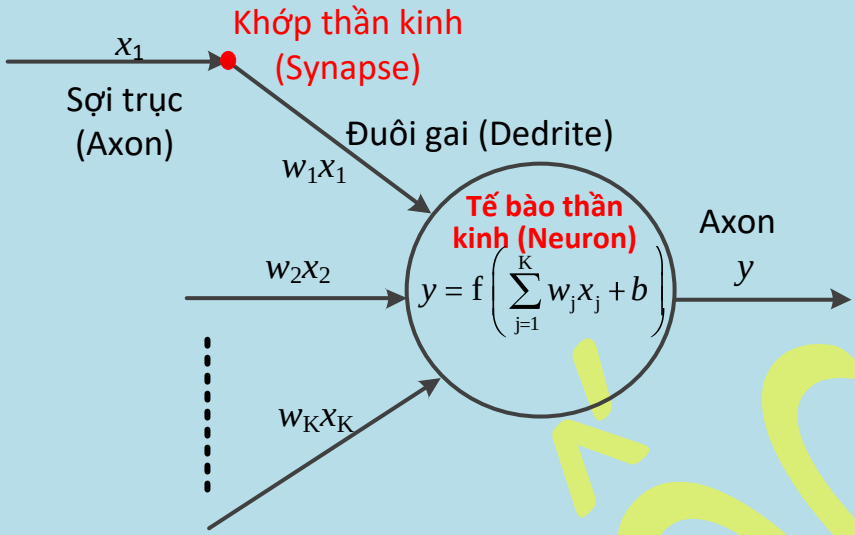
Spiking: điện toán tăng vọt

Hình 3.1. Học sâu trong ngữ cảnh trí tuệ nhân tạo

Vào năm 1958 Frank Rosenblatt phát kiến ra perceptron . Perceptron là tế bào thần kinh nhân tạo thực tế đầu tiên được xây dựng dưa trên cảm hứng từ não. Với một số cải biên, perceptron trở thành khối cơ sở của các giải thuật học sâu. Phần dưới đây ta sẽ xét mô hình tế bào thần kinh nhân tạo này.

3.1.2. Mô hình tế bào thần kinh nhân tạo

Trong lĩnh vực ML, có một lĩnh vực thường được gọi là tính toán lấy cảm hứng từ não (brain-inspired computation), là một chương trình nhằm mô phỏng một số khía cạnh của cách chúng ta hiểu não hoạt động. Vì người ta tin rằng các yếu tố tính toán chính của bộ não con người là 86 tỷ tế bào thần kinh, hai phân khu vực của tính toán lấy cảm hứng từ não đều được lấy cảm hứng từ kiến trúc của tế bào thần kinh, như thể hiện trong hình 3.2.



Hình 3.2. Kết nối với tế bào thần kinh trong não. x_i, w_i, f(•) và b lần lượt là các kích hoạt, trọng số, hàm phi tuyến tính và độ lệch

Bản thân các tế bào thần kinh (Neuron) được kết nối với nhau với một số nguyên tố đi vào chúng được gọi là đuôi gai (Dendrite) và một nguyên tố rời khỏi chúng được gọi là sợi trục như trong Hình 3.2. Tế bào thần kinh chấp nhận các tín hiệu đi vào nó thông qua các đuôi gai, thực hiện tính toán trên các tín hiệu đó và tạo ra tín hiệu trên sợi trục (Axon). Các tín hiệu đầu vào và đầu ra này được gọi là kích hoạt (Activation). Sợi trục của một tế bào thần kinh phân nhánh và được kết nối với các đuôi gai của nhiều tế bào thần kinh khác. Các kết nối giữa một nhánh của sợi trục và một đuôi gai được gọi là khớp thần kinh (Synapse). Ước tính có 10^{14} đến 10^{15} khớp thần kinh trong não người trung bình.

Một đặc điểm chính của khớp thần kinh là nó có thể chia tỷ lệ tín hiệu (x_j) đi qua nó như thể hiện trong Hình 3.2. Hệ số tỷ lệ đó có thể được gọi là trọng số (w_j) và cách não được cho là học là thông qua những thay đổi đối với trọng số liên quan đến khớp thần kinh. Do đó, các trọng số khác nhau dẫn đến các phản hồi khác nhau đối với một đầu vào. Lưu ý rằng học tập là sự điều chỉnh các trọng số để đáp ứng với một kích thích học tập, trong khi tổ chức (những gì có thể được coi là chương trình) của não không thay đổi. Đặc điểm này làm cho bộ não trở thành nguồn cảm hứng tuyệt vời cho thuật toán kiểu học máy.

Trong mô hình máy tính lấy cảm hứng từ não bộ có một lĩnh vực phụ được gọi là điện toán spiking (Spike: gai hay tăng vọt). Trong tiểu khu vực này, cảm hứng được lấy từ thực tế là truyền thông trên các đuôi gai và sợi trục là các xung giống gai và thông tin được truyền tải không chỉ dựa trên biên độ của gai. Thay vào đó, nó cũng phụ thuộc vào thời gian xung đến và tính toán xảy ra trong tế bào thần kinh là một hàm không chỉ của

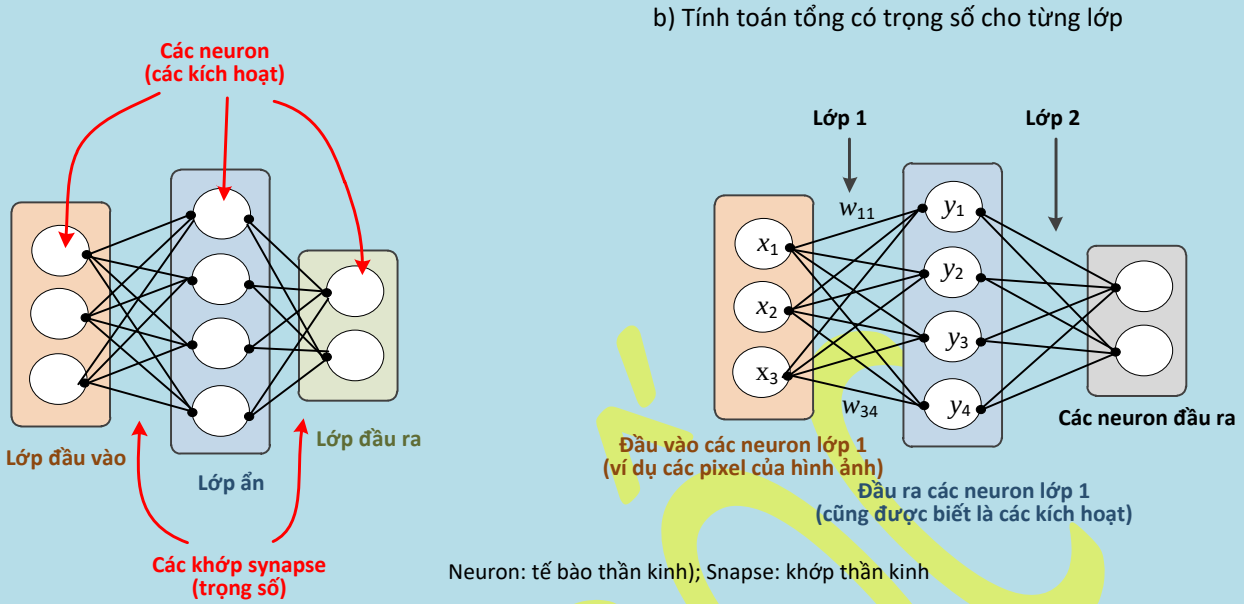
một giá trị duy nhất mà còn là độ rộng của xung và mối quan hệ thời gian giữa các xung khác nhau. Một ví dụ về một dự án được lấy cảm hứng từ sự tăng đột biến của não là IBM TrueNorth . Trái ngược với điện toán gai, một lĩnh vực phụ khác của điện toán lấy cảm hứng từ não bộ được gọi là mạng neuron được tập trung trình bầy trong các phần dưới đây.

So với các phương pháp tiếp cận điện toán tăng vọt (Spike Approach), các phương pháp ML phổ biến hơn là sử dụng "mạng neuron" làm mô hình. Dưới đây ta sẽ xét các mạng neuron.

3.1.3. Mạng neuron và mạng neuron sâu (DNN: Deep Learning Network)

Mạng neuron lấy cảm hứng từ quan niệm rằng tính toán của tế bào thần kinh liên quan đến tổng có trọng số của các giá trị đầu vào. Các tổng có trọng số này tương ứng với tỷ lệ giá trị được thực hiện bởi các khớp thần kinh và sự kết hợp của các giá trị đó trong tế bào thần kinh. Hơn nữa, tế bào thần kinh không chỉ xuất ra tổng có trọng số đó, vì tính toán được liên kết với một tầng tế bào thần kinh sau đó sẽ là một phép toán đại số tuyến tính đơn giản. Thay vào đó, có một hoạt động chức năng trong tế bào thần kinh được thực hiện trên các đầu vào kết hợp. Phép toán này dường như là một hàm phi tuyến tính khiến tế bào thần kinh chỉ tạo ra đầu ra nếu các đầu vào vượt qua một số ngưỡng nào đó. Do đó, bằng cách tương tự, mạng neuron áp dụng một hàm phi tuyến tính cho tổng trọng số của các giá trị đầu vào. Chúng ta sẽ xem xét một số hàm phi tuyến tính đó là gì trong các phần sau.

Hình 3.3 (a) cho thấy một bức tranh sơ đồ của một mạng neuron tính toán. Các tế bào thần kinh trong lớp đầu vào (Input Layer) nhận một số giá trị và truyền chúng đến các tế bào thần kinh ở lớp giữa của mạng, thường được gọi là 'lớp ẩn' (Hidden Layer). Các tổng có trọng số (Weighted Sumes) từ một hoặc nhiều lớp ẩn cuối cùng được truyền đến lớp đầu ra (Output Layer), lớp này trình bày đầu ra cuối cùng của mạng cho người dùng. Để điều chỉnh thuật ngữ lấy cảm hứng từ não bộ với mạng neuron, đầu ra của các tế bào thần kinh thường được gọi là kích hoạt và các khớp thần kinh (Synapse) thường được gọi là trọng số như trong Hình 3.3 (a). Trong các phần dưới đây các các thuật ngữ kích hoạt/trọng số sẽ được sử dụng.



Neuron: tế bào thần kinh); Snapse: khớp thần kinh.

Hình 3.3. Thí dụ về mạng neuron đơn giản và thuật ngữ

Hình 3.3 (b) cho thấy một ví dụ về mạng neuron trong đó lớp đầu vào có ba neuron với cá pixel ảnh l là x_1, x_2 và x_3 và lớp ẩn có bốn neuron với các đầu ra là y_1, y_2, y_3 và y_4. Đầu ra của neuron j được tính như sau:

$$y_j = f\left(\sum_{i=1}^{3} w_{ij} \times x_i + b\right) = f\left(\boldsymbol{w}_j^T \times \boldsymbol{x} + b\right) = f\left(\boldsymbol{x}^T \boldsymbol{w}_j \times + b\right) \qquad (3.1)$$

Trong đó

$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$, $\boldsymbol{w}_j = \begin{bmatrix} w_{1,j} & w_{2,j} & w_{3,j} \end{bmatrix}^T$, b là độ lệch

i=1,2,3 và j=1,2,3,4

$\boldsymbol{x}$, $\boldsymbol{w}_j$,và y_j là vectơ kích hoạt đầu vào, vectơ trọng số, kích hoạt đầu ra của neuron j và f (•) là một hàm phi tuyến tính được mô tả trong phần sau.

Tổng quát cho mạng neuron trong đó lớp đầu vào có K neuron và lớp ẩn a có N neuron, đầu ra của neuron j được tính như sau: :

$$y_j = f\left(\sum_{i=1}^{K} w_{ij} \times x + b\right) = f\left(\boldsymbol{w}_j \times \boldsymbol{x} + b\right)$$

Nếu kết hợp độ lệch *b* vào vectơ trọng số như đã xét trong chương trước, ta có:

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_1 & \dots & x_{K-1} & x_K \end{bmatrix}^T, \boldsymbol{w}_j = \begin{bmatrix} b & w_{1,j} & \dots & w_{K-1,j} & w_{K,j} \end{bmatrix}^T$$

i=1,2,…, K và j=1,2,…,N

Ta có thể biểu diễn phương trình (3.1) vào dạng tích của các ma trận như sau:

$$\boldsymbol{y} = \mathrm{f}\left(\boldsymbol{W}^T \boldsymbol{x}\right) = \mathrm{f}\left(\boldsymbol{x}^T \boldsymbol{W}\right) \qquad (3.2)$$

Trong đó $\boldsymbol{y} = \left[y_1, \dots, y_j, \dots, y_N \right]^T$ là vectơ của N đầu ra của lớp ẩn và $\boldsymbol{x} = \left[1 \;\; x_1, \dots, x_i, \dots, x_K \right]^T$ là vectơ của K đầu vào của lớp đầu vào, T ký hiệu cho đảo vị và ma trận trọng số:

$$\boldsymbol{W} = \begin{bmatrix} w_{0,1} & \dots & w_{0,j} & \dots & w_{0,N} \\ w_{1,1} & & w_{1,j} & & w_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i,1} & \dots & w_{i,j} & \dots & w_{i,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{K,1} & \dots & w_{K,j} & \dots & w_{K,N} \end{bmatrix}$$

Trong đó $w_{0,j}$ là độ lệch cho từng đầu vào của lớp ẩn.

Trong lĩnh vực mạng neuron, có một lĩnh vực được gọi là học sâu (Deep Learning), trong đó mạng neuron có nhiều hơn ba lớp, tức là nhiều hơn một lớp ẩn. Ngày nay, số lượng lớp mạng điển hình được sử dụng trong deep learning dao động từ năm đến hơn một nghìn. Trong phần này, ta sẽ sử dụng thuật ngữ mạng neuron sâu (DNN: Deep Learning Network) để chỉ các mạng neuron được sử dụng trong học sâu.

DNN có khả năng học các tính năng cấp cao với độ phức tạp và trừu tượng hơn so với mạng neuron nông hơn. Một ví dụ minh họa điểm này là sử dụng DNN để xử lý dữ liệu thị giác. Trong các ứng dụng này, các pixel của hình ảnh được đưa vào lớp đầu tiên của DNN và đầu ra của lớp đó có thể được hiểu là đại diện cho sự hiện diện của các đặc điểm cấp thấp khác nhau trong hình ảnh, chẳng hạn như đường thẳng và cạnh. Ở các lớp tiếp theo, các đặc điểm này sau đó được kết hợp thành một thước đo về sự hiện diện có thể xảy ra của các đặc điểm cấp cao hơn, ví dụ: các đường thẳng được kết hợp thành các hình

BẢN GỐC

dạng, được kết hợp thêm thành các tập hợp các hình dạng. Và cuối cùng, với tất cả thông tin này, mạng cung cấp xác suất rằng các tính năng cấp cao này bao gồm một đối tượng hoặc cảnh cụ thể. Hệ thống phân cấp tính năng sâu này cho phép DNN đạt được hiệu năng vượt trội trong nhiều nhiệm vụ.

3.143. Đào tạo và suy luận

Vì DNN là một ví dụ của thuật toán học máy, chương trình cơ bản không thay đổi khi nó học cách thực hiện các nhiệm vụ nhất định của nó. Trong trường hợp cụ thể của DNN, việc học này liên quan đến việc xác định giá trị của trọng số (và độ lệch: Bias) trong mạng và được gọi là đào tạo mạng. Sau khi được đào tạo, chương trình có thể thực hiện nhiệm vụ của mình bằng cách tính toán đầu ra của mạng bằng cách sử dụng trọng số được xác định trong quá trình đào tạo. Chạy chương trình với các trọng số này được gọi là suy luận.

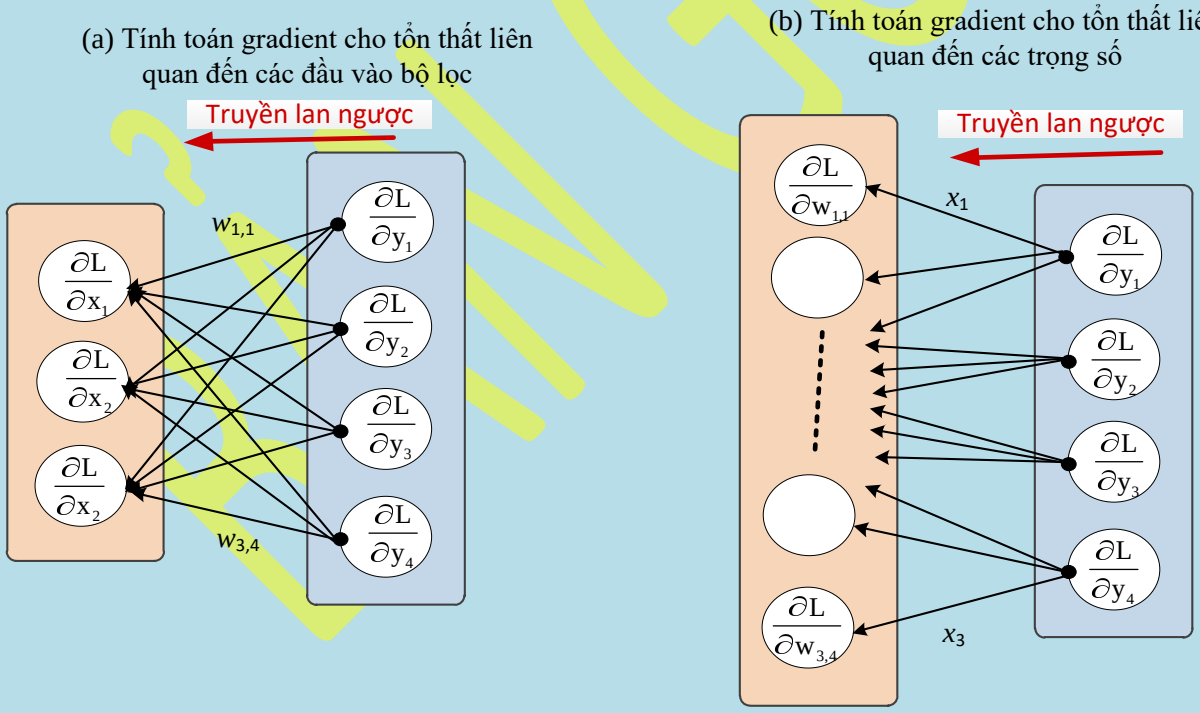
Khi thực hiện suy luận sử dụng DNN, một ảnh đầu vào được cho trước và đầu ra của DNN là một vectơ của các điểm số (Scores), mỗi điểm số cho một loại đối tượng. Loại có có điểm số cao nhất chỉ thị rằng loại này có khả năng cao nhất là đối tượng trong ảnh. Mục đích bao trùm đối với đào tạo là xác định trong số dẫn đến cực đại hóa điểm số của loại đúng trong ảnh. Khi đào tạo mạng, loại đúng thường đã được biết, nó được cho trước đối các ảnh được sử dụng cho đào tạo (ví dụ: tập đào tạo của mạng). Sự khác biệt giữa các điểm số đúng lý tưởng và các điểm số được tính toán bởi DNN dựa trên các trọng số của nó được gọi là mất mát (L: Loss). Như vậy, mục đích của đào tạo các mạng DNN là tìm được tập các trọng số cho phép giảm thiểu mất mát trung bình trên một tập đào tạo lớn.

Đào tạo (Traning) là một quá trình trong đó mô hình AI/ML học cách thực hiện các nhiệm vụ nhất định của nó, cụ thể hơn là bằng cách tối ưu hóa giá trị của trọng số trong DNN. DNN được đào tạo bằng cách nhập một tập huấn luyện, thường là các mẫu đào tạo được dán nhãn chính xác. Ví dụ, phân loại hình ảnh, bộ đào tạo bao gồm các hình ảnh được phân loại chính xác. Khi đào tạo một mạng, các trọng số (w_{ij}) thường được cập nhật bằng cách sử dụng quy trình tối ưu hóa leo đồi (hill-climbing) được gọi là giảm độ dốc (gradient descent). Độ dốc cho biết trọng số sẽ thay đổi như thế nào để giảm mất mát (khoảng cách giữa đầu ra chính xác và đầu ra do DNN tính toán dựa trên trọng số hiện tại của nó). Bội số của gradient của mất mát so với từng trọng số là đạo hàm riêng được sử dụng để cập nhật trọng số (ví dụ: trọng số cập nhật $\mathbf{w}_{ij}^{t+1} = \mathbf{w}_{ij}^{t} - \alpha \frac{\partial L}{\partial \mathbf{w}_{ij}}$, trong đó α được gọi là tốc độ học) . Lưu ý rằng gradient chỉ thị rằng các trọng số cần thay đổi như thế nào

để giảm mất mát. Quá trình đào tạo được lặp đi lặp lại để liên tục giảm mất mát tổng thể. Cho đến khi mất mát dưới ngưỡng xác định trước, DNN với độ chính xác cao sẽ thu được.

Cách tính toán các đạo hàm riêng của gradient hiệu quả là thông qua một quá trình được gọi là truyền lan ngược (Backpropagation). Truyền lan ngược, tính toán được rút ra từ một quy tắc tính toán dây truyền (Chain Rule), hoạt động bằng cách chuyển ngược qua mạng các giá trị để tính toán mất mát đã chịu ảnh hưởng của từng trọng số như thế nào.

Thực ra, tính toán truyền lan ngược có dạng rất giống với tính toán được sử dụng cho suy luận như thấy trên hình 3.4. Như vậy các kỹ thuật thực hiện hiệu quả suy luận có thể đôi khi hữu ích cho thực hiện hiệu quả đào tạo. Tuy nhiên cần lưu ý hai điểm. Thứ nhất, truyền lan ngược yêu cầu các đầu ra trung gian của mạng được dành trước cho tính toán ngược,vì thế đào tạo sẽ tăng các yêu cầu lưu trữ. Thứ hai, do sử dụng các gradient cho leo đồi (Hill-Climbing), yêu cầu độ chính xác cho đào tạo nói chung sẽ cao hơn suy luận.



Hình 3.4. Thí dụ về truyền lan ngược

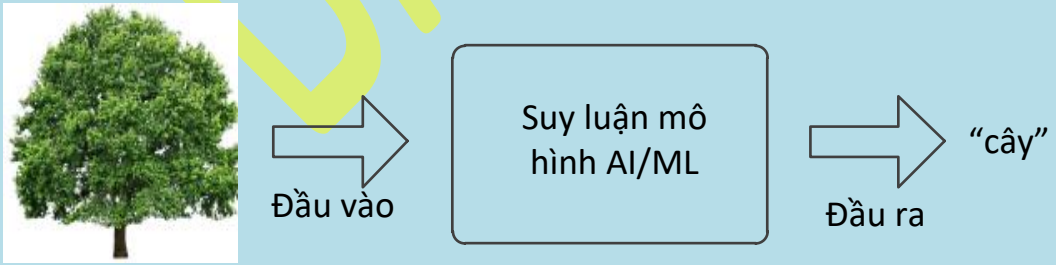
Nhiều loại kỹ thuật được sử dụng để cải thiện hiệu quả cũng như độ bền vững của đào tạo. Thí dụ, thường thì mất mát từ nhiều tập dữ liệu đầu vào (ví dụ: một lô) được thu thập

trước khi thực hiện chuyển một lần cập nhật trọng số. Điều này giúp tăng tốc và ổn định quá trình đào tạo.

Có nhiều cách để đào tạo mạng (đào tạo trọng số) cho các mục tiêu khác nhau. Giới thiệu ở trên là học có giám sát (supervised learning) sử dụng các mẫu đào tạo được gắn nhãn để tìm đầu ra chính xác cho một nhiệm vụ. Học không giám sát (unsupervised learning) sử dụng các mẫu đào tạo không được gắn nhãn để tìm cấu trúc hoặc cụm trong dữ liệu. Học tăng cường (Reinforcement learning) có thể được sử dụng để đưa ra hành động mà tác nhân nên thực hiện tiếp theo để tối đa hóa phần thưởng mong đợi. Học chuyển giao (transfer learning) là điều chỉnh các trọng số đã được đào tạo trước đó (ví dụ: trọng số trong mô hình toàn cục) bằng cách sử dụng một tập huấn luyện mới, được sử dụng để đào tạo nhanh hơn hoặc chính xác hơn cho một mô hình được cá nhân hóa.

Mộ cách tiếp cận khác thường được sử dụng để xác định các trọng số là điều chỉnh tinh (Fine-Tuning), trong đó các trọng số được đào tạo trước đây có sẵn và được sử dụng như là một điểm bắt đầu và sau đó các trọng số này được điều chỉnh cho một tập dữ liệu mới (ví dụ: học chuyển giao (Transfer Learning)). Điều này dẫn đến đào tạo nhanh hơn và đôi khi có thể dẫn đến độ chính xác tốt hơn.

Sau khi DNN được đào tạo, nó có thể thực hiện nhiệm vụ của mình bằng cách tính toán đầu ra của mạng bằng cách sử dụng trọng số được xác định trong quá trình đào tạo, được gọi là suy luận (Inference). Trong quá trình suy luận mô hình, đầu vào từ thế giới thực được truyền qua DNN. Sau đó, dự đoán (Prediction) cho nhiệm vụ là đầu ra, như trong hình 3.5. Ví dụ: đầu vào có thể là pixel của hình ảnh, biên độ lấy mẫu của sóng âm thanh hoặc biểu diễn số của trạng thái của một số hệ thống hoặc trò chơi. Tương ứng, đầu ra của mạng có thể là xác suất mà một hình ảnh chứa một đối tượng cụ thể, xác suất mà một chuỗi âm thanh chứa một từ cụ thể hoặc một hộp giới hạn trong hình ảnh xung quanh một đối tượng hoặc hành động được đề xuất nên được thực hiện .



Hình 3.5. Thí dụ suy luận mô hình AI/ML

Hiệu năng của DNN đạt được với chi phí của độ phức tạp tính toán cao. Do đó, các công cụ tính toán hiệu quả hơn thường được sử dụng, ví dụ như đơn vị xử lý đồ họa (GPU:

graphics processing unit) và bộ xử lý mạng (NPU: network processing unit). So với suy luận chỉ liên quan đến quá trình chuyển thẳng (feedforward process), việc đào tạo thường đòi hỏi nhiều tài nguyên tính toán và lưu trữ hơn vì nó cũng liên quan đến quá trình lan truyền ngược.

Những năm gần đây các bộ khung học sâu (Deep Framework) trở nên rất phổ biên thu hút rất nhiều quan tâm từ cộng đồng nghiên cứu. Các bộ khung này cung cấp các sơ đồ học máy mà không cần kỹ thuật tạo tính năng (Feature Engineering), nhưng vẫn khá linh hoạt. Ban đầu được phát triển cho các nhiệm vụ giam sát, ngày nay chúng được mở rộng cho nhiều kịch bản khác.
Học sâu , nói một cách chặt chẽ, liên quan đến sử dụng nhiều lớp các tế bào thần kinh (neuron) nhân tạo. Các mạng thần kinh đầu tiên được phát triển vào cuối thập kỷ 1950 với việc trình bày các giải thuật perceptron. Tuy nhiên do các hạn chế liên quan đến chi phí tính toán các giải thuật này trong giai đoạn này và các hiểu nhầm rằng các perceptron không thể học các hàm phi tuyến tính như XOR dẫn đến giảm dần mối quan tâm đến nghiên cứu tiếp các giải thuật này. Mặc dù trong giai đoạn này không còn nhiều quan tâm nghiên cứu đối với các perceptron, một số giải thuật vô cùng quan trọng như giai thuật truyền lan ngược (Backpropagation) và các mạng tế bào thần kinh hồi quy đã được đưa ra.
Sau giai đoạn này, nhiều nghiên cứu đã chỉ ra các phương pháp hiệu quả đào tạo các perceptron nhiều lớp theo từng lớp bằng cách xử lý từng lớp như là một máy Bolztmann hạn chế không được giám sát sau đó sử dụng truyền lan ngược được giám sát để vi chỉnh. Các tiến bộ như vậy trong các giải thuật tối ưu hóa và đặc biệt trong các đơn vị xử lý đồ họa (GPU: Graphics Processing Unit) đã tăng tốc độ tính toán của các hệ thống học sâu àm cho việc đào tạo chũng dễ hơn và nhanh hơn. Hơn nữa năm 2010, các tập dữ liệu lớn, với ImageNet là một trong số các tập dữ liệu phổ biến nhất đã được cung cấp, đóng góp cho thành công của các giải thuật học máy sâu và cho phép chứng minh thực nghiêm hiệu suất vượt trột của chúng đối với một số nhiệm vụ so với các giải thuật học máy thường được sử dụng khác. Cuối cùng một nhân tố quan trọng đòng góp cho sự phổ biến của học máy sâu là sự hỗ trợ của các thư viện có thể tiếp cận công khai và dễ sử dụng như Theno, Caffe, Tensorflow, Keras và Pythorch. Tất nhiên, hiện nay, nhờ tất cả các thư viện có thể tiếp cận công khai này hỗ trợ nghiên cứu cộng tác và khả lặp (Reproduciple Reseach) và tiếp cập đến các nguồn tài nguyên của các công ty lớn như Kaggle, Google Golab và Amason Web Services, dậy và nghiên cứu các giải thuật này đã trở nên dễ ràng hơn nhiều.

DNN đã tạo ra những bước đột phá đáng kinh ngạc kể từ những năm 2010 trong nhiều lĩnh vực ứng dụng thiết yếu vì chúng có thể đạt được độ chính xác ở cấp độ con người hoặc thậm chí vượt quá độ chính xác của con người. Các kỹ thuật học sâu sử dụng các chiến lược có giám sát và / hoặc không giám sát để tự động học các biểu diễn phân cấp trong kiến trúc sâu để phân loại. Với một số lượng lớn các lớp ẩn, hiệu suất vượt trội của DNN đến từ khả năng trích xuất các đặc điểm cấp cao từ dữ liệu cảm giác thô sau khi sử dụng học thống kê trên một lượng lớn dữ liệu để đạt được trình bày hiệu quả một không gian đầu vào. Trong những năm gần đây, nhờ dữ liệu lớn (Big Data) thu được từ thế giới thực, khả năng tính toán tăng nhanh và các thuật toán phát triển liên tục, các DNN đã trở thành các mô hình phổ biến bật nhất cho nhiều ứng dụng AI.

3.2. CÁC MẠNG SÂU NỐI THẲNG (DEEP FEEDFORWARD NETWORKS)

Các mạng sâu nối thẳng là một tập các mô hình tham số phi tuyến tính và thể hiện phân cấp, nó được tối ưu hóa bằng cách giảm gradient hỗn loạn. Trong định nghĩa này thuật ngữ mạng tham số vì các tham số cần được sử dụng trong quá trình đào tạo, phi tuyến tính vì các hàm phi tuyến tính cần được tạo ra và thể hiện phân cấp vì đầu ra của một hàm được sử dụng như làm đầu vào của của hàm tiếp theo theo cách phân cấp.

3.2.1. Perceptron

Perceptron ban đầu được phát triển cho các vấn đề phân loại nhị phân có giám sát. Nó được xây dựng xung quanh một neuron phi tuyến tính, mô hình một neuron của McCulloch-Pitts. Chính thức hơn, ta tìm một hàm $f(\boldsymbol{x};\boldsymbol{w},b)$, sao cho $f(.;\boldsymbol{w},b):\boldsymbol{x}=\mathbb{R}^K\rightarrow +1,-1$, trong đó $\boldsymbol{w}$, b là vectơ trọng số và độ lệch được gọi là các tham số của f và vectơ $\boldsymbol{x}=x_1,x_2,\ldots,x_K^{\ T}$ là đầu vào. Tập đào tạo là $\boldsymbol{x}_i,y_i$. Cụ thể, perceptron dựa trên mô hình tuyến tính để thực hiện phân loại:

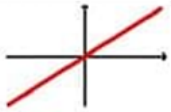
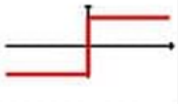
$$f(\boldsymbol{x};\boldsymbol{w},b)=f(z)=\begin{matrix}+1 \text{ if } z\geq 0\\ -1 \text{ otherwwise}\end{matrix}, \qquad (3.5)$$

Trong đó $z=\boldsymbol{w}^T\boldsymbol{x}$, trong đó $\boldsymbol{w}^T$ là chuyển vị vectơ $\boldsymbol{w}$, $\boldsymbol{w}=[w_1,w_2,\ldots,w_K]^T$ là vecơ trọng số,$\boldsymbol{x}=[x_1, x_2,\ldots, x_K]^T$ là vectơ đầu vào, b là độ lệch..

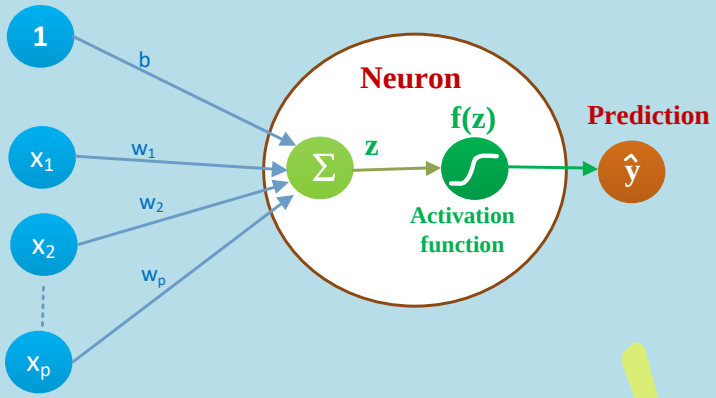
Có thể diễn giải hình học mô hình này như là một siêu mặt phẳng có thể chia các điểm dữ liệu tương ứng mà các điểm này có thể phân tách tuyến tính. Ngoài ra, ta có thể quan sát thấy rằng, trong định nghĩa trên , một perceptron là là một kết hợp của một tổng được đánh trọng số giữa các phần tử của vectơ đầu vào $\boldsymbol{x}$ được kết hợp với hàm kích hoạt và một quy định ngưỡng ấn định loại +1 hoặc -1thực hiện quyết định cho phân loại. Các

hàm kích hoạt có thể là các hàm được cho trong bảng 3.1 , ví dụ: hàm bước (step), sigmoid, hyperbolic trangent hay hàm dấu (sign) .

Bảng 3.1. Các loại hàm kích hoạt khác nhau có thể được sử dụng cho perceptron

Activation Function	Equation	Example	1D Graph
Linear	$\phi(z) = z$	Adaline, linear regression	
Unit Step (Heaviside Function)	$\phi(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Sign (signum)	$\phi(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Piece-wise Linear	$\phi(z) = \begin{cases} 0 & z \leq -\frac{1}{2} \\ z + \frac{1}{2} & -\frac{1}{2} \leq z \leq \frac{1}{2} \\ 1 & z \geq \frac{1}{2} \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multilayer NN	
Hyperbolic Tangent (tanh)	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multilayer NN, RNNs	
ReLU	$\phi(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$	Multilayer NN, CNNs	

Về mặt đồ thị, perceptron được trình bày trên hình 3.6, trên hình này mỗi phần tử đầu vào được mô tả như là một neuron và tất các phần tử này được kết hợp bằng cách đánh trọng số với các tham số của mô hình và sau đó chuyển đến một hàm kích hoạt cho quyết định cuối cùng.



Neuron: tế bào thần kinh;
Activation Function: hàm kích hoạt
Prediction: dự đoán

Hình 3.6. Một mô hình perceptron đơn giản. Các phần tử đầu vào được trình bày như các neuron và được kết hợp cho dự báo cuối cùng $\hat{y}$. Dự đoán cuối cùng được soạn thảo từ một tổng được đánh trọng số và một hàm kích hoạt.

Trong quá trình đào tạo và giống như các giải thuật học máy khác, ta cần tìm ra các tham số tối ưu $\boldsymbol{w}$ và b cho mô hình perceptron. Một trong các ý tưởng mới của Rosenbatt là đề xuất giải thuật học (Learning Algorithm) bằng một quá trình lặp

Đầu tiên các trọng số được khởi đầu một cách ngẫu nhiên và sau đó bằng cách sử dụng một mẫu $\boldsymbol{x}_i, y_i$ của tập đào tạo, dự báo của perceptron được tính toán. Nếu dự báo đúng, không cần hành động tiếp theo và điểm số liệu tiếp theo được xử lý. Nếu dự báo sai, các trọng số được cập nhật theo quy tắc sau: các trọng số được tăng trong trường hợp dự báo nhỏ hơn nhãn sự thật cơ bản (ground truth label) y_i và giảm nếu dự báo cao hơn nhãn sự thật cơ bản. Quá trình này được lặp cho đến khi không còn lỗi. Mã giả (pseudocode) của đào tạo hay giải thuật hội tụ được trình bày trong bảng 3.2 (lưu ý rằng phiên bản này cho trường hợp giả thết là số liệu là phải có thể phân tách tuyến tính).

Bảng 3.2. Mã giả (pseudocode) của đào tạo

Algorithm 1 Train Perception

Procedure TRAIN $(\{\boldsymbol{x}_i, y_i\})$

Initialization: initialize randomly the weights **w** and bias b

while $\exists i \in \{1,\ldots,N\}, f(\boldsymbol{x}_i, \boldsymbol{w}, b) \neq y_i$ **do**

Pick i randomly

error= $y_i - f(\boldsymbol{x}_i, \boldsymbol{w}, b)$

if error $\neq 0$ **then**

$\mathbf{w} \leftarrow \mathbf{w} + \text{error}.\boldsymbol{x}_i$

$b \leftarrow b + \text{error}$

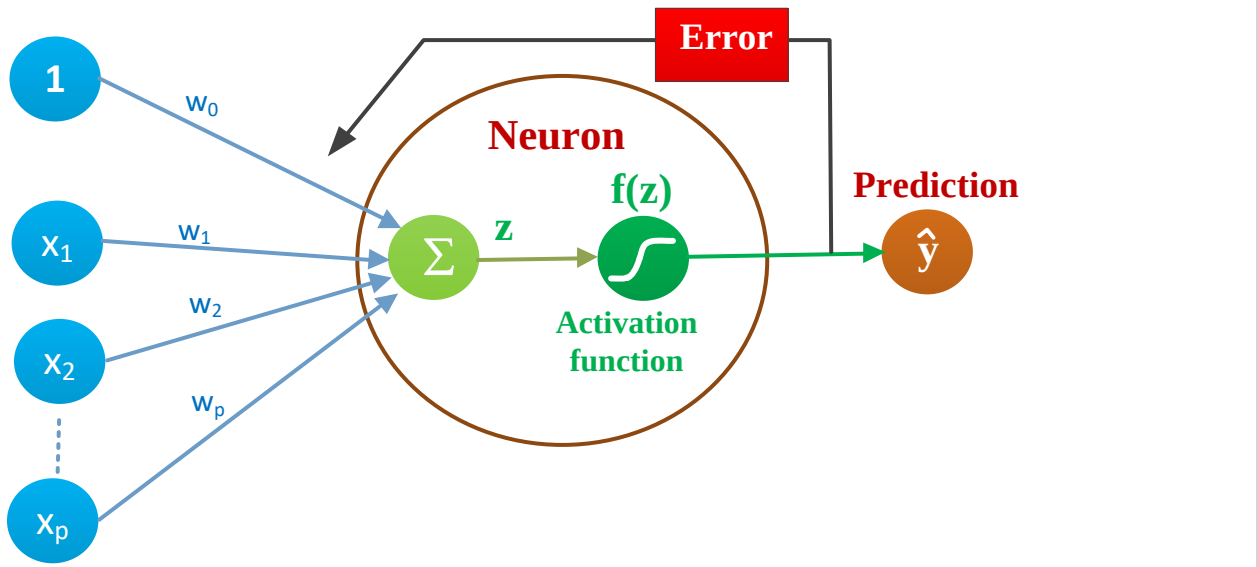
Lúc đầu, được đề xuất cho các nhiệm vụ phân loại nhị phân. Tuy nhiên, giải thuật này có thể được tổng quát hóa cho trường phợp phân loại nhiều loại: $f_c(\boldsymbol{x},\boldsymbol{w},b)$ trong đó $c\in\{1,\ldots.,C\}$ là các loại khác nhau. Có thể dễ ràng đạt được điều này bằng cách bổ sung thêm các neuron vào lớp đầu ra của perceptron. Có thể thực hiện quyết định cuối cùng bằng cách chọn cực đại của các neuron đầu ra khác nhau $f_n = \max_{c\in 1,\ldots,C} f_c(\boldsymbol{x},\boldsymbol{w},b)$.

Cuối cùng, ta sẽ tích hợp độ lệch b vào các trọng số $\boldsymbol{w}$ (và bổ sung 1 như là phần tử đầu tiên của vectơ đầu vào $\boldsymbol{x} = 1, x_1,\ldots,x_K{}^T$). Sau đó có thể viết lại mô hình này như sau: f $\boldsymbol{x},\boldsymbol{w}$ sao cho f $.,\boldsymbol{w} : \boldsymbol{x}\in\mathbb{R}^{K+1}\rightarrow +1,-1$.

Tổng kết lại, các thành phần cơ bản của perceptron như sau:

1. Đầu vào: là một neuron
2. Trọng số (w): neuron đầu vào được liên kết với một trọng số, biểu thị cường độ kết nối giữa neuron đầu vào và neuron đầu ra.
3. Độ lệch (b): Một thuật ngữ độ lệch được thêm vào lớp đầu vào để cung cấp cho perceptron tính linh hoạt bổ sung trong việc mô hình hóa các mẫu phức tạp trong dữ liệu đầu vào.
4. Hàm kích hoạt: Hàm kích hoạt xác định đầu ra của perceptron dựa trên tổng trọng số của các đầu vào và thuật ngữ bias (độ lệch). Các hàm kích hoạt phổ biến được sử dụng trong perceptron được cho trong bảng 3.1.
5. Nếu error khác dự báo khác khác với nhãn sự thật cơ bản, tính toán error (sai số): hiệu số giữa dự báo và nhãn sự thật cơ bản y_i
6. Điều chỉnh trọng số và độ lệch b, lặp lại từ đầu cho đến khi error bằng không (hoặc nhỏ hơn mức yêu cầu)
7. Đầu ra: Đầu ra của perceptron là một giá trị nhị phân duy nhất, là +1 hoặc 1, biểu thị lớp hoặc thể loại mà dữ liệu đầu vào thuộc về.
8. Thuật toán đào tạo: Perceptron thường được đào tạo bằng thuật toán học có giám sát như thuật toán học perceptron trong bảng 3.2 hoặc truền lan ngược (backpropagation). Trong quá trình đào tạo, trọng số và độ lệch của perceptron được điều chỉnh để giảm thiểu lỗi giữa đầu ra dự đoán và đầu ra thực cho một tập hợp các ví dụ đào tạo nhất định.

Hình 3.7 cho thấy mô hình perceptron trong đó độ lệch b được tích hợp vào các trọng số $\boldsymbol{w}$ và bổ sung 1 như là phần tử đầu tiên của vectơ đầu vào $\boldsymbol{x} = 1, x_1, \ldots, x_K^T$.



Hình 3.7. Mô hình perceptron trong đó độ lệch *b* **được tích hợp vào các trọng số của** $\boldsymbol{w}$

Trong trường hợp này ta có:

$$z = \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{w}$$

trong đó $\boldsymbol{w} = w_0, w_1, \ldots, w_K^T$, $\boldsymbol{x} = 1, x_1, \ldots, x_K^T, b = w_0$.

Lưu ý: ta có thể tích hợp *b* và vectơ $\boldsymbol{w}$ bằng cách đặt nó vào cuối vectơ $\boldsymbol{w}$ và 1 vào cuối vectơ $\boldsymbol{x}$ như đã xét ở chương 2.

3.2.2. Mạngc perceptron đa lớp

Có thể khắc phục hạn chế của các perceptron đối với các vấn đề tuyến tính bằng cách sử dụng mạng perceptron đa lớp, thường được ký hiệu là MLP (Multilayer Perceptron). Một MLP bao gồm ít nhất ba lớp perceptron: lớp đầu vào, một lớp ẩn và một lớp đầu ra. Ngoại trừ các neuron đầu vào, mỗi neuron sử dụng một hàm kích hoạt phi tuyến tính để nó có thể phân biệt dữ liệu mà dữ liệu này không thể phân tách tuyến tính. Các lớp này cũng có thể được gọi là các lớp kết nối toàn bộ vì chúng kết nối tất cảcác neuron của lớp trước và của lớp sau. Đây là điều tuyệt đối quan trọng mà ta cần lưu tâm, vì các hàm phi tuyến tính cần thiết để tìm được các phân tách phi tuyến tính trong dữ liệu (nếu không tất cả các lớp đơn giản là sẽ cùng nhau sụp đổ vào một hàm đơn khổng lỗ).

3.2.1.1. Mạng đa lớp đơn giản

Ta xét một mô hình tổng quát MLP với một lớp ẩn như trên hình 3.8.

Trong trường hợp này ta có các ma trận trọng số $\mathbf{W}^1$ và $\mathbf{W}^2$ như sau:

$$\boldsymbol{W}^1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \cdots & w^1_{1,i} & \cdots & w^1_{1,N} \\ w^1_{2,1} & w^1_{2,2} & \cdots & w^1_{2,i} & \cdots & w^1_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w^1_{j,1} & w^1_{i,2} & \cdots & w^1_{j,i} & \cdots & w^1_{j,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w^1_{K,1} & w^1_{K,2} & \cdots & w^1_{K,i} & \cdots & w^1_{K,N} \end{bmatrix}, \boldsymbol{W}^2 = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} \\ w^2_{2,1} & w^2_{2,2} \\ \vdots & \vdots \\ w^2_{j,1} & w^2_{j,2} \\ \vdots & \vdots \\ w^2_{N,1} & w^2_{N,2} \end{bmatrix},$$

Với z là hàm tổng và g là hàm kích hoạt, ta có:

$$z\ (\boldsymbol{x}) = \boldsymbol{z} = \boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x} = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & \cdots & w^1_{1,i} & \cdots & w^1_{1,N} \\ w^1_{2,1} & w^1_{2,2} & \cdots & w^1_{2,i} & \cdots & w^1_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w^1_{j,1} & w^1_{j,2} & \cdots & w^1_{j,i} & \cdots & w^1_{j,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w^1_{K,1} & w^1_{K,2} & \cdots & w^1_{K,i} & \cdots & w^1_{KN} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_1 \\ \vdots \\ x_j \\ \vdots \\ x_K \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_N \end{bmatrix}$$

trong đó $\boldsymbol{z} = \begin{bmatrix} z_1 & z_2 & \cdots & z_N \end{bmatrix}^{\mathrm{T}}$ la vectơ tổng, T ký hiệu cho đảo vị ma trận, và các hàm kích hoạt g đưa lên lớp hai:

$$g\ \boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x} = g\boldsymbol{z} = \begin{bmatrix} gz_1 & gz_2 & \ldots & gz_i & \ldots & gz_N \end{bmatrix}^{\mathrm{T}}$$

Ký hiệu z^2 là hàm tổng. ta được vectơ dự báo:

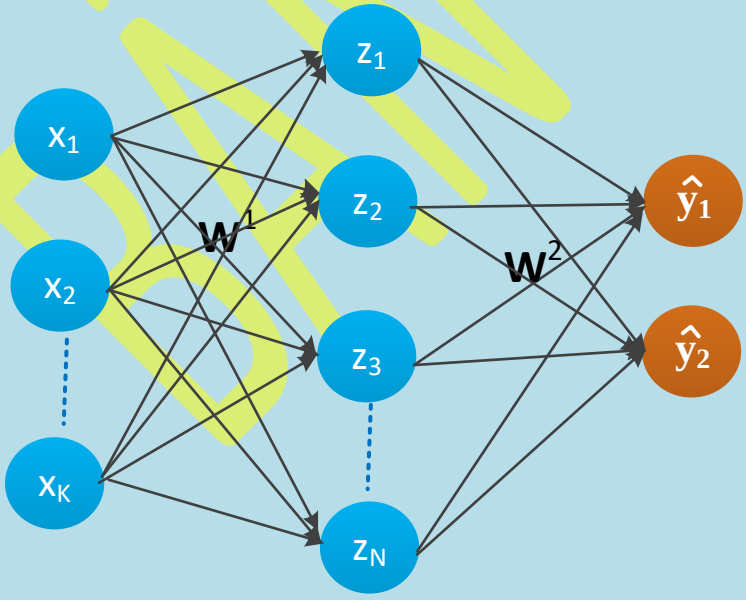
$$z^2(gz^{\mathrm{T}}) = \hat{\boldsymbol{y}} = \boldsymbol{W}^{2\mathrm{T}}(gz^{\mathrm{T}}) = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} \\ w^2_{21} & w^2_{2,2} \\ \vdots & \vdots \\ w^2_{\mathrm{i},1} & w^2_{\mathrm{j},2} \\ \vdots & \vdots \\ w^2_{\mathrm{N},1} & w^2_{\mathrm{N},2} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} gz_1 \\ gz^1_2 \\ \\ gz_{\mathrm{i}} \\ \\ gz_{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}$$

trong đó $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 \end{bmatrix}^{\mathrm{T}}$ la vectơ dự báo và $\hat{\mathrm{y}}_{\mathrm{j}} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w}^2_{\mathrm{ij}} \mathrm{gz}_{\mathrm{i}}$, $\mathrm{j} = 1,2$.

Không mất tính tổng quát, có thể định nghĩa một MLP với một lớp ẩn như sau:

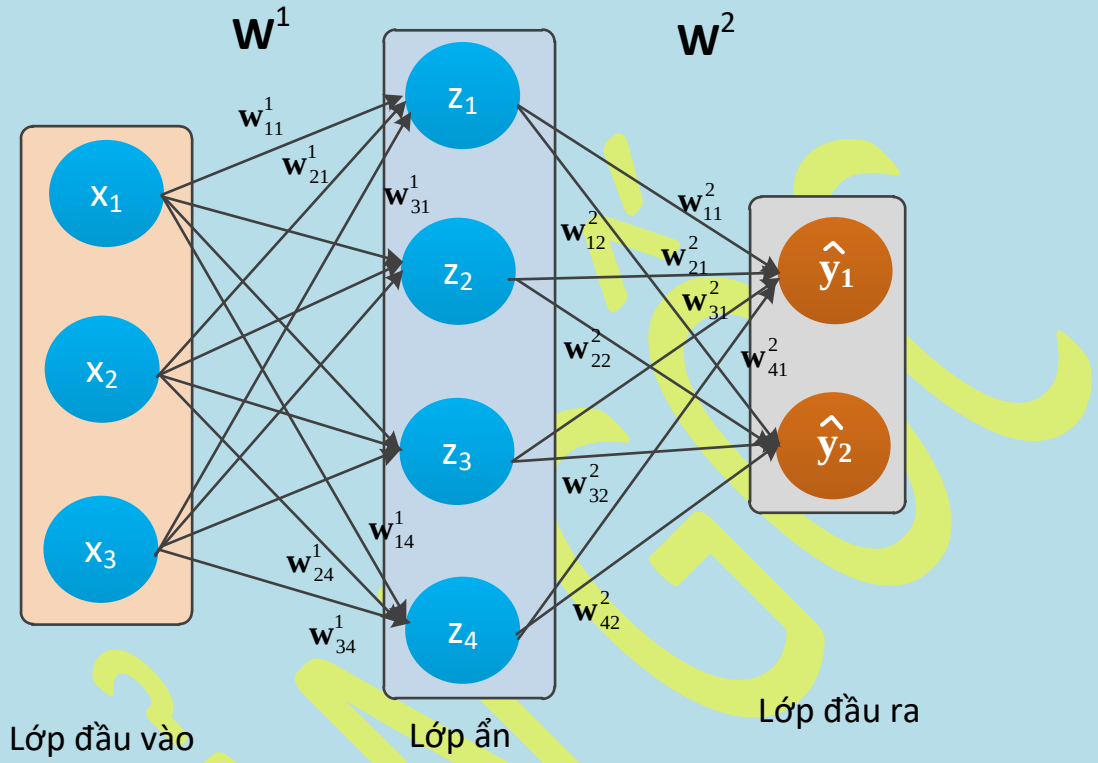
$$\begin{cases} z(\boldsymbol{x}) = g(\boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x}) \\ \hat{\boldsymbol{y}} = \mathrm{f}(\boldsymbol{x}, \boldsymbol{W}^1, \boldsymbol{W}^2) = \boldsymbol{W}^{2\mathrm{T}} z(\boldsymbol{x}) \end{cases}$$

Trong đó $g(\boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x}) : \mathbb{R} \to \mathbb{R}$ ký hiệu cho hàm phi tuyến tính (hàm áp dụng cho từng phần tử vectơ), $\boldsymbol{W}^1$ là ma trận các hệ số của lớp thứ nhất và $\boldsymbol{W}^2$ là ma trận các hệ số của lớp thứ hai, T ký hiệu cho đảo vị ma trận.



Hình 3.8. Thí dụ về một mô hình perceptron hai lớp đơn giản. Thông tin của lớp thứ nhất được cung cấp cho một lớp ẩn (z), sau đó thông tin của lớp này được kết hợp cho lớp đầu ra để cung cấp dự báo cuối cùng

Ta xét một mô hình mạng perceptron đa lớp đơn giản trong đó lớp đầu vào có ba neuron (K=3), lớp ẩn có năm neuron (N=4) và lớp đầu ra có hai neuron như trên hình 3.9.



Hình 3.9. mô hình perceptron đơn giản trong đó lớp đầu vào có ba neuron (p=3), lớp ẩn có năm neuron (d_1=5) và lớp đầu ra có hai neuron

Trong trường hợp này ta có các ma trận trọng số $\mathbf{W}^1$ và $\mathbf{W}^2$ như sau:

$$\boldsymbol{W}^1 = \begin{bmatrix} w^1_{1,1} & w^1_{1,2} & w^1_{1,3} & w^1_{1,4} \\ w^1_{2,1} & w^1_{2,2} & w^1_{2,3} & w^1_{2,4} \\ w^1_{3,1} & w^1_{3,2} & w^1_{3,3} & w^1_{3,4} \end{bmatrix}, \boldsymbol{W}^2 = \begin{bmatrix} w^2_{1,1} & w^2_{1,2} \\ w^2_{2,1} & w^2_{2,2} \\ w^2_{3,1} & w^2_{3,2} \\ w^2_{4,1} & w^2_{4,2} \end{bmatrix},$$

Với z là hàm tổng và g là hàm kích hoạt, ta có:

$$z(\boldsymbol{x}) = \boldsymbol{z} = \boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x} = \begin{bmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 & w_{14}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 & w_{24}^1 \\ w_{31}^1 & w_{32}^1 & w_{33}^1 & w_{34}^1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

trong đó $\boldsymbol{z} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^{\mathrm{T}}$ la vecto tổng , $z_{\mathrm{j}} = \sum_{\mathrm{i}=1}^{3} w_{\mathrm{i}\,\mathrm{j}}^1 x_{\mathrm{i}}$, $\mathrm{j} = 1,2,3,4$ và các hàm kích hoạt g đưa lên lớp hai:

$$g\ \boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x}\ = g\boldsymbol{z} = \begin{bmatrix} gz_1 & gz_2 & gz_3 & gz_4 \end{bmatrix}^{\mathrm{T}}$$

Ký hiệu z^2 là hàm tổng đầu ra, ta được vecto dự báo:

$$z^2(g\mathbf{z}\ ) = \hat{\boldsymbol{y}} = \boldsymbol{W}^{2\mathrm{T}}(g\boldsymbol{z}\ ) = \begin{bmatrix} w_{11}^2 & w_{12}^2 \\ w_{21}^2 & w_{22}^2 \\ w_{31}^2 & w_{32}^2 \\ w_{41}^2 & w_{42}^2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} gz_1 \\ gz_2 \\ gz_3 \\ gz_4 \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}$$

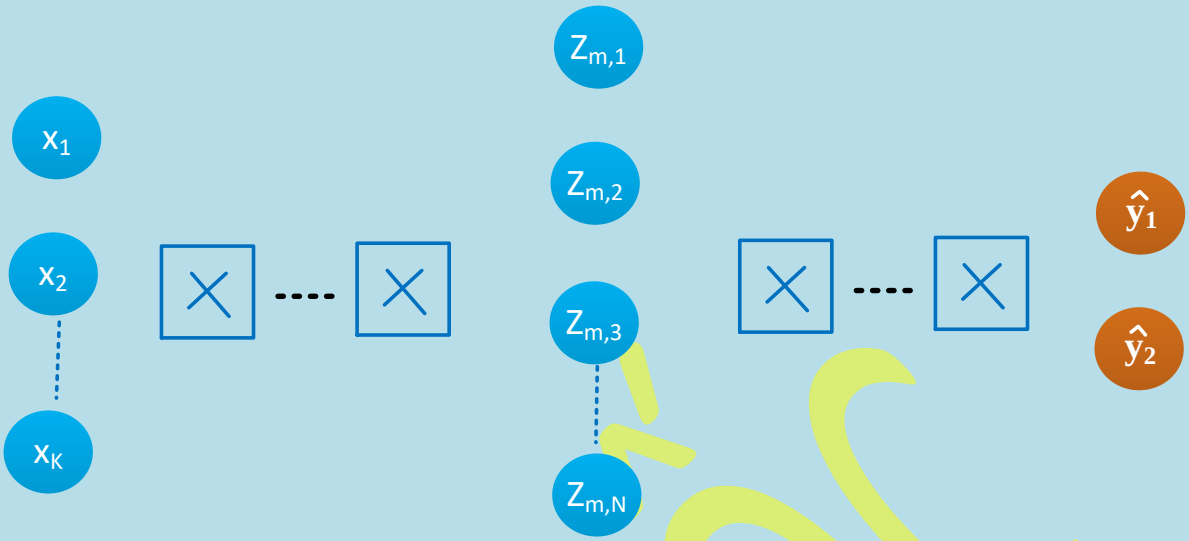
trong đó $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 \end{bmatrix}^{\mathrm{T}}$ la vecto dự báo và $\hat{y}_{\mathrm{j}} = \sum_{\mathrm{i}=1}^{4} w_{\mathrm{i}\,\mathrm{j}}^2 gz_{\mathrm{i}}$, $\mathrm{j} = 1,2$.

3.2.1.2. Mạng neuron sâu (DNN: Deep Neural Network)

Có thể tổng quát hóa mạng MLP đơn giản thành các mạng sâu hơn với nhiều hơn một lớp ân để từng bước một tổng quát hoá các tính năng mức cao hơn từ dữ liệu thô đầu vào. Có thể viết các mạng này như sau:

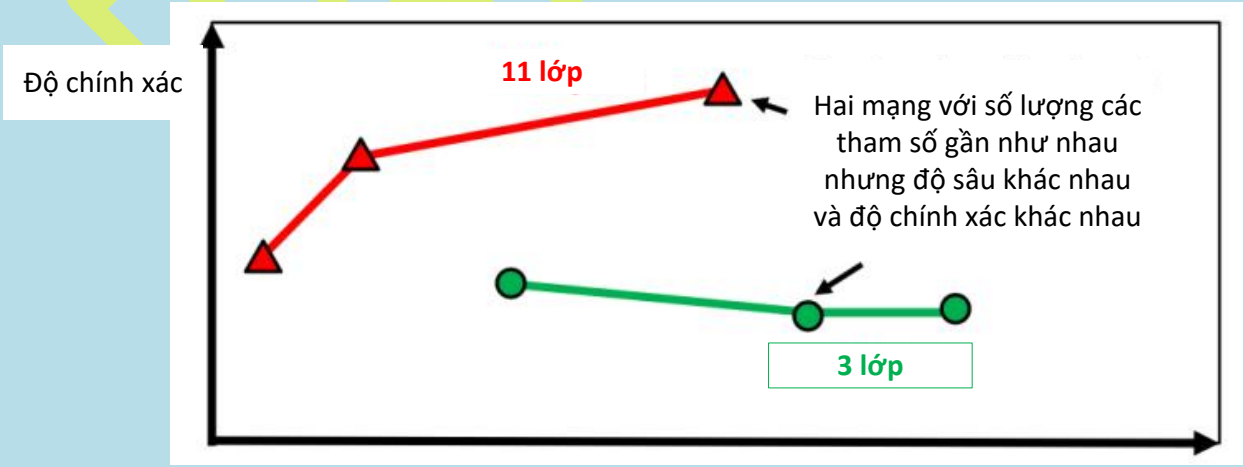
$$\begin{cases} z_1(\boldsymbol{x}) = g\ \boldsymbol{W}^{1\mathrm{T}}\boldsymbol{x} \\ \ldots \\ z_{\mathrm{m}}(\boldsymbol{x}) = g\ \boldsymbol{W}^{\mathrm{m}\mathrm{T}} z_{\mathrm{m}-1}(\boldsymbol{x}) \\ \ldots \\ \hat{\mathrm{y}} = \mathrm{f}\ \ \boldsymbol{x}, \boldsymbol{W}^1, \ldots, \boldsymbol{W}^{\mathrm{M}}\ = z_{\mathrm{M}}\ \ z_{\mathrm{M}-1}(\ldots(z_1(\boldsymbol{x}))) \end{cases} \tag{3.6}$$

Trong đó M ký hiệu cho số lớp của mạng tế bào thần kinh, nó định nghĩa độ sâu của mạng. Hình 310 thể hiệu đồ họa của perceptron sâu nhiều lớp. Một lần nữa ta thấy, lớp đầu vào được đưa đến các lớp các lớp ẩn khác nhau theo cách phân cấp sao cho đầu ra của một lớp là đầu vào của lớp sau. Lớp cuối cùng, tương ứng với lớp đầu ra thực hiện dự báo cuối cùng của mô hình.



Hình 3.10. Một thí dụ về một mạng neuron sâu. Lớp đầu vào, lớp m của mạng neuron và lớp đầu ra được thể hiện trên hình vẽ

Cho đến nay, ta đã mô tả các mạng neuron dạng các chuỗi các lớp mẫu được áp dụng theo phân cấp với xem xét chính là độ sâu của mạng (số lượng các lớp M) và độ rộng của từng lớp m (số lượng neuron K). Nói chung, không có quy tắc lựa chọn các tham số M và K để định nghĩa kiến trúc MLP. Tuy nhiên kinh nghiệm cho thấy rằng các mô hình sâu hơn thực hiện tốt hơn. Hình 3.11 trình bầy tổng quan các mạng khác nhau với 3 lớp và 11 lớp liên quan đến các tham số và độ chính xác của chúng. Đối với từng kiến trúc, số lướng các tham số thay đổi bằng cách thay đổi số lượng K tế bào thần kinh.Ta có thể nhận thấy rằng, theo kinh nghiệm, các mạng sâu hơn đạt được hiệu suất tốt hơn khi sử dụng một cách gần đúng số lượng các tham số như nhau hoặc thấp hơn.



Hình 3.11. So sánh hai mạng khác nhau với số các tham số gần như nhau nhưng độ sâu khác nhau

Các mạng tế bào thần kinh có thể có các dạng mô hình và các kiến trúc khác nhau. Lựa chọn kiến trúc và kiểu mạng neuron hợp lý phụ thuộc vào kiểu ứng dụng và kiểu dữ liệu. Đa phần, kiến trúc tốt nhất được định nghĩa theo kinh nghiệm. Phần sau ta sẽ bàn luận về các hàm chính được sử dụng trong các mạng neuron.

3.3. CÁC HÀM CHÍNH

Một mạng neuron là một kết hợp của nhiều hàm được gọi là các module. Đa phần các modul này được áp dụng lần lượt. Tuy nhiên trong các thiết kế phức tạp hơn (ví dụ mạng dư sâu (Deep Residual Network)), các cách kết hợp khác nhau có thể được thiết kế. Trong các hần sau, ta sẽ xét các chức năng được sử dụng phổ biến nhất tạo thành sương sống của hầu hết các perceptron và các kiến trúc perceptron nhiều lớp. Tuy nhiên cần lưu ý rằng các dạng chức năng khác nhau có thể được đề suất và sử dụng cho các kiến trúc học sâu khác nhau với ràng buộc là có thể phân biệt được ở hầu như mọi nơi. Lý do chính là cách thức đào tạo mạng. Vấn đề này sẽ được bàn luận trong phần dưới đây (sơ lược về các hàm này đã được tổng kết trong bảng 3.1.

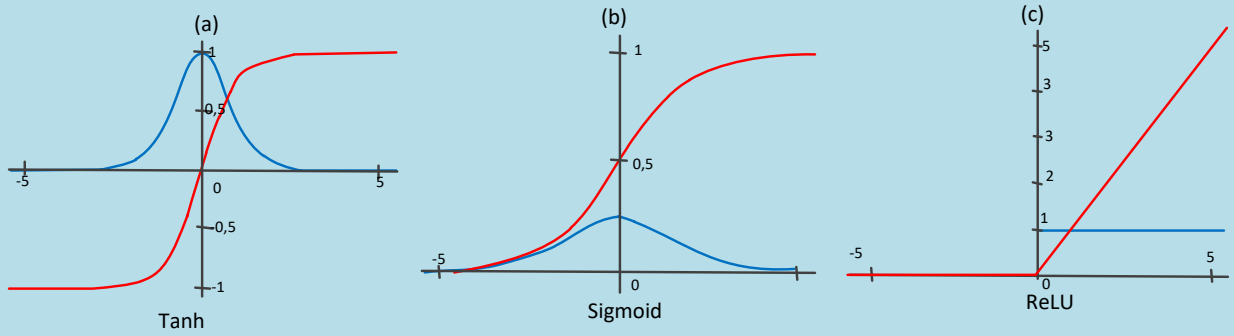
3.3.1. Các hàm tuyến tính

Một trong các hàm cơ bản nhất được sử dụng trong các mạng neuron sâu là hàm tuyến tính đơn giản. Các hàm tuyến tính tạo ra một kết hợp tuyến tính của tất cả các nút của một lớp được được đánh trọng số bằng các tham số $\boldsymbol{W}$. Tín hiệu đầu ra của hàn tuyến tính là $W^T\mathbf{x}$, là một đa thức bậc một. Trong khi nó cho phép dễ ràng giải quyết các phương trình tuyến tính, nhưng chúng có nawmh lực kém hơn để học lập các ánh xạ chức năng phức tạp từ số liệu. Chính vì thế cần kết hợp chúng với các hàm phi tuyến tính được gọi là các hàm kích hoạt (tên gọi kích hoạt ban đầu được lấy cảm hứng từ sinh vật học, vì neuron sẽ hoạt động hay không phụ thuộc vào đầu ra của hàm này).

3.3.2. Các hàm phi tuyến tính

Một trong các phần tử quan trọng nhất của các mạng neuron sâu là các hàm phi tuyến tính còn được gọi là các hàm kích hoạt. Chúng chuyển đổi các đầu vào tuyến tính thành các đầu ra phi tuyến tính để tạo điều kiện cho học các đa thức bậc cao. Trong các mục dưới đây ta sẽ bàn thảo về các hàm phi tuyến tính kinh điển nhất (hình 3.12).

BẢN GỐC



Hình 3.12. Tổng quan các hàm phi tuyến tính (đỏ) và các đạo hàm bậc một của chúng (xanh da trời). (a) hàm tangent hyperbol, (b) hàm Sigmoid, (c) hàm đơn vị tuyến tính chỉnh lưu (RLU: Rectified Linear Unit)

Hàm tangent hyperbolic (tanh: hyperbolic trangent). Một trong số các hàm phi tuyến tính tiêu chuẩn nhất, còn được biết là hàm tanh. Tanh đối xứng qua gốc tọa độ và có dải giá trị thay đổi từ -1 đến +1. Ưu điểm lớn nhất của hàm tanh là nó cho đầu ra có tâm bằng 0 (hình 3.12 (a)), vì thế hỗ trợ quá trình truyền ngược (sẽ xét trong phần sau). Tanh được sử dụng nhiều cho đào tạo các mạng neuron nhiều lớp. Về mặt công thức, hàm Tanh cùng với gradient của nó được định nghĩa như sau:

$$g = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$
$$\frac{\partial g}{\partial x} = 1 - \tanh^{2} x \qquad (3.7)$$

Một trong các bất lợi của hàm này là bão hòa của gradient xảy ra đối với các đầu vào lớn hoặc nhỏ. Điều này có thể làm chậm đào tạo các mạng.

Sigmoid. Giống như tanh, sigmoid là một trong số các hàm phi tuyến tính đầu tiên đã được sử dụng để cấu trúc các kiến trúc học sâu. Một trong số các lợi điểm chính là nó có dải các giá trị thay đổi từ 0 đến 1 (hình 3.12(b) và vì thế nó được sử dụng cho các mô hình hướng đến dự đoán xác suất như là một đầu ra. Về mặt công thức, hàm sigmoid được định nghĩa như sau:

$$g = \sigma(x) = \frac{1}{1 - e^{-x}}$$
$$\frac{\partial y}{\partial x} = \sigma(x)(1 - \sigma(x)) \qquad (3.8)$$

Lưu ý rằng , thực chất đây là hàmlogistic, nó là một trường hợp đặc biệt của lớp hàm sigmoid tổng quát. Như thấy trên hình 3.12(b), gradient của sigmoid biến mất đối với các

đầu vào lớn và nhỏ khiến cho quá trình đào tạo gặp khó. Tuy nhiên trong trường hợp nó được sử dụng cho các đơn vị đầu ra không phải là các biến tiềm ẩn (Latent Variable) và trên hàm này ta có quyền truy nhập đến các nhãn sự thật cơ bản, sigmoatd có thể là một tùy chọn tốt.

Đơn vị truyến tính chỉnh lưu (RLU: Rectified Linear Unit). ReLU được xét đến trong lựa chọn mặc định các hàm phi tuyến tính. Một số lợi điểm của ReLU bao gồm tính toán hiệu quả và truyền lan gradient tốt hơn do các vấn đề biến mất gradient ít hơn so với hai hàm kích hoạt được xét ở trên. Về mặt công thức, hàm này cùng với gradient của nó được định nghĩa như sau:

$$g = \max(0, x)$$
$$\frac{\partial y}{\partial x} = \begin{cases} 0, \text{if } x \le 1 \\ 1, \text{if } x > 0 \end{cases} \qquad (3.9)$$

Như thấy trên hình 3.12(c), ReLU cóthể tích phân mọi nơi khác không. Tuy nhiên đây không thành vấn đề vì giá trị tích phân tại không có thể tùy chọn là 0 hoặc 1. Nghiên cứu cho thấy, theo kinh nghiệm số lần lặp yêu cầu để đạt được lỗi đào tạo 25% trên tập dữ liệu CIFAR-10 đối với mạng tích chập bốn lớp với ReLU sáu lần nhanh hơn các neuron tanh. Mặt khác, các mạng neuron kiểu ReLU mang đến một hàm phân loại tuyến tính xác định theo từng khoảng (Piece-wise Linear Classifier Function) hầu như luôn luôn tạo ra các dự báo tin cậy cao từ dữ liệu đào tạo. Tuy nhiên, do sự hiệu quả và nổi tiếng của nó, rất nhiều biến thể của ReLU đã được đề xuất như ReLU bị dò rỉ (Leaky ReLU) hay ReLU tham số (Parametric ReLU). Hai biến thể này đều nhắm đến vấn đề của các neuron chết, trong đó một số nearon chết đối với tất cả các đầu vào duy trì không hoạt động bất kể đầu vào được cung cấp. Trong trường hợp này không có dòng gradient từ các neuron này và đào tạo kiến trúc mạng neuron bị ảnh hưởng. Leaky ReLU và parametric ReLU thay đổi phần g(x)=0 bằng cách bổ sung một độ dốc và mở rộng dải của ReLU.

Swish. Việc lựa chọn hàm kích hoạt không luôn luôn dễ ràng và nó có thể ảnh hưởng hiệu suất rất lớn. Một số tác giả đã thực hiện kết hợp các nghiên cứu dựa trên học tăng cường và toàn diện để khám phá các hàm kích hoạt mới. Thực nghiệm của họ đã khám phá một hàm kích hoạt mới được gọi là Swish, được định nghĩa như sau:

$$g = x\sigma(\beta x)$$
$$\frac{\partial y}{\partial x} = \beta g(x) + \sigma(\beta x)(1 - \beta g(x)) \qquad (3.10)$$

Trong đó σ là hàm sigmoid và β hoặc là một hằng số hoặc là một tham số khả đào tạo.

Thí nghiệm cho thấy rằng Swish có xu hướng hoạt động tốt hơn ReLU trên các mô hình sâu hơn trong các lĩnh vực khác nhau.

Softmax. Softmax thường được sử dụng như là một hàm kích hoạt của một mạng neuron. Trong thực tế nó chuẩn hóa đầu ra của một mạng đến một phân bố xác suất trên các loại (Class) đầu ra được chuẩn hóa. Softmax được định nghĩa như sau:

$$\text{Softmax} = \frac{e_i^x}{\sum_{j}^{C} e_j^x} \qquad (3.11)$$

Softmax nhận như đầu vào một vectơ **x** của C các số thực và chuẩn hóa nó đến một phân bố xác suất gồm C xác suất tỷ lệ với các hàm mũ của các số đầu vào. Tuy nhiên hạn chế của softmax là nó giả thiết rằng vectơ đầu vào $\boldsymbol{x}$ thuộc về ít nhất một trong số C các loại (đây không phải trường hợp trong thực tế, nghĩa là mạng có thể được áp dụng cho một đầu vào không thuộc bất kỳ trong số các loại này).

3.3.3. Các hàm mất mát (Loss Function)

3.3.3.1. Tổng quan

Ngoài các hàm kích hoạt, các hàm mất mát định định nghĩa hàm chi phí (cost function), là một trong số các phần tử chính của các mạng neuron. Hàm này trình bày lỗi đối với một dự báo cho trước. Để vậy, đối với một mẫu dự báo cho trước, nó so sánh dự báo $f(\boldsymbol{x}_i, \boldsymbol{W})$ với sự thật cơ bản y_i (để đơn giản ta ký hiệu $\boldsymbol{W}$ là tất cả các tham số $\boldsymbol{W}^1$, …. , $\boldsymbol{W}^M$ trong perceptron nhiều lớp đã xét ở trên). Hàm mất mát được ký hiệu như sau $\ell\ y, f(\boldsymbol{x}, \boldsymbol{W})$. Mất mát trung bình trên tất các các mẫu đào tạo (N) được gọi là hàm chi phí (Cost Function) và được định nghĩa như sau:

$$L(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} \ell\ y_i, f(\boldsymbol{x}_i; \boldsymbol{W}) \qquad (3.12)$$

Trong đó $\boldsymbol{x}_i, y_i$ bao gồm tập đào tạo. Mục đích của đào tạo là tìm ra $\boldsymbol{W}$ sao cho $L(\boldsymbol{W})$ cực tiểu. Lưu ý rằng trong học máy sâu người ta thường gọi hàm chi phí là hàm mất mát, mặc dù mất mát dành cho một mẫu cho trước còn chi phí dành cho lấy trung bình trên nhiều mẫu. Ngoài ra hàm mục tiêu (Objective Function) là hàm tổng thể để giảm thiểu bao gồm chi phí và các thuật ngữ đại diện có thể có khác. Lưu ý rằng, trong các phần

dưới đây, theo sự sử dụng phổ biến trong học sâu, đôi khi ta sẽ sử dụng thuật ngữ hàm mất mát thay cho hàm chi phí.

Trong các phần dưới đây hàm mất mát có thể hầu như là bất kỳ hàm nào có thể lấy đạo hàm được. Dưới đây ta sẽ trình bày hai hàm mất mát phổ biến nhất, chúng được sử dụng cho các vấn đề phân loại (Classfication) và hồi quy (Regression). Tuy nhiên cũng tồn tại các hàm mất mát đặc thù dành cho các nhiệm vụ khác như phân đoạn (Segmentation) sẽ được xét trong phần tương ứng dưới đây.

3.3.3.2. Mất mát entropy chéo (Cross Entropy Loss). Một trong số các hàm mất mát cơ bản nhất cho các vấn đề phân loại tương ứng với entroy chéo giữa hai giá trị được kỳ vọng và các giá trị được dự đoán. Nó dẫn đến hàm chi phí sau đây:

$$\mathrm{L}(\boldsymbol{W}) = -\sum_{\mathrm{i}=1}^{\mathrm{N}} \log \ \mathrm{P}(y = y_{\mathrm{i}} \big| \mathbf{x} = \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{W}) \qquad (3.13)$$

Trong đó $\mathrm{P}(y = y_{\mathrm{i}} \big| \mathbf{x} = \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{W})$ là xác suất mà một mẫu cho trước được phân loại đúng.

Entropy chéo cũng có thể được nhìn nhận như là logarrit khả năng xẩy ra (log-likelihoood) âm của tập đào tạo khi cho trước các dự báo của mạng. Nói một cách khác, cực tiểu hóa hàm mất mát tương ứng với cực đại hóa khả năng xẩy ra:

$$\mathrm{L}(\boldsymbol{W}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P} \ \ y = y_{\mathrm{i}} \big| \mathbf{x} = \boldsymbol{x}_{\mathrm{i}}; \boldsymbol{W} \qquad (3.14)$$

3.3.3.3. Mất mát sai số bình phương trung bình (Mean Squared Error Loss)

Đối với các vấn đề hồi quy, sai số bình phương trung bình quân phương là một trong số các hàm chi phí cơ sở nhất, đo trung bình của các bình phương sai số, là trung bình hiệu số bình phương giữa các giá trị được dự báo và các giá trị thực tế. Sai số bình phương trung bình được định nghĩa như sau:

$$\mathrm{L}(\boldsymbol{W}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \left\| y_{\mathrm{i}} - \mathrm{f} \ \ \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{W} \ \right\|^2 \qquad (3.15)$$

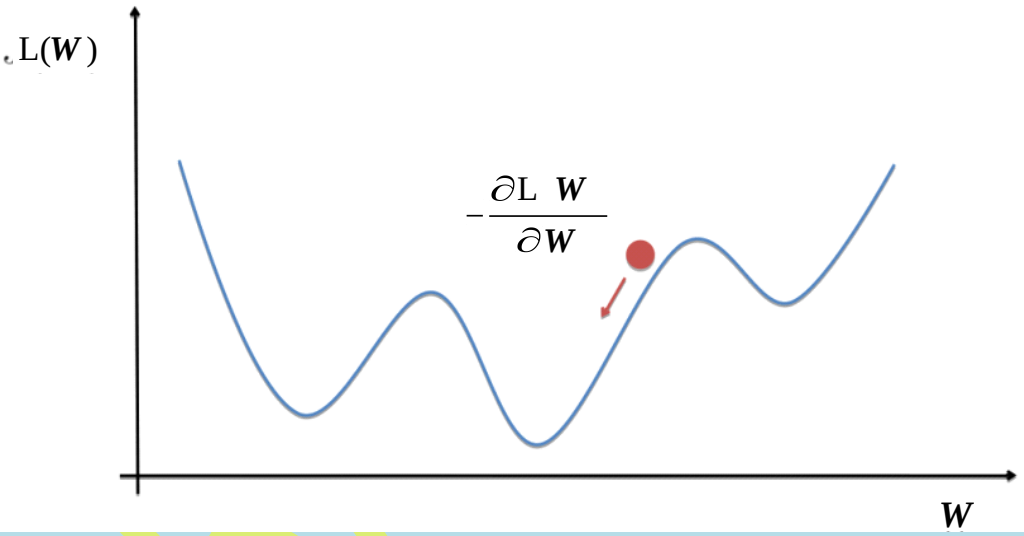
3.4. TỐI ƯU HÓA CÁC MẠNG NEURON SÂU

Tối ưu hóa là một trong số các phần tử quan trọng nhất của các mạng neuron sâu, nó tập trung lên tìm kiếm các tham số $\boldsymbol{W}$ cho phép giảm thiểu hàm mất mát $\mathrm{L}(\boldsymbol{W})$. Tổng thể, tối ưu hóa là một nhiệm vụ khó khăn. Theo truyền thống, quá trình tối ưu hóa được thực hiện

bằng cách thiết kế cẩn thận hàm mất mát và tích hợp các ràng buộc nó để đảm bảo quá trình tối ưu hóa là lồi (Covex) và như vậy ta có thể chắc chắn tìm được cực tiểu toàn cục. Tuy nhiên các mạng neuron là các mô hình không lồi dẫn đến thách thức khi tối ưu hóa chúng và nói chung ta không thể tìm được cực tiểu toàn cục mà chỉ tìm được cực tiểu cục bộ. Trong các phần sau, các phần tử chính của tối ưu hóa sẽ được trình bầy để cho ta góc nhìn tổng quát của quá trình tối ưu hóa, các thách thức của nó và các thực tế chung.

3.4.1. Sự giảm độ dốc (Gradient Descent)

Sự giảm độ dốc là một giải thuật tối ưu hóa lặp nằm trong số các giải thuật cơ sở phổ biến nhất trong học máy. Nó là một giải thật tối ưu hóa bậc 1 để tìm kiếm cực tiểu cục bộ của một hàm khả đạo hàm. Ý tưởng chính của giảm độ dốc là thực hiện các bước lặp tiến về phương ngược lại của độ dốc (gradient) của hàm mà ta cần tối ưu (hình 3.13).



Hình 3.13. Giải thuật gradient descent (giảm độ dốc). Giải thuật tối ưu hóa bậc một này để tìm một cực tiểu cục bộ bằng cách thực hiện các bước về phía ngược với độ dốc (gradient)

Theo cách này, các tham số $\boldsymbol{W}$ của mô hình được cập nhật bởi:

$$\boldsymbol{W}^{t+1} \leftarrow \boldsymbol{W}^{t} - \eta \frac{\partial L(\boldsymbol{W}^{t})}{\partial \boldsymbol{W}^{t}} \qquad (3.15)$$

Trong đó t là lần lặp và η được họi là tốc độ học, là một siêu thông số chỉ ra độ lớn của bước lặp mà giải thuật sẽ chọn.

Ngoài sự đơn giản của mình, giảm độ dốc còn là một trong số các giải thuật được sử dụng phổ biến. Các giải thuật phức tạp hơn đòi hỏi tính toán Hessian (hay một xấp xỉ hóa) (hoặc) và đảo ngược nó (hay một xấp xỉ hóa). Ngay cả khi các biến thể này có thể đảm bảo xấp xỉ hóa tốt hơn, chúng thường đòi hỏi tính toán tốn kém hơn, vì thế phương pháp giảm độ dốc là phương pháp mặc định cho tối ưu hóa.

Trong trường hợp các hàm lồi có thể giảm vấn đề tối ưu xuống vấn đến tìm một cực tiểu cục bộ. Khi này mọi cực tiểu tối ưu đều được đảm bảo là cực tiểu toàn cục và giảm gradient có thể nhận dạng nó. Tuy nhiên khi gặp các hàm không lồi, như các mạng neuron, có thể có nhiều cực tiểu cục bộ gây thách thức cho việc sử dụng giảm gradient. Nói chung các mạng neuron là các mạng không thể nhận biết. Một mô hình được coi là khả nhận biết nếu về mặt lý thuyết, khi cho trước một tập dữ liệu đào tạo đủ lớn, có thể loại bỏ tất cả trừ một tập các tham số của mô hình. Các mô hình với các biến tiềm ẩn, chẳng hạn các lớp ẩn, thường là không thể nhận biết, vì ta không thể nhận được các mô hình tương đương bằng cách trao đổi các biến tiềm ẩnvới nhau. Tuy nhiên, các cực tiểu này thường hầu như tương đương với nhau trong giá trị của hàm chi phí. Trong trường hợp này, các cực tiểu cục bộ không phải là dạng không lồi xấu. Vẫn còn là một câu hỏi rằng liệu việc tồn tại nhiều cực tiểu cục bộ với một hàm chi phí cản trở đào tạo thích hợp các mạng neuron. Tuy nhiên, hiện nay người ta tin rằng hầu hết các cực tiểu cục bộ, ít nhất như đã tìm được bởi các thủ tục tối ưu hóa, luôn tương ứng với chi phí thấp.

Đối với $\boldsymbol{W}^*$ là một cực tiểu cục bộ, Ta cần chủ yếu hai điều kiện phải được thực hiện:

- $\left\| \frac{\partial \mathrm{L}}{\partial \mathrm{W}} \boldsymbol{W}* \right\| = 0$ và
- Tất các giá trị của eigen $\left(\frac{\partial^2 \mathrm{L}}{\partial \mathrm{W}^2} \boldsymbol{W}* \right)$ phải dương.

Đối với các hàm ngẫu nhiên kích thước N, xác suất đối với các giá trị eigen phải tất cả là dương và bằng 1/N. Nói một cách khác, tỷ số giữa số lượng các điểm yên ngựa và các cực tiểu cục bộ tăng theo hàm mũ với N. Điểm yên ngựa hay điểm tới hạn là một điểm mà tại đó các đạo hàm bằng không mà không phải cực tiểu của hàm. Các điểm như vậy có thể dẫn đến lỗi cao gây thách thức cho tối ưu hóa bằng cách giảm gradient.

3.4.2. Giảm gradient ngẫu nhiên (Stochastic Gradient Descent)

Hiệu quả của giảm gradient không đủ khi nó gặp các vấn đề học máy với số lượng các mẫu đào tạo lớn. Dĩ nhiên, đây là các trường hợp đối với các mạng neuron và học sâu thường dựa trên hàng trăm hoặc hàng nhìn mẫu đào tạo. Cập nhật các tham số $\boldsymbol{W}$ sau tính toán gradient với sử dụng tất cả các mẫu đào tạo có thể dẫn đến mức độ phức tạp tính toán cùae giải thuật tối ưu hóa vô cùng lớn. Để xử lý điều này, giải thuật giảm gradient ngẫu nhiên (SGD: Stochastic Gradient Descent) cho phép đơn giản hóa rất lớn. Thay vì tính toán $\frac{\partial \mathrm{L}(\boldsymbol{W})}{\partial \boldsymbol{W}}$ chính xác, mỗi lần lặp đánh giá gradient này trên cơ sở một tập nhỏ các mẫu được chọn một các ngẫu nhiên như sau:

$$\boldsymbol{W}^{t+1} \leftarrow \boldsymbol{W}^{t} - \eta_t G\ \boldsymbol{W}^{t} \qquad (3.16)$$

Trong đó

$$G\ \boldsymbol{W}^{t} = \frac{1}{\mathrm{K}} \sum_{k=1}^{K} \frac{\partial}{\partial \boldsymbol{W}} \mathrm{L}\ y_{i_k}, \mathrm{f}\ \boldsymbol{x}_{i_k}, \boldsymbol{W}^{t} \qquad (3.17)$$

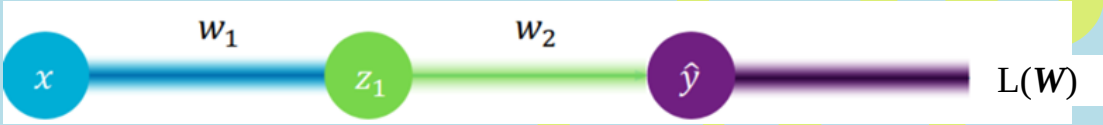
Trong đó L_{i_k} là hàm mất mát tại mẫu đào tạo i_k, $\boldsymbol{x}_{ik}, y_{ik}$, $k = 1, \ldots, K$ là một tập con gồm K mẫu đào tạo (K<<N). Tập K mẫu này được gọi là lô mini hai đôi khi là lô (Batch). Theo cách như vậy, chi phí lặp của giảm gradient sẽ là "$\mathcal{O}$(K)" và đối với giảm gradient là $\mathcal{O}$(N). Lựa chọn lý tưởng cho kích thước lô là vấn đề cần bàn luận. Trước hết, giới hạn trên đối với kích thước lô thường được xác định bởi bộ nhớ khả dụng của GPU (Graphic Processing Unit: đơn vị xử lý đồ họa), đặc biệt là khi kích thước dữ liệu đầu vào lớn (ví dụ: các ảnh 3D).

Ngoài ra chọn K là một giá trị mũ 2 thường dẫn đến tính toán hiệu quả hơn. Cuối cùng các kích thước lô nhỏ có xu thế có hiệu ứng chính quy hóa (Regularizing Effect) có lợi. Trong mọi trường hợp, kích thước lô lý tưởng thường phụ thuộc vào ứng dụng tìm kiếm các kích thước lô khác nhau không phải là việc hiếm thấy (nghĩa là mỗi mẫu đào tạo được sử dụng một lần). Cuối cùng ta gọi epoch (kỷ nguyên) lầ một lần truyền hoàn chỉnh (Complete Pass) trên toàn bộ tập đào tạo nói một cách khác epoch đề cập đến một lần truyền dữ liệu đào tạo hoàn chỉnh qua thuật toán. Số lượng kỷ nguyên (Number of Epochs) là số lần truyền hoàn chỉnh trên tập dữ liệu đào tạo. Không được nhầm lẫn số lần lặp với số lô mini được xử lý.

Lưu ý rằng các cải thiện khác nhau đối với SGD (Stochastic Gradient Descent) truyền thống đã được đưa ra dẫn đến các phương pháp tối ưu hiểu quả hơn. Các phương pháp tối ưu hóa hiện đại sẽ được trình bày trong phần sau.

3.5. TRUYỀN LAN NGƯỢC (BACKPROPAGATION)

Đào tạo các mạng neuron được thực hiện với truyền lan ngược. Truyền lan ngược tính toán gradient của hàm mất mát liên quan đến các tham số của mạng theo một cách cục bộ và hiệu quả. Giải thuật này lần đầu tiên được đưa ra vào năm 1970. Tuy nhiên nó chỉ bắt đầu nổi tiếng sau các công bố cho thấy truyền lan ngược làm việc nhanh hơn các phương pháp khác đã được đề xuất cho đào tạo các mạng neuron.
Giải thuật truyền lan ngược hoạt đồng bằng cách tính gradient của hàm mất mát (L) liên quan đến từng trọng số bởi quy tắc chuỗi (Chain Rule) và tính toán gradient một lớp một lần và lặp ngược từ lớp cuối cùng để tránh các tính toán dư các thành phần trung gian trong quy tắc chuỗi. Hình 3.14 trình bày một thí dụ về perceptron nhiều lớp có một lớp ẩn.



Hình 3.14. Một perceptron với một lớp ẩn

Trong mạng này, truyền lan ngược được tính toán như sau:

$$\frac{\partial \mathrm{L}\ \boldsymbol{W}}{\partial w_2} = \frac{\partial \mathrm{L}\ \boldsymbol{W}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_2}$$

$$\frac{\partial \mathrm{L}\ \boldsymbol{W}}{\partial w_1} = \frac{\partial \mathrm{L}\ \boldsymbol{W}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_1} = \frac{\partial \mathrm{L}\ \boldsymbol{W}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \mathrm{z}_1} \times \frac{\partial \mathrm{z}_1}{\partial w_1}$$

Nhìn chung, truyền lan ngược rất đơn giản và cục bộ. Tuy nhiên, nguyên nhân vì sao ta có thể đào tạo một máy không lồi cao có nhiều cực tiểu cục bộ, giống như các mạng neuron, bằng một giải thuật cục bộ mạnh thậm chí đến nay thực sự vẫn chưa hiểu được. Trong thực tế, truyền lan ngược có thể được tính toán theo các cách khác nhau như nhân công (Manual Calculation), phân biệt số (Numerical Differentiation) sử dụng xấp xỉ hóa hiệu số hữu hạn, phân biệt tượng trưng (Symbolic Differentiation). Hiện nay, các bộ khung học máy sử dụng các phân biệt tự động (Automatic Differenciation) cho ứng dụng truyền lan ngược.

3.5. TỔNG QUÁT HÓA VÀ QUÁ KHỚP (GENERALIZATION AND OVERFITTING)

Giống như các giải thuật học máy (đã xét trong chương 2), các mạng neuron có thể gặp vấn đề tổng quát hóa (Generalization**)** và quá khớp (Overfitting**)**. Nguyên nhân gây ra các vấn đề là là tối ưu hóa các tham số của các mô hình được thực hiện trong tập dữ liệu đào tạo $x_i, y_i \ _{i=1,\ldots,N}$ trong khi ta cần mô hình thực hiện tốt trên dữ liệu chưa thấy (Unseen Data) khác, mà các dữ liệu này không có sẵn trong quá trình đào tạo. Một cách chính thức hơn, trong trường hợp entropy chéo, mất mát mà ta muốn giảm thiểu là:

$$L(\boldsymbol{W}) = -\log \prod_{x,y \in T_T} P\ \ y = y \middle| x = \boldsymbol{x}; \boldsymbol{W} \qquad (3.18)$$

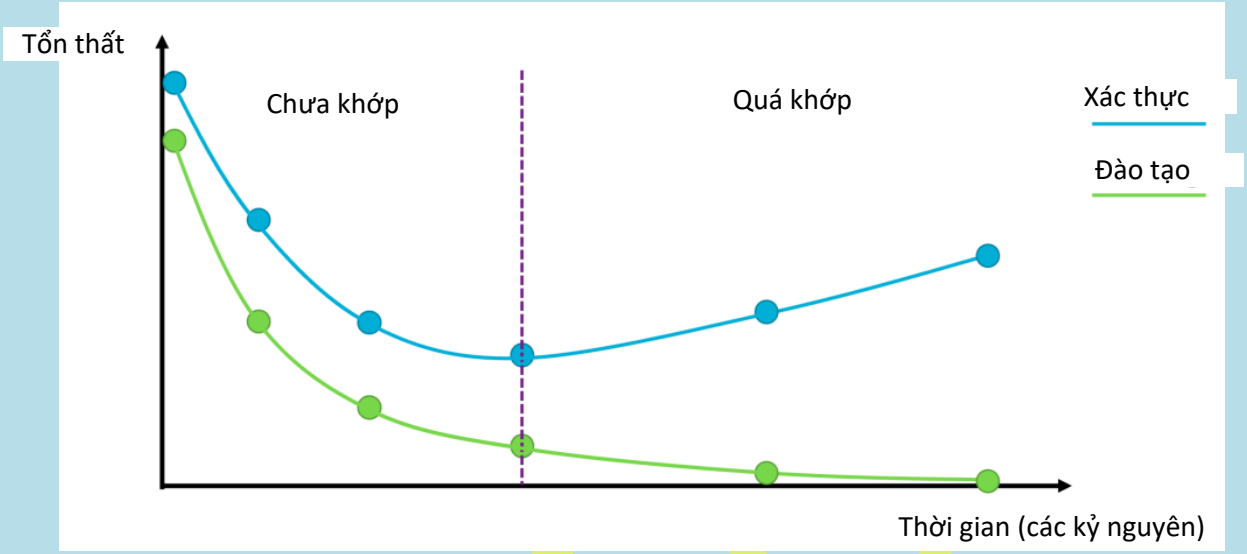
Trong đó T_T là tập dữ liệu bất kỳ không có sẵn trong quá trình đào tạo.

Trong thực tế một tập dữ liệu xác thực (Vadidation Data Set) T_T được sử dụng để đánh giá mất mát trên dữ liệu chưa thấy. Tất nhiên tập xác thực này phải khác với tập đào tạo. Điều cực kỳ quan trọng cần nhớ là hiệu suất nhận được từ tập xác thực nói chung có xu hướng bị đánh giá cao hơn vì tập xác thực được sử dụng để thực hiện dừng sớm hoặc để chọn các tham số chính quy hóa (Regularization Parameter). Vì thế cần sử dụng một tập kiểm thử độc lập được cách ly ngay từ đầu và nhất thiết không được sử dụng trong quá trình đào tạo và chỉ được sử dụng để báo cáo hiệu suất. Trong trường hợp không thể có tập kiểm thử độc lập bổ sung do thiếu dữ liệu, cần hiểu rằng hiệu suất có thể bị đánh giá sai và đây là hạn chế của nghiên cứu đặc thù này.

Để tránh quá khớp và cải thiện hiệu suất tổng quát hóa của mô hình, tập xác thực được sử dụng để giám sát mất mát trong quá trình đào tạo mạng. Theo dõi đào tạo và mất mát xác thực trên một số kỷ nguyên là quan trọng và nó cung cấp hiểu biết sâu sắc quan trọng về quá trình đào tạo và các siêu tham số được chọn (ví dụ: chọn tốc độ học). Các công cụ trực quan hóa hiện nay như TensorBoard hay Weights&Biases cho phép theo dõi dễ ràng. Trong phần sau, ta sẽ đề cập đến một số kỹ thuật tối ưu được áp dụng phổ biến để hỗ trợ ngăn chặn quá khớp.

Dừng sớm (Early Stopping). Mô hình tốt nhất liên quan đến hiệu suất và khả năng tổng quát hóa được chọn bằng cách sử dụng báo cáo về các lỗi xác thực và đào tạo. Đặc biệt, dừng sớm tương ứng với chọn một mô hình tương ứng với thời điểm sớm hơn kỷ nguyên cuối cùng là cách phổ biến để ngăn chặn quá khớp. Dừng sớm là một dạng chính quy hóa cho các mô hình được đào tạo bằng phương pháp lặp, như giảm gradient và các biến thể của nó. Dừng sớm có thể được thực hiện băng các tiêu chuẩn khác nhau. Tuy nhiên, nói chung, nó đòi hỏi giám sát hiệu suất của mô hình trên một tập xác thực và mô hình được chọn khi hiệu suất giảm và mất mát của nó tăng. Nhìn chung, dừng sớm cần được sử

dụng gần như phổ biến cho đào tạo các mạng neuron. Khái niệm dừng sớm được minh họa trên hình 3.15.



Hình 3.15. Minh họa khái niệm dừng sớm. Mô hình phải được chọn tương ứng với điểm cắt cuả đường mất mát xác thực và đào tạo với đường đứt nét, là điểm mà mất mát xác thực bắt đầu tăng. Trước điểm này mô hình chưa khớp (Underfitting) và sau điểm này là mô hình quá khớp.

Chính quy hóa trọng số (Weight Regulaization)

Giống như các phương pháp học máy khác (chương 2), Chính quy hóa học máy cũng là một phương pháp được sử dụng rộng rãi để tránh quá khớp trong các mạng nueron. Đặc biệt hơn, trong quá trình đào tạo mô hình các trọng số của mô hình mạng bắt đầu tăng kích thước để đặc tả mô hình cho dữ liệu đào tạo. Tuy nhiên, các trọng số lớn có khuynh hướng gây ra chuyển đổi đột ngột trong các lớp khác nhau của mạng và theo cách đó, gây ra các thay đổi lớn tại đầu ra cho các thay đổi nhỏ tại đầu vào. Để khắc phục điều này, trong quá trình đào tạo, các trọng số có thể được cập nhật theo một cách mà chúng được khuyến khích để có giá trị nhỏ bằng cách bổ xung một khoản phạt (Penalty), chẳng hạn một chuẩn các tham số $\lambda\|\boldsymbol{W}\|^2$ của ℓ_2 trong đó λ là tham số chấp nhận đánh đổi giữa mất mát và chính quy hóa. Do chính quy hóa như vậy khá phổ biến trong các mạng neuron, các bộ tối ưu hóa khác nhau đã tích hợp chúng vào quá trình tối ưu của chúng ở dạng tiêu biến trọng số (Weight Decay).

Khởi tạo trọng số (Weight Initialization)

Cách thức mà các trọng số của các mạng neuron được khởi tạo đóng rất quan trọng, nó quyết định giải thuật có hội tụ hay không, với một số điểm ban đầu không ổn định giải thuật có thể gập một số khó khăn dẫn đến thất bại.Thường xuyên, các trọng số được khởi tạo ngẫu nhiên từ phân bố Gauss hay đồng nhất (Unifform Distribution). Việc lựa chọn phân bố Gauss hay đồng nhất có vẻ không quan trọng lắm, nhưng, quy mô/tỷ lệ (scale) gây ảnh hưởng lớn lên cả kết cục của thủ tục tối ưu hóa lẫn khả năng mạng tổng quát hóa. Tuy nhiên các phương pháp tiếp cận được thiết kế riêng (Tailored Aproche) đã được phát triển trong nhiều thập kỷ gần đây và đã trở thành các điểm khởi đầu tiêu chuẩn. Một trong số chúng là khởi đầu Xavier, nó cân bằng giữa tất cả các lớp để có cùng một phương sai kích hoạt và cùng một phương sai gradient. Một cách chính thức hơn, các trọng số được khởi đầu như sau:

$$\boldsymbol{W}_{i,j} \sim \text{Unifform}\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right) \qquad (3.19)$$

Trong đó m là số lượng đầu vào và n là số lượng đầu ra của ma trận $\boldsymbol{W}_{i,j}$. Ngoài ra , các độ lệch **b** được khởi đầu bằng không.

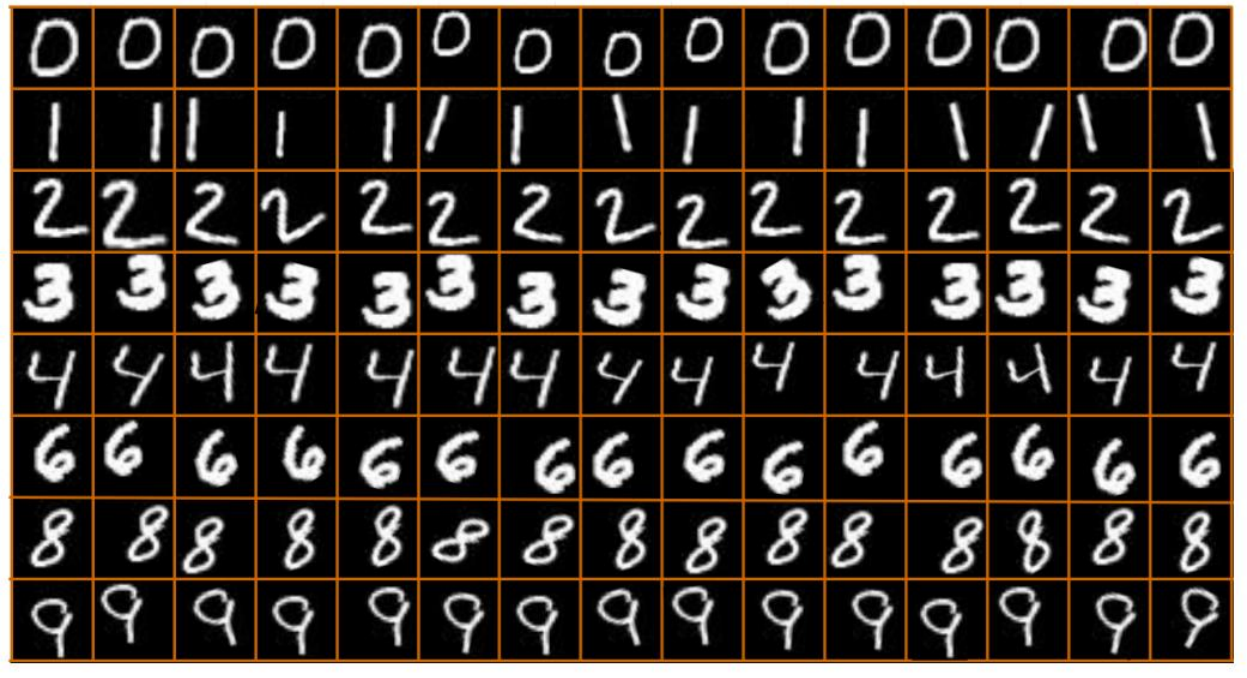
Thả rơi (Drop Out).

Tồn tại các kỹ thuật khác để ngăn chặn quá khớp, như thả rơi (Drop Out), nó bao gồm hủy ngẫu nhiên các neuron trong quá trình đào tạo, nhờ vậy giảm sự phức tạp của mô hình. Drop-out là một phương pháp học theo nhóm (học tổng hợp) không cần xây dựng các mô hình một cách rõ ràng. Trong thực tế, tại mỗi lần lặp của tối ưu hóa, các mặt nạ nhị phân ngẫu nhiên trên các đơn vị được xét. Xác suất loại bỏ một đơn vị (p) được định nghĩa như là một siêu tham số trong quá trình đào tạo mạng. Trong quá trình suy luận, tất cả các đơn vị được kích hoạt, tất cả các tham số nhận được được nhân với xác suất p này. Thả rơi là một phương pháp khá hiệu quả và phổ biến của các kiến trúc mạng neuron.

Tăng cường dữ liệu (Data Augmentation)

Vì các mạng neuron là các phương pháp vận hành bởi dữ liệu, hiệu suất của chúng phụ thuộc vào dữ liệu đào tạo. Để tăng khối lượng dữ liệu trong quá trình đào tạo, tăng cường dữ liệu có thể được thực hiện. Nó tạo ra các bản sao hơi bị biến thể của dữ liệu đào tạo hiện có để làm giầu dữ liệu đào tạo. Các kỹ thuật này hoạt động giống như một bộ chính quy hóa (Regularizer) và hỗ trợ giảm quá khớp. Một số trong các biến đổi được sử dụng phổ biến nhất được áp dụng trong quá trình tăng cường dữ liệu bao gồm quay (Rotation),

phiên dịch (Translation), cắt (Cropping), nhiễu mầu (Color Jitering), thay đổi kích thước (Resizing), làm mờ Gauss (Gaussian Blurring) ngẫu nhiên và nhiều nữa. Trên hình 3.16 ta thấy các thí dụ về biến đổi khác nhau trên các chữ số khác nhau (cột 1) của tập dữ liệu MIST.



Hình 3.16. Các thí dụ về biến đổi dữ liệu được áp dụng trong tập dữ liệu MIST. Mỗi mẫu được tạo ra được coi như là dữ liệu đào tạo bổ sung.

Chuẩn hóa lô (Batch Normalization)

Để đảm bảo đào tạo mạng ổn định và nhanh hơn, chuẩn hoá lô đã được đề xuất. Trong thực tế, chuẩn hóa lô tái định tâm (recenter) và tái lập tỷ lệ (rescale) đầu vào của lớp để loại bỏ vấn đề dịch chuyển hiệp biến (Covariate Shift), vì sự dịch chuyển này làm thay đổi phân bố của các đầu vào của một lớp gây ảnh hưởng lên tốc độ học của mạng. Mặc dù phương pháp này khá phổ biến, sự cần thiết và sử dụng nó vẫn còn là một câu hỏi.

3.6. CÁC BỘ TỐI ƯU HÓA

Trong nhiều năm, các bộ tối ưu hóa đã được đề xuất và được sử dụng nhằm cung cấp các cải tiến so với giảm gradient kinh điển. Các giải huật này được thúc đẩy bởi các thách thức cần giải quyết vấn đề giảm gradient ngẫu nhiên và trập trung lên chọn tốc độ học

thích hợp, sự thay đổi động của nó trong quá trình đào tạo cũng như sự thật rằng nó giống vây cho tất cả các cập nhật tham số. Ngoài ra việc chọn phù hợp bộ tối ưu hóa có thể tăng tốc hội tụ đến giải pháp tối ưu. Trong phần này chúng ta sẽ xét một số bộ tối ưu hóa được sử dụng phổ biến nhất hiện nay.

3.6.1. Giảm gradient ngẫu nhiên với xung lượng (Stochastic Gradient Descent with Momentum)

Một trong số các hạn chế của giảm gradient ngẫu nhiên là do phương của garadient mà ta chon bị ngẫu nhiên. Nó có thể giao động và có thể bị mắc kẹt tại một điểm đèo. Để xử lý vấn đề này, giảm gradient ngẫu nhiên với xung lượng duy trì lịch sử của các gradient trước và cập nhật các trong số có xét đến các cập nhật trước đó. Cụ thể là:

$$\begin{aligned} g^t &\leftarrow \rho g^{t-1} + (1-\rho) G^t \quad W^t \\ \Delta W^t &\leftarrow -\eta_t g^t \\ W^{t+1} &\leftarrow W^t + \Delta W^t \end{aligned} \tag{3.20}$$

trong đó g^t là phương cập nhật các trọng số trong bước thời gian t và $\rho \in 0,1$ là một siêu tham số điều khiển đóng góp của các gradient trước và gradient hiện thời. Khi ρ=0, vẫn giống như giảm gradent ngẫu nhiên kinh điển. giá trị ρ lớn nghĩa là cập nhật bị ảnh hưởng mạnh bởi các cập nhật trước.

Giải huật xung lượng tích lũy đường trung bình động suy giảm hàm mũ (Expotencially Decying moving Average) của các gradient quá khứ và tiếp tục chuyển động theo hướng của chúng. Xung lượng tăng tốc độ hội tụ, trong khi nó cũng giúp tránh bị mắc kẹt tại các vị trí mà không gian tìm kiếm bị phẳng (các điểm đồi có gradient bằng không), vì xung lượng sẽ theo đuổi tìm kiếm theo cùng hướng xẩy ra trước vùng phẳng.

3.6.2. Bộ tối ưu hóa gradient thích ứng (Adagrad: Adaptive Gradient)

Để tạo điều kiện thuận lợi và tăng tốc hơn nữa quá trình đào tạo, các bộ ngẫu nhiên hóa với tốc độ học thíco ứng theo tham số đã được đề suất. Cac bộ tối ưu hóa gradient thích ứng (AdaGrad) là một trong số đó. Nó cập nhật từng tham số riêng tỷ lệ với thành phần của chúng (và xung lượng) theo gradient. Cụ thể là:

$$
\begin{aligned}
g^t &\leftarrow G\ W^t \\
r^t &\leftarrow r^{t-1} + g^t \odot g^t \\
\Delta W^t &\leftarrow \frac{\eta}{\delta + \sqrt{r^t}} \odot g^t \\
W^{t+1} &\leftarrow W^t + \Delta W^t
\end{aligned}
\qquad (3.21)
$$

Trong đó g^t là vectơ ước tính gradient tại bước thời gian t, r^t là thành phần điều khiển cập nhật theo tham số và δ là một đại lượng nhỏ được sử dụng để tránh chia cho không. Lưu ý rằng r^t đóng góp vào tích theo từng thành phần (element-wise product) của gradient bởi chính nó và thành phần r^{t-1} trước đó tích lũy gradient của các thành phần trước.

Giải thuật thực hiện rất tốt đối với dữ liệu thưa, vì nó giảm tốc độ học nhanh hơn đối với các tham số thường xuyên và chậm hơn đối với các tham số không thường xuyên. Tuy nhiên, vì cập nhật tích lũy các gradient của các bước trước, nên các cập nhât có thể giảm rất nhanh dẫn đến cản trở quá trình học. Hạn chế này được loại bỏ bằng các mở rộng giải thuật AdaGrad mà ta sẽ xét trong phần sau.

3.6.3. RMSrop (Robot Mean Square Propogation)

Một giải thuật với các tốc độ học thích ứng theo tham số là giải thuật lan truyền bình phương trung bình robot (RMSProp: Robot Mean Square Proropogation) được đề xuất bởi Geoffrey Hinton. Mặc dù sự nổi tiếng và ứng dụng của nó, giải thuật nay không được công bố. RMSrop là một mở rộng của của giải thuật AdaGrad để giải quyết vấn đề các tốc độ học biến mất hoàn toàn do ít bị ảnh hưởng hơn bởi các lần lặp đầu tiên của giải thuật. Một cách chính thức:

$$
\begin{aligned}
g^t &\leftarrow G\ W^t \\
r^t &\leftarrow \rho r^{t-1} + (1-\rho) g^t \odot g^t \\
\Delta W^t &\leftarrow \frac{\eta}{\delta + \sqrt{r^t}} \odot g^t \\
W^{t+1} &\leftarrow W^t + \Delta W^t
\end{aligned}
\qquad (3.22)
$$

Trong đó ρ là một siêu tham số điều khiển sự đóng góp của các gradient trước và gradient hiện thời trong cập nhật hiện thời. Lưu ý rằng RMSrop ước tính các gradient bình phương theo cùng cách thức của AdamGrad, nhưng thay vì để ước tính này tích lũy liên tục trên dữ liệu đào tạo ta duy trì trung bình của nó chuyển động đồng thời tích hợp với xung

lượng (momentum). Theo kinh nghiệm, RMSrop thể hiện là một giải thuật tối ưu hóa thực tế và hiệu quả đối với các mạng học sâu.

3.6.4. Adam (Adaptive Moment Estimation: ước tính momen thích ứng)

Tính hiệu quả và các ưu điểm của các giải thuật AdaGrad và RMS rop được tích hợp trong bộ tối ưu hóa Adam (Adaptive Moment Estimation: ước tính momen thích ứng**).** Phương pháp này tính toán các tốc độ thích ứng riêng lẻ cho các tham số khác nhau của các momen thứ nhất và thứ hai của các gradient. Một cách chính thức hơn:

$$
\begin{aligned}
g^t &\leftarrow G\ W^t \\
s^t &\leftarrow \rho_1 s^{t-1} + (1-\rho_1) g^t \\
r^t &\leftarrow \rho_2 r^{t-1} + (1-\rho_2) g^t \odot g^t \\
\hat{s}^t &\leftarrow \frac{s^t}{1-(\rho_1)^t} \\
\hat{r}^t &\leftarrow \frac{r^t}{1-(\rho_2)^t} \\
\Delta W^t &\leftarrow \frac{\lambda}{\delta + \sqrt{\hat{r}^t}} \odot s^t \\
W^{t+1} &\leftarrow W^t + \Delta W^t
\end{aligned}
\qquad (3.23)
$$

Trong đó s^t là gradient với momentum, r^t tích lũy các gradient bình phương với momentum như trong RMSrop và $\hat{s}^t, \hat{r}^t$ nhỏ hơn s^t và r^t, nhưng chúng hội tụ vào s^t và r^t. Ngoài ra, δ là một đại lượng nhỏ nào đó được sử dụng để tránh chia cho không, trong khi ρ_1 và ρ_2 là các siêu tham số của giải thuật. Các tham số ρ_1 và ρ_2 điều khiển các tốc độ giảm của từng trung bình chuyển động và hía trị của chúng gần với 1. Các kết quả thực nghiệm chứng tỏ rằng Adam làm việc tốt trong thực tế và tôt hơn các phương pháp tối ưu hóa ngẫu nhiên khác dẫn đến nó là bộ tối ưu hóa được ưa chuộng cho các vấn đề học sâu.

3.6.5. Các bộ tối ưu hóa khác

Phát triển các bộ tối ưu hiệu quả (liên quan đến tốc độ và độ ổn định) vẫn là một hương nghiên cứu tích cực. RAdam là một biến thể của Adam dựa trên đưa ra một thành phần để chỉnh lưu phương sai của tốc độ học thích ứng. Cụ thể là, RAdam tận dụng một bộ chỉnh lưu động (Dynamics Retifier) để điều chỉnh momentum của Adam dựa trên phương sai và cung cấp hiệu quả khởi động tự động được tùy chỉnh phù hợp theo tập dữ liệu hiện thời để đảm bảo một khởi đầu vững chắc. Ngoài ra LookAhead được cảm hứng

từ các tiến bộ hiện nay trong hiểu biết về các bề mặt tổn hao của các mạng neuron sâu, cung cấp một đột phá trong khám phá mạnh mẽ và ổn định trong toàn bộ quá trình đào tạo. Một cách trực giác, giải thuật này lựa chọn một hướng tìm kiếm bằng cách xem xét chuỗi các trọng số nhanh được tạo bởi một bộ tối ưu hóa khác. Đây chỉ là một số bộ tối ưu hóa có trong các tài liệu, phụ thuộc vào vấn đề và ứng dụng, các bộ tối ưu hóa khác nhau có thể được chọn và áp dụng.

3.7. TỔNG KẾT

Các mạng tế bào thần kinh sâu (DNN: Deep Neural Netwwork) hiên nay là nền tảng của nhiều ứng dụng trí tuệ nhân tạo (AI). Học sâu là một lĩnh vực phát triển rất nhanh cùng với rấ nhiều câu hỏi lý thuyết chưa thể trả lời. Tuy nhiên, các mô hình dựa trên học sâu đã trở thành các phương pháp hiện đại nhất cho nhiều lĩnh vực và nhiệm vụ. Chương này đã trình bày các nguyên lý cở bản của perceptron và các mạng perceptron. Mạng tế bào thần kinh tích chập (CNN: Convolution Neural Network) là một thể loại rất quan trọng của DNN sẽ được trình bày trong chương sau.

Chương 4
PHÂN LOẠI HÌNH ẢNH

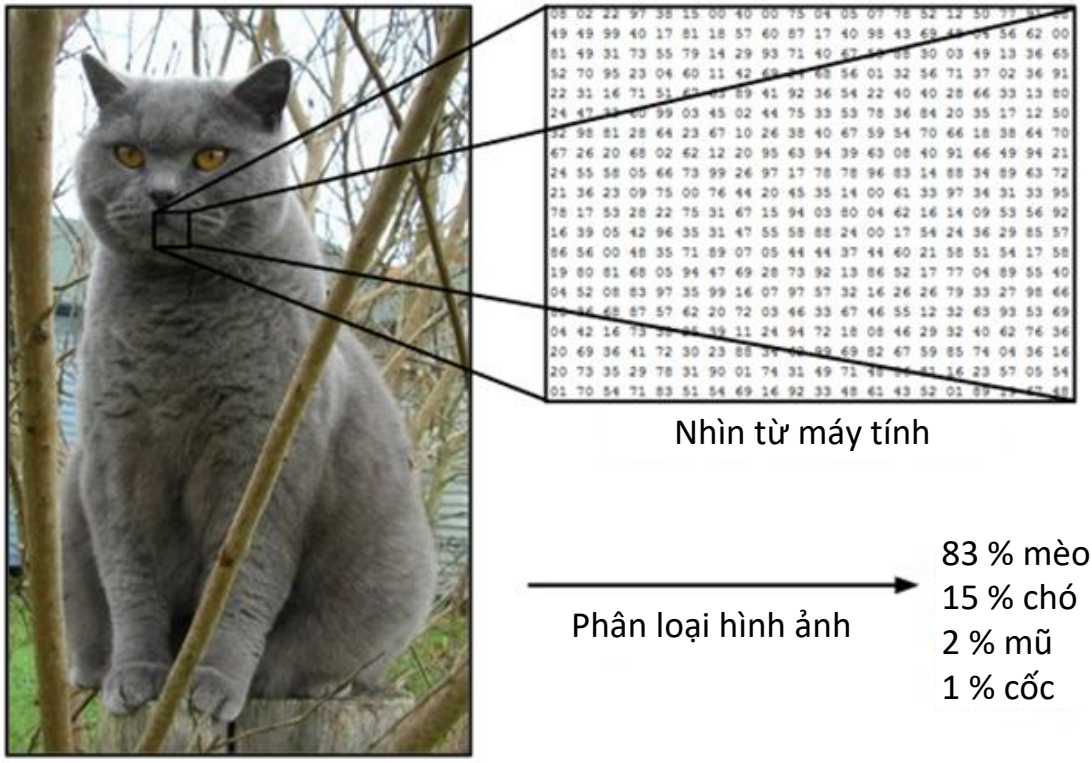
Chương này sẽ sử dụng các kiến thức nhận được từ chương 2 và 3 trước đây cho phân loại hình ảnh.

4.1. TỔNG QUAN

Phân loại ảnh, là nhiệm vụ gán nhãn cho một ảnh đầu vào từ một tập hợp các danh mục cố định. Đây là một trong những bài toán cốt lõi của thị giác máy tính (Computer Vision), mặc dù đơn giản nhưng có rất nhiều ứng dụng thực tế. Hơn nữa, nhiều nhiệm vụ thị giác máy tính khác nhau có vẻ riêng biệt (chẳng hạn như phát hiện đối tượng, phân đoạn) có thể được rút gọn thành phân loại ảnh.

Ví dụ, trong hình ảnh bên dưới, một mô hình phân loại hình ảnh lấy một hình ảnh duy nhất và gán xác suất cho 4 nhãn *{mèo, chó, mũ, cốc}* . Như thể hiện trong hình, đối với máy tính, một hình ảnh được biểu diễn dưới dạng một mảng số 3 chiều lớn. Trong ví dụ này, hình ảnh con mèo rộng 248 pixel, cao 400 pixel và có ba kênh màu Đỏ, Xanh lục,

Xanh lam (hay gọi tắt là RGB). Do đó, hình ảnh bao gồm 248 x 400 x 3 số, hay tổng cộng là 297.600 số. Mỗi số là một số nguyên nằm trong khoảng từ 0 (đen) đến 255 (trắng). Nhiệm vụ của chúng ta là chuyển đổi phần tư triệu số này thành một nhãn duy nhất (hình 4.1), chẳng hạn như *"mèo"*.



Hình 4.1. Nhiệm vụ trong Phân loại hình ảnh là dự đoán một nhãn duy nhất (hoặc phân phối của các nhãn như được hiển thị ở đây để thể hiện độ tin cậy) cho một hình ảnh nhất định. Hình ảnh là các mảng 3 chiều gồm các số nguyên từ 0 đến 255, có kích thước Chiều rộng $\times$ Chiều cao $\times$ 3. Số 3 đại diện cho ba kênh màu đỏ, xanh lục, xanh lam.

4.1.1. Các thách thức khi phân loại hình ảnh

Vì nhiệm vụ nhận dạng một khái niệm trực quan (ví dụ: mèo) tương đối dễ dàng đối với con người, nên việc xem xét những thách thức liên quan từ góc nhìn của thuật toán thị giác máy tính là cần thiết. Khi xem xét danh sách (không đầy đủ) các thách thức dưới đây, ta cần lưu ý đến việc biểu diễn hình ảnh thô dưới dạng một mảng 3 chiều của các giá trị độ sáng (hình 4.2):

- **Sự thay đổi góc nhìn** . Một vật thể có thể được định hướng theo nhiều cách so với máy ảnh.
- **Sự thay đổi về tỷ lệ** . Các loạitrực quan thường thể hiện sự thay đổi về kích thước (kích thước trong thế giới thực, không chỉ về phạm vi của chúng trong hình ảnh).

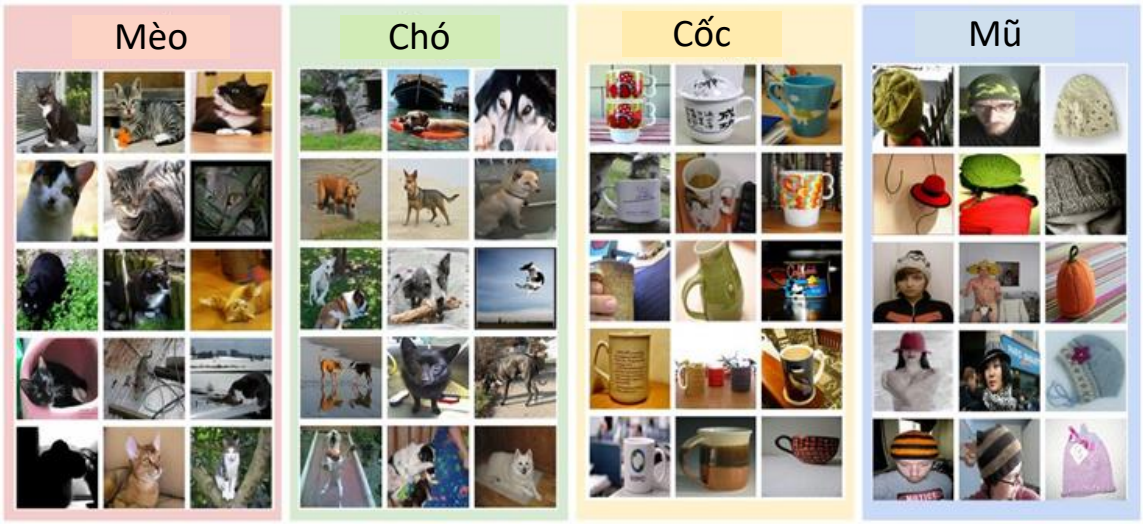
- **Biến dạng**. Nhiều vật thể quan tâm không phải là vật thể rắn và có thể bị biến dạng theo những cách cực đoan.
- **Che khuất** . Các đối tượng quan tâm có thể bị che khuất. Đôi khi chỉ một phần nhỏ của đối tượng (chỉ vài pixel) có thể được nhìn thấy.
- **Điều kiện chiếu sáng** . Hiệu ứng chiếu sáng có tác động rất lớn đến từng pixel.
- **Bối cảnh lộn xộn** . Các đối tượng quan tâm có thể *hòa* vào môi trường xung quanh, khiến chúng khó được xác định.
- **Biến thể nội loại**. Các loạiquan tâm thường có thể tương đối rộng, chẳng hạn như *ghế* . Có nhiều loại đồ vật này, mỗi loại có hình dáng riêng.

Một mô hình phân loại hình ảnh tốt phải bất biến với tích có hướng của tất cả các biến thể này, đồng thời vẫn giữ được độ nhạy với các biến thể giữa các loại (Class).

Hình 4.2. Cần lưu ý đến việc biểu diễn hình ảnh thô dưới dạng một mảng 3 chiều của các giá trị độ sáng

4.1.2. Học dựa trên tập dữ liệu đào tạo

Làm thế nào có thể viết một thuật toán có thể phân loại hình ảnh thành các thể loại (Category) riêng biệt?. Không giống như viết một thuật toán để sắp xếp một danh sách các số, chẳng hạn, không rõ ràng cách người ta có thể viết một thuật toán để nhận dạng mèo trong hình ảnh. Do đó, thay vì cố gắng chỉ định từng thể loại quan tâm trông như thế nào trực tiếp trong mã (code), cách tiếp cận mà chúng ta sẽ thực hiện không khác gì cách ta cần thực hiện với một đứa trẻ: chúng ta sẽ cung cấp cho máy tính nhiều ví dụ về từng loại và sau đó phát triển các thuật toán học tập để xem xét các ví dụ này và tìm hiểu về hình thức trực quan của từng loại. Cách tiếp cận này được gọi là *cách tiếp cận dựa trên dữ liệu* , vì nó dựa vào việc đầu tiên tích lũy một *tập dữ liệu đào tạo* gồm các hình ảnh được gắn nhãn.Hình 4.3 dưới đây là một ví dụ về cách một tập dữ liệu như vậy có thể trông như thế nào:



Hình 4.3. Một ví dụ về tập huấn luyện cho bốn danh mục hình ảnh. Trên thực tế, chúng ta có thể có hàng nghìn danh mục và hàng trăm nghìn hình ảnh cho mỗi danh mục

4.1.3. Quy trình phân loại hình ảnh

Chúng ta đã thấy nhiệm vụ trong phân loại hình ảnh là lấy một mảng pixel đại diện cho một hình ảnh duy nhất và gán nhãn cho nó. Quy trình hoàn chỉnh của chúng ta có thể được hình thức hóa như sau:

- **Đầu vào (Input):** Đầu vào của ta bao gồm một tập hợp N hình ảnh, mỗi hình ảnh được gắn nhãn với một trong M loại (Class) khác nhau. Ta gọi dữ liệu này là *tập huấn luyện* .
- **Học tập (Learning) :** Nhiệm vụ của chúng ta là sử dụng tập huấn luyện để tìm hiểu xem từng loại trông như thế nào. Ta gọi bước này là *huấn luyện bộ phân loại* (Training a Classifier) hoặc *học một mô hình* (Learning a Model).
- **Đánh giá (Evaluation) :** Cuối cùng, ta đánh giá chất lượng của bộ phân loại bằng cách yêu cầu nó dự đoán nhãn cho một tập ảnh mới mà nó chưa từng thấy trước đây. Sau đó, ta sẽ so sánh nhãn thực (True Label) của những ảnh này với nhãn do bộ phân loại dự đoán. Theo trực giác, ta hy vọng rằng nhiều dự đoán sẽ khớp với câu trả lời đúng (mà ta gọi là sự thật cơ bản (Ground Truth) hay dữ liệu thực).

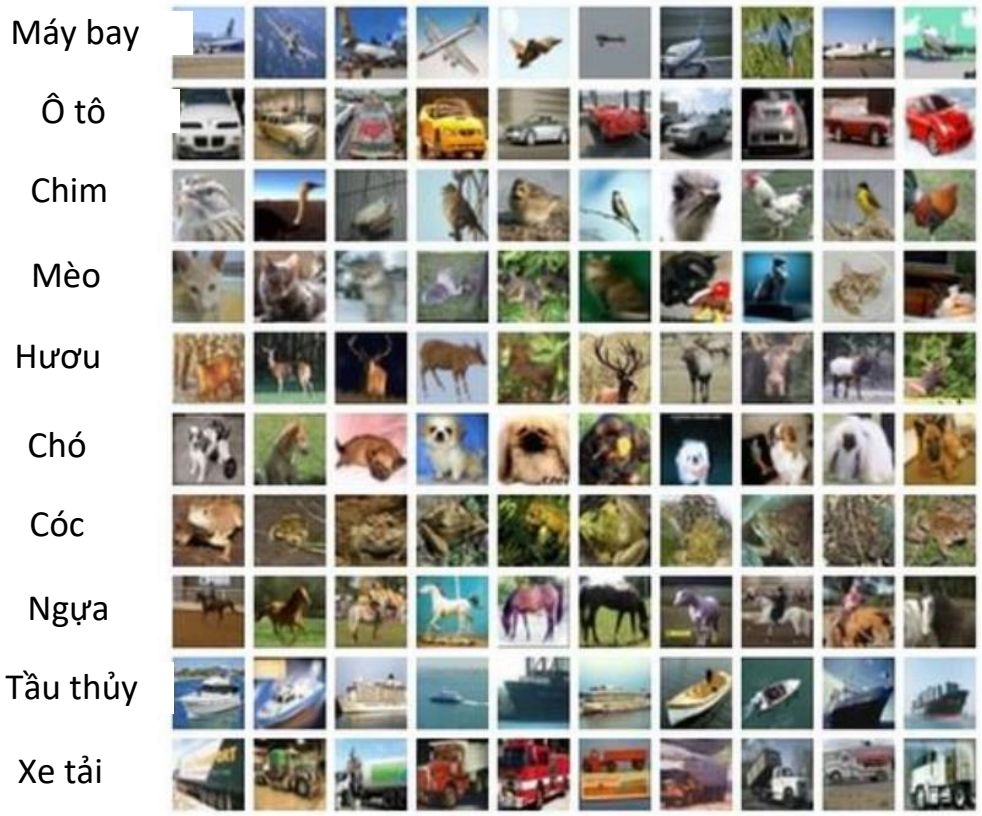
4.2. BỘ PHÂN LOẠI LÁNG GIỀNG GẦN NHẤT (NEAREST NEIGHBOR CLASSIFIER)

Trong phương pháp tiếp cận đầu tiên, chúng ta sẽ phát triển một bộ phân loại gọi là Nearest Neighbor Classifier (Bộ phân loại láng giềng gần nhất). Bộ phân loại này không liên quan gì đến mạng Neuron Tích chập và rất hiếm khi được sử dụng trong thực

tế, nhưng nó sẽ cho phép chúng ta hình dung về cách tiếp cận cơ bản đối với bài toán phân loại hình ảnh.

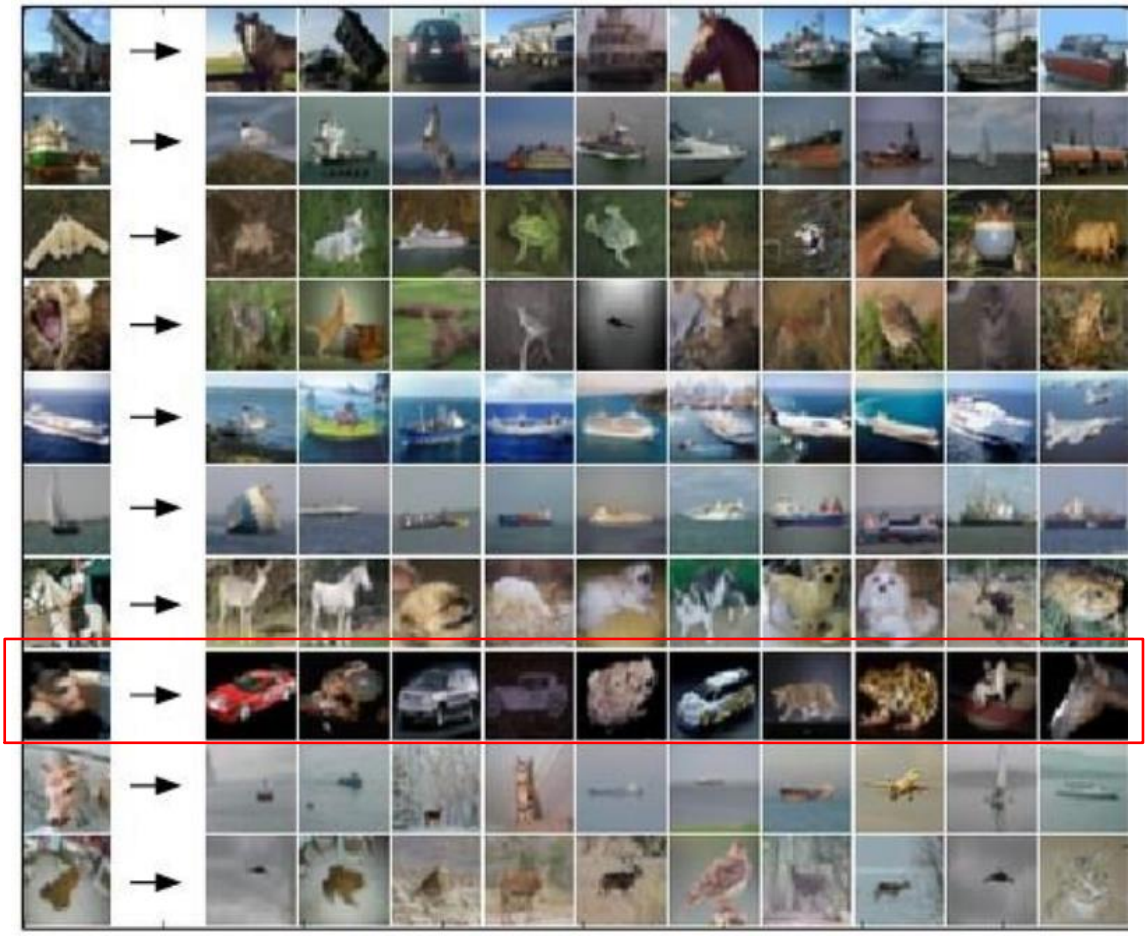
4.2.1. Ví dụ về tập dữ liệu phân loại hình ảnh: CIFAR-10

Một tập dữ liệu phân loại hình ảnh đồ chơi phổ biến là tập dữ liệu CIFAR-10 . Tập dữ liệu này bao gồm 60.000 hình ảnh nhỏ, cao và rộng 32 pixel. Mỗi hình ảnh được gắn nhãn với một trong 10 loại (ví dụ: "máy bay, ô tô, chim, v.v."). 60.000 hình ảnh này được phân chia thành một tập huấn luyện gồm 50.000 hình ảnh và một tập kiểm tra gồm 10.000 hình ảnh. Trong hình 4.4, ta có thể thấy 10 hình ảnh ví dụ ngẫu nhiên từ mỗi một trong số 10 loại.



Hình 4.4. Hình ảnh ví dụ ngẫu nhiên từ mỗi một trong số 10 loại từ tập dữ liệu CIFAR-10.

Hình 4.5. Minh họa ảnh kiểm tra cho ảnh trong tập dữ liệu đào tao: cột đầu tiên hiển thị một vài hình ảnh kiểm tra và bên cạnh mỗi hình ảnh, ta hiển thị 10 ảnh lân cận gần nhất trong tập dữ liệu huấn luyện theo sự khác biệt từng pixel.



Hình 4.5. Cột đầu tiên hiển thị một vài hình ảnh kiểm tra và bên cạnh mỗi hình ảnh, ta hiển thị 10 ảnh lân cận gần nhất trong tập dữ liệu huấn luyện theo sự khác biệt từng pixel.

4.2.2. Phân loại ảnh láng giềng gần nhất (Nearest Neighbor)

Giả sử bây giờ ta được cung cấp bộ dữ liệu huấn luyện CIFAR-10 gồm 50.000 hình ảnh (5.000 hình ảnh cho mỗi nhãn) và chúng ta muốn gắn nhãn cho 10.000 hình ảnh còn lại. Bộ phân loại láng gần gần nhất sẽ lấy một hình ảnh kiểm tra, so sánh nó với từng hình ảnh huấn luyện và dự đoán nhãn của hình ảnh huấn luyện gần nhất. Trong hình 4.5, ta có thể thấy kết quả ví dụ của quy trình như vậy đối với 10 hình ảnh thử kiểm ta ví dụ. Lưu ý rằng chỉ có khoảng 3 trong số 10 ví dụ thu được hình ảnh cùng loại, trong khi 7 ví dụ còn lại thì không. Ví dụ, ở hàng thứ 8, hình ảnh huấn luyện gần đầu ngựa nhất là một chiếc ô tô màu đỏ, có lẽ là do nền đen đậm. Do đó, trong trường hợp này, hình ảnh con ngựa này sẽ bị gắn nhãn nhầm là ô tô.

Cách so sánh hai hình ảnh, trong trường hợp này chỉ là hai khối 32 x 32 x 3. Một trong những cách đơn giản nhất là so sánh từng điểm ảnh của các hình ảnh và cộng tất cả các điểm khác biệt. Nói cách khác, cho hai hình ảnh đơn mầu được thể hiện trong dạng mảng pixel, $\boldsymbol{X} \in \mathbb{R}^{M \times K}$, $\boldsymbol{X}' \in \mathbb{R}^{M \times K}$ cho ảnh kiểm tra và cho ảnh đào tạo, ta kéo dãn hai mảng

này thành các vectơ tương ứng $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{x}' \in \mathbb{R}^D$, $D = M \times K$ một lựa chọn hợp lý để so sánh chúng có thể là khoảng cách d_1 :

$$d_1(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{D} \left| \boldsymbol{x}_j - \boldsymbol{x}'_j \right| \quad (4.1)$$

Ta cũng có thể biểu diễn (4.1) theo một cách khác:

$$d_1(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{M} \sum_{i=1}^{K} \left| x_{ij} - x'_{ij} \right| \quad (4.2)$$

trong đó j=1,…N và i=1,…,K là các chỉ số hàng và cột của các mảng pixel ảnh tương ứng.

Để tính d_1 ta trừ theo phần tử rồi cộng các hiệu số này với nhau.

Thí dụ

Gia sử mảng ảnh kiểm tra và mảng ảnh ảnh đào tạo như sau:

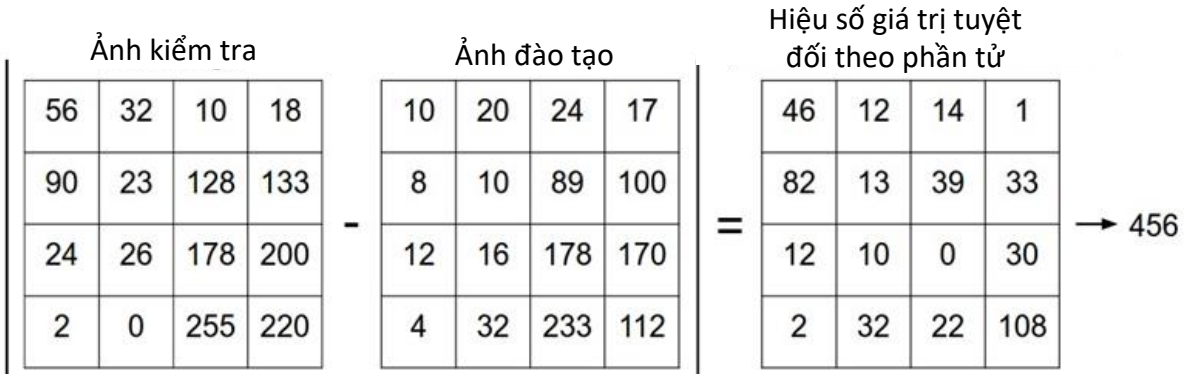
$$\mathbf{X} = \begin{bmatrix} 56 & 32 & 10 & 18 \\ 90 & 23 & 128 & 133 \\ 24 & 26 & 178 & 200 \\ 2 & 0 & 255 & 220 \end{bmatrix}, \mathbf{X}' = \begin{bmatrix} 10 & 20 & 24 & 17 \\ 8 & 10 & 89 & 100 \\ 12 & 16 & 178 & 170 \\ 4 & 32 & 233 & 112 \end{bmatrix}$$

Ta có thể kéo dài các mảng trên vào hai vectơ x và x’ tương ứng:

$$x = \begin{matrix} 56 & 90 & 24 & 2 & 32 & 23 & 26 & 0 & 10 & 128 & 178 & 255 & 18 & 133 & 200 & 220 \end{matrix}^T$$

$$x' = \begin{matrix} 10 & 8 & 12 & 4 & 20 & 10 & 16 & 32 & 24 & 89 & 178 & 233 & 17 & 100 & 170 & 112 \end{matrix}^T$$

Quy trình tính d_1 theo phưpng trình (4.2) được minh họa như trong hình 4.6.



Hình 4.6. Một ví dụ về việc sử dụng chênh lệch từng pixel để so sánh hai hình ảnh với khoảng cách d_1 (cho một kênh màu trong ví dụ này). Hai hình ảnh được trừ đi từng phần tử, sau đó tất cả chênh lệch được cộng lại thành một số duy nhất. Nếu hai hình ảnh giống hệt nhau, kết quả sẽ bằng 0. Nhưng nếu các hình ảnh rất khác nhau, kết quả sẽ rất lớn.

Chúng ta cũng hãy xem cách triển khai bộ phân loại trong mã. Trước tiên, hãy tải dữ liệu CIFAR-10 vào bộ nhớ dưới dạng 4 mảng: dữ liệu/nhãn huấn luyện và dữ liệu/nhãn kiểm tra. Trong mã dưới đây, Xtr (kích thước 50.000 x 32 x 32 x 3) chứa tất cả các hình ảnh trong tập huấn luyện, và một mảng 1 chiều tương ứng Ytr (độ dài 50.000) chứa các nhãn huấn luyện (từ 0 đến 9), sau đó chúng ta chuyển tất cả tất cả các mảng hình ảnh thành các hàng Xtr-rows, Xte-row bằng đoạn mã sau(sử dụng thư viện mã nguồn mở pyton : numpy):

```
Xtr, Ytr, Xte, Yte = load_CIFAR10('data/cifar10/') # a magic function we provide
# flatten out all images to be one-dimensional
Xtr_rows = Xtr.reshape(Xtr.shape[0], 32 * 32 * 3) # Xtr_rows becomes 50000 x 3072
Xte_rows = Xte.reshape(Xte.shape[0], 32 * 32 * 3) # Xte_rows becomes 10000 x 3072
```

Cuối cùng chúng ta có thể đào tạo và đánh giá bộ phân loại bằng đoạn mã sau::

```
nn = NearestNeighbor() # create a Nearest Neighbor classifier class
nn.train(Xtr_rows, Ytr) # train the classifier on the training images and labels
Yte_predict = nn.predict(Xte_rows) # predict labels on the test images
# and now print the classification accuracy, which is the average number
# of examples that are correctly predicted (i.e. label matches)
print 'accuracy: %f' % ( np.mean(Yte_predict == Yte) )
```

Lưu ý rằng tiêu chí đánh giá thường sử dụng độ chính xác (accuracy), đo lường tỷ lệ dự đoán đúng. Lưu ý rằng tất cả các bộ phân loại mà chúng ta sẽ xây dựng đều đáp ứng một API chung này: chúng có một hàm đào tạo train(X,y) lấy dữ liệu và nhãn để học. Về mặt nội bộ, loại (Class) nên xây dựng một mô hình nào đó về các nhãn và cách chúng có thể được dự đoán từ dữ liệu. Và sau đó là một hàm dự đoán predict(X) lấy dữ liệu mới và dự đoán các nhãn. Tất nhiên, chúng ta đã bỏ qua phần quan trọng nhất - chính bộ phân loại. Dưới đây là một triển khai của một bộ phân loại d_1 đơn giản với khoảng cách d_1 thỏa mãn mẫu này. Trong bộ phân loại này ta có: $X \in \mathbb{R}^{N \times D}; y \in \mathbb{R}^{D}$ trong đó mỗi hàng là một thí dụ và D là kích thước được kéo dãn của ảnh và mã tương ứng như sau:

```
import numpy as np “nhập thư viện numppy và đặt bí anh là np”

class NearestNeighbor(object):
  def __init__(self):  “định nghĩa phương pháp của bộ cấu trúc cho một loại”
    pass

  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
```

```
    self.ytr = y

  def predict(self, X):
    """ X is N x K where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in range(num_test):
      # find the nearest training image to the i'th test image
      # using the d1 distance (sum of absolute value differences)
      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)
      min_index = np.argmin(distances) # get the index with smallest distance
      Ypred[i] = self.ytr[min_index] # predict the label of the nearest example

    return Ypred
```

Nếu chạy đoạn mã này, ta sẽ thấy bộ phân loại này chỉ đạt 38,6% trên CIFAR-10. Con số này ấn tượng hơn so với việc đoán ngẫu nhiên (cho độ chính xác 10% vì có 10 loại), nhưng vẫn còn kém xa hiệu suất của con người (ước tính khoảng 94%) hoặc gần bằng các mạng neuron tích chập tiên tiến đạt khoảng 95%, tương đương với độ chính xác của con người (xem bảng xếp hạng của cuộc thi Kaggle về CIFAR-10).

4.2.3. Lựa chọn khoảng cách

Có nhiều cách khác để tính khoảng cách giữa các vectơ. Một lựa chọn phổ biến khác có thể là sử dụng khoảng cách d_2 , được diễn giải theo hình học là tính khoảng cách Euclid giữa hai vectơ. Khoảng cách có dạng:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{I}, d_{2=} \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|_2 = \sqrt{\sum_{j=1}^{D} x_j - x'_j}^2$$

Trong đó các vectơ $\boldsymbol{x}$, $\boldsymbol{x}$’ là các vectơ đầu vào có kích thước (D×1) nhận được từ kéo dãn các mảng ảnh đầu vào $\boldsymbol{X}$, $\boldsymbol{X}$’.

Nói cách khác, chúng ta sẽ tính toán chênh lệch từng pixel như trước, nhưng lần này chúng ta bình phương tất cả, cộng lại và cuối cùng lấy căn bậc hai. Trong numpy (Numerical Python: thư viện mã nguồn mở trong Python), sử dụng đoạn mã trên, chúng ta chỉ cần thay thế một dòng mã duy nhất. Dòng mã tính toán khoảng cách:

```
distances = np.sqrt(np.sum(np.square(self.Xtr - X[i,:]), axis = 1))
```

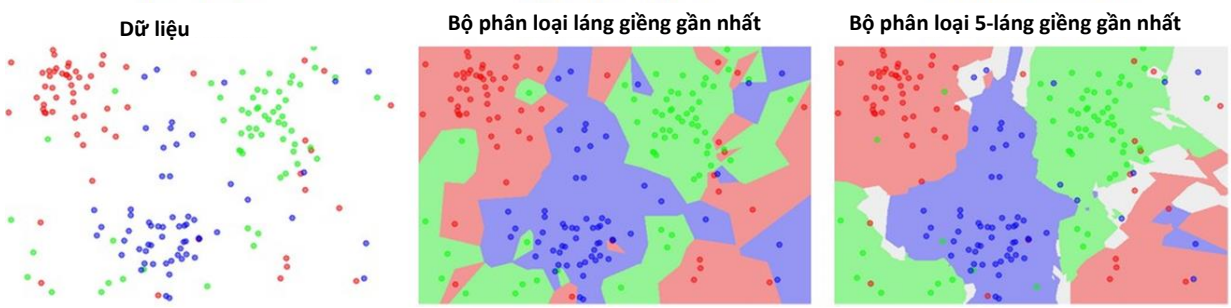
Lưu ý rằng ở tên ta đã bao gồm sử dụng np.sqrt, nhưng trong một ứng dụng láng giềng gần nhất thực tế, chúng ta có thể bỏ qua phép toán căn bậc hai vì căn bậc hai là một hàm đơn điệu. Nghĩa là, phép toán này tỷ lệ kích thước tuyệt đối của các khoảng cách nhưng vẫn giữ nguyên thứ tự, do đó các lân cận gần nhất có hoặc không có nó đều giống hệt nhau. Nếu tan chạy bộ phân loại Lân cận gần nhất trên CIFAR-10 với khoảng cách này, bạn sẽ đạt được độ chính xác 35,4% (thấp hơn một chút so với kết quả khoảng cách L1 của chúng tôi).

Thật thú vị khi xem xét sự khác biệt giữa hai phép đo d_1 và d_2. Cụ thể, khoảng cách d_2 khó chấp nhận hơn nhiều so với khoảng cách d_1 khi xét đến sự khác biệt giữa hai vectơ. Nghĩa là, khoảng cách d_2 ưu tiên nhiều sự khác biệt trung bình hơn là một sự khác biệt lớn. Khoảng cách d_1 và d_2 (hay tương đương là chuẩn d_1/d_2 (d_1/d_2 Norm) của sự khác biệt giữa một cặp ảnh) là những trường hợp đặc biệt được sử dụng phổ biến nhất của chuẩn p, trong đps p=1 đối với d_1 và p=2 đối với d_2.

4.3. BỘ PHÂN LOẠI K-LÁNG GIỀNG GẦN NHẤT (K-NEAREST NEIGHBOR CLASSIFIER)

4.3.1. Tổng quan

Ta có thể nhận thấy việc chỉ sử dụng nhãn của ảnh gần nhất khi muốn đưa ra dự đoán là khá kỳ lạ. Thực tế, hầu như luôn có thể làm tốt hơn bằng cách sử dụng bộ phân loại k - láng giềng gần nhất. Ý tưởng rất đơn giản: thay vì tìm một ảnh gần nhất trong tập huấn luyện, chúng ta sẽ tìm k ảnh gần nhất hàng đầu và để chúng bỏ phiếu cho nhãn của ảnh kiểm tra. Cụ thể, khi k = 1 , chúng ta sẽ khôi phục lại bộ phân loại lang giềng gần nất (Neares Neighbor Classifier). Theo trực giác, các giá trị k cao hơn có hiệu ứng làm mịn, giúp bộ phân loại chống lại các giá trị ngoại lai tốt hơn(hình 4.7).



Hình 4.7. Một ví dụ về sự khác biệt giữa bộ phân loại láng giềng gần nhất (NN: Nearest Neighbor) và bộ phân loại 5- láng giềng gần nhất (5-NN: 5-Nearest Neighbor Classifier), sử dụng các điểm 2 chiều và 3 loại (đỏ, xanh lam, xanh lục). Các vùng màu cho thấy ranh giới quyết định do bộ phân loại tạo ra với khoảng cách d_2. Các vùng màu trắng cho thấy các điểm được phân loại mơ hồ (tức là các phiếu

bầu loại được hòa cho ít nhất hai loại). Lưu ý rằng trong trường hợp bộ phân loại NN, các điểm dữ liệu ngoại lệ (ví dụ: điểm xanh lục ở giữa một đám mây điểm xanh lam) tạo ra các đảo nhỏ chứa các dự đoán có khả năng không chính xác, trong khi bộ phân loại 5-NN làm mịn các điểm bất thường này, có khả năng dẫn đến khái quát hóa tốt hơn trên dữ liệu thử nghiệm (không hiển thị). Cũng lưu ý rằng các vùng màu xám trong ảnh 5-NN là do các phiếu bầu hòa giữa các láng giềng gần nhất (ví dụ: 2 láng giềng có màu đỏ, hai hàng giềng tiếp theo có màu xanh lam, láng giềng cuối cùng có màu xanh lục).

Trên thực tế, ta hầu như luôn muốn sử dụng k-Nearest Neighbor. Nhưng ta nên sử dụng giá trị *k* nào ? Chúng ta sẽ xem xét bài toán này ở phần sau.

4.3.2. Các tập xác thực để điều chỉnh siêu tham số (Validation sets for Hyperparameter tuning)

Bộ phân loại k-láng giềng gần nhất yêu cầu một thiết lập cho k. Nhưng con số nào hoạt động tốt nhất? Ngoài ra, chúng ta thấy rằng có rất nhiều hàm khoảng cách khác nhau mà chúng ta có thể sử dụng: chuẩn d_1, chuẩn d_2, và còn nhiều lựa chọn khác mà chúng ta thậm chí chưa cân nhắc đến (ví dụ: tích vô hướng). Những lựa chọn này được gọi là siêu tham số và chúng xuất hiện rất thường xuyên trong thiết kế của nhiều thuật toán học máy học từ dữ liệu. Thường không rõ ràng nên chọn giá trị/thiết lập nào.
Một ý trưởng gợi ý rằng chúng ta nên thử nhiều giá trị khác nhau và xem giá trị nào hiệu quả nhất. Đó là một ý tưởng hay và thực sự là điều chúng ta sẽ làm, nhưng điều này phải được thực hiện rất cẩn thận. Cụ thể, chúng ta không thể sử dụng tập kiểm tra cho mục đích tinh chỉnh siêu tham số. Bất cứ khi nào thiết kế các thuật toán học máy, ta nên coi tập kiểm tra là một tài nguyên rất quý giá mà lý tưởng nhất là không bao giờ được động đến cho đến một lần vào thời điểm cuối. Nếu không, mối nguy hiểm thực sự là ta có thể tinh chỉnh siêu tham số của mình để hoạt động tốt trên tập kiểm tra, nhưng nếu ta triển khai mô hình của mình, ta có thể thấy hiệu suất giảm đáng kể. Trên thực tế, chúng ta sẽ nói rằng ta đã quá khớp (Overfit) với tập kiểm tra. Một cách nhìn khác là nếu ta tinh chỉnh siêu tham số của mình trên tập kiểm tra, về cơ bản ta đang sử dụng tập kiểm tra làm tập huấn luyện và do đó hiệu suất đạt được trên đó sẽ quá lạc quan so với những gì ta thực sự có thể quan sát được khi triển khai mô hình của mình. Nhưng nếu ta chỉ sử dụng tập kiểm tra một lần ở cuối, thì nó vẫn là một giải pháp thay thế tốt để đo lường mức độ tổng quát hóa của bộ phân loại (Generalization of Classifier) của mình.

4.3.2.1. Đánh giá trên bộ kiểm tra chỉ một lần duy nhất, vào thời điểm cuối cùng

May mắn thay, có một cách chính xác để tinh chỉnh các siêu tham số mà không hề ảnh hưởng đến tập kiểm tra. Ý tưởng là chia tập huấn luyện của chúng ta thành hai: một tập huấn luyện nhỏ hơn một chút, và cái mà chúng ta gọi là **tập xác thực** (Vadiation Set). Lấy ví dụ về CIFAR-10, chúng ta có thể sử dụng 49.000 ảnh huấn luyện để huấn luyện và để lại 1.000 ảnh cho việc xác thực. Tập xác thực này về cơ bản được sử dụng như một tập kiểm tra giả để tinh chỉnh các siêu tham số.

Đoạn mã dưới cho thấy điều này sẽ như thế nào trong trường hợp của CIFAR-10:

```
# assume we have Xtr_rows, Ytr, Xte_rows, Yte as before
# recall Xtr_rows is 50,000 x 3072 matrix
Xval_rows = Xtr_rows[:1000, :] # take first 1000 for validation
Yval = Ytr[:1000]
Xtr_rows = Xtr_rows[1000:, :] # keep last 49,000 for train
Ytr = Ytr[1000:]

# find hyperparameters that work best on the validation set
validation_accuracies = []
for k in [1, 3, 5, 10, 20, 50, 100]:

  # use a particular value of k and evaluation on validation data
  nn = NearestNeighbor()
  nn.train(Xtr_rows, Ytr)
  # here we assume a modified NearestNeighbor class that can take a k as input
  Yval_predict = nn.predict(Xval_rows, k = k)
  acc = np.mean(Yval_predict == Yval)
  print 'accuracy: %f' % (acc,)

  # keep track of what works on the validation set
  validation_accuracies.append((k, acc))
```

Đến cuối quy trình này, chúng ta có thể vẽ đồ thị thể hiện giá trị k nào hoạt động tốt nhất. Sau đó, chúng ta sẽ giữ nguyên giá trị này và đánh giá một lần trên tập kiểm tra thực tế.

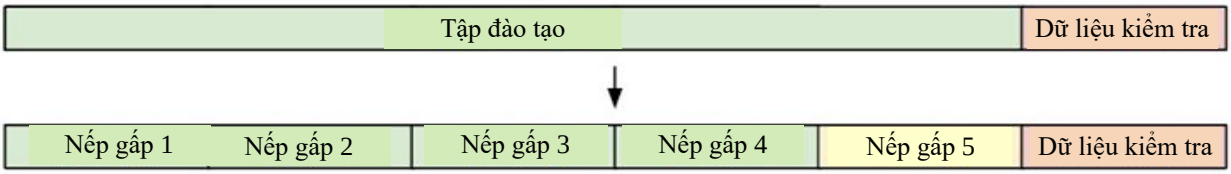
Ta có thể chia thành tập huấn luyện và tập xác thực. Sử dụng tập xác thực để tinh chỉnh tất cả các siêu tham số. Cuối cùng, chạy một lần duy nhất trên tập kiểm tra và báo cáo hiệu suất.

4.3.2.2. Xác thực chéo (Cross Validarion)

Ta có thể chia thành tập huấn luyện và tập xác thực. Sử dụng tập xác thực để tinh chỉnh tất cả các siêu tham số. Cuối cùng, chạy một lần duy nhất trên tập kiểm tra và báo cáo hiệu suất

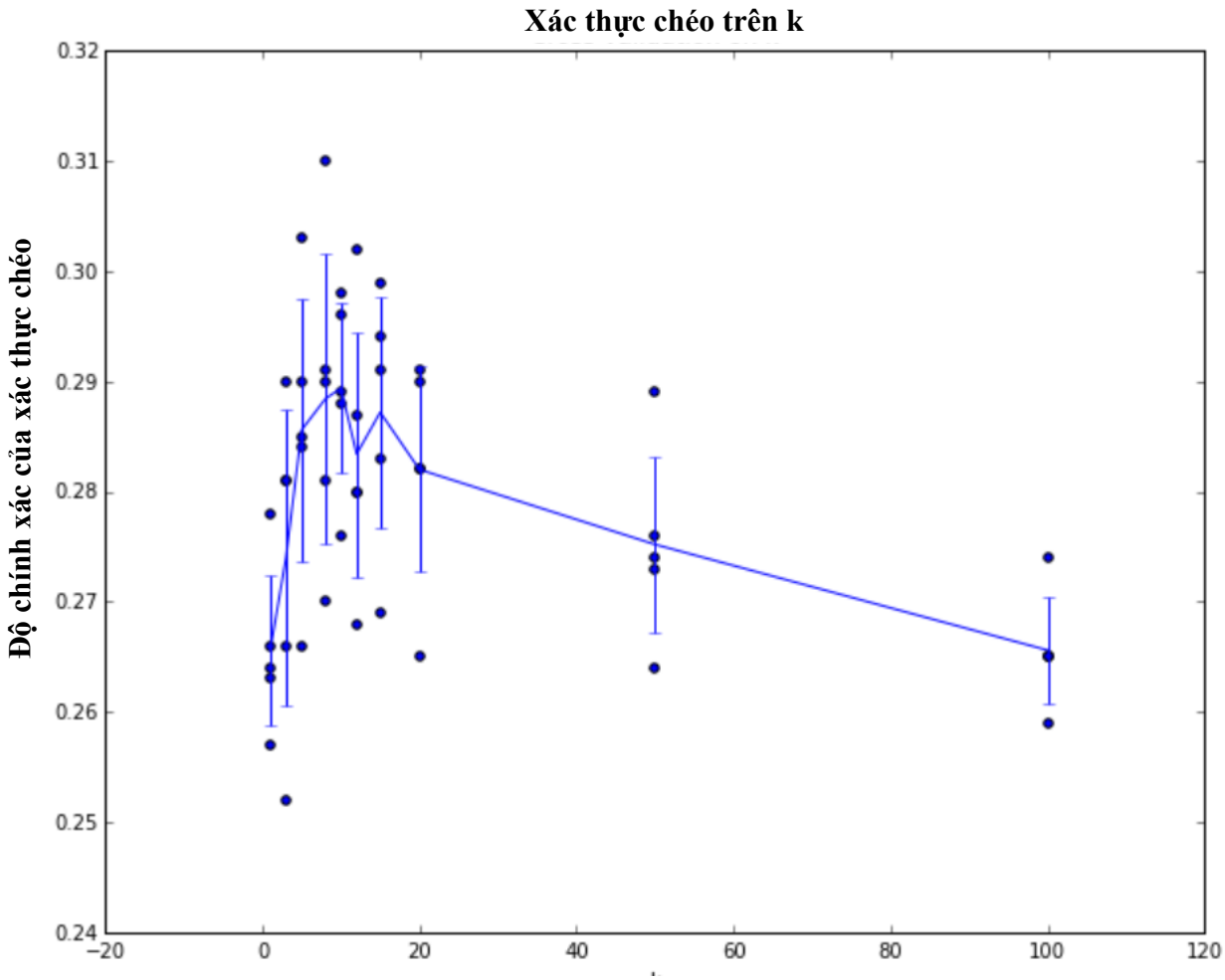
Trong trường hợp kích thước dữ liệu huấn luyện của ta (và do đó cả dữ liệu kiểm tra có thể nhỏ, đôi khi người ta sử dụng một kỹ thuật tinh vi hơn để điều chỉnh siêu tham số được gọi là xác thực chéo. Sử dụng ví dụ trước của ta với ý tưởng là thay vì chọn tùy ý 1000 điểm dữ liệu đầu tiên làm tập kiểm tra và đặt phần còn lại là tập huấn luyện, ta có thể có được ước tính tốt hơn và ít nhiễu hơn về mức độ hiệu quả của một giá trị k nhất định bằng cách lặp trên các tập xác thực khác nhau và tính trung bình hiệu suất trên các

tập này. Ví dụ: trong xác thực chéo 5 nếp gấp (5- Fold Validation), chúng ta sẽ chia dữ liệu huấn luyện thành 5 nếp gaaps bằng nhau, sử dụng 4 nếp gấp để huấn luyện và 1 nếp gấp để xác thực. Khi này, chúng ta sẽ lặp lại để xem nếp gấp nào là nếp gấp xác thực xác thực, đánh giá hiệu suất và cuối cùng tính trung bình hiệu suất trên các nếp gấp khác nhau (hình 4.8).



Hình 4.8. Phân tách dữ liệu phổ biến. Một tập huấn luyện và một tập kiểm tra được đưa ra. Tập huấn luyện được chia thành các nếp gấp (ví dụ: 5 nếp gấp ở đây). Các nếp gấp từ 1 đến 4 trở thành tập huấn luyện. Một nếp gấp (ví dụ: nếp gấp 5 ở đây màu vàng) được ký hiệu là nếp gấp xác thực và được sử dụng để tinh chỉnh các siêu tham số. Kiểm chứng chéo tiến thêm một bước nữa và lặp lại việc lựa chọn nếp gấp nào là nếp gấp xác thực, tách biệt với các nếp gấp từ 1 đến 5. Điều này được gọi là kiểm chứng chéo 5 nếp gấp. Cuối cùng, sau khi mô hình được huấn luyện và tất cả các siêu tham số tốt nhất đã được xác định, mô hình sẽ được đánh giá một lần duy nhất trên dữ liệu kiểm tra (màu đỏ).

Hình 4.9 cho thấy ví dụ về một lần chạy xác thực chéo 5 nếp gấp cho tham số k . Với mỗi giá trị k , chúng ta huấn luyện trên 4 nếp gấp và đánh giá ở nếp gấpthứ 5. Do đó, với mỗi k , chúng ta nhận được 5 độ chính xác trên nếp gấp xác thực (độ chính xác là trục y, mỗi kết quả là một điểm). Đường xu hướng được vẽ qua giá trị trung bình của các kết quả cho mỗi k và các thanh lỗi biểu thị độ lệch chuẩn. Lưu ý rằng trong trường hợp cụ thể, xác thực chéo cho thấy giá trị khoảng k = 7 hoạt động tốt nhất trên tập dữ liệu cụ thể này (tương ứng với đỉnh trên biểu đồ). Nếu chúng ta sử dụng nhiều hơn 5 nếp gấp xác thực, chúng ta có thể mong đợi thấy một đường cong mượt mà hơn (tức là ít nhiễu hơn).



Hình 4.9. Ví dụ về một lần chạy xác thực chéo 5 nếp gấp cho tham số k

4.4. PHÂN LOẠI TUYẾN TÍNH

4.4.1. Tổng quan

Trong phần trước, ta đã xét bài toán phân loại hình ảnh, là nhiệm vụ gán một nhãn duy nhất cho một hình ảnh từ một tập hợp các danh mục cố định. Hơn nữa, ta đã đã mô tả bộ phân loại k-Nearest Neighbor (k-NN) dán nhãn hình ảnh bằng cách so sánh chúng với các hình ảnh (đã được gán nhãn) từ tập huấn luyện. Như đã thấy, k-NN có một số nhược điểm:

- Bộ phân loại phải ghi nhớ tất cả dữ liệu huấn luyện và lưu trữ để so sánh với dữ liệu kiểm tra trong tương lai. Điều này không hiệu quả về mặt không gian vì các tập dữ liệu có thể dễ dàng lên đến hàng gigabyte.
- Việc phân loại hình ảnh thử nghiệm rất tốn kém vì nó đòi hỏi phải so sánh với tất cả hình ảnh đào tạo.

Bây giờ chúng ta sẽ xét một phương pháp phân loại hình ảnh mạnh mẽ hơn, mà cuối cùng ta sẽ tự nhiên mở rộng cho toàn bộ mạng neuron và mạng meuron tích chập. Phương pháp này sẽ bao gồm hai thành phần chính: một hàm điểm số (Score Function) ánh xạ dữ liệu thô thành điểm số loại (Class Score), và một hàm mất mát định lượng sự phù hợp giữa điểm số dự đoán và nhãn thực cơ bản (Ground Truth Label). Sau đó, chúng ta sẽ đưa bài toán này vào một bài toán tối ưu hóa, trong đó chúng ta sẽ tối thiểu hóa hàm mất mát theo các tham số của hàm điểm số.

4.4.2. Ánh xạ được tham số hóa từ các hình ảnh đến các điểm số nhãn (Parameterized Mapping from Images to Label Scores)

Thành phần đầu tiên của phương pháp này là định nghĩa hàm điểm số (Score Function), ánh xạ các giá trị pixel của ảnh thành điểm số tin cậy cho từng loại. Chúng ta sẽ phát triển phương pháp này bằng một ví dụ cụ thể. Như trước đây, hãy giả sử một tập dữ liệu huấn luyện gồm các hình ảnh $x_i \in \mathbb{R}^D$, mỗi ảnh được liên kết với một nhãn y_i . Trong đó $y_i = 1 \ldots M$ và $i \in 1 \ldots N$, nghĩa là, ta có N ví dụ (mỗi ví dụ có kích thước D) và M thể loại (Category) riêng biệt. Ví dụ, trong CIFAR-10, chúng ta có một tập huấn luyện gồm **N** = 50.000 ảnh, mỗi ảnh có **D** = 32 x 32 x 3 = 3072 pixel, và **M** = 10, vì có 10 loại riêng biệt (chó, mèo, ô tô, v.v.). Bây giờ, chúng ta sẽ định nghĩa hàm điểm số $f : \mathbb{R}^D \to \mathbb{R}^K$ ánh xạ các pixel ảnh của hình ảnh thô thành điểm số của loại (Class Score).

4.4.3. Phân loại tuyến tính.

Trong mô-đun này, chúng ta sẽ bắt đầu với hàm có lẽ là đơn giản nhất, một ánh xạ tuyến Tính (xem phương trình (3.1) và (3.2)):

$$f\left(\boldsymbol{x}_i, \boldsymbol{W}, \boldsymbol{b}\right) = \boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{b} = \boldsymbol{x}_i^T \boldsymbol{W} + \boldsymbol{b} \qquad (4.3)$$

Trong đó $\boldsymbol{x}_i \in \mathbb{R}^{D\times 1}, \boldsymbol{b} \in \mathbb{R}^{M\times 1}, \boldsymbol{W} \in \mathbb{R}^{D\times M}$ và $\boldsymbol{W}^T \in \mathbb{R}^{M\times D}$

Trong đó D là kích thước ảnh đầu vào được làm phẳng (được kép dãn), M là số loại. Tổng quát a có thể biểu diễn vectơ đầu vào $\boldsymbol{x}_i$,vecttơ độ lệch $\boldsymbol{b}$ và ma trận trong số $\boldsymbol{W}$ như sau:

$$\boldsymbol{x}_i = \begin{bmatrix} x_1 & x_2 & \ldots & x_j & \ldots & x_D \end{bmatrix}^T, \boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & \ldots & b_i & \ldots & b_M \end{bmatrix}^T$$

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,i} & \cdots & w_{1,M} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,i} & \cdots & w_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{j,1} & w_{i,2} & \cdots & w_{j,i} & \cdots & w_{j,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{D,1} & w_{D,2} & \cdots & w_{D,i} & \cdots & w_{D,M} \end{bmatrix}, \boldsymbol{W}^T = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,i} & \cdots & w_{1,D} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,i} & \cdots & w_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{j,1} & w_{i,2} & \cdots & w_{j,i} & \cdots & w_{j,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{M,1} & w_{M,2} & \cdots & w_{M,i} & \cdots & w_{M,D} \end{bmatrix}$$

Trong phương trình trên, chúng ta giả định rằng hình ảnh $\boldsymbol{x}_i$ có tất cả các pixel ảnh được làm phẳng thành một vectơ cột đơn $\boldsymbol{x}_i$ kích thước [D × 1]. Ma trận $\boldsymbol{W}$ kích thước [D xM] và vectơ $\boldsymbol{b}$ kích thước [M x 1] là các tham số của hàm và $\boldsymbol{W}^T$ là ma trận chuyển vị của W. Trong CIFAR-10, $\mathbf{x}_i$ với D= 32×32×3= 3072 chứa tất cả các pixel ảnh trong ảnh thứ i được làm phẳng trong một cột tạo thành vecơ với kích thước [3072 x 1], $\boldsymbol{W}$ là một ma trận với kích thước [3072×10] với M=10, $\boldsymbol{W}^T$ là một ma trận với kích thước [10 x 3072] và $\boldsymbol{b}$ là một vectơ với kích thước [10 x 1], do đó 3072 số được đưa vào hàm (giá trị pixel của ảnh thô) và 10 số được đưa ra (điểm số của loại). Các tham số trong $\boldsymbol{W}$ thường được gọi là trọng số , và $\boldsymbol{b}$ được gọi là vectơ độ lệch vì nó ảnh hưởng đến điểm số đầu ra, nhưng không tương tác với dữ liệu thực tế $\boldsymbol{x}_i$.Tuy nhiên, thường thì các thuật ngữ trọng số và tham số thay thế cho nhau.
Một số điểm lưu ý:

- Đầu tiên, lưu ý rằng phép nhân ma trận đơn$\boldsymbol{W}^T \times \boldsymbol{x}_i$ thực sự đang đánh giá 10 bộ phân loại (Classifier) riêng biệt song song (một bộ cho mỗi loại), trong đó mỗi bộ phân loại là một cột của $\boldsymbol{W}$ hay một hàng của $\boldsymbol{W}^T$ tương ứng.
- Lưu ý rằng ta coi rằng dữ liệu đầu vào ($\boldsymbol{x}_i$,$\boldsymbol{y}_i$) đã cho trước và cố định, nhưng ta có quyền kiểm soát việc thiết lập các tham số $\boldsymbol{W}, \boldsymbol{b}$. Mục tiêu của ta là thiết lập các tham số này sao cho điểm số được tính toán khớp với nhãn thực cơ bản trên toàn bộ tập huấn luyện, nghĩa là $\hat{\boldsymbol{y}}$ khớp nhất với $\boldsymbol{y}_i$. Chúng ta sẽ đi sâu hơn vào cách thực hiện việc này, nhưng trực giác mách bảo chúng ta rằng loại đúng sẽ có điểm số cao hơn điểm số của các loại sai.
- Ưu điểm của phương pháp này là dữ liệu huấn luyện được sử dụng để học các tham số $\boldsymbol{W}, \boldsymbol{b}$, nhưng sau khi quá trình học hoàn tất, chúng ta có thể loại bỏ toàn bộ tập huấn luyện và chỉ giữ lại các tham số đã học. Điều này là do một ảnh kiểm tra (Test Image) mới có thể được chuyển tiếp dễ dàng qua hàm và được phân loại dựa trên điểm số đã tính toán.
- Cuối cùng, lưu ý rằng việc phân loại ảnh kiểm tra bao gồm phép nhân và phép cộng ma trận đơn, nhanh hơn đáng kể so với việc so sánh ảnh kiểm tra với tất cả ảnh đào tạo.

Có thể nói trước rằng mạng neuron tích chập (mạng neuron tích chập sẽ xét trong chương sau) sẽ ánh xạ các điểm ảnh của hình ảnh thành điểm số chính xác như hiển thị ở trên, nhưng việc ánh xạ (f) sẽ phức tạp hơn và chứa nhiều tham số hơn.

4.4.3.1. Mô tả bộ phân loại tuyến tính (Linear Classifier)

Lưu ý rằng bộ phân loại tuyến tính tính điểm số của một loại là tổng có trọng số của tất cả các giá trị pixel của loạiđó trên cả 3 kênh màu. Tùy thuộc vào chính xác các giá trị chúng ta đặt cho các trọng số này, hàm có khả năng thích hoặc không thích (tùy thuộc vào dấu của từng trọng số) một số màu nhất định tại các vị trí nhất định trong ảnh. Ví dụ, ta có thể hình dung rằng loại "tàu thủy" có thể có nhiều khả năng hơn nếu có nhiều màu xanh lam ở hai bên ảnh (có thể tương ứng với nước). Ta có thể mong đợi rằng bộ phân loại "tàu thủy" khi đó sẽ có nhiều trọng số dương trên các trọng số kênh xanh lam của nó (sự hiện diện của màu xanh lam làm tăng điểm số của tàu thủy), và các trọng số âm trong các kênh đỏ/xanh lá cây (sự hiện diện của màu đỏ/xanh lá cây làm giảm điểm số của tàu thủy).

Thí dụ

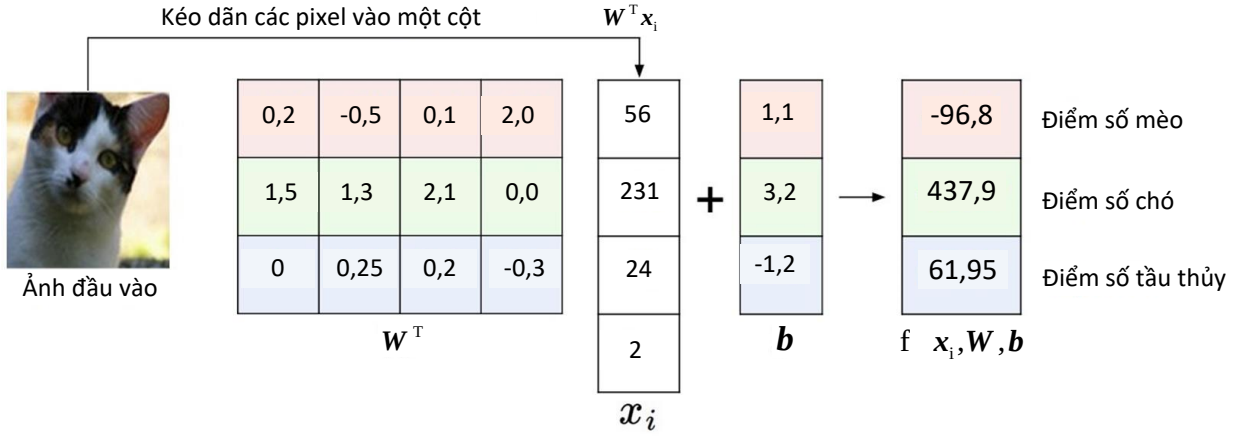
$$\text{cho: } \boldsymbol{W} = \begin{bmatrix} 0,2 & 1,5 & 0 \\ -0,5 & 1,3 & 0,25 \\ 0,1 & 2,1 & 0,2 \\ 2,0 & 0,0 & -0,3 \end{bmatrix} \rightarrow \boldsymbol{W}^{\mathrm{T}} = \begin{bmatrix} 0,2 & -0,5 & 0,1 & 2,0 \\ 1,5 & 1,3 & 2,1 & 0,0 \\ 0 & 0,25 & 0,2 & -0.3 \end{bmatrix},$$

$$\mathbf{X}_{\mathrm{i}} = \begin{bmatrix} 56 & 24 \\ 231 & 2 \end{bmatrix} \rightarrow \boldsymbol{x}_{\mathrm{i}} = \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 1,1 \\ 3,2 \\ -1,2 \end{bmatrix}$$

Ánh xạ của hình ảnh đầu vào vào các điểm số (Score) trong trường hợp này như sau:

$$\hat{\boldsymbol{y}} = \mathrm{f}\ \ \boldsymbol{x}_{\mathrm{i}}, \boldsymbol{W}, \boldsymbol{b}\ = \begin{bmatrix} 0,2 & -0,5 & 0,1 & 2,0 \\ 1,5 & 1,3 & 2,1 & 0,0 \\ 0 & 0,25 & 0,2 & -0.3 \end{bmatrix} \times \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \begin{bmatrix} 1,1 \\ 3,2 \\ -1,2 \end{bmatrix} = \begin{bmatrix} -96,8 \\ 437,9 \\ 61,2 \end{bmatrix}$$

Hình 4.10 minh họa ánh xạ một hình ảnh vào các điểm số của loại theo thí dụ trên.



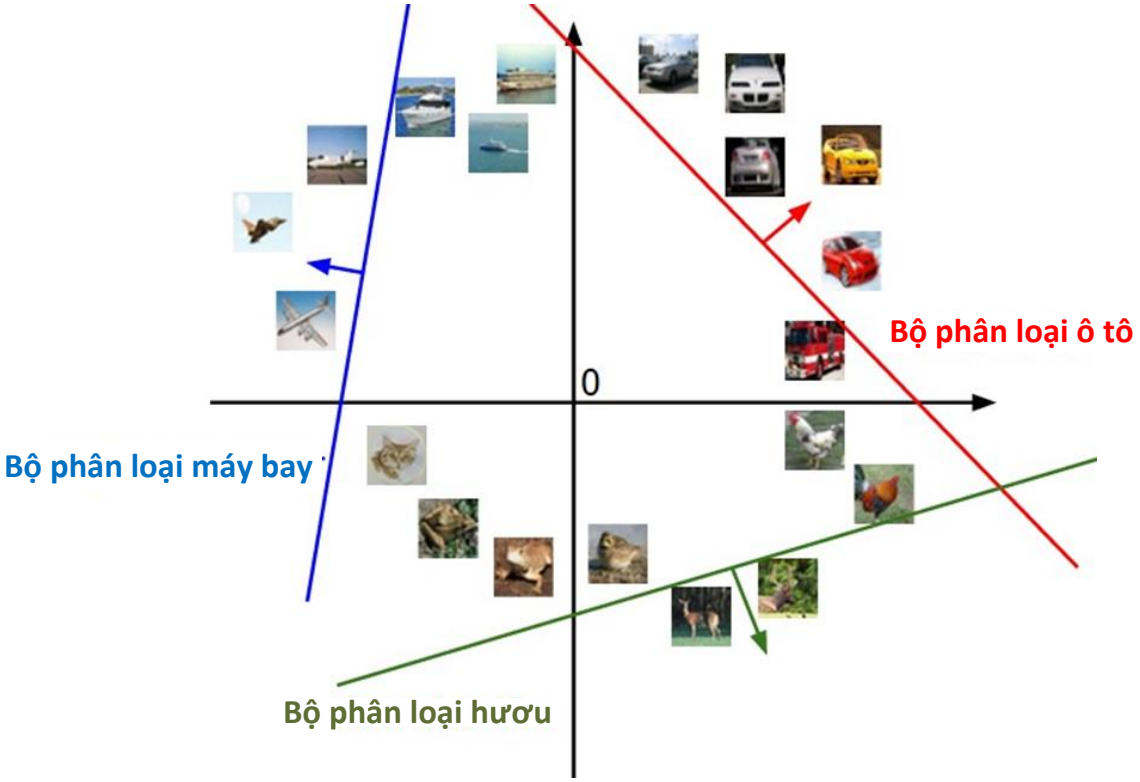
Hình 4.19. Một ví dụ về việc ánh xạ hình ảnh vào điểm số loại. Để dễ hình dung, chúng ta giả định hình ảnh chỉ có 4 pixel (4 pixel đơn sắc, chúng ta không xét đến các kênh màu trong ví dụ này cho ngắn gọn) và chúng ta có 3 loại (loại đỏ (mèo),

xanh lá cây (chó), xanh lam (tàu thủy)). (lưu ý: cụ thể, các màu ở đây chỉ biểu thị 3 loại và không liên quan đến các kênh RGB.) Chúng ta kéo dài các pixel hình ảnh thành một cột (x_i) và thực hiện phép nhân ma trận trọng số chuyển vị (W^T) , cộng độ lệch b để có được điểm số cho mỗi loại. Lưu ý rằng tập hợp các trọng số W cụ thể này không tốt chút nào: các trọng số gán cho hình ảnh con mèo của chúng ta một điểm số mèo rất thấp. Cụ thể, tập hợp trọng số này dường như tin chắc rằng nó đang nhìn vào một con chó.

4.4.3.2. Sự tương tự của hình ảnh như các điểm đa chiều

Vì các hình ảnh được kéo dài thành các vectơ cột kích thước lớn, chúng ta có thể diễn giải mỗi hình ảnh như một điểm duy nhất trong không gian này (ví dụ: mỗi hình ảnh trong CIFAR-10 là một điểm trong không gian kích thước 3072 của 32x32x3 pixel). Tương tự, toàn bộ tập dữ liệu là một tập hợp các điểm (được gắn nhãn).

Vì chúng ta đã định nghĩa điểm số của mỗi loại là tổng được đánh trọng số trọng số của tất cả các pixel của ảnh, nên điểm số của mỗi loại là một hàm tuyến tính trên không gian này. Chúng ta không thể hình dung được không gian 3072 chiều, nhưng nếu chúng ta hình dung việc nén tất cả các chiều đó lại thành chỉ hai chiều, thì chúng ta có thể thử hình dung xem bộ phân loại có thể đang làm gì (hình 4.111).



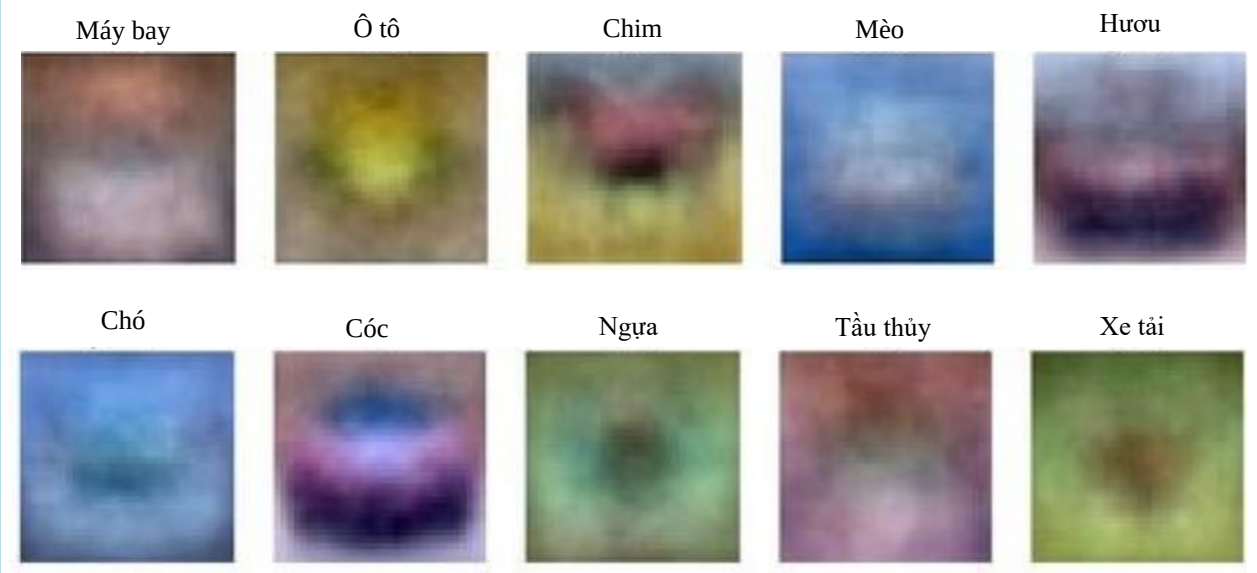
Hình 4.11. **Minh họa biểu diễn không gian hình ảnh, trong đó mỗi hình ảnh là một điểm duy nhất và ba bộ phân loại được trực quan hóa. Sử dụng ví dụ về bộ phân**

loại ô tô (màu đỏ), đường màu đỏ hiển thị tất cả các điểm trong không gian có điểm số bằng 0 cho loại ô tô. Mũi tên màu đỏ thể hiện hướng tăng, do đó tất cả các điểm bên phải đường màu đỏ đều có điểm dương (và tăng tuyến tính), và tất cả các điểm bên trái đều có điểm âm (và giảm tuyến tính)

Như chúng ta đã thấy ở trên, mỗi cộtg của $\boldsymbol{W}$ (tương ứng với mỗi hàng của $\boldsymbol{W}^T$) là một bộ phân loại cho một trong các loại. Giải thích hình học của những con số này là khi chúng ta thay đổi một trong các cột của $\boldsymbol{W}$, đường tương ứng trong không gian pixel ảnh sẽ xoay theo các hướng khác nhau. Mặt khác, các độ lệch $\boldsymbol{b}$ cho phép các bộ phân loại của chúng ta dịch các đường. Đặc biệt, lưu ý rằng nếu không có các thành phần độ lệch việc đưa vào $\boldsymbol{x}_i = 0$ sẽ luôn cho điểm số bằng 0 bất kể trọng số, do đó tất cả các đường thẳng sẽ buộc phải cắt qua gốc tọa độ.

4.4.3.3. Diễn giải các bộ phân loại tuyến tính như một phép so khớp mẫu

Một diễn giải khác cho các trọng số $\boldsymbol{W}$ đó là mỗi cột $\boldsymbol{W}$ tương ứng với một *mẫu* (Template) (hoặc đôi khi còn được gọi là *nguyên mẫu* (Protype)) cho một trong các loại. Điểm số (Score) của mỗi loại cho một hình ảnh sau đó được lấy bằng cách so sánh từng mẫu với hình ảnh bằng cách sử dụng *tích vô hướng* (inner product hay dot product) từng từng mẫu một để tìm ra loại "phù hợp" nhất. Với thuật ngữ này, bộ phân loại tuyến tính đang thực hiện khớp mẫu, trong đó các mẫu được học. Một cách khác để nghĩ về nó là chúng ta vẫn đang thực hiện Nearest Neighbor một cách hiệu quả, nhưng thay vì có hàng nghìn hình ảnh đào tạo, chúng ta chỉ sử dụng một hình ảnh duy nhất cho mỗi loại (mặc dù chúng ta sẽ học nó và nó không nhất thiết phải là một trong những hình ảnh trong tập đào tạo) và chúng ta sử dụng tích vô hướng làm khoảng cách thay vì khoảng cách d_1 hoặc d_2 (hình 4.12).



Hình 4.12. Ví dụ về các trọng số đã được học ở cuối quá trình học cho CIFAR-10. Lưu ý rằng, ví dụ, mẫu tàu thủy (Ship Template) chứa rất nhiều điểm ảnh màu xanh lam như mong đợi. Do đó, mẫu này sẽ cho điểm cao khi được so sánh với hình ảnh tàu thủy trên biển bằng tích vô hướng

Ngoài ra, lưu ý rằng mẫu ngựa (Horse Template)dường như chứa một con ngựa hai đầu, điều này là do cả ngựa quay mặt trái và phải trong tập dữ liệu. Bộ phân loại tuyến tính *hợp nhất* hai chế độ ngựa này trong dữ liệu thành một mẫu duy nhất. Tương tự, bộ phân loại ô tô dường như đã hợp nhất một số chế độ thành một mẫu duy nhất phải xác định ô tô từ mọi phía và mọi màu sắc. Cụ thể, mẫu này cuối cùng có màu đỏ, điều này gợi ý rằng có nhiều ô tô màu đỏ trong tập dữ liệu CIFAR-10 hơn bất kỳ màu nào khác. Bộ phân loại tuyến tính quá yếu để tính toán chính xác các ô tô có màu khác nhau, nhưng như chúng ta sẽ thấy sau, mạng neuron sẽ cho phép chúng ta thực hiện nhiệm vụ này. Nhìn về phía trước một chút, một mạng neuron sẽ có thể phát triển các neuron trung gian trong các loạiẩn của nó có thể phát hiện các loại ô tô cụ thể (ví dụ: ô tô màu xanh lá cây quay mặt trái, ô tô màu xanh lam quay mặt về phía trước, v.v.) và các neuron trên loạitiếp theo có thể kết hợp những điều này thành một điểm số ô tô chính xác hơn thông qua tổng trọng số của các bộ phát hiện ô tô riêng lẻ.

4.4.3.4. Thủ thuật kết hợp ma trận trọng số và vectơ độ lệch

Trước khi tiếp tục, ta xet đến một thủ thuật đơn giản hóa phổ biến để biểu diễn hai tham số $\boldsymbol{W}$, $\boldsymbol{b}$ như một tham số. Nhắc lại rằng chúng ta đã định nghĩa hàm điểm số như sau:

$$f\ \boldsymbol{x}_i, \boldsymbol{W}, \boldsymbol{b}\ = \boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{b} = \boldsymbol{x}_i^T \boldsymbol{W} + \boldsymbol{b}$$

Việc theo dõi riêng biệt hai tập tham số (độ lệch $\boldsymbol{b}$ và trọng số $\boldsymbol{W}$) hơi rườm ra. Một mẹo thường được sử dụng là kết hợp hai tập tham số thành một ma trận duy nhất chứa cả hai bằng cách mở rộng vectơ $\boldsymbol{x}_i$ với một chiều bổ sung luôn giữ nguyên hằng số 1- một *chiều* độ lệch mặc định. Ta có thể chèn $\boldsymbol{b}$ vào hàng thứ nhất của ma trận W và 1 vào phần tử

đầuu tiên của vecơ $\boldsymbol{x}_i$ như đã thấy trong hình 3.7, hoặc tương tự chèn $\boldsymbol{b}$ vào hàng cuối cùng của ma trận $\boldsymbol{W}$ và 1 vào phần tử cuối cùng của vecơ $\boldsymbol{x}_i$, hàm điểm số mới sẽ được đơn giản hóa thành phép nhân ma trận đơn:

$$f\left(\boldsymbol{x}_i, \boldsymbol{W}\right) = \boldsymbol{W}^T \boldsymbol{x}_i = \boldsymbol{x}_i^T \boldsymbol{W} \qquad (4.4)$$

trong đó

$$\boldsymbol{x}_i = \begin{bmatrix} x_1 & x_2 & \dots & x_i & \dots & x_D & 1 \end{bmatrix}^T$$

$$\boldsymbol{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,i} & \cdots & w_{1,M} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,i} & \cdots & w_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{j,1} & w_{i,2} & \cdots & w_{j,i} & \cdots & w_{j,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{D,1} & w_{D,2} & \cdots & w_{D,i} & \cdots & w_{D,M} \\ b_1 & b_2 & \dots & b_i & \dots & b_M \end{bmatrix}$$

Với ví dụ CIFAR-10 của chúng ta, $\boldsymbol{x}_i$ bây giờ là [3073 x 1] thay vì [3072 x 1] - (với chiều bổ sung có hằng số 1) và $\boldsymbol{W}$ bây giờ là [3073×10] thay vì [3072×10] và $\boldsymbol{W}^T$ bây giờ là [10 × 3073] thay vì [10 × 3072].

Thí dụ.

Giả sử cho trước

$$\boldsymbol{W} = \begin{bmatrix} 0,2 & 1,5 & 0 \\ -0,5 & 1,3 & 0,25 \\ 0,1 & 2,1 & 0,2 \\ 2,0 & 0,0 & -0,3 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 1,1 \\ 3,2 \\ -1,2 \end{bmatrix}, \boldsymbol{x}_i = \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}$$

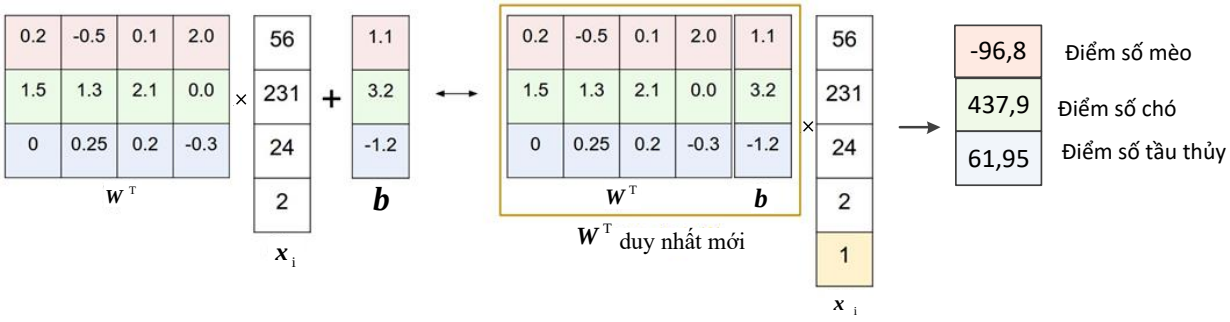
Chèn $\boldsymbol{b}$ vào cột cuối cùng của $\boldsymbol{W}$ và 1 vào phần tử cuối của $\boldsymbol{x}_i$, ta được:

$$\boldsymbol{W} = \begin{bmatrix} 0,2 & 1,5 & 0 \\ -0,5 & 1,3 & 0,25 \\ 0,1 & 2,1 & 0,2 \\ 2,0 & 0,0 & -0,3 \\ 1,1 & 3,2 & -1,2 \end{bmatrix} \rightarrow \boldsymbol{W}^T = \begin{bmatrix} 0,2 & -0,5 & 0,1 & 2,0 & 1,1 \\ 1,5 & 1,3 & 2,1 & 0,0 & 3,2 \\ 0 & 0,25 & 0,2 & -0,3 & -1,2 \end{bmatrix}, \boldsymbol{x}_i = \begin{bmatrix} 56 \\ 231 \\ 21 \\ 2 \\ 1 \end{bmatrix}$$

Ánh xạ hình ảnh với điểm số loại theo phương trình (4.4) như sau:

$$f(x_i, W) = \begin{bmatrix} 0,2 & -0,5 & 0,1 & 2,0 & 1,1 \\ 1,5 & 1,3 & 2,1 & 0,0 & 3,2 \\ 0 & 0,25 & 0,2 & -0,3 & -1,2 \end{bmatrix} \times \begin{bmatrix} 56 \\ 231 \\ 21 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -96,8 \\ 437,9 \\ 61,95 \end{bmatrix}$$

Hình 4.13 minh họa ánh xạ một hình ảnh vào các điểm số của loại theo thí dụ trên.



Hình 4.13. Minh họa thủ thuật kết hợp ma trận trọng số và vectơ độ lệnh

4.4.4. Tiền xử lý dữ liệu hình ảnh

Lưu ý, trong các ví dụ trên, ta đã sử dụng các giá trị pixel thô (nằm trong khoảng từ [0…255]). Trong học máy, việc luôn thực hiện chuẩn hóa các đặc trưng đầu vào là một thực hành rất phổ biến (trong trường hợp hình ảnh, mỗi pixel được coi là một tính năng (Feature)). Đặc biệt, điều quan trọng là phải định tâm dữ liệu (Center Data) của ta bằng cách trừ giá trị trung bình khỏi mọi tính năng. Trong trường hợp hình ảnh, điều này tương ứng với việc tính toán một hình ảnh trung bình trên các hình ảnh đào tạo và trừ nó khỏi mọi hình ảnh để có được các hình ảnh có các pixel nằm trong khoảng từ xấp xỉ [-127 … 127]. Tiền xử lý phổ biến hơn nữa là chia tỷ lệ từng đặc trưng đầu vào sao cho các giá trị của nó nằm trong khoảng từ [-1, 1]. Từ đó, ta thấy việc định tâm trung bình trung bình bằng 0 có thể được cho là quan trọng hơn nhưng chúng ta sẽ phải đợi để chứng minh cho đến khi chúng ta hiểu được động lực của quá trình giảm dần độ dốc.

4.5. HÀM MẤT MÁT (LOSS FUNCTION) VÀ CHÍNH QUY HÓA (REGULARIZATION)

4.5.1. Tổng quan

Trong phần trước, chúng ta đã định nghĩa một hàm ánh xạ các giá trị pixel đến điểm số của loại, được tham số hóa bởi một tập hợp các trọng số $\boldsymbol{W}$. Hơn nữa, ta thấy rằng không kiểm soát được dữ liệu $(\boldsymbol{x}_i, y_i)$ (được cố định và được cho trước), nhưng ta có thể kiểm soát các trọng số này và ta muốn thiết lập chúng sao cho điểm số của dự đoán phù hợp với nhãn thực cơ bản (Ground Truth Label) trong dữ liệu đào tạo.

Ví dụ, quay lại hình ảnh ví dụ về một con mèo và điểm số của nó cho các loại "mèo", "chó" và "tàu thủy", chúng ta thấy rằng tập hợp trọng số cụ thể trong ví dụ đó không tốt

chút nào: ta đã đưa vào các pixel mô tả một con mèo nhưng điểm số của mèo lại rất thấp (-96,8) so với các loại khác (điểm chó là 437,9 và điểm tàu tầu thủy là 61,95). Chúng ta sẽ đo lường mức độ không hài lòng của mình với các kết quả như thế này bằng hàm mất mát (hoặc đôi khi còn được gọi là hàm chi phí hoặc mục tiêu). Theo trực giác, mức mất mát sẽ cao nếu chúng ta phân loại dữ liệu đào tạo kém và sẽ thấp nếu chúng ta làm tốt.

4.5.1.2. Hàm mất mát của máy vectơ hỗ trợ nhiều loại (Multiclass Support Vector Machine)

Có một số cách để xác định chi tiết hàm mất mát. Ví dụ đầu tiên, chúng ta sẽ phát triển một hàm mất mát thường được sử dụng, gọi là hàm mất mát của máy vectơ hỗ trợ (SVM: Support Vector Machine) nhiều loại. Hàm mất mát SVM được thiết lập sao cho SVM "muốn" loại đúng cho mỗi ảnh có điểm số cao hơn loại sai theo một biên độ cố định Δ. Lưu ý rằng đôi khi việc nhân cách hóa hàm mất mát như chúng ta đã làm ở trên sẽ hữu ích: SVM "muốn" một kết quả nhất định theo nghĩa là kết quả đó sẽ mang lại mức mất mát thấp hơn (điều này là tốt).

Bây giờ chúng ta hãy đi sâu hơn vào chi tiết. Nhắc lại đối với ví dụ thứ i, chúng ta được cho trước các pixel của hình ảnh $\boldsymbol{x}_i$ và nhãn y_i là một số nguyên, đặc tả chỉ số của loại chính xác (Corect Class). Hàm điểm số lấy các pixel ảnh và tính toán vectơ $f(\boldsymbol{x}_i, \boldsymbol{W})$ của các điểm số của loại. Ta sẽ ký hiệu vecơ này như sau:

$$\hat{y} = f\left(\boldsymbol{x}_i, \boldsymbol{W}\right)$$

Điểm số của loại thứ j (Score of j-th class) là phần tử thứ j

$$\hat{y}_j = f\left(\boldsymbol{x}_i, \boldsymbol{W}\right)_j$$

Mất mát SVM đa loạicho ví dụ thứ i có thể được viết như sau:

$$L_i = \sum_{j \neq y_i} \max\left(0, (\hat{y}_j - \hat{y}_{y_i} + \boldsymbol{\Delta})\right)$$

Lưu ý y_i là một số nguyên được gán cho từng nhãn.

Thí dụ.

Giả sử chúng ta có ba loại nhận các điểm số $\hat{y} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \hat{y}_3 \end{bmatrix}^T = \begin{bmatrix} 13 & -7 & 11 \end{bmatrix}^T$, và loại đầu tiên là loạiđúng (tức là $y_i=1$). Cũng giả sử Δ (một siêu tham số mà chúng ta sẽ tìm hiểu chi tiết hơn sau này) bằng 10. Biểu thức trên tính tổng của tất cả các loạikhông đúng ($j \neq y_i$, cụ thể là $j \neq 1$). Trong trường hợp này $y_i=1$, $j=2,3$, do đó chúng ta có hai thành phần:

$$L_i = \sum_{j=2}^{3} \max\left(0, (\hat{y}_j - \hat{y}_1 + 10)\right)$$

$$L_i = \max\left(0, (-7 - 13 + 10)\right) + \max\left(0, (11 - 13 + 10)\right)$$

$$L_i = \max\ 0, -10\ + \max\ 0, 8)\ = 0 + 8 = 8$$

Ta có thể thấy rằng thành phần đầu tiên đưa ra số không vì [-7 - 13 + 10] đưa ra một số âm, sau đó được ngưỡng hóa thành số không bằng max hàm {0,-}. Chúng ta nhận được mất mát bằng không cho cặp này vì điểm loạiđúng (13) lớn hơn điểm loạikhông đúng (-7) ít nhất là biên độ 10. Trên thực tế, hiệu số là 20, lớn hơn 10 rất nhiều nhưng SVM chỉ quan tâm đến việc hiệu số ít nhất là 10; Bất kỳ hiệu số nào vượt quá biên độ đều được giữ ở mức không bằng phép toán max. thành phần thứ hai tính toán [11 - 13 + 10] đưa ra kết quả là 8. Nghĩa là, mặc dù loạiđúng có điểm số cao hơn loạikhông đúng (13 > 11), nhưng nó không lớn hơn biên độ mong muốn là 10. Hiệu số chỉ là 2, đó là lý do tại sao mất mát là 8 (tức là hiệu số phải cao hơn bao nhiêu để đạt được biên độ). Tóm lại, hàm mất mát SVM muốn điểm của loạiđúng lớn hơn điểm của loạisai ít nhất là Δ (delta). Nếu không, chúng ta sẽ tích lũy mất mát.

Lưu ý rằng trong mô-đun cụ thể mà chúng ta đang làm việc với nó là các hàm điểm số tuyến tính $f\ \boldsymbol{x}_i, \boldsymbol{W}\ = \boldsymbol{W}^T \boldsymbol{x}_i$, do đó chúng ta cũng có thể viết lại hàm mất mát theo dạng tương đương sau:

$$L_i = \sum_{j \neq y_i} \max\ 0, (\boldsymbol{w}_j^T \boldsymbol{x}_i - \boldsymbol{w}_{y_i}^T \boldsymbol{x}_i + \Delta)$$

Trong đó $\boldsymbol{w}_j$ là cột thứ j của ma trận $\boldsymbol{W}$.

Tuy nhiên, điều này sẽ không nhất thiết đúng khi chúng ta bắt đầu xem xét các dạng phức tạp hơn của hàm điểm số f.

Máy vectơ hỗ trợ nhiều loại "muốn" điểm số của loại đúng phải cao hơn tất cả các điểm khác ít nhất một khoảng delta (hình 4.12). Nếu bất kỳ loại nào có điểm nằm trong vùng đỏ (hoặc cao hơn), thì sẽ có mất mát tích lũy. Nếu không, mất mát sẽ bằng không. Mục tiêu của chúng ta là tìm các trọng số đồng thời thỏa mãn ràng buộc này cho tất cả các ví dụ trong dữ liệu huấn luyện và đưa ra tổng mất mát thấp nhất có thể.

Hình 4.12. Máy vectơ hỗ trợ nhiều loại "muốn" điểm số của loại đúng phải cao hơn tất cả các điểm khác ít nhất một khoảng delta

Một thuật ngữ cuối cùng cần lưu ý trước khi kết thúc phần này là hàm ngưỡng tại 0 thường được gọi là mất mát bản lề (hinge loss). Đôi khi ta sẽ nghe nói về những người sử dụng SVM mất mát bản lề bình phương (hoặc L2-SVM), sử dụng dạng $\max(0, -)^2$ để phạt các biên độ (Margin) vi phạm mạnh hơn (theo phương pháp bậc hai thay vì tuyến tính). Phiên bản không bình phương chuẩn hơn, nhưng trong một số tập dữ liệu, mất mát

bản lề bình phương có thể hoạt động tốt hơn. Điều này có thể được xác định trong quá trình xác thực chéo.

4.5.2. Chính quy hóa

Có một lỗi (Bug) với hàm mất mát mà ta đã xét ở trên. Giả sử rằng chúng ta có một tập dữ liệu và một tập hợp các tham số $\boldsymbol{W}$ phân loại chính xác mọi ví dụ (tức là tất cả các điểm số đều đạt các giá trị sao cho tất cả các biên độ đều được đáp ứng là $L_i=0$ cho mọi i). Vấn đề là tập hợp $\boldsymbol{W}$ này không nhất thiết phải duy nhất: có thể có nhiều $\boldsymbol{W}$ tương tự phân loại chính xác các ví dụ. Một cách dễ dàng để thấy điều này là nếu một số tham số $\boldsymbol{W}$ phân loại chính xác tất cả các ví dụ (do đó mất mát bằng không cho mỗi ví dụ), thì bất kỳ bội số nào của các tham số này $\lambda \boldsymbol{W}$, trong đó $\lambda>1$, cũng sẽ cho mất mát bằng không vì phép biến đổi này kéo dài đồng đều tất cả các độ lớn của điểm số và do đó cũng kéo dài cả sự khác biệt tuyệt đối của chúng. Ví dụ: nếu sự khác biệt về điểm số giữa một loại đúng và một loại sai gần nhất là 15, thì việc nhân tất cả các phần tử của $\boldsymbol{W}$ với 2 sẽ tạo ra sự khác biệt mới là 30.

Nói cách khác, ta muốn mã hóa một ưu tiên nào đó cho một tập các trọng số $\boldsymbol{W}$ nhất định so với các tập hợp khác để loại bỏ sự mơ hồ này. Chúng ta có thể làm như vậy bằng cách mở rộng hàm mất mát với một phạt chính quy hóa (Regulization Penalty). Phạt chính quy hóa phổ biến nhất là chuẩn ℓ_2 bình phương để ngăn cản các trọng số lớn thông qua một hình phạt bậc hai theo từng phần tử trên tất cả các tham số:

$$R(\boldsymbol{W}) = \sum_k \sum_\ell \boldsymbol{W}_{k,\ell}^2$$

Trong biểu thức trên, ta đang tính tổng tất cả các phần tử được bình phương của $\boldsymbol{W}$. Lưu ý rằng hàm chính quy hóa không phải là một hàm của dữ liệu, nó chỉ dựa trên các trọng số. Việc bao gồm cả phạt chính quy hóa sẽ hoàn thiện toàn bộ mất mát của máy vectơ hỗ trợ nhiều loại, bao gồm hai thành phần: mất mát dữ liệu (là mất mát trung bình trên tất cả các ví dụ) và mất mát chính quy hóa. Nghĩa là, mất mát SVM nhiều loại đầy đủ sẽ trở thành:

$$L = \underbrace{\frac{1}{N}\sum_i L_i}_{\text{data loss}} + \underbrace{\lambda R \quad \mathbf{W}}_{\text{regularization loss}}$$

trongđó: data loss: mất mát dữ liệu, regularization losa: mất mát chính quy hóa.

Hay triển khai phương trình trên và dạng đầy đủ:

$$L = \frac{1}{N}\sum_i \sum_{j \equiv y_i} \Big[\max(0,\ f(\boldsymbol{x}_i;\boldsymbol{W})_j - f(\boldsymbol{x}_i;\boldsymbol{W})_{y_i} + \boldsymbol{\Delta} \ \Big] + \lambda \sum_k \sum_\ell \boldsymbol{W}_{k,\ell}^2$$

Trong đó N là số lượng ví dụ huấn luyện. Như có thể thấy, ta thêm phạt chính quy vào mục tiêu mất mát, được tính trọng số bởi một siêu tham số λ. Không có cách đơn giản

nào để thiết lập siêu tham số này và nó thường được xác định bằng xác thực chéo (Cross-Validation).

Ngoài động lực đã nêu ở trên, còn có nhiều tính chất đáng mong muốn khác cần được đưa vào hình phạt chính quy hóa, nhiều tính chất trong số đó chúng ta sẽ đề cập lại ở các phần sau. Ví dụ, việc đưa hình phạt ℓ_2 vào sẽ dẫn đến tính chất biên độ cực đại (Max Margin) hấp dẫn trong SVM .

Thuộc tính hấp dẫn nhất là việc phạt các trọng số lớn có xu hướng cải thiện khả năng tổng quát hóa, vì điều đó có nghĩa là không có chiều đầu vào nào có thể tự nó có ảnh hưởng rất lớn đến điểm số. Ví dụ, giả sử rằng chúng ta có một số vectơ đầu vào $\boldsymbol{x} = 1 \quad 1 \quad 1 \quad 1^{\mathrm{T}}$ và hai vectơ trọng số $\boldsymbol{w}_1 = 1 \quad 0 \quad 0 \quad 0^{\mathrm{T}}$, $\boldsymbol{w}_2 = 0{,}25 \quad 0{,}25 \quad 0{,}25 \quad 0{,}25^{\mathrm{T}}$. Khi này $\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}$, nên cả hai vectơ trọng số đều dẫn đến cùng một tích vô hướng, nhưng phạt ℓ_2 của $\boldsymbol{w}_1$ là 1,0 trong khi phạt L2 của $\boldsymbol{w}_2$ chỉ là 0,25. Do đó, theo hình phạt ℓ_2, vectơ trọng số w_2 sẽ được ưu tiên hơn vì nó đạt được mất mát chính quy thấp hơn. Theo trực giác, điều này là do các trọng số trong $\boldsymbol{w}_2$ nhỏ hơn và khuếch tán hơn. Vì hình phạt ℓ_2 ưu tiên các vectơ trọng số nhỏ hơn và khuếch tán hơn, bộ phân loại cuối cùng được khuyến khích tính đến tất cả các chiều đầu vào với số lượng nhỏ thay vì chỉ một vài chiều đầu vào và rất mạnh. Như chúng ta sẽ thấy dưới đây, hiệu ứng này có thể cải thiện hiệu suất tổng quát hóa của bộ phân loại trên ảnh thử nghiệm và dẫn đến ít *quá khớp (Overfitting) ít hơn* .

Lưu ý rằng độ lệch không có cùng tác động, vì không giống như trọng số, chúng không kiểm soát cường độ ảnh hưởng của kicha thước đầu vào (Input Dumention). Do đó, thông thường chỉ điều chỉnh các trọng số $\boldsymbol{W}$ chứ không phải độ lệch $\boldsymbol{b}$. Tuy nhiên, trên thực tế, điều này thường có tác động không đáng kể. Cuối cùng, lưu ý rằng do ảnh hưởng của độ lệch điều chỉnh, chúng ta không bao giờ có thể đạt được độ lệch chính xác bằng 0,0 trên tất cả các ví dụ, vì điều này chỉ khả thi trong bối cảnh bệnh lý $\boldsymbol{W}$= 0

Dưới đây là đoạn mã cho hàm mất mát (không có chính quy hóa) được triển khai bằng Python, ở cả dạng không vectơ hóa và nửa vectơ hóa:

```
def L_i(x, y, W):
  """
  unvectorized version. Compute the multiclass svm loss for a single example (x,y)
  - x is a column vector representing an image (e.g. 3073 x 1 in CIFAR-10)
    with an appended bias dimension in the 3073-rd position (i.e. bias trick)
  - y is an integer giving index of correct class (e.g. between 0 and 9 in CIFAR-10)
  - W is the weight matrix (e.g. 10 x 3073 in CIFAR-10)
  """
  delta = 1.0 # see notes about delta later in this section
  scores = W.dot(x) # scores becomes of size 10 x 1, the scores for each class
  correct_class_score = scores[y]
  D = W.shape[0] # number of classes, e.g. 10
  loss_i = 0.0
```

```
  for j in range(D): # iterate over all wrong classes
    if j == y:
      # skip for the true class to only loop over incorrect classes
      continue
    # accumulate loss for the i-th example
    loss_i += max(0, scores[j] - correct_class_score + delta)
  return loss_i

def L_i_vectorized(x, y, W):
  """
  A faster half-vectorized implementation. half-vectorized
  refers to the fact that for a single example the implementation contains
  no for loops, but there is still one loop over the examples (outside this function)
  """
  delta = 1.0
  scores = W.dot(x)
  # compute the margins for all classes in one vector operation
  margins = np.maximum(0, scores - scores[y] + delta)
  # on y-th position scores[y] - scores[y] canceled and gave delta. We want
  # to ignore the y-th position and only consider margin on max wrong class
  margins[y] = 0
  loss_i = np.sum(margins)
  return loss_i

def L(X, y, W):
  """
  fully-vectorized implementation :
  - X holds all the training examples as columns (e.g. 3073 x 50,000 in CIFAR-10)
  - y is array of integers specifying correct class (e.g. 50,000-D array)
  - W are weights (e.g. 10 x 3073)
  """
  # evaluate loss over all examples in X without using any for loops
  # left as exercise to reader in the assignment
```

Bài học rút ra từ phần này là SVM sử dụng một phương pháp cụ thể để đo lường mức độ nhất quán giữa các dự đoán trên dữ liệu huấn luyện với các nhãn thực tế cơ bản. Ngoài ra, việc đưa ra các dự đoán tốt trên tập huấn luyện cũng tương đương với việc giảm thiểu mất mát.

4.5.3. Thiết lập Delta

Lưu ý rằng chúng ta đã xét qua siêu tham số Δ và thiết lập của nó. Giá trị nên đặt cho nó là bao nhiêu và chúng ta có phải kiểm tra chéo không? Hóa ra siêu tham số này có thể

được đặt an toàn thành Δ=1.0 trong mọi trường hợp. Các siêu tham số Δ và λ có vẻ như là hai siêu tham số khác nhau, nhưng trên thực tế, cả hai đều kiểm soát cùng một sự đánh đổi: sự đánh đổi giữa mất mát dữ liệu và mất mát chính quy trong mục tiêu. Chìa khóa để hiểu điều này là độ lớn của các trọng số $\boldsymbol{W}$ có tác động trực tiếp đến các điểm số (và do đó cũng ảnh hưởng đến sự khác biệt của chúng): khi chúng ta thu nhỏ tất cả các giá trị bên trong $\boldsymbol{W}$, sự khác biệt về điểm số sẽ trở nên thấp hơn và khi chúng ta tăng tỷ lệ các trọng số, tất cả sự khác biệt về điểm số sẽ trở nên cao hơn. Do đó, giá trị chính xác của biên độ giữa các điểm số (ví dụ Δ=1 hoặc Δ=100) theo một nghĩa nào đó là vô nghĩa vì các trọng số có thể thu nhỏ hoặc kéo dài sự khác biệt một cách tùy ý. Do đó, sự đánh đổi thực sự duy nhất là chúng ta cho phép trọng số tăng trưởng lớn đến mức nào (thông qua cường độ chính quy hóa λ).

4.5.4. Mối quan hệ với máy vectơ hỗ trợ nhị phân

Ta có thể liên hệ hiểu biết về máy vectơ hỗ trợ nhị phân mà ta đã xét trong chương trước, trong đó mất mát cho ví dụ thứ i có thể được viết như sau:

$$L_i = C\max\left[0,\ 1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i\right] + R\ \boldsymbol{W}$$

trong đó là C một siêu tham số, và $y_i \in -1,1$. Ta có thể thấy rằng công thức được trình bày trong phần này chứa SVM nhị phân như một trường hợp đặc biệt khi chỉ có hai loại. Nghĩa là, nếu chúng ta chỉ có hai loại thì mất mát sẽ giảm xuống SVM nhị phân được chỉ ra ở trên. Ngoài ra, trong C công thức trên và trong công thức cho nhận dạng ảnh kiểm soát cùng một sự đánh đổi và có liên quan thông qua quan hệ nghịch đảo $C \propto \frac{1}{\lambda}$.

4.6. CÁC CÔNG THỨC SVM NHIỀU LOẠI

Cần lưu ý rằng SVM nhiều loại được trình bày trong phần này là một trong số ít cách xây dựng SVM trên nhiều loại. Một dạng thường được sử dụng khác là SVM một đối trọi tấp cả (OVA: One-vs-All) để huấn luyện một SVM nhị phân độc lập cho mỗi loại so với tất cả các loại khác. Có liên quan nhưng ít phổ biến hơn trong thực tế là chiến lược tất cả đối trọi tất cả (AVA: All-vs-All)). Công thức của được xét ở đây tuân theo phiên bản Weston và Watkins 1999, đây là phiên bản mạnh hơn OVA (theo nghĩa là bạn có thể xây dựng các tập dữ liệu đa loạitrong đó phiên bản này có thể đạt được mức mất dữ liệu bằng không, nhưng OVA thì không). Công thức cuối cùng là SVM có cấu trúc (Structured SVM), giúp tối đa hóa biên độ giữa điểm số của loại đúng và điểm số của loại á quân không đúng có điểm cao nhất.

4.6.1. Bộ phân loại Softmax (Softmax Classifier)

Thực ra SVM là một trong hai bộ phân loại thường thấy. Lựa chọn phổ biến khác là bộ phân loại Softmax , có hàm mất mát khác. Trong chương trước ta đã đề cập đến bộ phân loại hồi quy Logistic nhị phân, thì bộ phân loại Softmax là sự tổng quát hóa của nóa của nó cho nhiều loại. Không giống như SVM, vốn xử lý các đầu ra f $\boldsymbol{x}_i, \boldsymbol{W}$ như các điểm số (chưa được hiệu chuẩn và có thể khó diễn giải) cho mỗi loại, bộ phân loại Softmax đưa ra đầu ra trực quan hơn một chút (xác suất loại được chuẩn hóa) và cũng có cách diễn giải xác suất mà ta sẽ xét ngắn gọn dưới đây. Trong bộ phân loại Softmax, hàm ánh xạ vẫn không thay đổi, nhưng giờ đây chúng ta diễn giải các điểm số này như các xác suất logarit chưa được chuẩn hóa cho mỗi loại và thay thế mất mát bản lề bằng mất mát entropy chéo (Cross-Entropy Loss) từ phương trình (3.13) có dạng:

$$L(\boldsymbol{W}) = -\sum_{i=1}^{N} L_i = \sum_{i=1}^{N} \log\ P(y = y_i | \mathbf{x} = \boldsymbol{x}_i, \boldsymbol{W})$$

Trong đó $L_i = \log\ P(y = y_i | \mathbf{x} = \mathbf{x}_i, \boldsymbol{W})$, $P(y = y_i | \mathbf{x} = \boldsymbol{x}_i, \boldsymbol{W})$ là xác suất mà một mẫu cho trước được phân loại đúng.

Nếu sử dụng hàm softmax cho xác suất ta được:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

Hay tương đương

$$L_i = -e^{f_{y_i}} + \log \sum_j e^{f_j}$$

Như trước đây, mất mát đầy đủ cho tập dữ liệu là giá trị trung bình của trên tất cả các ví dụ đào tạo cùng với một thành phần chính quy hóa $R(\mathbf{W})$. Hàm $f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$ được gọi là hàm softmax : Hàm này lấy một vectơ các điểm số có giá trị thực tùy ý (theo z) và nén nó thành một vectơ các giá trị từ 0 đến 1 có tổng bằng 1. Mất mát entropy chéo đầy đủ liên quan đến hàm softmax có thể trông phức tạp nếu ta nhìn thấy nó lần đầu tiên nhưng tương đối dễ hiểu.

Để hiểu được mất mát entropy chéo ta cần ôn lại một số kiến thức về lý thuyết thông tin.

Entropy chéo giữa phân phối "thực" p và phân phối ước tính q được định nghĩa như sau:

$$H(p,q) = \sum p(x) \log q(x)$$

Vì thế, bộ phân loại Softmax giảm thiểu entropy chéo giữa các xác suất của loại được ước tính ($q = e^{f_{y_i}} \Big/ \sum_j e^{f_j}$ như đã thấy ở trên) và phân phối "thực", theo cách giải thích này là phân phối mà mọi đại lượng xác suất đều nằm trên loại chính xác (tức

là $p = 0 \ \ldots \ 1 \ \cdots \ 0^{T}$ chứa một số 1 duy nhất ở vị trí y_i). Hơn nữa, vì entropy chéo có thể được viết theo entropy và độ phân kỳ Kullback-Leibler là $H(p,q) = H(p) + D_{KL}\ p \| q$ và entropy của hàm delta bằng không, điều này cũng tương đương với việc giảm thiểu độ phân kỳ KL giữa hai phân phối (một thước đo khoảng cách). Nói cách khác, mục tiêu entropy chéo muốn phân phối dự đoán có mọi đại lượng của nó trên câu trả lời đúng.

4.6.2. Giải thích theo xác suất

Nhìn vào biểu thức, chúng ta thấy rằng

$$P\ \ y_i \left| \boldsymbol{x}_i, \boldsymbol{W} \right. = \frac{e^{f_{y1}}}{\sum_j e^{f_j}}$$

có thể được diễn giải là xác suất (được chuẩn hóa) được gán cho nhãn đúng y_i cho hình ảnh và được tham số hóa bởi $\boldsymbol{W}$. Để thấy điều này, hãy nhớ rằng bộ phân loại Softmax diễn giải các điểm số bên trong vectơ đầu ra là xác suất logarit không chuẩn hóa. Do đó, việc lũy thừa các đại lượng này sẽ cung cấp xác suất (không chuẩn hóa) và phép chia thực hiện chuẩn hóa để tổng các xác suất bằng một. Trong diễn giải xác suất, vì thế chúng ta đang giảm thiểu logarit likelihood (logarit khả giống) âm của loại đúng, có thể được diễn giải như là thực hiện *Ước lượng khả giống cực đại (MLE: Maximum Likelihood Estimation).* Một tính năng hay của quan điểm này là giờ đây chúng ta cũng có thể diễn giải thuật ngữ chính quy hóa trong hàm mất mát đầy đủ là đến từ một tiên nghiệm Gauss (Gaussian Prior) trên ma trận trọng số $\boldsymbol{W}$, trong đó thay vì MLE, chúng ta đang thực hiện ước *lượng hậu nghiệm cực đại* (MAP: Maximum of Posteriori) . Lưu ý được đề cập đến những diễn giải này giúp ta trực giác tốt hơn, nhưng toàn bộ chi tiết của phép suy luận này nằm ngoài phạm vi được xét ở đây.

4.6.3. Các vấn đề thực tế: Tính ổn định số (Numeric Sability)

Khi viết mã để tính toán hàm Softmax trong thực tế, các thành phần trung gian $e^{f_{y_i}}$ và $\sum_j e^{f_j}$ có thể rất lớn do các hàm mũ. Việc chia các số lớn có thể không ổn định về mặt số, vì vậy điều quan trọng là phải sử dụng thủ thuật chuẩn hóa. Lưu ý rằng nếu chúng ta nhân phần trên và phần dưới của phân số với một hằng số và đưa nó vào tổng, chúng ta sẽ có biểu thức sau (tương đương về mặt toán học):

$$\frac{e^{f_{y_i}}}{\sum_j e^{f_j}} = \frac{Ce^{f_{y_i}}}{C\sum_j e^{f_j}} = \frac{e^{f_{y_i} + \log C}}{\sum_j e^{f_j + \log C}}$$

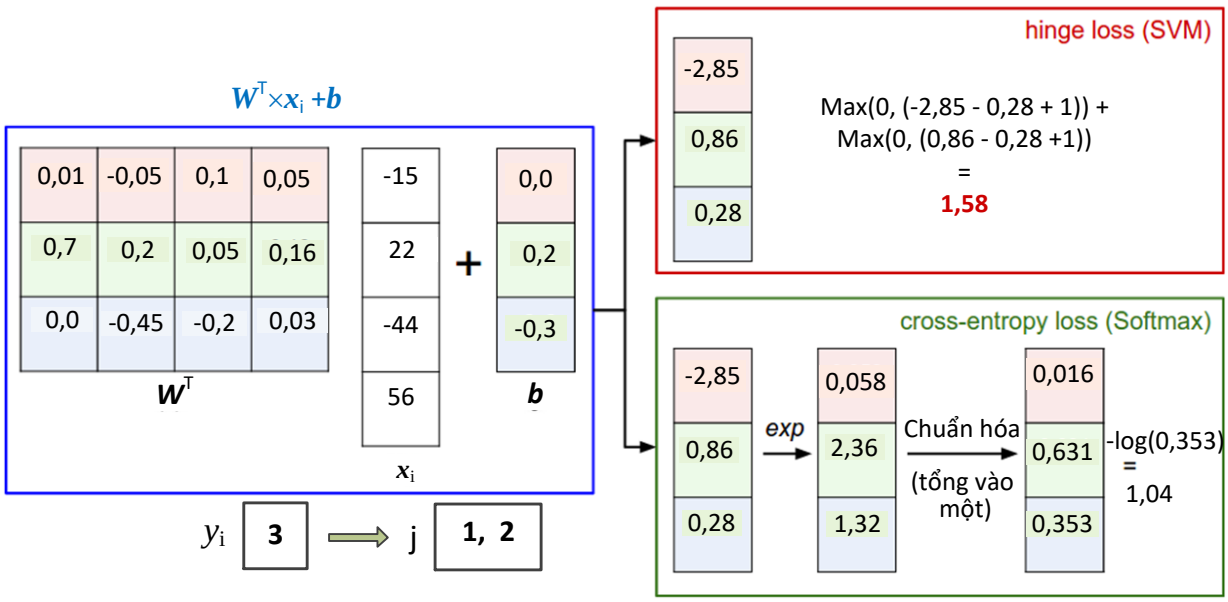
Chúng ta có thể tự do chọn giá trị của C. Điều này sẽ không làm thay đổi bất kỳ kết quả nào, nhưng chúng ta có thể sử dụng giá trị này để cải thiện tính ổn định số của phép tính. Một lựa chọn phổ biến cho C là đặt $\log C = -\max_j f_j$. Điều này đơn giản có nghĩa là chúng ta nên dịch chuyển các giá trị bên trong vectơ sao cho giá trị cao nhất bằng 0 theo mã:

```
f = np.array([123, 456, 789]) # example with 3 classes and each having large scores
p = np.exp(f) / np.sum(np.exp(f)) # Bad: Numeric problem, potential blowup

# instead: first shift the values of f so that the highest number is 0:
f -= np.max(f) # f becomes [-666, -333, 0]
p = np.exp(f) / np.sum(np.exp(f)) # safe to do, gives the correct answer
```

Quy ước đặt tên có thểdễ gây nhầm lẫn. Chính xác hơn, bộ phân loại SVM sử dụng hàm mất mát bản lề (hinge loss), hay đôi khi còn được gọi là hàm mất mát biên độ cực đại (max-margin loss). Bộ phân loại Softmax sử dụng hàm mất mát entropy chéo (cross - entropy loss). Tên của bộ phân loại Softmax bắt nguồn từ hàm softmax, được sử dụng để nén điểm số loạithô thành các giá trị dương được chuẩn hóa có tổng bằng một, nhờ đó có thể áp dụng hàm mất mát entropy chéo. Đặc biệt, cần lưu ý rằng về mặt kỹ thuật, việc nói về "lỗi mất mát softmax" là không hợp lý, vì softmax chỉ là hàm nén, nhưng đây là một cách viết tắt khá phổ biến.

4.6.3. So sánh SVM với Softmax

Hình 4.31 có thể giúp làm rõ sự khác biệt giữa bộ phân loại Softmax và SVM.
Hình 4.13 cho thấy ví dụ về sự khác biệt giữa bộ phân loại SVM và Softmax cho một điểm dữ liệu. Trong cả hai trường hợp, ta tính toán cùng một vectơ điểm số f (ví dụ bằng phép nhân ma trận trong phần này). Sự khác biệt nằm ở cách diễn giải các điểm trong f : SVM diễn giải chúng thành điểm số loại và hàm mất mát của nó khuyến khích loại đúng (loại 3, màu xanh lam) có điểm số cao hơn một biên độ (Margin) so với các điểm số loại khác. Thay vào đó, bộ phân loại Softmax diễn giải các điểm số thành xác suất logarit (không chuẩn hóa) cho mỗi loại và sau đó khuyến khích xác suất logarit (chuẩn hóa) của loại đúng là cao (tương đương với giá trị âm của nó là thấp). Mất mát cuối cùng trong ví dụ này là 1,58 đối với SVM và 1,04 (lưu ý đây là 1,04 khi sử dụng logarit tự nhiên, không phải cơ số 2 hoặc cơ số 10) đối với bộ phân loại Softmax, nhưng lưu ý rằng các số này không thể so sánh được; Chúng chỉ có ý nghĩa liên quan đến mất mát được tính toán trong cùng một bộ phân loại và với cùng một dữ liệu.



Hinge Loss: mất mát bản lề, Cross-Entropy Loss: mất mát entropy chéo

Hình 4.13 THí dụ về sự khác biệt giữa bộ phân loại Softmax và SVM

Sofmax tương đương với xác suất trong mất mát entropy chéo cho thí dụ trên hình 4.13 được tính như sau. Trước tiên ta tính các hàm mũ (exp: expotential):

- J=1: $e^{-2,85} = 0,058$
- J=2: $e^{0,86} = 2,36$
- J=3: $e^{0,28} = 1,32$

Tiếp theo ta chuẩn hóa các giá trị trên để được xác suất mất mát bằng cách chuẩn hóa các giá trị trên. Để chuẩn hóa trước tiên ta tính tổng của cả ba giá trị trên:

0,058+2,36+1,32=3,738

Chia các giá trị của hàm mũ nói trên nói trên cho tổng ta được xác suất sau chuẩn hóa:

- J=1: P=0,058/3,738=0,016
- J=2: P=2,36/3,738=0,31
- J=3: 1,32/3,738=0,353

Cuỗi cùng mất entropy chéo là

L_i=-log(0,353)= 1,04

trong đó logarit là logarit tự nhiên (theo cơ số e).

4.6.4. Bộ phân loại Softmax cung cấp "xác suất" cho mỗi loại

Không giống như SVM tính toán điểm số chưa hiệu chuẩn và khó diễn giải cho tất cả các loại, bộ phân loại Softmax cho phép chúng ta tính toán "xác suất" cho tất cả các nhãn. Ví dụ: với một hình ảnh, bộ phân loại SVM có thể cho ta các điểm số $[12,5, 0,6, -23,0]^T$ cho các loại"mèo", "chó" và "tàu thủy". Thay vào đó, bộ phân loại Softmax có thể tính toán

xác suất của ba nhãn là $[0,9, 0,09, 0,01]^T$, cho phép ta diễn giải độ tin cậy của nó trong mỗi loại. Tuy nhiên, lý do chúng ta đặt từ "xác suất" trong dấu ngoặc kép là vì mức độ đỉnh hoặc khuếch tán của các xác suất này phụ thuộc trực tiếp vào cường độ chính quy hóa λ mà ta chịu trách nhiệm làm đầu vào cho hệ thống. Ví dụ: giả sử rằng log-xác suất chưa chuẩn hóa cho ba loạinào đó là $[1, -2, 0]^T$. Khi đó, hàm Softmax sẽ tính toán:

$$\begin{bmatrix}1 & -2 & 0\end{bmatrix}^T \rightarrow \begin{bmatrix}e^1 & e^{-2} & e^0\end{bmatrix}^T = \begin{bmatrix}2,71 & 0,14 & 1\end{bmatrix}^T \rightarrow \frac{1}{2,71+0,14+1}\begin{bmatrix}2,71 & 0,14 & 1\end{bmatrix}^T$$
$$= \frac{1}{3.85}\begin{bmatrix}2,71 & 0,14 & 1\end{bmatrix}^T = \begin{bmatrix}0,7 & 0,04 & 0,26\end{bmatrix}^T$$

Các bước được thực hiện là hàm mũ và chuẩn hóa để tổng bằng một. Bây giờ, nếu cường độ chính quy hóa cao λ hơn, các trọng số sẽ bị phạt nhiều hơn và điều này sẽ dẫn đến các trọng số nhỏ hơn. Ví dụ, giả sử các trọng số nhỏ hơn một nửa ([0,5, -1, 0]). Lúc này, softmax sẽ tính:

$$\begin{bmatrix}0,5 & -1 & 0\end{bmatrix}^T \rightarrow \begin{bmatrix}e^{0,5} & e^{-1} & e^0\end{bmatrix}^T = \begin{bmatrix}1,65 & 0,37 & 1\end{bmatrix}^T \rightarrow \frac{1}{1,65+0,37+1}\begin{bmatrix}1,65 & 0,37 & 1\end{bmatrix}^T$$
$$= \frac{1}{3.02}\begin{bmatrix}1,65 & 0,37 & 1\end{bmatrix}^T = \begin{bmatrix}0,55 & 0,12 & 0,33\end{bmatrix}^T$$

trong đó các xác suất giờ đây phân tán hơn. Hơn nữa, trong giới hạn mà trọng số hướng về các số nhỏ do cường độ chính quy hóa rất λ mạnh , xác suất đầu ra sẽ gần như đồng đều. Do đó, xác suất được tính toán bởi bộ phân loại Softmax nên được coi là độ tin cậy, trong đó, tương tự như SVM, thứ tự của các điểm số có thể diễn giải được, nhưng về mặt kỹ thuật, các số tuyệt đối (hoặc hiệu số của chúng) thì không.

Trên thực tế, SVM và Softmax thường tương đương nhau**.** Sự khác biệt về hiệu suất giữa SVM và Softmax thường rất nhỏ và những người khác nhau sẽ có ý kiến khác nhau về bộ phân loại nào hoạt động tốt hơn. So với bộ phân loại Softmax, SVM là một mục tiêu cục bộ hơn , có thể được coi là một lỗi (bug) hoặc một tính năng. Hãy xem xét một ví dụ đạt được điểm số $[10, -2, 3]^T$ và trong đó loại đầu tiên là chính xác. Một SVM (ví dụ với biên độ mong muốn là Δ=1) sẽ thấy rằng loạichính xác đã có điểm số cao hơn biên độ so với các loạikhác và nó sẽ tính toán mất mát bằng không. SVM không quan tâm đến chi tiết của các điểm số riêng lẻ: nếu thay vào đó chúng là $[10, -100, -100]^T$ hoặc $[10, 9, 9]^T$ thì SVM sẽ không quan tâm vì biên độ 1 được thỏa mãn và do đó mất mát bằng không. Tuy nhiên, các kịch bản này không tương đương với bộ phân loại Softmax, bộ này sẽ tích lũy mất mát cao hơn nhiều đối với các điểm số $[10, 9, 9]^T$ so với $[10, -100, -100]^T$. Nói cách khác, bộ phân loại Softmax không bao giờ hoàn toàn hài lòng với các điểm số mà nó tạo ra: loại đúng luôn có thể có xác suất cao hơn và các loại không đúng luôn có xác suất thấp hơn và mất mát sẽ luôn tốt hơn. Tuy nhiên, SVM hài lòng khi các biên độ được thỏa mãn và nó không quản lý vi mô các điểm số chính xác vượt quá ràng buộc này. Điều này có thể được coi một cách trực quan là một tính năng: Ví dụ, một bộ phân loại ô tô có khả

năng dành phần lớn "nỗ lực" của mình vào bài toán khó là tách ô tô khỏi xe tải không nên bị ảnh hưởng bởi các ví dụ về con ếch, mà nó đã gán điểm số rất thấp và có khả năng tập trung xung quanh một phía hoàn toàn khác của đám mây dữ liệu.

Bây giờ chúng ta đã thấy một cách để lấy một tập dữ liệu hình ảnh và ánh xạ từng hình ảnh với điểm số loại dựa trên một tập hợp các tham số, và chúng ta đã thấy hai ví dụ về hàm mất mát mà chúng ta có thể sử dụng để đo lường chất lượng dự đoán. Nhưng làm thế nào để xác định hiệu quả các tham số mang lại mức mất mát tốt nhất (thấp nhất)? Quá trình này chính là *tối ưu hóa* , và nó là chủ đề của phần tiếp theo.

4.7. TỐI ƯU HÓA

4.7.1. Tổng quan

Trong phần trước, chúng ta đã nghiên cứu hai thành phần chính trong ngữ cảnh của nhiệm vụ phân loại hình ảnh:

1. Hàm điểm số (có tham số) ánh xạ các pixel ảnh hình ảnh thô thành điểm số loại (ví dụ như hàm tuyến tính)
2. Hàm mất mát đo lường chất lượng của một tập hợp tham số cụ thể dựa trên mức độ phù hợp của điểm số cảm ứng với nhãn dữ liệu thực tế trong dữ liệu huấn luyện. Chúng ta đã thấy rằng có nhiều cách và phiên bản khác nhau của hàm này (ví dụ: Softmax/SVM).

Cụ thể, nhắc lại rằng hàm tuyến tính có dạng $f(\boldsymbol{x}_i, \boldsymbol{W}) = \boldsymbol{W}^T \boldsymbol{x}_i$ và SVM mà ta phát triển có dạng sau:

$$L = \frac{1}{N} \sum_{j \neq y_i} \left[\max(0, (f(x_i, \boldsymbol{W})_j - f(\boldsymbol{x}_i, \boldsymbol{W})_i + 1) \right] + \alpha R(\boldsymbol{W})$$

Ta thấy rằng một thiết lập các thông số $\boldsymbol{W}$ đã tạo ra những dự đoán cho các ví dụ $\boldsymbol{x}_i$ phù hợp với nhãn sự thật cơ bản y_i của chúng cũng sẽ có lất mát L rất thấp. Bây giờ ta sẽ đưa ra thành phần quan trọng thứ ba và cũng là thành phần cuối cùng: tối ưu hóa . Tối ưu hóa là quá trình tìm tập hợp các tham số $\boldsymbol{W}$ để giảm thiểu hàm mất mát.

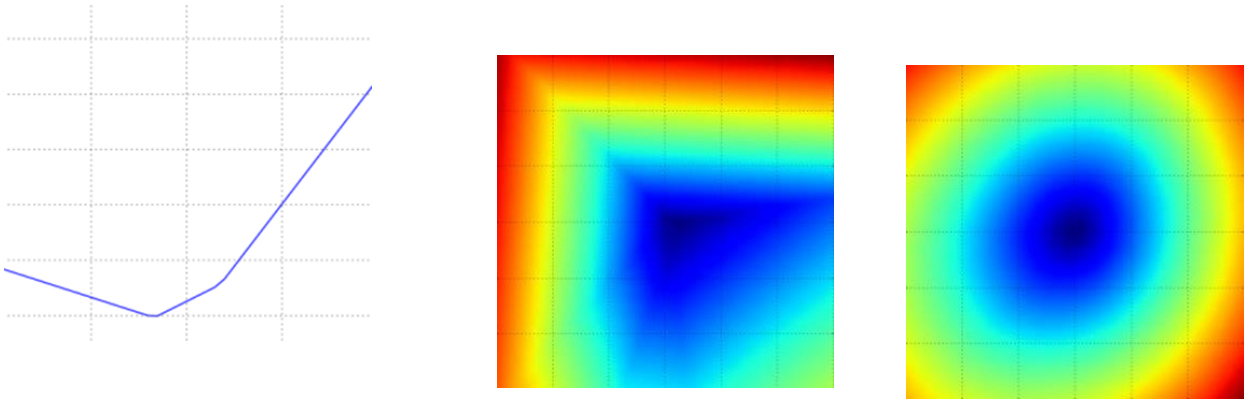
Khi đã hiểu cách ba thành phần cốt lõi này tương tác với nhau, chúng ta sẽ xem xét lại thành phần đầu tiên (ánh xạ hàm tham số) và mở rộng nó sang các hàm phức tạp hơn nhiều so với ánh xạ tuyến tính: Đầu tiên là toàn bộ mạng Neuron, sau đó là mạng Neuron tích chập. Các hàm mất mát và quy trình tối ưu hóa sẽ gần như không thay đổi.

4.7.2. Trực quan hóa hàm mất mát

Các hàm mất mát mà chúng ta sẽ xem xét ở đây thường được định nghĩa trên các không gian có số chiều rất lớn (ví dụ: trong CIFAR-10, một ma trận trọng số $\boldsymbol{W}$ phân loại tuyến tính có kích thước $[3073 \times 10]$ tương đương với W^T có kích thước $[10 \times 3073]$ với tổng cộng 30.730 tham số), khiến chúng khó hình dung. Tuy nhiên, chúng ta vẫn có thể có

được một số trực giác về một ma trận mất mát bằng cách cắt ngang không gian có số chiều rất lớn theo các tia (1 chiều) hoặc theo các mặt phẳng (2 chiều). Ví dụ, chúng ta có thể tạo ra một ma trận trọng số ngẫu nhiên $\boldsymbol{W}$ (tương ứng với một điểm duy nhất trong không gian), sau đó di chuyển dọc theo một tia và ghi lại giá trị hàm mất mát trên đường đi. Nghĩa là, chúng ta có thể tạo ra một hướng ngẫu nhiên $\boldsymbol{W}_1$ và tính toán mất mát theo hướng này bằng cách đánh giá L ($\boldsymbol{W}$+ a$\boldsymbol{W}_1$) cho các giá trị khác nhau a. Quá trình này tạo ra một biểu đồ đơn giản với giá trị của a như là trục x và giá trị của hàm mất mát như là trục y. Chúng ta cũng có thể thực hiện quy trình tương tự với hai chiều bằng cách đánh giá hàm mất mát L ($\boldsymbol{W}$+ a$\boldsymbol{W}_1$+ b$\boldsymbol{W}_2$)khi chúng ta thay đổi a, b. Khi này trong một hình vẽ, a, b tương ứng với trục x và trục y, và giá trị của hàm mất mát có thể được trực quan hóa bằng màu sắc (hình 4.14).

Hình 4.14 cho thấy toàn cảnh hàm mất mát cho SVM nhiều loại (không có chính quy hóa) cho một ví dụ duy nhất (trái, giữa) và cho một trăm ví dụ (phải) trong CIFAR-10. Trái: mất mát một chiều chỉ bằng cách thay đổi a . Giữa, Phải: lát cắt mất mát hai chiều, xanh lam = mất mát thấp, đỏ = mất mát cao. Lưu ý cấu trúc tuyến tính từng phần của hàm mất mát. Các mất mát cho nhiều ví dụ được kết hợp với giá trị trung bình, do đó hình dạng bát (Bowl) bên phải là trung bình của nhiều bát tuyến tính theo từng phần (chẳng hạn như bát ở giữa).



Hình 4.14 cho thấy oàn cảnh hàm mất mát cho SVM đa lớp

Chúng ta có thể giải thích cấu trúc tuyến tính theo từng phần của hàm mất mát bằng cách xem xét toán học. Ví dụ, đối với một thí dụ đơn lẻ chúng ta có:

$$L_i = \sum_j \left[\max\ 0,\ \boldsymbol{w}_j^T \boldsymbol{x}_i - \boldsymbol{w}_{y_i}^T \boldsymbol{x}_i + 1 \right]$$

Rõ ràng từ phương trình này theo đó mất mát dữ liệu cho mỗi ví dụ là tổng của (hàm được định ngưỡng bằng max (0 , −)) các hàm tuyến tính của $\boldsymbol{W}$. Hơn nữa, mỗi cột của $\boldsymbol{W}$ (tức là $\boldsymbol{w}_j$) đôi khi có dấu dương ở phía trước (khi nó tương ứng với một loại sai đối với một ví dụ), và đôi khi có dấu âm (khi nó tương ứng với loại đúng đối với ví dụ đó).

Để làm rõ hơn, hãy xem xét một tập dữ liệu đơn giản chứa ba điểm 1 chiều và ba loại. Tổng mất mát SVM (không có chính quy hóa) sẽ là:

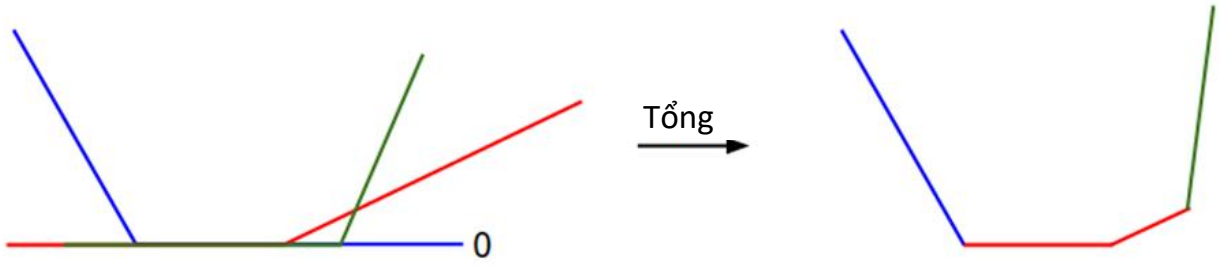
$$L_1 = \max\ 0,\ \boldsymbol{w}_2^T\boldsymbol{x}_1 - \boldsymbol{w}_1^T\boldsymbol{x}_1 + 1 \quad + \max\ 0,\ \boldsymbol{w}_3^T\boldsymbol{x}_1 - \boldsymbol{w}_1^T\boldsymbol{x}_1 + 1$$

$$L_2 = \max\ 0,\ \boldsymbol{w}_1^T\boldsymbol{x}_2 - \boldsymbol{w}_2^T\boldsymbol{x}_2 + 1 \quad + \max\ 0,\ \boldsymbol{w}_3^T\boldsymbol{x}_2 - \boldsymbol{w}_2^T\boldsymbol{x}_2 + 1$$

$$L_3 = \max\ 0,\ \boldsymbol{w}_1^T\boldsymbol{x}_3 - \boldsymbol{w}_3^T\boldsymbol{x}_3 + 1 \quad + \max\ 0,\ \boldsymbol{w}_2^T\boldsymbol{x}_3 - \boldsymbol{w}_3^T\boldsymbol{x}_3 + 1$$

$$L = \ L_1 + L_2 + L_3\ \ / 3$$

Vì những ví dụ này là 1 chiều, dữ liệu $\boldsymbol{x}_i$và trọng số $\boldsymbol{w}_j$ là những con số. Xem xét, ví dụ, $\boldsymbol{x}_1$, một số số hạng trên là các hàm tuyến tính của $\boldsymbol{w}_j^T$ và mỗi hàm bị xén ở mức 0. Chúng ta có thể hình dung điều này như trong hình 4.15.



Hình 4.15. Minh họa một chiều về mất mát dữ liệu. Trục x là một trọng số duy nhất và trục y là giá trị mất mát. Mất mát dữ liệu là tổng của nhiều số hạng, mỗi số hạng độc lập với một trọng số cụ thể, hoặc là một hàm tuyến tính của trọng số đó, có ngưỡng bằng 0. Mất mát dữ liệu SVM đầy đủ là phiên bản 30.730 chiều của hình này

Ngoài ra, ta có thể đoán từ hình dạng hình bát của nó rằng hàm chi phí SVM là một ví dụ về Khi chúng ta mở rộng các hàm điểm số f của mình cho Mạng neuron, các hàm mục tiêu của chúng ta sẽ trở nên không lồi và hình ảnh trực quan ở trên sẽ không có hình bát mà là địa hình gồ ghề, phức tạp.

Các hàm mất mát không khả vi . Về mặt kỹ thuật, ta cũng có thể thấy rằng *các điểm nút* trong hàm mất mát (do phép toán max) về mặt kỹ thuật khiến hàm mất mát không khả vi vì tại các điểm nút này, gradient không được xác định. Tuy nhiên, gradient dưới (subgradient) vẫn tồn tại và thường được sử dụng thay thế. Trong phần dưới đây này, các thuật ngữ *subgradient* và *gradient* sẽ được sử dụng thay thế cho nhau.

4.7.3. Tối ưu hóa

Nhắc lại, hàm mất mát cho phép chúng ta định lượng chất lượng của bất kỳ tập trọng số $\boldsymbol{W}$ cụ thể nào. Mục tiêu của tối ưu hóa là tìm $\boldsymbol{W}$ để giảm thiểu hàm mất mát. Bây giờ, chúng ta sẽ thúc đẩy và dần dần phát triển một phương pháp tối ưu hóa hàm mất mát.

Để hiểu được nguyên lý đào tạo ta nhắc lại quy trình đào tọ cho perceptron đã xét trong chương 3. Đầu tiên các trọng số được khởi đầu một cách ngẫu nhiên và sau đó bằng cách

sử dụng một mẫu $\boldsymbol{x}_i, y_i$ của tập đào tạo, dự báo của perceptron được tính toán. Nếu dự báo đúng, không cần hành động tiếp theo và điểm số liệu tiếp theo được xử lý. Nếu dự báo sai, các trọng số được cập nhật theo quy tắc sau: các trọng số được tăng trong trường hợp dự báo nhỏ hơn nhãn sự thật cơ bản (ground truth label) y_i và giảm nếu dự báo cao hơn nhãn sự thật cơ bản. Quá trình này được lặp cho đến khi không còn lỗi. Mã giả (pseudocode) của đào tạo hay giải thuật hội tụ được trình bày trong bảng 4.1 (lưu ý rằng phiên bản này cho trường hợp giả thết là số liệu là phải có thể phân tách tuyến tính).

Bảng 4.1. Mã giả (pseudocode) của đào tạo

Algorithm 1 Train Perception

Procedure TRAIN $(\{\boldsymbol{x}_i, y_i\})$

Initialization: initialize randomly the weights **w** and bias b

while $\exists i \in \{1,\ldots,N\}, f(\boldsymbol{x}_i, \boldsymbol{w}, b) \neq y^{(i)}$ **do**

Pick i randomly

error= $y_i - f(\boldsymbol{x}_i, \boldsymbol{w}, b)$

if error $\neq 0$ **then**

$\boldsymbol{w} \leftarrow \boldsymbol{w} + \text{error}.\boldsymbol{x}$

$b \leftarrow b + \text{error}$

Quy trình này có thể được mở rộng cho nhiều perceptron (Muliti Perceps tron) và nhiều loại, trong đó vectơ $\boldsymbol{w}$ được thai thế bằng ma trận $\boldsymbol{W}$ và độ lệch b được thay thể bằng vectơ **b**. Đối với CIFAR-10, ma trận trọng số **W** có kích thước $[3072 \times 10]$ tương đương với $\boldsymbol{W}^T$ có kích thước $[10 \times 3072]$ và vectơ $\boldsymbol{b}$ có kích thước $[10 \times 1]$. Tuy nhiên nếu tích hợp ma trận trọng số $\boldsymbol{W}$ với vectơ độ lệch **b** ta sẽ được ma trận trọng số kết hợp $\boldsymbol{W}$ có kích thước $[3073 \times 10]$ hay $\boldsymbol{W}^T$ kích thước $[10 \times 2073]$.

4.7.4. Các chiến lược đào tạo

4.7.4.1. Tìm kiếm ngẫu nhiên

Vì việc kiểm tra độ tốt của một tập hợp tham số $\boldsymbol{W}$ cho trước rất đơn giản , nên ý tưởng đầu tiên (rất tệ) có thể nảy ra là thử nhiều trọng số ngẫu nhiên khác nhau và theo dõi xem trọng số nào hiệu quả nhất. Quy trình này có thể như sau:

```
# assume X_train is the data where each column is an example (e.g. 3072 x 50,000)
# assume Y_train are the labels (e.g. 1D array of 50,000)
# assume the function L evaluates the loss function
```

```
bestloss = float("inf") # Python assigns the highest possible float value
for num in range(1000):
  W = np.random.randn(10, 3072) * 0.0001 # generate random parameters
  loss = L(X_train, Y_train, W) # get the loss over the entire training set
  if loss < bestloss: # keep track of the best solution
    bestloss = loss
    bestW = W
  print 'in attempt %d the loss was %f, best %f' % (num, loss, bestloss)

# prints:
# in attempt 0 the loss was 9.401632, best 9.401632
# in attempt 1 the loss was 8.959668, best 8.959668
# in attempt 2 the loss was 9.044034, best 8.959668
# in attempt 3 the loss was 9.278948, best 8.959668
# in attempt 4 the loss was 8.857370, best 8.857370
# in attempt 5 the loss was 8.943151, best 8.857370
# in attempt 6 the loss was 8.605604, best 8.605604
# ... (trunctated: continues for 1000 lines)
```

Trong đoạn mã trên, chúng ta thấy rằng chúng ta đã thử nghiệm một số vectơ trọng số ngẫu nhiên ***W*** , và một số trong số chúng hoạt động tốt hơn những vectơ khác. Chúng ta có thể lấy các trọng số ***W*** tốt nhất tìm được từ tìm kiếm này và thử nghiệm trên tập kiểm tra:

```
# Assume X_test is [3072 x 10000], Y_test [10000 x 1]
scores = Wbest.dot(Xte_cols) # 10 x 10000, the class scores for all test examples
# find the index with max score in each column (the predicted class)
Yte_predict = np.argmax(scores, axis = 0)
# and calculate accuracy (fraction of predictions that are correct)
np.mean(Yte_predict == Yte)
# returns 0.1555
```

Với ***W*** tốt nhất, độ chính xác sẽ đạt khoảng 15,5**%** . Xét đến việc đoán các loạihoàn toàn ngẫu nhiên chỉ đạt được 10%, thì đây không phải là một kết quả tệ cho một giải pháp tìm kiếm ngẫu nhiên tồi tệ.

4.7.4.2. Tinh chỉnh lặp (Intative Refinement)

Tất nhiên, hóa ra chúng ta có thể làm tốt hơn nhiều. Ý tưởng cốt lõi là việc tìm ra tập trọng số *W* tốt nhất là một bài toán rất khó, thậm chí là bất khả thi (đặc biệt là khi ***W*** chứa trọng số của toàn bộ mạng neuron phức tạp), nhưng việc tinh chỉnh một tập trọng số **W** cụ thể để tốt hơn một chút lại dễ dàng hơn đáng kể. Nói cách khác, cách tiếp cận

của chúng ta sẽ là bắt đầu với một W ngẫu nhiên và sau đó tinh chỉnh nó theo từng bước, mỗi lần lại tốt hơn một chút.
Như vậy, ta sẽ bắt đầu với ma trận trọng số **W** ngẫu nhiên và tinh chỉnh chúng theo thời gian để giảm thiểu mất mát. Ta có thể tưởng tượng cách tìm kiếm này như là một phép so sánh với người đi bộ đường dài bịt mắt. Một phép so sánh có thể hữu ích cho ta khi tiến về phía trước và tưởng tượng mình đang đi bộ trên một địa hình đồi núi với đôi mắt bịt kín và cố gắng chạm đến chân đồi. Trong ví dụ về CIFAR-10, các ngọn đồi có kích thước 3073 chiều, vì kích thước của $\boldsymbol{W}^T$ là 10 x 3073. Tại mỗi điểm trên đồi, chúng ta đạt được một độ lệch nhất định (chiều cao của địa hình).

4.7.4.3. Tìm kiếm cục bộ ngẫu nhiên (Random Local Search)

Chiến lược đầu tiên ta có thể nghĩ đến là thử duỗi một chân theo một hướng ngẫu nhiên và chỉ bước nếu hướng đó dẫn xuống dốc. Cụ thể, chúng ta sẽ bắt đầu với một bước ngẫu nhiên, tạo ra nhiễu loạn ngẫu nhiên δ$\boldsymbol{W}$ với nó và nếu mất mát ở nhiễu loạn $\boldsymbol{W}$+ δ**W**thấp hơn, chúng ta sẽ thực hiện cập nhật. Mã cho quy trình này như sau:

```
W = np.random.randn(10, 3073) * 0.001 # generate random starting W
bestloss = float("inf")
for i in range(1000):
  step_size = 0.0001
  Wtry = W + np.random.randn(10, 3073) * step_size
  loss = L(Xtr_cols, Ytr, Wtry)
  if loss < bestloss:
    W = Wtry
    bestloss = loss
  print 'iter %d loss is %f' % (i, bestloss)
```

Lưu ý trong đoạn mã trên để tiện lợi ta sử dụng ký hiệu W thay cho W^T.
Sử dụng cùng số lượng đánh giá hàm mất mát như trước (1000), phương pháp này đạt được độ chính xác phân loại tập kiểm tra là **21,4%** . Phương pháp này tốt hơn, nhưng vẫn lãng phí và tốn kém về mặt tính toán.

4.7.4.4. Theo dõi độ dốc (Following Gradient)

Trong phần trước, chúng ta đã cố gắng tìm một hướng trong không gian trọng số để cải thiện vectơ trọng số (và giảm thiểu mất mát). Hóa ra không cần phải tìm kiếm ngẫu nhiên một hướng tốt: chúng ta có thể tính toán hướng *tốt nhất* để thay đổi vectơ trọng số, hướng này được đảm bảo về mặt toán học là hướng xuống dốc nhất (Steepest Descent) (ít nhất là trong giới hạn khi kích thước bước tiến về 0). Hướng này sẽ liên quan đến độ dốc (Gradient) của hàm mất mát. Trong phép so sánh đi bộ đường dài của chúng ta, cách tiếp cận này gần giống với việc cảm nhận độ dốc của ngọn đồi dưới chân và bước xuống theo hướng có cảm giác dốc nhất.

Trong các hàm một chiều, độ dốc là tốc độ thay đổi tức thời của hàm tại bất kỳ điểm nào ta quan tâm. Độ dốc là một khái quát của độ dốc cho các hàm không lấy một số duy nhất mà là một vectơ số. Ngoài ra, độ dốc chỉ là một vectơ độ dốc (thường được gọi là đạo hàm) cho mỗi chiều trong không gian đầu vào. Biểu thức toán học cho đạo hàm của một hàm một chiều theo đầu vào của nó là:

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f\ x+h\ - f(x)}{h} \qquad (4.5)$$

Khi các hàm quan tâm lấy một vectơ số thay vì một số duy nhất, chúng ta gọi các đạo hàm là đạo hàm riêng (ký hiệu là $\frac{\partial f(x)}{\partial x}$**)** và độ dốc chỉ đơn giản là vectơ của các đạo hàm riêng trong mỗi chiều.

Có hai cách để tính toán độ dốc: Một cách chậm, gần đúng nhưng dễ thực hiện (độ dốc số) và một cách nhanh, chính xác nhưng dễ xảy ra lỗi hơn, đòi hỏi phải có phép tính (độ dốc giải tích). Bây giờ chúng ta sẽ xét cả hai.

4.7.5. Tính toán độ dốc theo số với các chênh lệch hữu hạn

Công thức (4.5) cho phép chúng ta tính toán độ dốc theo phương pháp số. Dưới đây là một hàm tổng quát nhận một hàm f, một vectơ ***x*** để đánh giá độ dốc (Gradient) trên, và cho ra kết quả độ dốc (Gradient) của f tại ***x***:

```
def eval_numerical_gradient(f, x):
  """
  a naive implementation of numerical gradient of f at x
  - f should be a function that takes a single argument
  - x is the point (numpy array) to evaluate the gradient at
  """

  fx = f(x) # evaluate function value at original point
  grad = np.zeros(x.shape)
  h = 0.00001

  # iterate over all indexes in x
  it = np.nditer(x, flags=['multi_index'], op_flags=['readwrite'])
  while not it.finished:

    # evaluate function at x+h
    ix = it.multi_index
    old_value = x[ix]
```

```
    x[ix] = old_value + h # increment by h
    fxh = f(x) # evalute f(x + h)
    x[ix] = old_value # restore to previous value (very important!)

    # compute the partial derivative
    grad[ix] = (fxh - fx) / h # the slope
    it.iternext() # step to next dimension

  return grad
```

Theo công thức gradient đã nêu ở trên, đoạn mã trên sẽ lặp lại từng chiều một, thực hiện một thay đổi nhỏ h dọc theo chiều đó và tính toán đạo hàm riêng của hàm mất mát dọc theo chiều đó bằng cách xem hàm đã thay đổi bao nhiêu. Biến này grad cuối cùng sẽ chứa toàn bộ gradient.

Lưu ý rằng trong công thức toán học, gradient được định nghĩa trong giới hạn khi h tiến về 0, nhưng trên thực tế, thường chỉ cần sử dụng một giá trị rất nhỏ (chẳng hạn như $1e^{-5}$ như trong ví dụ). Lý tưởng nhất là bạn nên sử dụng kích thước bước nhỏ nhất mà không dẫn đến các vấn đề về số. Ngoài ra, trên thực tế, việc tính gradient số bằng công thức hiệu số trng tâm thường hiệu quả hơn: [f(x + h) − f(x − h)] / 2 h. Chúng ta có thể sử dụng hàm trên để tính gradient tại bất kỳ điểm nào và cho bất kỳ hàm nào. Hãy tính gradient cho hàm mất mát CIFAR-10 tại một điểm ngẫu nhiên nào đó trong không gian trọng số:

```
# to use the generic code above we want a function that takes a single argument
# (the weights in our case) so we close over X_train and Y_train
def CIFAR10_loss_fun(W):
  return L(X_train, Y_train, W)

W = np.random.rand(10, 3073) * 0.001 # random weight vector
df = eval_numerical_gradient(CIFAR10_loss_fun, W) # get the gradient
```

Độ dốc cho chúng ta biết độ dốc của hàm mất mát theo mọi chiều, chúng ta có thể sử dụng để thực hiện cập nhật:

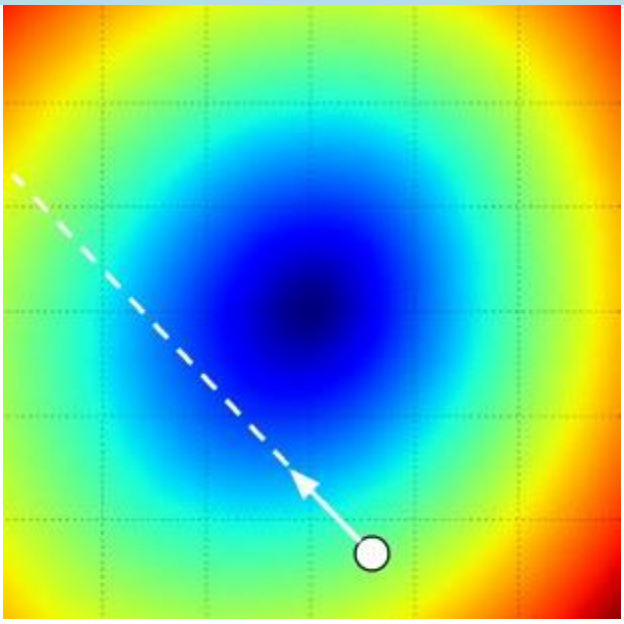
```
loss_original = CIFAR10_loss_fun(W) # the original loss
print 'original loss: %f' % (loss_original, )

# lets see the effect of multiple step sizes
for step_size_log in [-10, -9, -8, -7, -6, -5,-4,-3,-2,-1]:
  step_size = 10 ** step_size_log
  W_new = W - step_size * df # new position in the weight space
  loss_new = CIFAR10_loss_fun(W_new)
  print 'for step size %f new loss: %f' % (step_size, loss_new)
```

```
# prints:
# original loss: 2.200718
# for step size 1.000000e-10 new loss: 2.200652
# for step size 1.000000e-09 new loss: 2.200057
# for step size 1.000000e-08 new loss: 2.194116
# for step size 1.000000e-07 new loss: 2.135493
# for step size 1.000000e-06 new loss: 1.647802
# for step size 1.000000e-05 new loss: 2.844355
# for step size 1.000000e-04 new loss: 25.558142
# for step size 1.000000e-03 new loss: 254.086573
# for step size 1.000000e-02 new loss: 2539.370888
# for step size 1.000000e-01 new loss: 25392.214036
```

Trong đoạn mã trên, lưu ý rằng để tính toán, `W_new` chúng ta đang thực hiện cập nhật theo hướng âm của gradient `df` vì chúng ta muốn hàm mất mát giảm chứ không phải tăng.

Độ dốc cho chúng ta biết hướng mà hàm có tốc độ tăng dốc nhất, nhưng không cho chúng ta biết chúng ta nên bước bao xa theo hướng này. Như chúng ta sẽ thấy sau này, việc chọn kích thước bước (còn gọi là *tốc độ học*) sẽ trở thành một trong những thiết lập siêu tham số quan trọng nhất (và gây đau đầu nhất) trong quá trình huấn luyện mạng neuron. Tương tự như xuống đồi bịt mắt của chúng ta, chúng ta cảm thấy ngọn đồi dưới chân mình dốc theo một hướng nào đó, nhưng độ dài bước mà chúng ta nên thực hiện là không chắc chắn. Nếu chúng ta xáo trộn đôi chân của mình một cách cẩn thận, chúng ta có thể mong đợi đạt được tiến bộ nhất quán nhưng rất nhỏ (điều này tương ứng với việc có kích thước bước nhỏ). Ngược lại, chúng ta có thể chọn thực hiện một bước lớn, tự tin để cố gắng xuống nhanh hơn, nhưng điều này có thể không mang lại hiệu quả. Như có thể thấy trong ví dụ mã ở trên, tại một số thời điểm, việc thực hiện một bước lớn hơn sẽ mang lại mất mát cao hơn khi chúng ta "bước quá".

Hình 4.16 cho phép ta hình dung tác động của kích thước bước. Chúng ta bắt đầu tại một điểm $\boldsymbol{W}$ cụ thể và đánh giá gradient (hay đúng hơn là giá trị âm của nó - mũi tên trắng) cho chúng ta biết hướng giảm mạnh nhất của hàm mất mát. Các bước nhỏ có thể dẫn đến tiến triển nhất quán nhưng chậm. Các bước lớn có thể dẫn đến tiến triển tốt hơn nhưng rủi ro hơn. Lưu ý rằng cuối cùng, với kích thước bước lớn, chúng ta sẽ vượt quá và làm cho mất mát trở nên tồi tệ hơn. Kích thước bước (hay như chúng ta sẽ gọi sau này - tốc độ học) sẽ trở thành một trong những siêu tham số quan trọng nhất mà chúng ta phải tinh chỉnh cẩn thận.



Hình 4.14. Tác động của kích thước bước

Ta có thể nhận thấy rằng việc đánh giá gradient số có độ phức tạp tuyến tính theo số lượng tham số. Trong ví dụ của chúng ta, chúng ta có tổng cộng 30.730 tham số và do đó phải thực hiện 30.731 lần đánh giá hàm mất mát để đánh giá gradient và chỉ thực hiện một lần cập nhật tham số duy nhất. Vấn đề này càng trở nên tồi tệ hơn, vì các mạng Neuron hiện đại có thể dễ dàng có hàng chục triệu tham số. Rõ ràng, chiến lược này không thể mở rộng quy mô và chúng ta cần một giải pháp tốt hơn.

4.7.6. Tính toán độ dốc (gradient) theo cách giải tích

Độ dốc số rất dễ tính toán bằng cách sử dụng phép xấp xỉ sai phân hữu hạn, (Finite Difference Approximation) nhưng nhược điểm là nó chỉ là gần đúng (vì ta phải chọn một giá trị h nhỏ, trong khi độ dốc thực được định nghĩa là giới hạn khi h tiến tới 0), và việc tính toán rất tốn kém về mặt tính toán. Cách thứ hai để tính độ dốc là tính toán theo giải tích, cho phép ta suy ra công thức trực tiếp cho độ dốc (không cần xấp xỉ) và cũng rất nhanh. Tuy nhiên, không giống như độ dốc số, nó có thể dễ xảy ra lỗi hơn khi triển khai, đó là lý do tại sao trong thực tế, việc tính toán độ dốc giải tích và so sánh với độ dốc số để kiểm tra tính chính xác của việc triển khai là rất phổ biến. Đây được gọi là kiểm tra độ dốc.

Chúng ta hãy sử dụng ví dụ về hàm mất mát SVM cho một điểm dữ liệu duy nhất:

$$L_i = \sum_{j \neq y_i} \left[\max\ 0,\ \boldsymbol{w}_j^T \boldsymbol{x}_i - \boldsymbol{w}_{y_i}^T \boldsymbol{x}_i \right]$$

Ta có thể phân biệt hàm theo trọng số. Ví dụ, lấy gradient theo $\boldsymbol{w}_{y_i}$ ta thu được:

$$\nabla_{w_{y_i}} L_i = -\left(\sum_{j \neq y_i} \mathbf{1} \ \ w_j^T x_i - w_{y_i}^T x_i + \Delta > 0 \right) x_i$$

Trong đó ***1*** là hàm chỉ báo, có giá trị bằng 1 nếu điều kiện bên trong là đúng hoặc bằng 0 nếu không. Mặc dù biểu thức này trông có vẻ đáng sợ khi viết ra, nhưng khi triển khai nó trong mã, ta chỉ cần đếm số loại không đạt được biên độ mong muốn (Desired Margin) (và do đó góp phần vào hàm mất mát) và sau đó lấy vectơ dữ liệu $\boldsymbol{x}_i$ chia cho số này là được độ dốc. Lưu ý rằng đây chỉ là độ dốc đối với cột $\boldsymbol{W}$ (hay hàng $\boldsymbol{W}^T$) tương ứng với loại chính xác. Đối với các cột khác, nơi $j \neq y_i$ độ dốc là:

$$\nabla_{w_j} L_i = \ \mathbf{1} \ \ w_j^T x_i - w_{y_i}^T x_i + \Delta > 0 \quad x_i$$

Khi đã có được biểu thức cho gradient, ta có thể dễ dàng triển khai các biểu thức và sử dụng chúng để thực hiện cập nhật gradient.

4.7.7. Độ dốc giảm dần

Bây giờ chúng ta đã có thể tính toán gradient của hàm mất mát, quy trình đánh giá gradient nhiều lần và sau đó thực hiện cập nhật tham số được gọi là *Gradient Descent theo phương trình* (3.15) như sau.

$$\mathbf{W}^{t+1} \leftarrow \mathbf{W}^t - \eta \frac{\partial L(\mathbf{W}^t)}{\partial \mathbf{W}^t}$$

Phiên bản gốc của nó trông như sau:

```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

Vòng lặp đơn giản này là cốt lõi của tất cả các thư viện mạng Neuron. Có nhiều cách khác để thực hiện tối ưu hóa (ví dụ: LBFGS), nhưng Gradient Descent hiện là cách phổ biến và đã được chứng minh nhất để tối ưu hóa các hàm mất mát của Mạng Neuron. Dưới đây, chúng ta sẽ bổ sung thêm một số chi tiết thú vị về vòng lặp này (ví dụ: chi tiết chính xác của phương trình cập nhật), nhưng ý tưởng cốt lõi là theo dõi gradient cho đến khi chúng ta hài lòng với kết quả sẽ vẫn như cũ.

4.7.8. Giảm dần gradient theo lô nhỏ (MGD: Mini-Batch Gradient Descent)

Trong các ứng dụng quy mô lớn (chẳng hạn như thử thách ILSVRC), dữ liệu huấn luyện có thể có hàng triệu ví dụ. Do đó, việc tính toán toàn bộ hàm mất mát trên toàn bộ tập huấn luyện để chỉ thực hiện một lần cập nhật tham số duy nhất có vẻ lãng phí. Một cách tiếp cận rất phổ biến để giải quyết thử thách này là tính toán gradient trên nhiều lô dữ liệu

huấn luyện. Ví dụ, trong các mạng ttích chập CNN hiện đại mà ta sẽ xét trong chương sau, một lô điển hình chứa 256 ví dụ từ toàn bộ tập huấn luyện gồm 1,2 triệu. Lô này sau đó được sử dụng để thực hiện cập nhật tham số:

```
# Vanilla Minibatch Gradient Descent

while True:
  data_batch = sample_training_data(data, 256) # sample 256 examples
  weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
  weights += - step_size * weights_grad # perform parameter update
```

Lý do phương pháp này hiệu quả là vì các ví dụ trong dữ liệu huấn luyện có tương quan. Để thấy rõ điều này, hãy xem xét trường hợp cực đoan khi tất cả 1,2 triệu hình ảnh trong ILSVRC thực chất chỉ bao gồm các bản sao chính xác của 1000 hình ảnh duy nhất (một bản sao cho mỗi loại, hay nói cách khác là 1200 bản sao giống hệt nhau của mỗi hình ảnh). Khi đó, rõ ràng là các gradient mà chúng ta tính toán cho tất cả 1200 bản sao giống hệt nhau sẽ giống nhau và khi chúng ta tính trung bình mức mất dữ liệu trên tất cả 1,2 triệu hình ảnh, chúng ta sẽ nhận được mức mất dữ liệu chính xác giống như khi chúng ta chỉ đánh giá trên một tập con nhỏ gồm 1000 hình ảnh. Trên thực tế, tất nhiên, tập dữ liệu sẽ không chứa các hình ảnh trùng lặp, gradient từ một lô nhỏ là một phép xấp xỉ tốt cho gradient của toàn bộ mục tiêu. Do đó, có thể đạt được sự hội tụ nhanh hơn nhiều trong thực tế bằng cách đánh giá các gradient của lô nhỏ để thực hiện cập nhật tham số thường xuyên hơn.

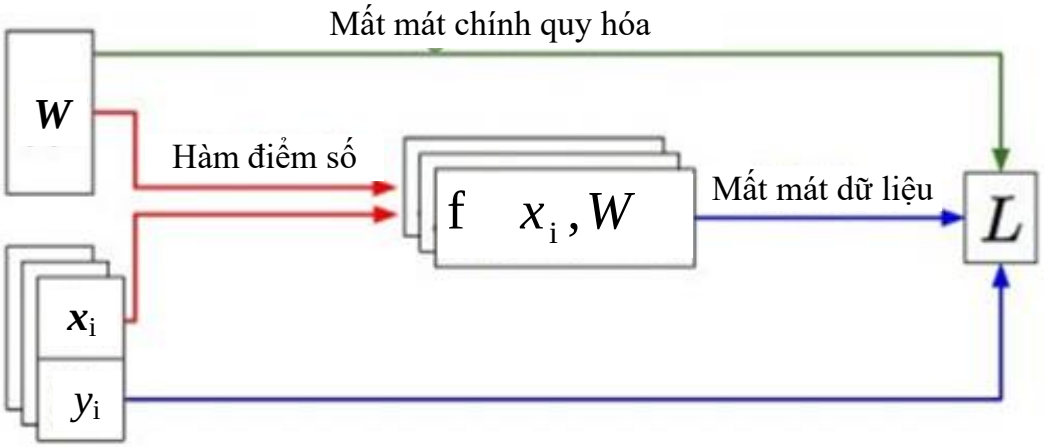
Trường hợp cực đoan của điều này là một thiết lập mà lô nhỏ chỉ chứa một ví dụ duy nhất. Quá trình này được gọi là giản gradient ngẫu nhiên (SGD Stochastic Gradient Descent) (hoặc đôi khi cũng là on-line gradient descent (giảm gradient trực tuyến)). Điều này tương đối ít phổ biến hơn vì trên thực tế, do tối ưu hóa mã vectơ hóa, về mặt tính toán, có thể hiệu quả hơn nhiều khi đánh giá gradient cho 100 ví dụ, so với gradient cho một ví dụ 100 lần. Mặc dù về mặt kỹ thuật, SGD đề cập đến việc sử dụng một ví dụ duy nhất tại một thời điểm để đánh giá gradient, mọi người cùng thường sử dụng thuật ngữ SGD ngay cả khi đề cập đến gradient descent lô nhỏ (tức là đề cập đến MGD cho "Minibatch Gradient Descent", hoặc BGD cho "Batch gradient descent" rất hiếm khi thấy), trong đó người ta thường cho rằng các lô nhỏ được sử dụng. Kích thước của lô nhỏ là một siêu tham số nhưng không phổ biến để xác thực chéo nó. Thông thường, nó dựa trên các ràng buộc về bộ nhớ (nếu có) hoặc được đặt thành một giá trị nào đó, ví dụ: 32, 64 hoặc 128. Trên thực tế, chúng ta sử dụng lũy thừa của 2 vì nhiều triển khai phép toán vectơ hóa hoạt động nhanh hơn khi đầu vào của chúng được định cỡ theo lũy thừa của 2.

4.8. TỔNG KẾT

Chương này đã trình bày các vấn đề sau.

- Định nghĩa một hàm điểm số (Score Fuction) từ các điểm ảnh của hình ảnh đến điểm số của loại (trong phần này, là một hàm tuyến tính phụ thuộc vào trọng số ***W*** và độ lệch **b**). Không giống như bộ phân loại kNN, ưu điểm của phương pháp tham số này là sau khi học các tham số, chúng ta có thể loại bỏ dữ liệu huấn luyện. Hơn nữa, việc dự đoán cho một ảnh thử nghiệm mới rất nhanh vì nó chỉ cần một phép nhân ma trận với ***W***, thay vì phải so sánh toàn diện với từng ví dụ huấn luyện.
- Ghép vectơ độ lệch vào ma trận trọng số để thuận tiện khi chỉ phải theo dõi một ma trận tham số.
- Định nghĩa các bộ phân loại tuyến tính: SVM và Softmax để đo lường mức độ tương thích của một tập tham số nhất định so với các nhãn thực tế trong tập dữ liệu huấn luyện. Cho thấy rằng hàm mất mát được định nghĩa theo cách mà việc đưa ra dự đoán tốt trên dữ liệu huấn luyện tương đương với việc có một mất mát nhỏ.
- So sánh hoạt động tối ưu hóa như là một người đi bộ đường dài bị bịt mắt muốn đi xuống đáy quả đồi. Cụ thể, cho thấy rằng hàm chi phí SVM là tuyến tính từng phần và có dạng hình bát.
- Đưa ra ý tưởng tối ưu hóa hàm mất mát bằng phương pháp tinh chỉnh lặp đi lặp lại, bắt đầu với một tập hợp trọng số ngẫu nhiên và tinh chỉnh chúng từng bước cho đến khi mất mát được giảm thiểu.
- Cho thấy rằng việc cập nhật tham số đòi hỏi một thiết lập phức tạp về kích thước bước (hay tốc độ học) cần được thiết lập chính xác: nếu quá thấp, tiến trình sẽ ổn định nhưng chậm. Nếu quá cao, tiến trình có thể nhanh hơn, nhưng rủi ro hơn. Chúng ta sẽ tìm hiểu sự đánh đổi này chi tiết hơn trong các phần sau.
- Cân nhắc về sự đánh đổi giữa việc tính toán gradient số và gradient phân tích. Gradient số đơn giản nhưng lại gần đúng và tốn kém. Gradient phân tích chính xác, tính toán nhanh nhưng dễ xảy ra lỗi hơn vì nó đòi hỏi phải suy ra gradient bằng toán học. Do đó, trong thực tế, người ta luôn sử dụng gradient phân tích và sau đó thực hiện kiểm tra gradient, trong đó việc triển khai của nó được so sánh với gradient số.
- Giới thiệu thuật toán Gradient Descent , thuật toán này tính toán gradient theo cách lặp đi lặp lại và thực hiện cập nhật tham số theo vòng lặp.

Ta có thể tóm tắt các vấn đã xét trong chương này trên trong hình 4.15. Tập dữ liệu gồm các cặp **(x,** *y***)** được cho sẵn và cố định. Các trọng số ban đầu là các số ngẫu nhiên và có thể thay đổi. Trong quá trình truyền tiếp, hàm điểm số sẽ tính điểm sô loại, được lưu trữ trong vectơ f . Hàm mất mát bao gồm hai thành phần: Mất mát dữ liệu tính toán tính tương thích giữa điểm số f và nhãn *y* . Mất mát chính quy hóa chỉ là một hàm của các trọng số. Trong quá trình Gradient Descent, chúng ta tính toán gradient trên các trọng số (và tùy chọn trên dữ liệu nếu muốn) và sử dụng chúng để thực hiện cập nhật tham số trong quá trình Gradient Descent.



Hình 4.15. tóm tắt các vấn đã xét trong chương này

Nội dung chính rút ra từ phần này là khả năng tính toán gradient của hàm mất mát theo trọng số của nó (và có một số hiểu biết trực quan về nó) là kỹ năng quan trọng nhất cần có để thiết kế, huấn luyện và hiểu mạng neuron.

Chương 5

CÁC MẠNG NEURON TÍCH CHẬP (CONVOLUTIONAL NEURON NETWORK)

Các mạng neuron tích chập (CNN: Covolutional Neural Network) là một thể loại đặc thù của các mạng neuron sâu sử dụng tích chập để xử lý dữ liệu đầu vào. Trái ngược với các mạng sâu kết nối hoàn toàn như đã xét, CNN vượt trội trong xử lý dữ liệu với một tổ chức không gian giống lưới (ví dụ: các dãy thời gian, ảnh, video v.v.), trong khi đồng thời giảm số lượng các tham số nhờ các thuộc tính chia sẻ trọng số của chúng. Phần này trước hết trình bày phép toán chập và động lực thúc đẩy sử dụng như là khối hay module cơ sở của các mạng neuron. Sau đó, trình bày một số các biến thể khác nhau cùng với các thí dụ về các kiến trúc quan trong nhất của CNN. Cuối cùng, tầm quan trọng của trường tiếp nhận (Receptive Field) - thuộc tính trung tâm của các mạng này được bàn luận.

5.1. PHÉP TOÁN CHẬP

Phép toán chập được định nghĩa là tích phân của tích hai hàm (f,g) sau khi một hàm được đảo ngược và dịch so với hàm kia. Chính thức, ta viết:

$$h(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau \qquad (5.1)$$

Pháp toán này có thể được ký hiệu bằng một dấu hoa thị (*), vì thế có thể được viết như sau:

$$h(t) = (f * g)(t) \qquad (5.2)$$

Thực chất phép toán tích chập chỉ ra việc một hàm tác động lên hàm khác như thế nào. Trực giác này nẩy sinh từ miền xử lý tín hiệu nơi mà thường quan trọng cần biết việc một tín hiệu chịu tác động của một bộ lọc như thế nào. Chẳng hạn, xét một tín hiệu liên tục môt chiều (uni-directional), giống như hoạt động của não bộ của một bệnh nhân trên một điện cực nào đó của điện tâm đồ và một bộ lọc Gauss. Kết quả phép toán chập giữa hai hàm này sẽ cho ra hiệu ứng của bộ lọc Gauss lên tín hiệu này, thực chất nó sẽ là một phiên bản được làm nhẵn của đầu vào.

Một cách nghĩ khác về phép toán chập là nó cho thấy cách thức mà hai hàm liên quan với nhau. Nói một cách khác, nó chỉ ra hai hàm giống hoặc khác nhau như thế nào tại các vị trí tương đối khác nhau. Thực chất, phép toán chập rất giống với phép toán tương quan chéo với sự khác biệt nhỏ là trong phép toán chập một trong hai hàm bị đảo. Đặc biệt trong ngữ cảnh học sâu, sự khác nhau chính xác giữa hai phép toán này chỉ là mối quan tâm thứ cấp; tuy nhiên, phép toán chập có nhiều thuộc tính hơn phép toán tương quan, chẳng hạn giao hoán. Lưu ý rằng khi các tín hiệu đối xứng, cả hai phép toán sẽ cho cùng một kết quả.

Để xử lý các tín hiệu rời rạc hữu hạn, ta có thể mở rộng định nghĩa phép toán tích chập. Cụ thể, cho trước hai tín hiệu rời rạc f(k) và g(k) với $k \in \mathbb{Z}$, phép toán tích chập được định nghĩa bởi

$$h[k] = \sum_n f[k-n]\,g[n] \qquad (5.3)$$

Tương tự có thể mở rộng phép toán tích chập cho các tín hiệu đa chiều. Chẳng hạn, ta có thể viết tích chập giữa hai tín hiệu hai chiều (ví dụ: I[i,j]. K[i,j]) như sau:

$$H[i,j] = \sum\sum I[i-m, j-n]\,K[m,n] \qquad (5.4)$$

Tích chập ảnh là biến đổi theo pixel được thực hiện bằng cách áp dụng một biến đổi nào đó được định nghĩa bằng một tập các trọng số, một pixel được biến đổi như sau:

$$y = \sum_{j=1}^{K} x_j w_j \qquad (5.5)$$

Rất thường xuyên, tín hiệu thứ nhất sẽ là đầu vào được quan tâm (ví dụ: một ảnh có kích thước lớn) trong khi tín hiệu thứ hai sẽ có kích thước tương đối nhỏ (ví dụ: một ma trận 3×3 hoặc 4×4) và sẽ thực hiện một phép toán đặc thù. Tín hiệu thứ hai thường được gọi là kenel (hạt nhân). Hình 5.1 minh họa tính toán tích chập cụ thể cho một bản đồ tính năng (Feature Map) của hình ảnh đầu vào trong đó bộ lọc tính năng có kích thước 3×3. Quá trình tích chập trên hình 5.1 được tính như sau:

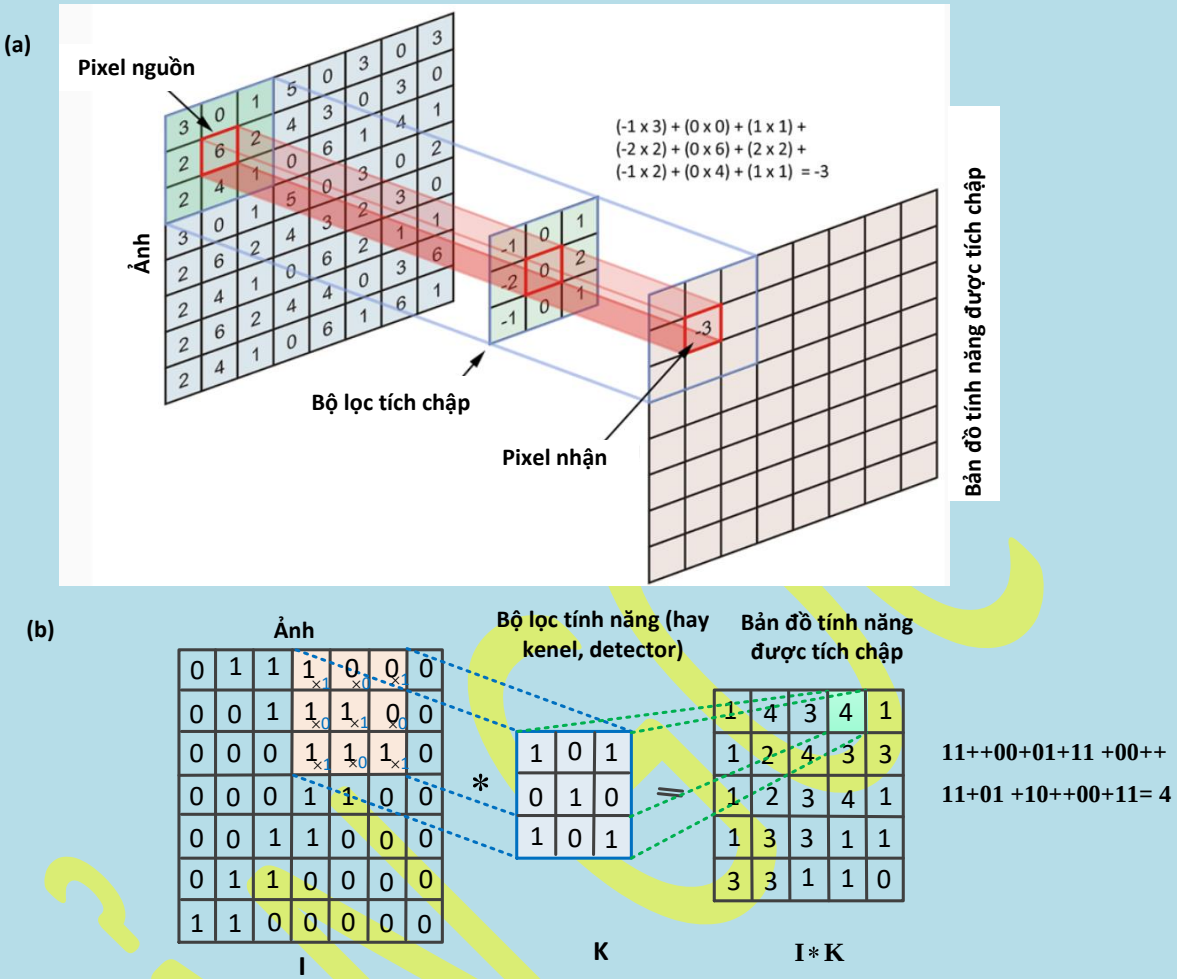
(a) (-1×3)++(0×0)+(1×1)+(-2×2) +(0×6)++(2×2)+(-1×2) +(0×4)++(1×1)= -3

(b) (1×1)+(0×0)+(1×0)+ (0×1) +(1×1)+(0×0)+(1×1) +(0×1)+(1×1)= 4

Chi tiết hơn, bộ lọc tính năng (Feature Filter) hay kenel chập sẽ dịch trên toàn bộ vị trí của bản đồ tính năng đầu vào (Input Feature Map), nhân phần tử và cộng để được tích chập đầu ra của vị trí tương ứng. Một cách chi tiết, kenel tích chập chuyển dịch trên tất cả các vị trí của đầu vào và nhân từng phần tử và cộng được thực hiện để tính toán đầu ra tích chập tại vị trí tương ứng.

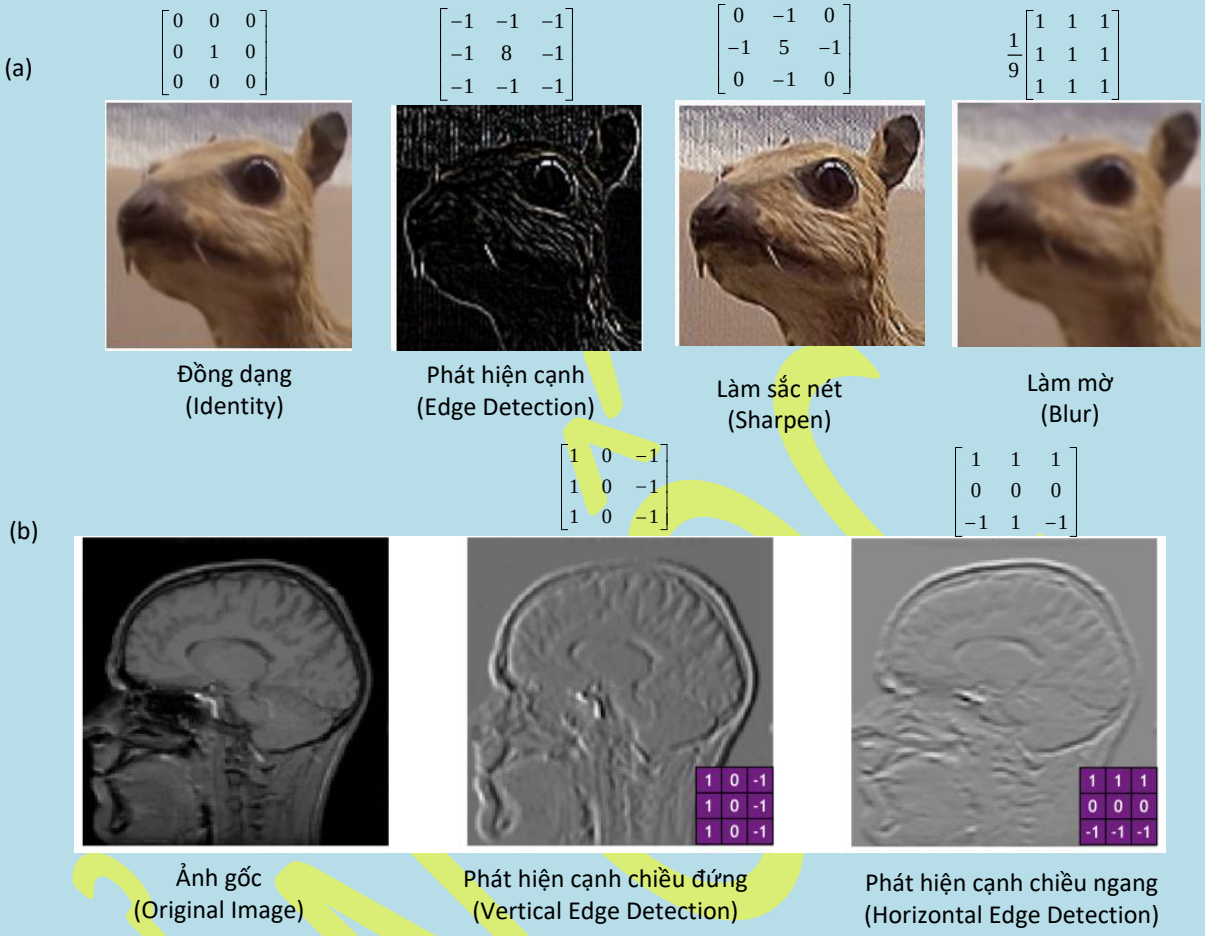
Các thí dụ cho các ứng dụng của các tích cập đến một ảnh được cho trên hình 5.2. Cuối cùng, cần lưu ý, trong các perceptron, nói chung theo sau một tích chập sẽ là một hàm kích hoạt phi tuyến tính, chẳng hạn ReLU (hình 5.3 và 5.4 cho thấy hai thí dụ của kích hoạt áp dụng cho một bản đồ tính năng).

BẢN GỐC

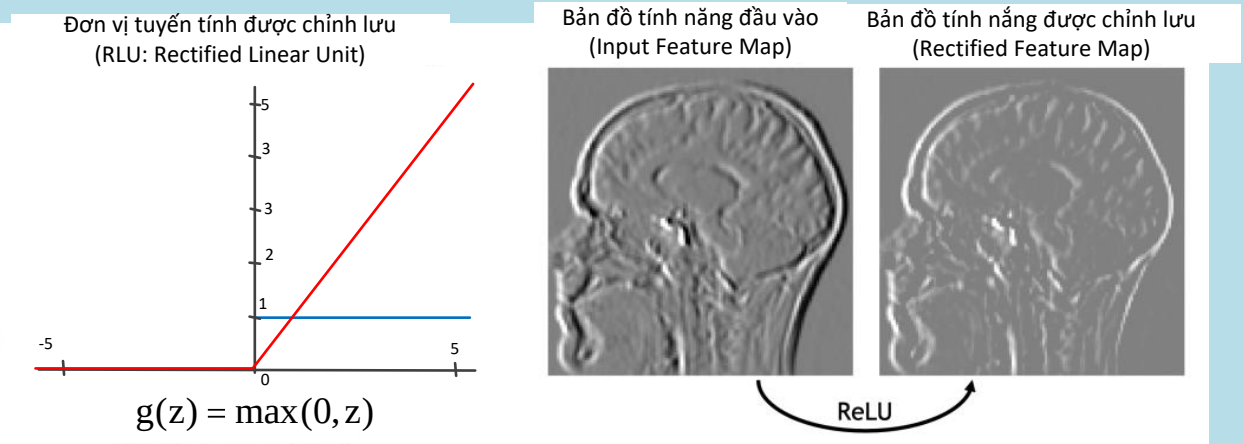


Bộ lọc tích chập thường được gọi là bộ lọc tính năng (Feature Filter), hay hạt nhân (Kenel), hsy bộ phát hiện (Detector)

Hình 5.1. Trực quan phép toán tích chập trong 2D



Hình 5.2.Hai thí dụ về các tích chập áp dụng cho hai ảnh . (a) các bộ lọc tính năng hoạt động như là các bộ: đồng dạng, phát hiện cạnh, làm sắc nét và làm mờ. (b) các bộ lọc tính năng hoạt động như là các bộ: phát hiện cạnh chiều đứng và phát hiện cạnh chiều ngang. Tất nhiên, trong CNN, các bộ lọc được học, không được quy định trước, nên không đảm bảo rằng trong số các bộ lọc được học này, sẽ có các bộ phát hiện cho trường hợp đứng/ngang, mặc dù đây sẽ thường xuyên là trường hợp trong thực tiễn, đặc biệt đối với các lớp thứ nhất của kiến trúc



Hình 5.3. Một thí dụ về áp dụng hàm kích hoạt phi tuyến tính (RLU trong trường hợp này) cho một ảnh

5.2. CÁC THUỘC TÍNH CỦA PHÉP TOÁN TICH CHẬP

Trong trường hợp của một vùng rời rạc, phép toán tích chập có thể được thực hiện bằng cách nhân ma trận đơn giản mà không cần dịch một tín hiệu trên tín hiệu khác.Đặc biệt có thể đạt được điều này bằng cách sử dụng biến đổi ma trận Toeplitz. Biến đổi Toeplitz tạo ra một ma trận thưa (Sparse Matrix) với các phần tử được lặp và khi nhân nó với các tín hiệu đầu vào tạo ra kết quả tích chập. Để minh họa cách thực hiện phép toán tích chập này bằng cách nhân ma trận, ta xét thí dụ gồm một 3×3 kenel (K) và một đầu vào 4×4 (I) như sau:

$$K = \begin{bmatrix} k_{00} & k_{00} & k_{00} \\ k_{00} & k_{00} & k_{00} \\ k_{00} & k_{00} & k_{00} \end{bmatrix}, I = \begin{bmatrix} i_{00} & i_{01} & i_{02} & i_{03} \\ i_{10} & i_{11} & i_{12} & i_{13} \\ i_{20} & i_{21} & i_{22} & i_{23} \\ i_{30} & i_{31} & i_{32} & i_{33} \end{bmatrix}$$

Sau đó, phép toán tích chập có thể được tính toán bằng cách nhân giữa ma trận kenel được chuyển đổi bởi Toeplitz:

$$\hat{K} = \begin{bmatrix} k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{00} & k_{01} & k_{02} & 0 & k_{10} & k_{11} & k_{12} & 0 & k_{20} & k_{21} & k_{22} \end{bmatrix}$$

và đầu vào được tạo lại hình dạng sau:

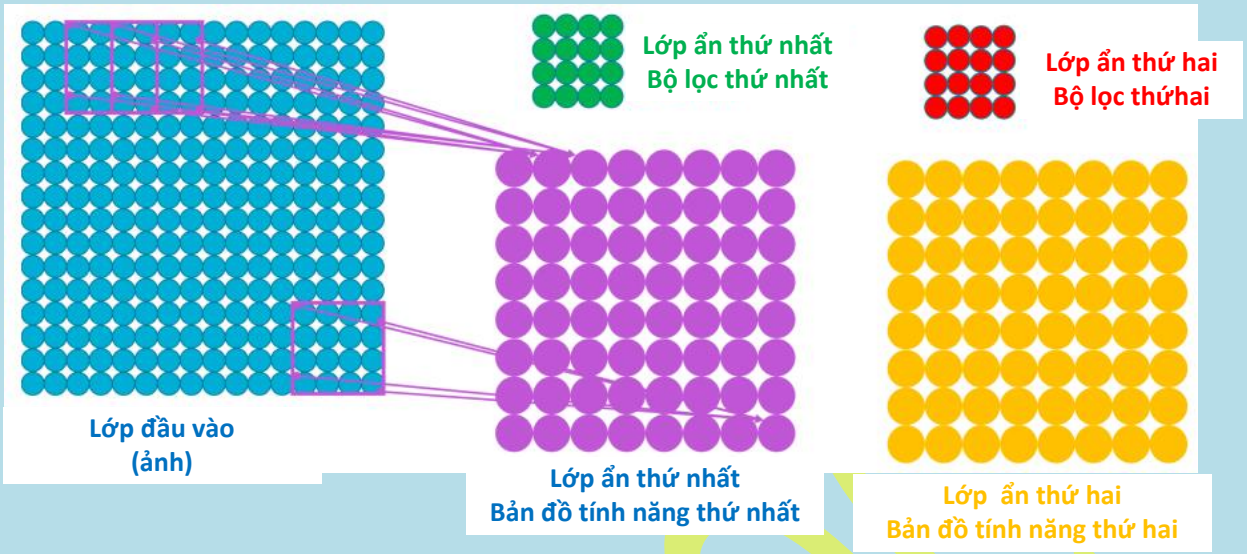
$$\hat{I} = \begin{matrix} i_{00} & i_{01} & i_{02} & i_{03} & i_{10} & i_{11} & i_{12} & i_{13} & i_{20} & i_{21} & i_{22} & i_{23} & i_{30} & i_{31} & i_{32} & i_{33} \end{matrix}^T$$

Đầu ra được tạo ra sẽ cần tạo lại dạng như là một ma trận 2×2 để được đầu ra tích chập.

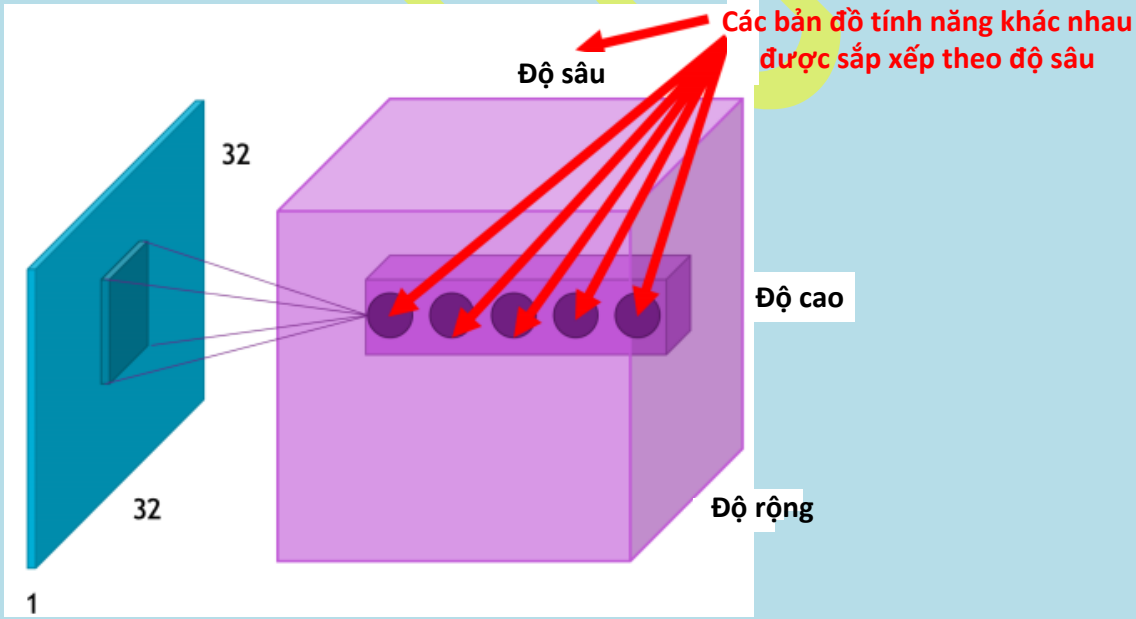
Thực hiện nhân ma trận chỉ mang tính minh họa về một số thuộc tính của phép toán tích chập. Các thuộc tính này là động lực chính cho sử dụng các phần tử này trong các mạng neuron.

Bằng cách biến đổi phép toán tích chập thành phép toán nhân ma trận, rõ ràng là điều này có thể phù hợp với chính thức hóa các hàm luyến tính như đã xét trong mục 3.3. Chính vì vậy, có thể thiết kế các kenel khả đào tạo. Trong thực tế, nhiều kenel có thể được xếp chồng lên đỉnh của từng khối khác để tạo thành các CNN sâu. Thường thì, đầu ra của một phép toán tích chập được gọi là bảm đồ tính năng (fmap: feature map) hay đơn giản là các tính năng (Features).

Một khía cạnh khác của phép toán tích chập là nó đòi hỏi các tham số ít hơn nhiều so với các mạng MLP dựa trên kết nối hoàn toàn. Như có thể nhìn thấy từ ma trận $\hat{K}$, các ma trận hoàn toàn như nhau được chia sẻ trên tất cả các vị trí. Cuối cùng, thay vì học một tập khác nhau các tham số cho các vị trí khác nhau của đầu vào, chỉ cần học một tập. Điều này ám chỉ là chia sẻ tham số hay chia sẻ trọng số và có thể giảm lớn khối lượng bộ nhớ cần để lưu trữ các tham số mạng. Hình 5.4 minh họa qua trình chia sẻ trọng số trên các vị trí , đồng thời với việc nhiều bộ lọc (để dẫn đến nhiều biểu đồ tính năng) được tính toán đối với một lớp cho trước. Nhiều bản đồ tính năng đối với một lớp cho trước được lưu trữ bằng cách sử dụng kích thước (dimenssion) khác (xem hình 5.5) dẫn đến một mảng 3D khi đầu vào là một ảnh 2D (và 4D khi đầu vào là một ảnh 3D).



Hình 5.4. Đối với một lớp cho trước, một số (thường là nhiều) bộ lọc được học, mỗi bộ lọc trong số chúng có khả năng phát hiện một đặc trưng đặc thù trong ảnh, dẫn đến một số bản đồ tính năng/lọc. Mặt khác, đối với một bộ lọc cho trước, các trọng số được chia sẻ trên tất cả các vị trí của ảnh



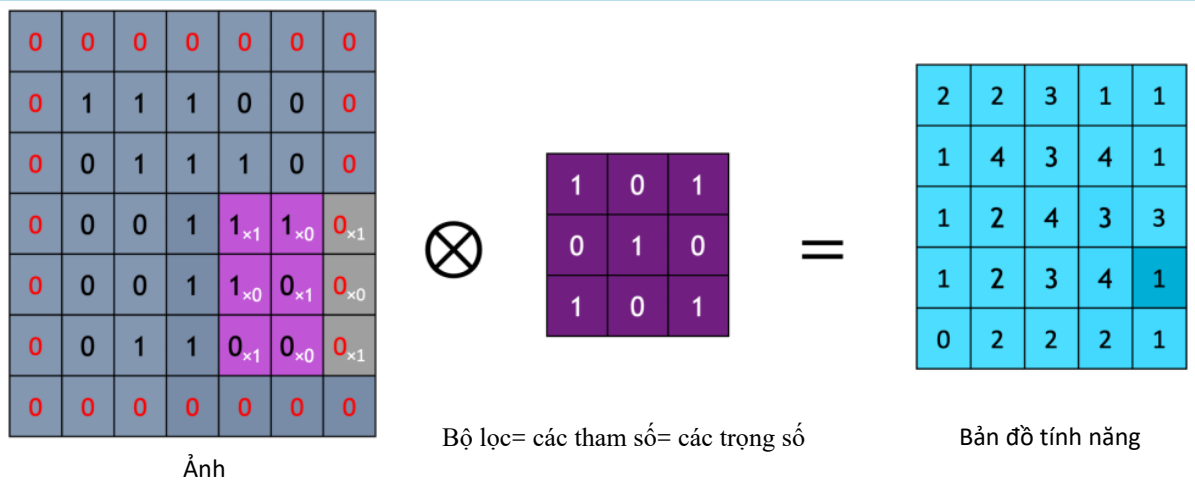
Hình 5.5. Các bản đồ tính năng khác nhau đối với một lớp cho trước được sắp xếp dọc theo một kích thước khác. Như vậy các bản đồ tính năng sẽ là một mảng 3D khi đầu vào là một ảnh 2D (và 4D nếu đầu vào là một ảnh 3D)

Các mạng neuron tích chập đã được chứng minh là khá mạnh trong xử lý dữ liệu có cấu trúc không gian (ví dụ các ảnh, các video, v.v). Điều này hiệu quả dựa trên thực tế là không có một kết nối cục bộ của các phần tử của kenel trong khi cùng một thời điểm cùng một kenel được áp dụng cho các vị trí khác nhau của đầu vào. Xử lý này cho ta một thuộc tính khá hữu ích được gọi là tương biến tịnh tiến (Translation Equivariance) cho phép mạng đưa ra các đáp ứng giống nhau cho các vị trí khác nhau của đầu vào. Một thí dụ về tính hữu ích của thuộc tính này có thể được nhận thấy ở các nhiệm vụ phát hiện ảnh.

Cuối cùng, một thuộc tính quan trọng khác của phép toán tích chập là nó giảm sự phụ thuộc lẫn nhau của kích thước đầu vào với các tham số khả đào tạo. Chẳng hạn trong trường hợp các MLP, kích thước của ma trận trọng số là hàm của kích thước đầu vào. Đặc biệt là, một lớp kết nối dầy đặc sắp xếp 256 tính năng lên 10 đầu ra sẽ có kích thước là $\boldsymbol{W} \in \mathbb{R}^{10\times256}$. Trái lại, trong các lớp tích chập, số các tham số khả đào tạo chỉ phụ thuộc vào kích thước kenel và số lượng kenel mà lớp này có. Điều này cuối cùng cho phép xử lý các đầu vào với các kích thước tùy ý, chẳng hạn trong trường hợp các mạng neuron tích chập kết nối hoàn toàn.

5.3. CÁC HÀM VÀ CÁC BIẾN THỂ

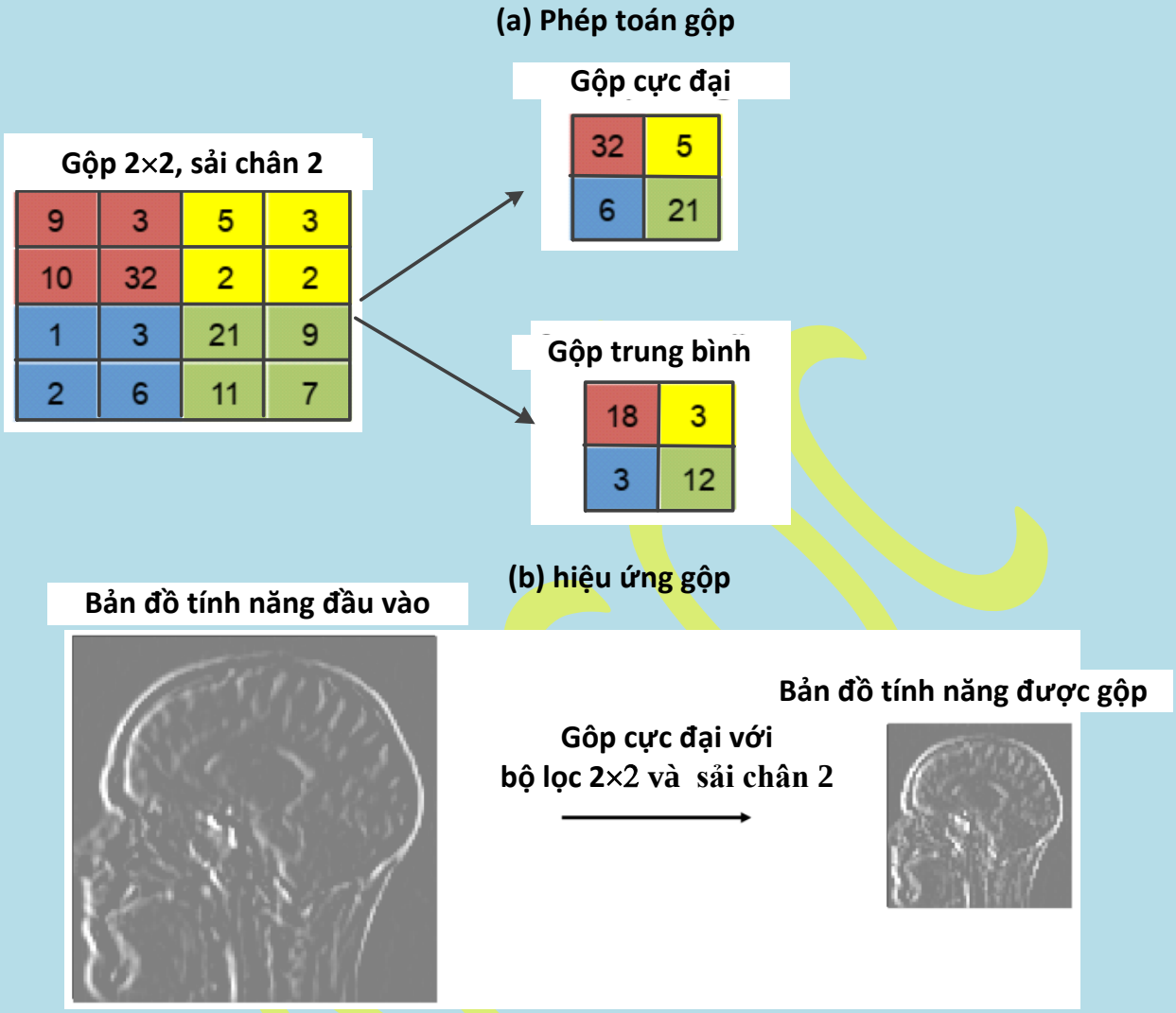
Ta có thể nhận thấy rằng phép toán tích chập thay đổi kích thước của đầu ra được tạo ra. Trong thí dụ được thể hiện trên hình 5.1, ảnh có kích thước 7×7 khi tích chập với một kenel kích thước 3×3, tạo ra một bản đồ tính năng kích thước 5×5. Thậm chí có thể thay đổi kích thước bằng cách đệm lót (Padđing) tương ứng (xem hình 5.6) trong một số trường thợp thực sự cần giảm các kích thước của đầu vào. Có thể đạt được giảm kích thước này bằng một số cách tùy thuộc vào nhiệm vụ cần thực hiện. Trong phần này ta sẽ bàn luận một số hàm điển hình nhất được sử dụng trong các CNN.



Hình 5.6. Phép toán đệm lót (Padđing Operration) bao gồm bổ xung các số không xung quanh ảnh để có thể nhận được các bản đồ tính năng có cùng kích thước như ảnh gốc

5.3.1. Lấy mẫu giảm (Downsampling), ví dụ: các lớp gộp (Pooling).

Trong rất nhiều kiến trúc của CNN, các phép toán lấy mẫu giảm tính toán, còn được gọi là các lớp gộp, các phép toán xử lý này tổng hợp các giá trị đầu vào của chúng tùy thuộc và thiết kế của chúng. Lớp gộp thực hiện chiết suất các giá trị đặc biệt của một tập giá trị thường là giá trị cực đại hoặc giá trị trung bình của tất cả các giá trị. Điều này giảm kích thước của bản đồ tính năng đầu ra. Một loạt các phép tính làm giảm kích thước của bản đồ tính năng được gọi là gộp (Pooling). Pooling, được áp dụng riêng biệt cho từng kênh, cho phép mạng mạnh mẽ và bất biến trước những thay đổi và biến dạng nhỏ. Pooling kết hợp, hoặc gộp, một tập hợp các giá trị trong trường tiếp nhận (receptive field) của nó thành một số giá trị nhỏ hơn. Một số trong số các lớplấy mẫu giảm phổ biến nhất là gộp cực đại (Maximum Pooling), gộp trung bình (Average Pooling) như được minh họa trên hình 5.7 hay gộp trung bình toàn cục (Global Average Pooling). Hình 5.7 (a) minh họa phép toán gộp, hình 5.7 (b) thể hiện hiệu ứng gộp với với bộ lọc gộp 2×2 và sải chân 2.

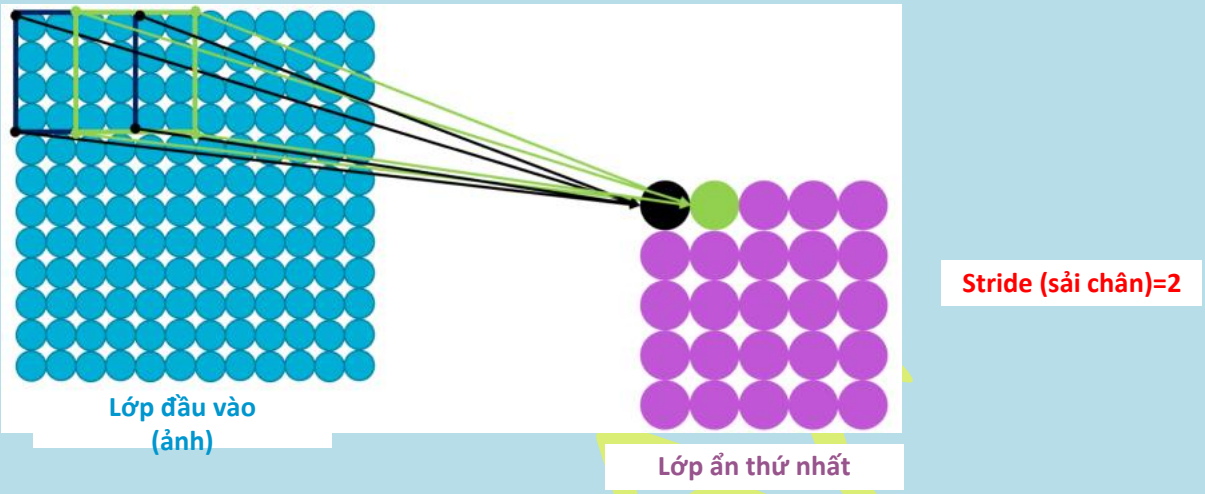


Hình 5.7. Các dạng gộp khác nhau

Trong hai trường hợp đầu, giá trị gộp cực đại và gộp trung bình được sử dụng là tính năng của đầu ra trên các vùng chồng lấn của một kích thước gộp được quy định trước. Trong trường hợp gôp trung bình toàn cục, tất cả các chiều không gian đều được trình bày với giá trị trung bình.

5.3.2. Tích chập theo sải chân (Strided Operation)

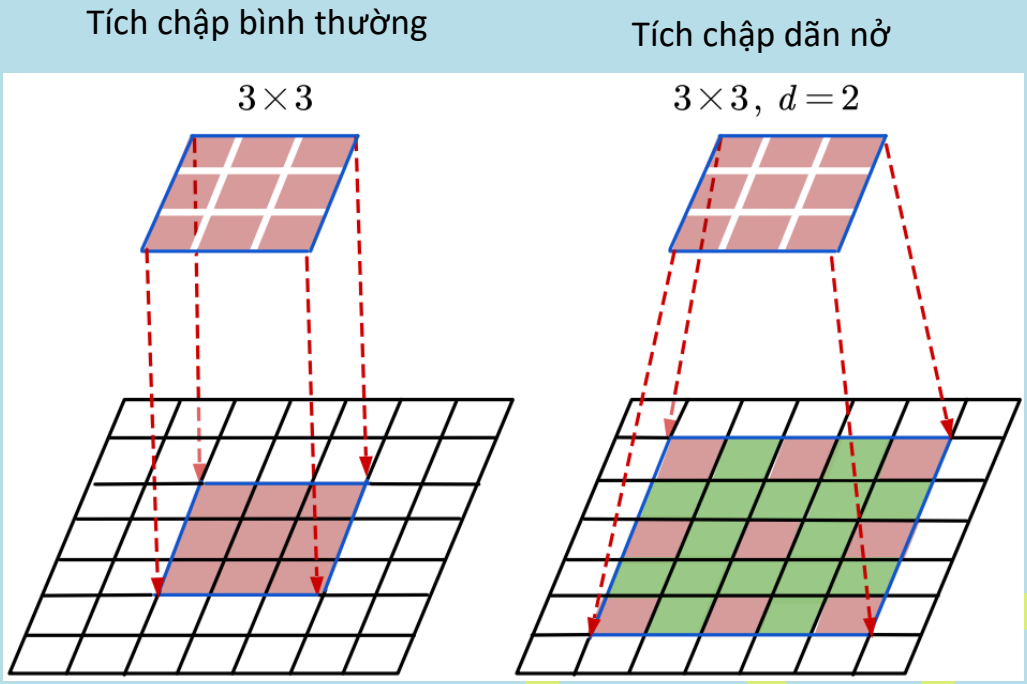
Tích chập theo sải chân đề cập đến trường hợp đặc thù trong đó thay vì áp dụng tích chập cho mọi vị trí với sử dụng một kích thước bước hay sải chân (Stride, s) bằng 1, các kích thước sải chân khác nhau có thể được xem xét (hình 5.8). Phép toán này dẽ tạo ra đầu ra tích chập với ít các phần tử hơn nhiều. Có thể coi các khối tích chập với s>1 trong các kiến trúc CNN là một cách giảm kích thước của các khối trung gian.



Hình 5.8. Phép toán sải chân, tại hình vẽ này sải chân bằng 2

5.3.3. Tích chập dãn nở hay tích chập được giãn nở (Atrous or Dilated Covolution)

Tích chập được dãn nở (Dilated Convolution) hay còn được gọi là tích chập dãn nở (Atrous Convolution) là tích chập các kenel được dãn nở bằng cách chèn các lô bằng không giữa các giá trị khác không của một kenel. Trong trường hợp này, một tham số (d) của phép toán tích chập bổ sung được bổ sung và nó thay đổi khoảng cách giữa các phần tử của kenel. Thực chất, nó thay đổi tiếp cận của kenel nhưng giữ nguyên số lượng các tham số khả đào tạo. Tích chập bình thường với kích thước $(2r+1)\times(2r+1)$ với độ dãn nở d sẽ tăng kích thước thành $(2r\times d+1)\times(2r\times d+1)$. Chẳng hạn, tích chập dãn nở với một kenel kích thước 3×3 và tốc độ dãn nở d=2 sẽ được sắp xếp thưa trên một lưới 5×5 như trên hình 5.9.



Hình 5.9.Minh họa tích chập dãn nở 2D với kích thước không gian 3×3 và thừa số dãn nở d=2. Cả hai tích chập không dãn nở và tích chập dãn nở đề có cùng số lượng các tham số trong khi tích chập dãn nở có kích thước lớn hơn.

Các kenel tích chập ãn nở có thể coi là thưa trong dạng bàn cờ như trên hình 5.9. NHư vậy phép toán tích chập dãn nở có thể làm mất một số thông tin cục bộ. Vấn đề này trở nên xấu hơn khi thừa số tích chập lớn. Kết quả là sử dụng tích chập dãn nở có thể dẫn đến các hiện vật tạo lưới (Gridđing Artifack) và vì thế làm hỏng một số chi tiết tần số cao của ảnh có độ phân giải cao (HR Image). Điều này có hại đối với học các hàm lập bản đồ phân giải thấp vào phân giải cao chính xác.

5.3.4. Tích chập được chuyển vị (Transposed Operation)

Trong một số trường hợp, ta không chỉ lấy mẫu giảm các chiều không gian của đầu vào mà mã cũng thường cần áp dụng phép toán lấy mẫu tăng (Upsampling) tại tầng cuối cùng của mạng. Trường hợp mang tính biểu tượng nhất là nhiệm vụ phân đoạn ảnh rong đó phân loại mức độ điểm ảnh (pixel level) được kỳ vọng và vì thế đầu ra mạng neuron phải có kích thước như đầu vào. Trong các trường hợp này, một số các phép tuán lấy mẫu tăng thường được áp dụng. Có thể đạt được lấy mẫu tăng một phép toán đượcchuyển vị (Transposed Operation), phép toán này cuối cùng sẽ tăng kích thước đầu ra. Chi tiết, tích

chập được chuyển vị được thực hiện bằng cách giãn nở đầu vào thay vì kenel trước khi áp dụng phép toán tích chập. Bằng cách này, một đầu vào kích thước 5×5 đạt được kích thước 10×10 sau giãn nở với d=2. Với đệm lót phù hợp và sử dụng kenel kích thước 3×3 đầu ra cuối cùng sẽ tăng gấp đôi kích thước.

5.3.4. Tính toán trường tiếp nhận

Trong ngữ cách các mạng neuron sâu và đặc biệt là các CNN, thuật ngữ trường tiếp nhận (Recepsive Field) được sử dụng để định nghĩa tỷ lệ của đầu vào để tạo ra một tính năng đặc thù. Chẳng hạn một CNN nhận một ảnh làm đầu vào và chỉ áp dụng phép toán tích chập đơn với một kenel kích thước 3×3 sẽ có trường tiếp nhận 3×3. Điều này có nghĩa rằng đối với từng pixel của bản đồ tính năng thứ nhất, một vùng 3×3 của đầu vào sẽ được xét. Bây giờ, nếu một lớp khác được bổ sung, lại với kích thước 3×3, trường tiếp nhận của bản đồ tính năng mới so với đầu vào của CNN sẽ là 5×5. Nói một cách khác tỷ lệ của đầu vào được sử dụng để tính toán từng phần tử của bản đồ tính năng của lớp thứ hai tăng.

Tính toán trường tiếp nhận tại các phần khác nhau của CNN rất quan trọng để hiểu biết hoạt động bên trong của một kến trúc đặc thù. Chẳng hạn, một CNN được thiết kế để nhận một ảnh kích thước 256×256 và đòi hỏi thông tin từ tất cả các phần của nó phải có trường tiếp nhận gần với kích thước đầu vào. Trường tiếp nhận có thể bị ảnh hưởng bởi tất cả các tham số tích chập khác nhau và các phép toán giảm mẫu/tăng mầu được mô tả trong phần trước.

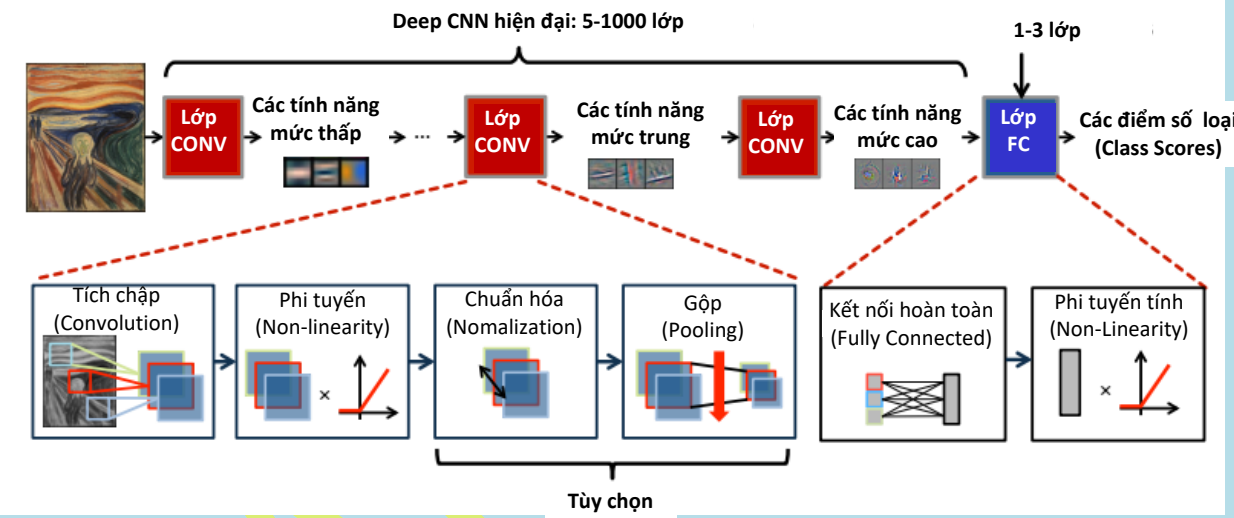
5.4. CÁC KIẾN TRÚC MẠNG NEURON TÍCH CHẬP KINH ĐIỂN

5.4.1. Tổng quan mạng CNN

Trong các thập kỷ gần đây, đa dạng các kiến trúc mạng neuron tích chập đã được đề xuất. Trong phần này ta sẽ chỉ xét một số kiến trúc kinh điển cho phân loại và hồi quy. Lưu ý rằng phân loại và hồi quy thường có thể được thực hiện bởi cùng một kiến trúc nhưng

thay đổi hàm mất mát (ví dụ: entropy chéo cho phân loại, sai số bình phương trung bình cho hồi quy).

Tổng quát, mạng thần kinh tích chập (CNN), bao gồm nhiều lớp tích chập (CONV) như trong hình 5.10. Trong các mạng như vậy, mỗi lớp tạo ra một sự trừu tượng hóa cấp độ cao hơn liên tiếp của dữ liệu đầu vào, được gọi là bản đồ tính năng (fmap: feature map) giữ lại thông tin thiết yếu nhưng duy nhất. CNN hiện đại có thể đạt được hiệu năng vượt trội bằng cách sử dụng hệ thống phân cấp các lớp rất sâu. CNN được sử dụng rộng rãi trong nhiều ứng dụng bao gồm nhận dạng hình ảnh, nhận dạng giọng nói, chơi trò chơi, robot v.v. Phần này sẽ tập trung vào việc sử dụng nó trong xử lý hình ảnh, đặc biệt cho nhiệm vụ phân loại hình ảnh (image classification).

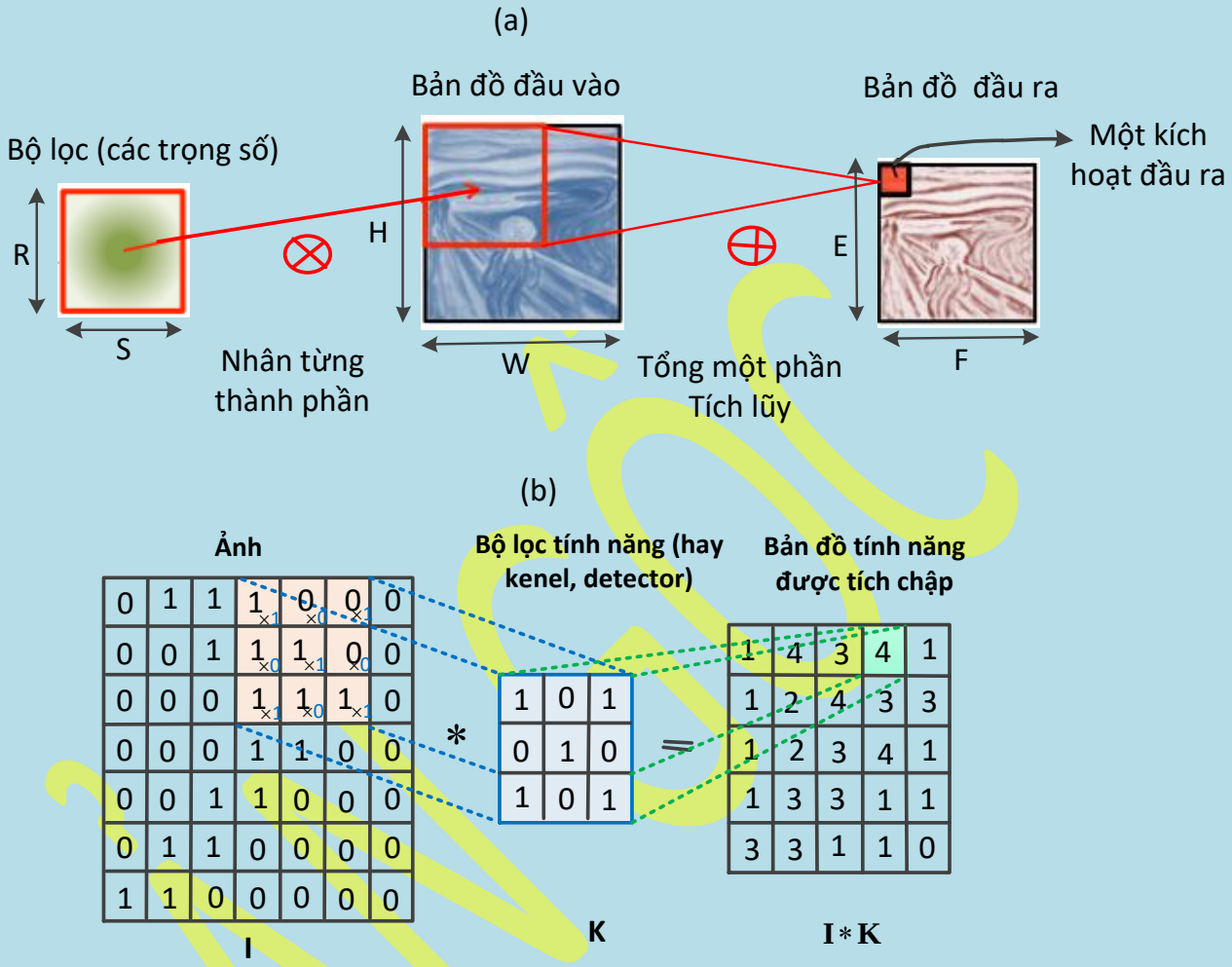


Hình 5.10. Mạng CNN

Hình 5.11 cho thấy tích chập 2D trong xử lý ảnh truyền thống. Hình 5.11(a), trong đó tổng trọng số cho mỗi kích hoạt đầu ra được tính toán chỉ bằng cách sử dụng một vùng lân cận nhỏ của các kích hoạt đầu vào (tức là tất cả các trọng số vượt ra ngoài vùng lân cận được đặt bằng không) và trong đó cùng một tập hợp trọng số được chia sẻ cho mọi đầu ra (tức là, bộ lọc là không gian bất biến). Hình 5.11(b) minh họa tính toán tích chập cụ thể cho một bản đồ tính năng của hình ảnh đầu vào trong đó bộ lọc tính năng có kích thước 3×3 . Quá trình tích chập trên hình 5.11(b) được tính như sau:

$$1\times1++0\times0+0\times1+1\times1\ +0\times0++1\times1+0\times1\ +1\times0++0\times0+1\times1=4$$

Chi tiết hơn, bộ lọc tính năng (kenel chập) sẽ dịch trên toàn bộ vị trí của bản đồ tính năng đầu vào, nhân phần tử và cộng để được tích chập đầu ra của vị trí tương ứng.

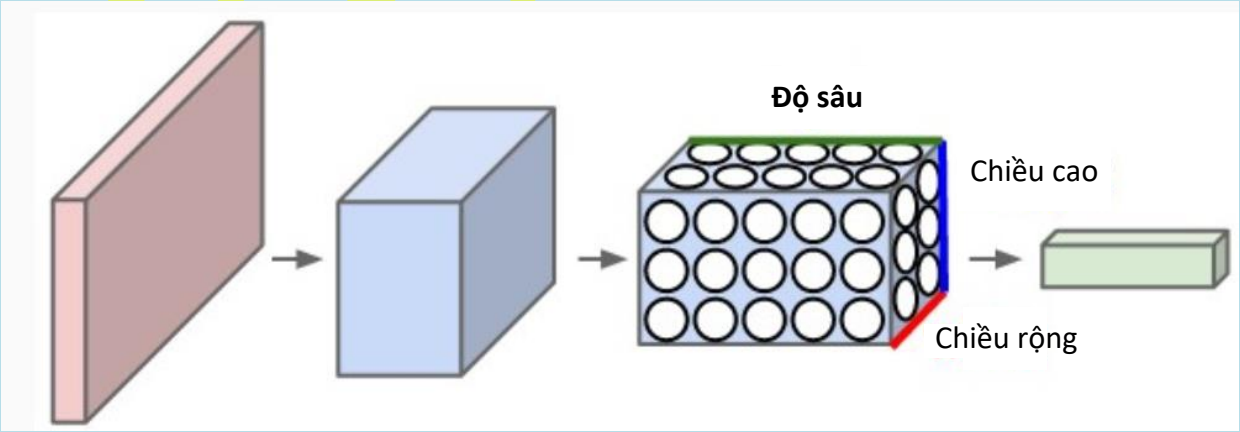


Hình 5.11. Tích chập 2D trong xử lý ảnh truyền thống

Một mạng neuron thông thường không có khả năng mở rộng cho toàn bộ các ảnh. Để - hiểu được điều này ta xét một tập con của tập dữ liệu CIFAR-10 (hình 15.2).

Hình 5.12. Một tập con của tập dữ liệu CIFAR-10

Tập dữ liệu này vao gồm các ảnh có kích thước 32×32×3 (Độ rộng×Độ cao×các kênh mầu RGB) được chia theo thể loại mà chúng chứa. Lưu ý rằng mặc dù đây là các ảnh 2D, do các kênh mầu, ta phải xử lý các hình khối (hình 5.13).



Mạng CNN sắp xếp các neuron theo ba chiều (độ rộng, độ cao, độ sâu) được thể hiện ở mooyj trong ba lớp. Mỗi lớp của của CNN biến đổi thể tích đầu vào 3D vào một thể tích đầu ra 3G các

các kích hoạt. Trong thí dụ này, , lớp đầu vào mầu đỏ chứa ảnh đầu vào, vì thế độ cao và độ rộng sẽ là các kích thước của ảnh nay và độ sâu là 3 (đỏ, luch, chàm).

Hình 5.13. Xử lý ảnh hình khối do có các kênh mầu RGB

Nếu ta có một perceptron kết nối đến tất cả các đầu vào, có nghĩa là phải có $32\times32\times3=3072$ trọng số. Số lượng này nhanh chóng vượt quá khả năng khi mở rộng đến các ảnh lớn.

Thông thường, các mạng neuron tích chập xử lý vấn đề này bằng các sử dụng mạng nối thẳng (Feedforward Network) và có kết nối cục bộ giữa các lớp, nghĩa là ta sẽ nối từng neuron chỉ đến một vùng cục bộ của thể tích đầu vào.

Dưới đây để đơn gian ta chỉ xét ba loại lớp cơ sở:

- Các lớp tích chập;
- Các lớp gộp
- Các lớp kết nối toàn bộ

Để mô tả các lớp này ta sẽ đề cập sự kết nối của chúng và kích thước của chúng: độ rộng, độ cao và độ sâu.

Lớp tích chập

Lớp tích chập bao gồm một tập các bộ lọc có thể học. Mỗi bộ lọc có kích thước không gian nhỏ theo chiều rộng và chiều cao, nhưng mở rộng xuyên suốt toàn bộ độ sâu của thể tích đầu vào. Đặc điểm của lớp này như sau:

- Các bộ lọc có thể học được áp dụng cho các vùng cục bộ
- Một bản đồ kích hoạt nhiều chiều (High Dimention) quyết định ảnh hưởng của bộ lọc
- Các trọng số có thể được chia sẻ trên các hàm kích hoạt để giảm số lượng các tham số có thể học
- Độ sâu (hay số kênh) về mặt khái niệm sẽ học các tính năng khác nhau
- Tương đương chuyển đổi: nếu đầu vào thay đổi, thì đầu ra cũng thay đổi theo cách tương tự

BẢN GỐC

5.4.2. Thí dụ kiến trúc đơn giản dựa trên CIFAR-10

Chúng ta sẽ đi sâu hơn vào chi tiết bên dưới, nhưng một CNN đơn giản cho phân loại CIFAR-10 có thể có kiến trúc [INPUT - CONV - RELU - POOL - FC]. Chi tiết hơn:

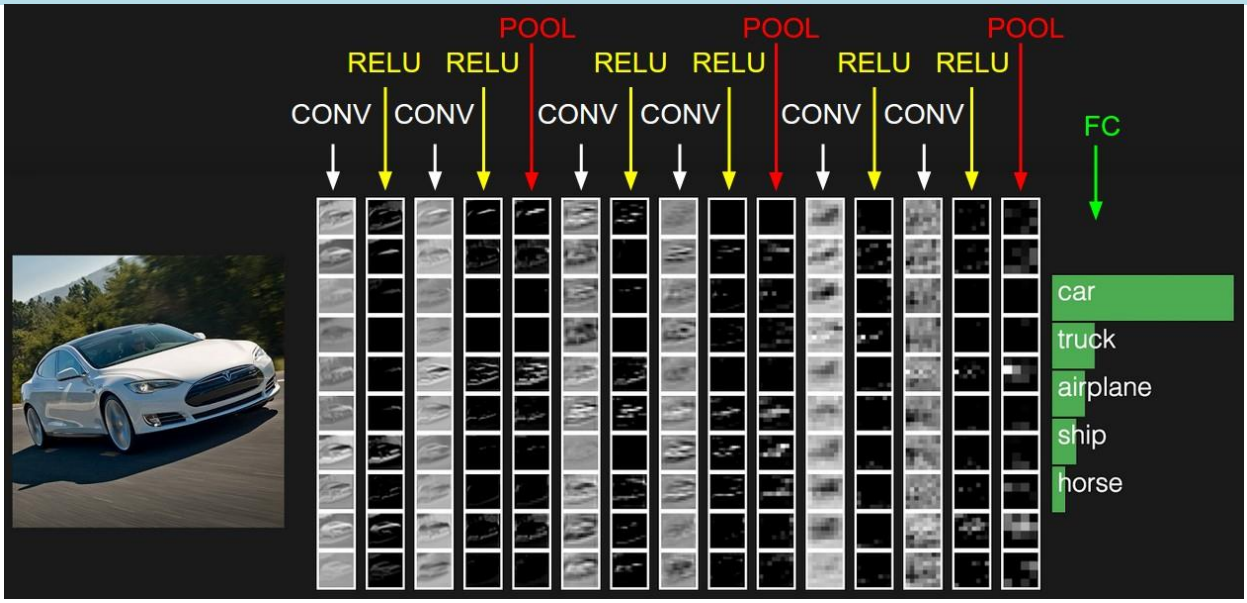
- INPUT [32x32x3] sẽ lưu giữ các giá trị pixel thô của hình ảnh, trong trường hợp này là hình ảnh có chiều rộng 32, chiều cao 32 và có ba kênh màu R, G, B.
- Lớp tích chập (CONV) sẽ tính toán đầu ra của các neuron được kết nối với các vùng cục bộ trong đầu vào, mỗi neuron tính toán tích vô hướng giữa trọng số của chúng và một vùng nhỏ mà chúng được kết nối trong thể tích đầu vào. Điều này có thể dẫn đến thể tích như [32x32x12] nếu chúng ta quyết định sử dụng 12 bộ lọc.
- Lớp RELU sẽ áp dụng một hàm kích hoạt từng phần tử, chẳng hạn như max (0 ,x) ngưỡng ở mức 0. Điều này giữ nguyên kích thước của khối lượng ([32x32x12]).
- Lớp POOL sẽ thực hiện thao tác lấy mẫu xuống theo các chiều không gian (chiều rộng, chiều cao), tạo ra thể tích như [16x16x12].
- Lớp FC (tức là lớp kết nối đầy đủ) sẽ tính toán điểm số củar loại, tạo ra thể tích có kích thước [1x1x10], trong đó mỗi số trong 10 số tương ứng với một điểm số của loại, chẳng hạn như trong 10 hạng mục của CIFAR-10. Giống như Mạng Neuron thông thường và đúng như tên gọi, mỗi neuron trong lớp này sẽ được kết nối với tất cả các số lượng neuron trong thể tích trước đó.

Theo cách này, CNN biến đổi từng lớp ảnh gốc từ các giá trị pixel gốc thành điểm số loại cuối cùng. Lưu ý rằng một số lớp chứa tham số, trong khi một số khác thì không. Cụ thể, các lớp CONV/FC thực hiện các phép biến đổi không chỉ phụ thuộc vào các kích hoạt trong khối lượng đầu vào mà còn phụ thuộc vào các tham số (trọng số và độ lệch của các neuron). Mặt khác, các lớp RELU/POOL sẽ triển khai một hàm cố định. Các tham số trong các lớp CONV/FC sẽ được huấn luyện với thuật toán giảm dần gradient sao cho điểm số của loại mà CNN tính toán nhất quán với các nhãn trong tập huấn luyện của mỗi ảnh.

Tóm lại:

- Kiến trúc CNN trong trường hợp đơn giản nhất là danh sách các lớp chuyển đổi khối lượng hình ảnh thành khối lượng đầu ra (ví dụ: lưu giữ các điểm số của loại)
- Có một số kiểu lớp riêng biệt (ví dụ: CONV/FC/RELU/POOL là kiểu phổ biến nhất)
- Mỗi lớp chấp nhận .14. cho thấy các kích hoạt của một thí dụ.hối lượng 3D đầu vào và chuyển đổi nó thành một khối lượng 3D đầu ra thông qua một hàm có thể phân biệt được
- Mỗi lớp có thể có hoặc không có tham số (ví dụ: CONV/FC có, RELU/POOL thì không)
- Mỗi lớp có thể có hoặc không có siêu tham số bổ sung (ví dụ: CONV/FC/POOL có, RELU thì không)

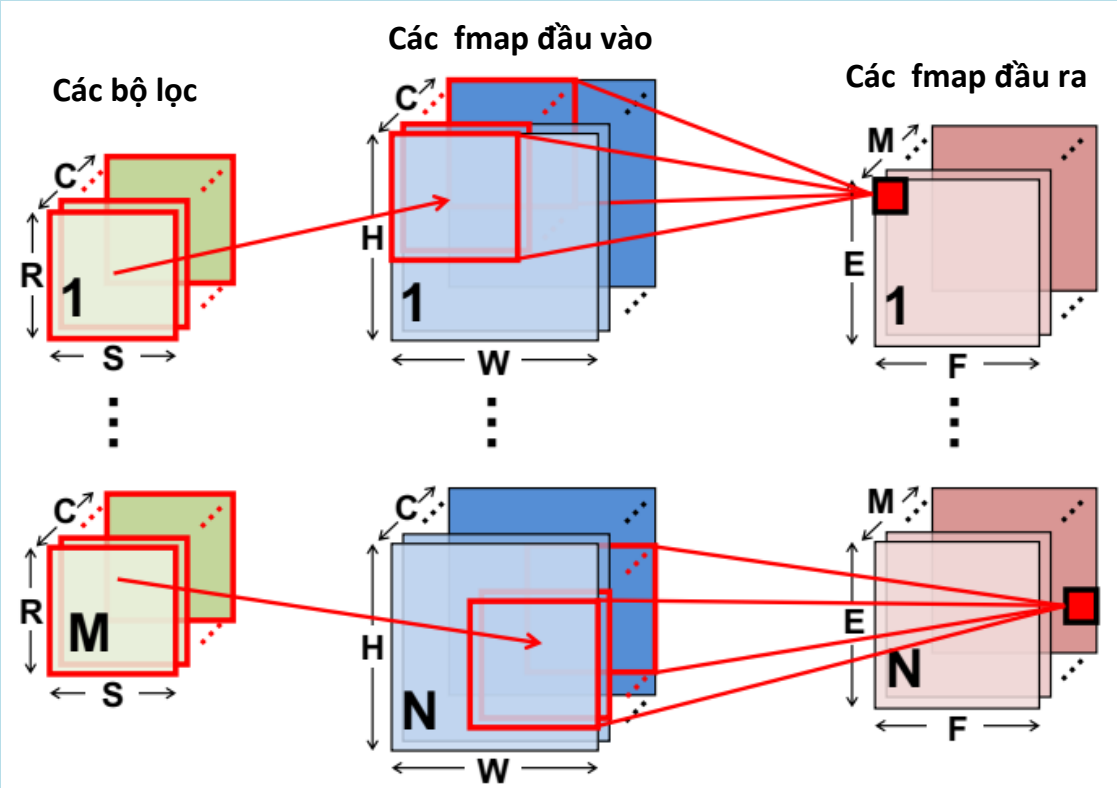
Hình 5.14 cho thí dụ về kiến trúc CNN đơn giản dựa trên CIFAR-10..



Hình 5.14. Ví dụ về kiến trúc CNN đơn giản trên cơ sở CIFAR-10. Khối lượng ban đầu lưu trữ các điểm ảnh ảnh thô (trái) và khối lượng cuối cùng lưu trữ điểm lớp (phải). Mỗi khối lượng kích hoạt dọc theo đường dẫn xử lý được hiển thị dưới dạng một cột. Vì khó hình dung các khối lượng 3D, hình vẽ sắp xếp các lát cắt của mỗi khối lượng thành hàng. Khối lượng lớp cuối cùng lưu trữ điểm cho mỗi lớp, nhưng ở đây chỉ thể hiện 5 điểm số cao nhất đã được sắp xếp và in nhãn của từng điểm. Kiến trúc được hiển thị ở đây là một VGG Net nhỏ.

Hình 5.15 cho thấy các tích chập kích thước cao trong CNN.

Như thể hiện trong hình 5.15, mỗi lớp tích chập (CONV) trong CNN chủ yếu bao gồm các tích chập kích thước cao (High- Dimentional Convolution). Trong tính toán này, các kích hoạt đầu vào của một lớp được cấu trúc thành một tập hợp các bản đồ tính năng đầu vào 2-D (ifmaps: input feature maps), mỗi bản đồ được gọi là một kênh. Mỗi kênh được ghép với một bộ lọc 2-D riêng biệt từ ngăn xếp bộ lọc, một bộ lọc cho mỗi kênh; ngăn xếp bộ lọc 2-D này thường được gọi là một bộ lọc 3-D duy nhất (single 3-D filter). Kết quả của tích chập tại mỗi điểm được tổng hợp (summed) trên tất cả các kênh. Ngoài ra, độ lệch 1-D (1-D bias) có thể được thêm vào kết quả lọc, nhưng một số mạng gần đây loại bỏ việc sử dụng nó khỏi các phần của lớp. Kết quả của tính toán này là các kích hoạt đầu ra bao gồm một kênh của bản đồ tính năng đầu ra (ofmap: output feature map). Các bộ lọc 3-D bổ sung có thể được sử dụng trên cùng một đầu vào để tạo các kênh đầu ra bổ sung. Cuối cùng, nhiều bản đồ tính năng đầu vào (multiple input feature maps) có thể được xử lý cùng nhau như một lô (batch) để có khả năng cải thiện việc tái sử dụng trọng số bộ lọc.



Hình 5.15. Các tích chập kích thước cao trong CNN

Với các tham số hình dạng (Shape Parameter) được cho trong Bảng 5.1, tính toán của một lớp CONV được định nghĩa như sau:

$$\mathbf{O}[z][x][y] = \mathbf{B}[u] + \sum_{k=0}^{C-1}\sum_{i=0}^{S-1}\sum_{j=0}^{R-1} \mathbf{I}[z][k][\mathbf{U}x+i][\mathbf{U}y+j] \times \mathbf{W}[u][k][i][j], \quad (5.1)$$

$0 \le z < N;\ 0 \le u < M;\ 0 \le x < F;\ 0 \le y < E;$

E = (H - R + U)=U; F = (W - S + U)=U.

Bảng 5.1. Các tham số hình dạng của một lớp CONV/FC

Tham số hình dạng (Shape Parameter)	**Mô tả**
N	Kích thước lô của các fmaps
M	Số lượng các bộ lọc 3-D/số lượng các kênh ofmap
C	Số lượng các kênh ifmap/lọc

H/W	Chiều cao/chiều rộng của mặt phẳng ifmap
R/S	Chiều cao/chiều rộng của mặt phẳng bô lọc (=H hoặc W trong FC)
E/F	Chiều cao/chiều rộng của mặt phẳng ofmap (=1 trong FC)

O, **I**, **W** và **B** lần lượt là ma trận của ofmaps, ifmaps, bộ lọc và bias (độ lệc). **U** là một kích thước sải chân nhất định. Hình 5.15 cho thấy hình ảnh trực quan của tính toán này (bỏ qua các độ lệch).

Để đồng bộ thuật ngữ của CNN với DNN chung,

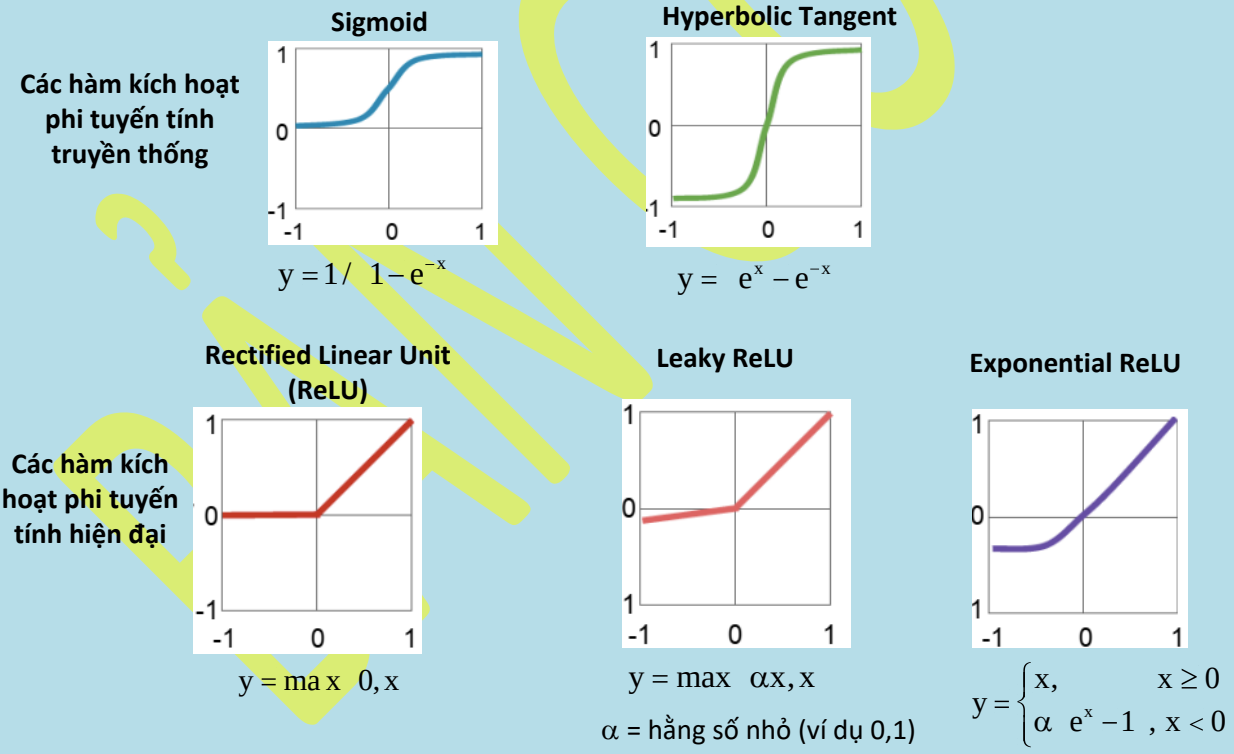
- Các bộ lọc bao gồm các trọng số (tức là synapses: các khớp thần kinh)
- Bản đồ tính năng đầu vào và đầu ra (ifmaps, ofmaps) bao gồm các kích hoạt (tức là các tế bào thần kinh đầu vào và đầu ra
- Sải chân (Stride): Trong tích chập theo sải chân, thay vì dịch một dòng hay một cột bộ lọc mỗi lần, ta có thể dịch nó 2 hay 3 dòng hay cột mỗi lần.

Từ năm đến hơn một nghìn lớp CONV thường được sử dụng trong các mô hình CNN gần đây. Một số lượng nhỏ, ví dụ: 1 đến 3, các lớp kết nối đầy đủ (FC) thường được áp dụng sau các lớp CONV cho mục đích phân loại. Một lớp FC cũng áp dụng các bộ lọc trên các ifmap như trong các lớp CONV, nhưng các bộ lọc có cùng kích thước với các ifmap. Do đó, nó không có thuộc tính chia sẻ trọng lượng của các lớp CONV. Phương trình (5.1) vẫn giữ nguyên cho việc tính toán các lớp FC với một vài ràng buộc bổ sung về các tham số hình dạng: H = R, F = S, E = F = 1 và U = 1.

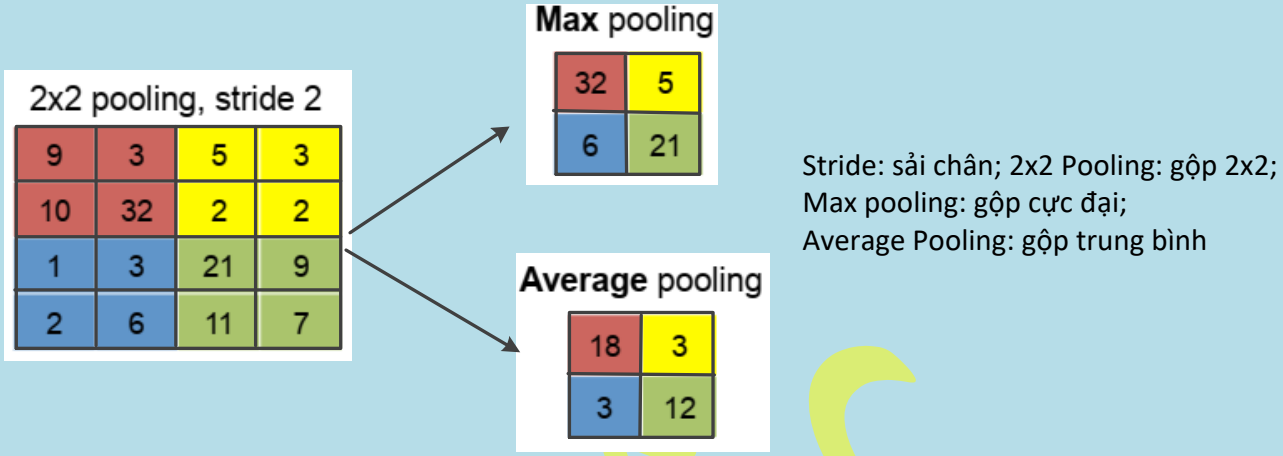
Ngoài các lớp CONV và FC, các lớp tùy chọn khác nhau có thể được tìm thấy trong DNN như phi tuyến tính, gộp và chuẩn hóa. Chức năng và tính toán cho từng lớp này sẽ được trình bày dưới đây.

1) *Phi tuyến tính*: Một hàm kích hoạt phi tuyến tính thường được áp dụng sau mỗi lớp CONV hoặc FC. Các hàm phi tuyến tính khác nhau được sử dụng để đưa tính phi tuyến tính vào DNN như trong Hình 5.16. Chúng bao gồm các hàm phi tuyến thông thường như sigmoid hoặc tiếp tuyến hyperbol cũng như đơn vị tuyến tính chỉnh lưu (ReLU: rectified linear unit) đã trở nên phổ biến trong những năm gần đây do tính đơn giản và khả năng cho phép đào tạo nhanh chóng. Các biến thể của ReLU, chẳng hạn như ReLU rò rỉ (LeakyRLU), ReLU tham số (Parametric RLU) và LU hàm mũ (exponential LU) cũng đã được khám phá để cải thiện độ chính xác. Cuối cùng, một phi tuyến tính được gọi là maxout, lấy giá trị tối đa của hai hàm tuyến tính giao nhau, đã được chứng minh là có hiệu quả trong các nhiệm vụ nhận dạng giọng nói.

2) *Gộp*: Lớp gộp thực hiện chiết suất các giá trị đặc biệt của một tập giá trị thường là giá trị cực đại hoặc giá trị trung bình của tất cả các giá trị. Điều này giảm kích thước của bản đồ tính năng đầu ra. Một loạt các phép tính làm giảm kích thước của bản đồ tính năng được gọi là gộp (Pooling). Pooling, được áp dụng riêng biệt cho từng kênh, cho phép mạng mạnh mẽ và bất biến trước những thay đổi và biến dạng nhỏ. Pooling kết hợp, hoặc gộp, một tập hợp các giá trị trong trường dễ tiếp nhận (receptive field) của nó thành một số giá trị nhỏ hơn. Nó có thể được cấu hình dựa trên kích thước của trường dễ tiếp nhận của nó (ví dụ: 2×2) và hoạt động gộp (ví dụ: tối đa hoặc trung bình), như thể hiện trong Hình 5.17. Thông thường, gộp lại xảy ra trên các khối không chồng lên nhau (tức là sải chân bằng kích thước của gộp). Thông thường một sải bước lớn hơn một được sử dụng sao cho có sự giảm kích thước của biểu diễn (tức là bản đồ tính năng).



Hình 5.16. Các dạng hàm kích hoạt phi tuyến tính khác nhau



Hình 5.17.Các dạng gộp khác nhau

3) *Chuẩn hóa*: Kiểm soát phân phối đầu vào trên các lớp có thể giúp tăng tốc độ đào tạo và cải thiện độ chính xác. Theo đó, sự phân bố của các kích hoạt đầu vào lớp (σ, μ) được chuẩn hóa sao cho nó có giá trị trung bình bằng không và độ lệch chuẩn đơn vị. Trong chuẩn hóa theo lô (BN: Batch Normalization), giá trị chuẩn hóa được chia tỷ lệ và dịch chuyển thêm, như thể hiện trong Phương trình (5.2), trong đó các tham số (γ, β) được học từ đào tạo . ε là một hằng số nhỏ để tránh các vấn đề số (Numerical Problem). Trước đó, chuẩn hóa phản ứng cục bộ (LRN: Local Response Normalization) được sử dụng, được lấy cảm hứng từ sự ức chế bên trong sinh học thần kinh, nơi các tế bào thần kinh bị kích thích (tức là kích hoạt có giá trị cao) sẽ khuất phục các tế bào lân cận của nó (tức là gây ra các kích hoạt có giá trị thấp); tuy nhiên, BN hiện được coi là thông lệ tiêu chuẩn trong thiết kế CNN trong khi LRN hầu như không được dùng nữa. Lưu ý rằng trong khi LRN thường được thực hiện sau hàm phi tuyến tính, BN chủ yếu được thực hiện giữa lớp CONV hoặc FC và hàm phi tuyến tính.

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}}\gamma + \beta \qquad (5.2)$$

Trong các phần tiếp theo ta sẽ phân tích mạng CNN chi tiết hơn.

5.5. LỚP TÍCH CHẬP CONV LAYER)

5.5.1. Tổng quan

Lớp Conv là khối xây dựng cốt lõi của Mạng tích chập, thực hiện hầu hết các nhiệm vụ tính toán nặng.

Trước tiên, hãy thảo luận về những gì lớp CONV tính toán mà không cần sự tương tự giữa não bộ và tế bào thần kinh. Các tham số của lớp CONV bao gồm một tập hợp các bộ lọc có thể học được. Mỗi bộ lọc đều nhỏ về mặt không gian (theo chiều rộng và chiều cao), nhưng mở rộng qua toàn bộ chiều sâu của khối lượng đầu vào. Ví dụ: một bộ lọc điển hình trên lớp đầu tiên của CNNt có thể có kích thước 5x5x3 (tức là chiều rộng và chiều cao là 5 pixel và 3, vì hình ảnh có chiều sâu là 3 (tức là các kênh màu). Trong quá trình chuyển tiếp, chúng ta trượt (chính xác hơn là tích chập) từng bộ lọc trên chiều rộng và chiều cao của khối lượng đầu vào và tính toán tích vô hướng giữa các mục của bộ lọc và đầu vào tại bất kỳ vị trí nào. Khi chúng ta trượt bộ lọc trên chiều rộng và chiều cao của khối lượng đầu vào, chúng ta sẽ tạo ra một bản đồ kích hoạt hai chiều cung cấp các phản hồi của bộ lọc đó tại mọi vị trí không gian. Theo trực giác, mạng sẽ học các bộ lọc được kích hoạt khi chúng nhìn thấy một số kiểu của tính năng trực quan, chẳng hạn như một cạnh có hướng hoặc một đốm màu trên lớp đầu tiên, hoặc cuối cùng là toàn bộ các mẫu hình tổ ong hoặc hình bánh xe trên các lớp cao hơn của mạng. Bây giờ, chúng ta sẽ có một tập hợp đầy đủ các bộ lọc trong mỗi lớp CONV (ví dụ: 12 bộ lọc), và mỗi bộ lọc sẽ tạo ra một bản đồ kích hoạt 2 chiều riêng biệt. Chúng ta sẽ xếp chồng các bản đồ kích hoạt này dọc theo chiều sâu và tạo ra thể tích đầu ra.

Nếu so sánh não bộ/neuron, thì mỗi mục trong khối dữ liệu đầu ra 3D cũng có thể được diễn giải là dữ liệu đầu ra của một neuron chỉ quan sát một vùng nhỏ trong dữ liệu đầu vào và chia sẻ các tham số với tất cả các neuron bên trái và bên phải về mặt không gian (vì tất cả các con số này đều là kết quả của việc áp dụng cùng một bộ lọc).

Bây giờ ta xét về chi tiết của các kết nối neuron, sự sắp xếp của chúng trong không gian và sơ đồ chia sẻ tham số của chúng.

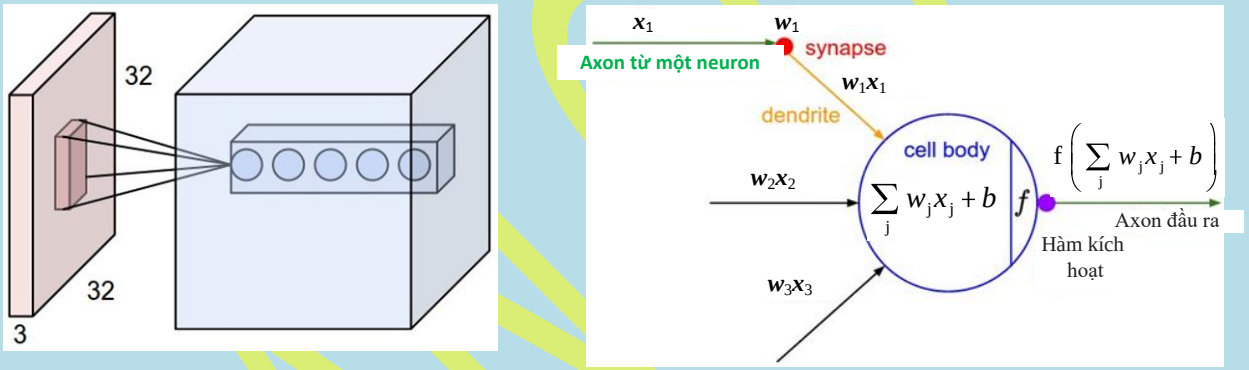
Khi xử lý các dữ liệu đầu vào có chiều cao như hình ảnh, như đã thấy ở trên, việc kết nối các neuron với tất cả các neuron trong khối dữ liệu trước đó là không thực tế. Thay vào đó, chúng ta sẽ kết nối mỗi neuron chỉ với một vùng cục bộ của khối dữ liệu đầu vào. Phạm vi không gian của kết nối này là một siêu tham số được gọi là trường tiếp nhận (Receptive Field) của neuron (tương đương với kích thước bộ lọc). Phạm vi kết nối dọc theo trục độ sâu luôn bằng với độ sâu của khối dữ liệu đầu vào. Điều quan trọng cần nhấn mạnh một lần nữa là tính bất đối xứng này trong cách xử lý các chiều không gian (chiều rộng và chiều cao) và chiều sâu: Các kết nối là cục bộ trong không gian 2D (theo chiều rộng và chiều cao), nhưng luôn đầy đủ theo toàn bộ chiều sâu của khối dữ liệu đầu vào.

Ví dụ 1. Giả sử thể tích đầu vào có kích thước [32×32×3] (ví dụ ảnh RGB CIFAR-10).

Nếu trường tiếp nhận (hoặc kích thước bộ lọc) là 5×5, thì mỗi neuron trong lớp tích chập sẽ có trọng số đến vùng [5×5×3] trong thể tích đầu vào, tổng cộng là 5×5×3 = 75 trọng số (và tham số bias +1). Lưu ý rằng phạm vi kết nối dọc theo trục độ sâu phải là 3, vì đây là độ sâu của thể tích đầu vào.

Hình 5.20 bên trái cho thấy một thể tích đầu của ví dụ về trong màu hồng (ví dụ: hình ảnh CIFAR-10 32x32x3) và một ví dụ về thể tích neuron trong lớp tích chập đầu tiên. Mỗi neuron trong lớp tích chập chỉ được kết nối với một vùng cục bộ trong thể tích đầu vào về mặt không gian, nhưng dọc theo độ sâu (tức là tất cả các kênh màu). Lưu ý, có nhiều neuron (5 trong ví dụ này) dọc theo độ sâu, tất cả đều nhìn vào cùng một vùng trong đầu vào: các đường nối cột 5 neuron này không biểu diễn các trọng số (tức là 5 neuron này không chia sẻ cùng một trọng số, nhưng chúng được liên kết với 5 bộ lọc khác nhau), chúng chỉ ra rằng các neuron này được kết nối với hoặc nhìn vào cùng một trường tiếp nhận hoặc vùng của khối lượng đầu vào, tức là chúng chia sẻ cùng một trường tiếp nhận nhưng không cùng trọng số.

Hình 5.20 bên phải các neuron từ mạng perceptron vẫn không thay đổi: Chúng vẫn tính toán tích vô hướng của các trọng số của chúng với đầu vào theo sau là một phi tuyến tính, nhưng kết nối của chúng hiện bị giới hạn ở mức cục bộ về mặt không gian.



Hình 5.20. Minh họa thí dụ 1

Ví dụ 2. Giả sử thể tích đầu vào có kích thước [16×16×16]. Khi này nếu sử dụng kích thước của trường tiếp nhận 3×3, thì mọi neuron ytong k[ps tích chập khi này sẽ có tổng số 3×3×3×20=180 kết nối đến thể tích đầu vào. Lưu ý một lần nứa là kết nối là cục bộ trong koong gian 2D (nghĩa là 3×3) nhưng dọc theo chiều sâu (20).

5.5.2. Sắp xếp không gian của các lớp tích chập

Chúng ta đã giải thích về khả năng kết nối của từng neuron trong lớp tích chập với thể tích đầu vào, nhưng vẫn chưa đề cập về số lượng neuron trong thể tích đầu ra hoặc cách chúng được sắp xếp. Ba siêu tham số kiểm soát kích thước của thể tích đầu ra: độ sâu (Deepth), bướcchân (Stride) và đệm số không (Sero-Padđing) . Dưới đây ta sẽ thảo luận về những điều này:

1. Đầu tiên, độ sâu của thể tích đầu ra là một siêu tham số: nó tương ứng với số lượng bộ lọc chúng ta muốn sử dụng, mỗi bộ lọc học cách tìm kiếm một thứ gì đó khác nhau trong đầu vào. Ví dụ, nếu lớp tích chập đầu tiên lấy ảnh thô làm đầu vào, thì các neuron khác nhau dọc theo chiều sâu có thể được kích hoạt khi có các cạnh được định hướng (Oriented Edges) khác nhau, hoặc các đốm màu (Blods of Color). Chúng ta sẽ gọi một tập hợp các neuron đều nhìn vào cùng một vùng đầu vào là một cột độ sâu (một số người cũng thích dùng thuật ngữ *sợi* (Fibre)).
2. Thứ hai, chúng ta phải xác định sải chân mà bộ lọc sẽ trượt. Khi sải chân là 1, chúng ta di chuyển bộ lọc từng pixel một. Khi sải chân là 2 (hoặc hiếm khi là 3 hoặc hơn, mặc dù điều này hiếm khi xảy ra trong thực tế) thì bộ lọc sẽ nhảy 2 pixel một khi chúng ta trượt chúng. Điều này sẽ tạo ra thể tích đầu ra nhỏ hơn về mặt không gian.
3. Như chúng ta sẽ sớm thấy, đôi khi việc thêm số 0 vào thể tích đầu vào xung quanh đường viền sẽ rất tiện lợi. Kích thước của phần đệm số 0 này là một siêu tham số. Tính năng tuyệt vời của việc đệm số 0 là nó cho phép chúng ta kiểm soát kích thước không gian của thể tích đầu ra (phổ biến nhất, như chúng ta sẽ thấy, là sử dụng nó để giữ nguyên kích thước không gian của thể tích đầu vào sao cho chiều rộng và chiều cao của đầu vào và đầu ra bằng nhau).
4. Ta có thể tính kích thước không gian của thể tích đầu ra như là một hàm của kích thước thể tích đầu vào W, kích thước bộ lọc F (Kích thước của trường tiếp nhận), sải chân S và đệm P như sau:

$$\frac{W - F + 2P}{S} + 1$$

5. Đối với từng lớp tích chập trong mạng, sẽ có các trọng số hoặc các tham số liên kết với từng neuron trong lớp này. Đối với lớp ℓ-1 nối đến lớp ℓ, với các độ sâu $D_{\ell-1}$ và D_ℓ tương ứng, và kích thước bộ lọc F, số lượng các tham số cần cho một lớp tích chập là::

Số lượng tham số = các trọng số kết nối + số lượng các trọng số độ lệch

$$= F \times F \times D_{\ell-1} \times D_\ell + D_\ell$$

Hình 5.21 minh họa về sự sắp xếp không gian. Trong ví dụ này chỉ có một chiều không gian (trục x), một neuron có kích thước trường tiếp nhận là F = 3, kích thước đầu vào là W = 5 và không có phần đệm nào là P = 1.

Bên trái: Sải chân S = 1, kích thước thể tích đầu ra sẽ là:

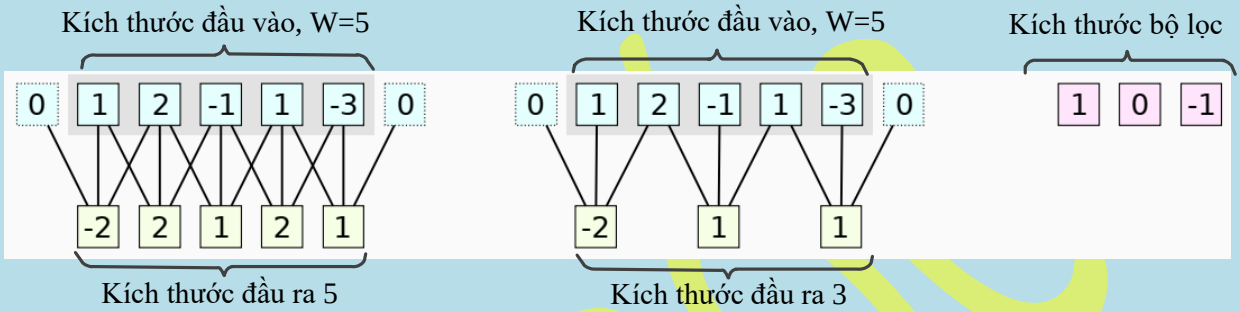
$$\frac{5 - 3 + 2.1}{1} + 1 = 5$$

Bên phải: sải chân S = 2, kích thước đầu ra sẽ là:

$$\frac{5 - 3 + 2.1}{2} + 1 = 3$$

Lưu ý rằng không thể sử dụng sải chân S = 3 vì nó sẽ không vừa khít trên toàn bộ thể tích. Về mặt phương trình, điều này có thể được xác định vì (5 - 3 + 2) = 4 không chia hết cho 3.

Trọng số của neuron trong ví dụ này là [1,0,-1] (hiển thị ở bên phải) và độ lệch của nó là 0. Các trọng số này được chia sẻ trên tất cả các neuron màu vàng (xem chia sẻ tham số bên dưới).



Hình 5.21 **Minh họa về sự sắp xếp không gian**

5.5.3. Sử dụng zero-padding

Trong ví dụ trên bên trái, lưu ý rằng chiều đầu vào là 5 và chiều đầu ra bằng nhau: cũng bằng 5. Điều này hoạt động như vậy vì các trường tiếp nhận của chúng ta là 3 và chúng ta đã sử dụng zero-padding là 1. Nếu không sử dụng zero-padding, thì thể tích đầu ra sẽ chỉ có chiều không gian là 3, vì đó là số lượng neuron sẽ "khớp" với đầu vào ban đầu. Nhìn chung, việc thiết lập zero-padding là$P = (F − 1) / 2$khi bước sải chân là$S = 1$đảm bảo rằng thể tích đầu vào và thể tích đầu ra có cùng kích thước về mặt không gian. Việc sử dụng zero-padding theo cách này rất phổ biến và chúng ta sẽ thảo luận đầy đủ lý do khi tìm hiểu thêm về kiến trúc ConvNet.

5.5.4. Ràng buộc về sải chân

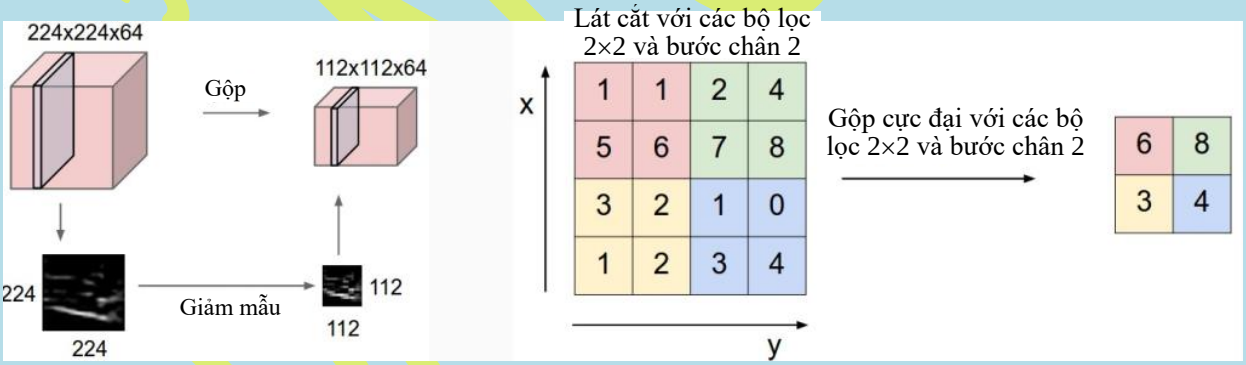
Lưu ý một lần nữa rằng các siêu tham số sắp xếp không gian có ràng buộc lẫn nhau. Ví dụ, khi đầu vào có kích thước$W = 10$, không sử dụng phần đệm số không$P = 0$và kích thước bộ lọc là$F = 3$, khi đó sẽ không thể sử dụng$S = 2$, từ$(W − F + 2P) / S + 1 = (10 − 3 + 0) / 2 + 1 = 4,5$, tức là không phải là một số nguyên, cho thấy các neuron không "khớp" chặt chẽ và đối xứng trên toàn bộ đầu vào. Do đó, việc thiết lập siêu tham số này được coi là không hợp lệ, và thư viện CNN có thể đưa ra một ngoại lệ hoặc thêm số 0 vào phần còn lại để khớp, hoặc cắt bớt đầu vào để khớp, hoặc tương tự. Đối với kiến trúc CNN, việc định cỡ CNN sao cho tất cả các chiều "ăn khớp" có thể thực sự là một vấn đề nan giải, nhưng việc sử dụng thêm số 0 và một số hướng dẫn thiết kế sẽ giúp giảm thiểu đáng kể vấn đề này.

Ví dụ thực tế. Kiến trúc Krizhevsky và cộng sự đã giành chiến thắng trong cuộc thi ImageNet năm 2012 đã chấp nhận hình ảnh có kích thước [227×227×3]. Trên tớp Tích chập đầu tiên, nó sử dụng các tế bào thần kinh có kích thước trường tiếp nhận (kích thước bộ lọc) F= 11, sải chân S= 4 và không có đệm số không P= 0. Vì (227 - 11)/4 + 1 = 55, và vì lớp tích chập có độ sâu là D= 96, khối lượng đầu ra của lớp tích chập có kích thước [55×55×96]. Mỗi neuron trong số 55×55×96 neuron trong thể tích này này được kết nối với một vùng có kích thước [11×11×3] trong thể tích đầu vào. Hơn nữa, tất cả 96 neuron trong mỗi cột độ sâu đều được kết nối với cùng một vùng [11×11×3] của đầu vào, nhưng tất nhiên là có trọng số khác nhau.

5.6. CÁC LỚP GỘP VÀ CÁC LỚP KHÁC

5.6.1. Các lớp gộp

Chức năng của một lớp gộp, còn được gọi là lớp giảm mẫu (Subsampling) là giảm mạnh kích thước không gian của thể hiện, để giảm số lượng các tham số trong mạng. Như đã xét trước đây, quá trình gộp thường bao gồm các hàm trung bình và cực đại được áp dụng cho tập con được xét (hình 5.21).



Hình 5.21. Gộp sử dụng hàm cực đại với các bộ lọc 2×2

5.6.2. Gộp chồng lấn cực đại (Overlapping MAX-POOL

Gộ chồng lấn cực đại cũng giống như gộp cực đại, ngoại trừ các cửa số gần nau mà trên đó gộp cực đại được tính toán bị chồng lấn lên nhau. Cồng lấn gộp cho phép giảm sai số top- 1 0,4% và sai số top-5 0,3% khi so sánh gộp không chồng lấn cho trường hợp kích thước cửa sổ 2×2 sải chân 2.

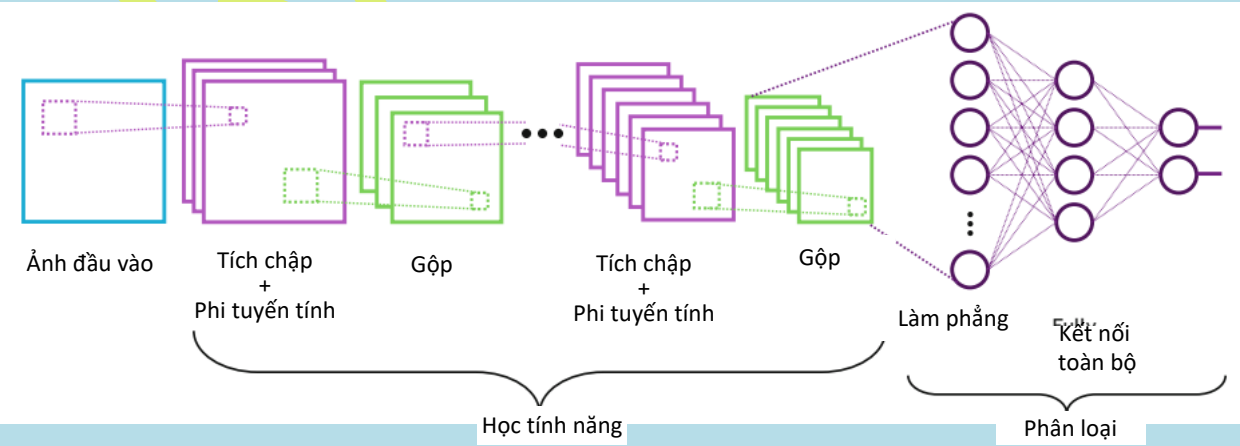
5.6.3. Các lớp khác

Ngoài các lớp trên, CNN có các lớp khác:

- Các lớp chuẩn hóa. Rất nhiều kiểu của các lớp chuẩn hóa đã được đề xuất, chẳng hạn cho mục đích thược hiện các sơ đồ ức chế quan sát được trong não bộ sinh học. Batchnorm là một kiểu phổ biến.
- Các lớp kết nối toàn bộ. Các neuron trong lớp kiết nối toàn bộ có toàn bộ các kết nối đến tất cả các kích hoạt trong lớp trước. Các lớp này thường được sử dụng trong các lớp đầu ra của các mạng CNN.
- Các lớp bỏ học (Dropout). Dropout là một kỹ thuật chính quy hóa để giảm quá khớp. Droupout bao gồm chỉ kiết nối ngẫu nhiên đến một tập con các neuron trong các bước truyền lan ngược.

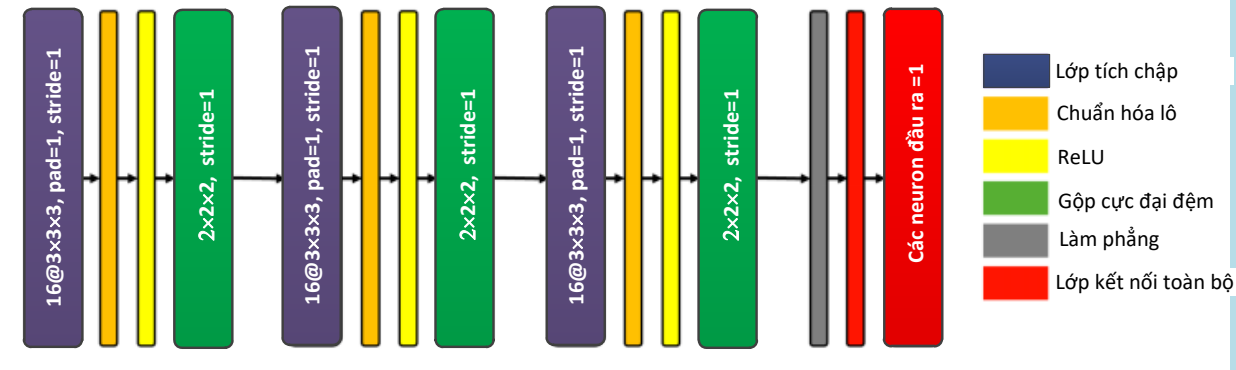
5.7. KIẾN TRÚC CƠ SỞ CỦA CNN

Ta hãy bắt đầu bằng một CNN đơn giản nhất gần giống với CNN gốc được đề xuất bởi Leun và một số tác giả khác, đôi khi được gọi là LeNET. Kiến trúc này gồm hai phần: phần thứ nhất dựa trên các phép toán tích chập và học các tính năng đối với ảnh và phần thứ hai làm phẳng các tính năng và đưa chúng đến một tập các lớp kết nối toàn bộ (nói một các khác, một perceptron đa lớp) để thực hiện phân loại/hồi quy (hình 5.22). Lưu ý rằng, tất nhiên, toàn bộ mạng được đào tạo đầu cuối đến đầu cuối: hai phần không được đào tạo độc lập. Trong phần thứ nhất ta có thể kết hợp một chuỗi khối bao gồm một phép toán tích chập (có thể là theo sải chân hoặc được dãn nở), một hàm kích hoạt phi tuyến tính (chẳng hạn ReLU) và một phép toán gộp. Ngoài ra cũng nên có cả hình vẽ các lớp khác nhau của kiến trúc được chọn.



Hình 5.22. Một kiến trúc CNN cơ sở. Kinh điển, kiến trúc gồm hai phần. Phần thứ nhất thực hiện học tính năng bằng các sử dụng các phép toán tích chập. Sau đó các tính năng được làn phẳng và đưa đến một tập các lớp được kết nối toàn bộ (nghĩa là một perceptron đa lớp), để thực hiện nhiệm vụ phân loại hoặc hồi quy.

Tiếc rằng không có định dạng hài hòa cho một mô tả như vậy. Một thí dụ được cung cấp trong hình 5.23.



Stride: sải chân, Pad: đệm lót

Hình 5.23. Mô tả một kiến trúc CNN. Kinh điển, kiến trúc này gồm hai phần chính. Trên hình này, 16@3×3×3 nghĩa là 16 bản đồ tính năng với kenel tích chập 3×3×3 sẽ được tính toán. Đối với phép toán gộp, kích thước kenel cũng được đề cập (2×2). Cuối cùng sải chân được chỉ ra một cách có hệ thống

Một trong số các kiến trúc đầu tiên tuân theo hình mẫu này là kiến trúc AlexNet. AleNets là một trong số các bài báo đầu tiên chỉ ra theo kinh nghiệm là hàm kích hoạt ReLU khiến hội tụ của CNN nhanh hơn so với các hàm phi tuyến tính khác, chẳng hạn hàm tanh. Hơn nữa, nó là kiến trúc đầu tiên đạt được tỷ lệ lỗi top-5 (Top-5 Error Rate) trên tập dữ liệu ImageNet, vượt trội hiệu suất hơn tất cả các phương pháp khác tại chuẩn mực này với biên độ lớn (khoảng 10%). Trước AlexNet, Các phương pháp hiệu suất nhất sử dụng các tính năng được chiết suất trước và học máy kinh điển. Sau tiến bộ này, học sâu nói chung và cụ thể các CNN nói riêng đã trở thành các hướng nghiên cứu tích cực để xử lý các vấn đề thị giác máy tính khác nhau. Điều này dẫn đến việc đưa ra nhiều dạng kiến trúc, chẳng hạn VGG16 được báo cáo tỷ lệ lỗi 7,3% trên ImageNet với việc đưa ra một

số thay đổi, chẳng hạn các bộ lọc kenel nhỏ hơn. Tiếp sau các tiến bộ này, và thậm chi nếu đã có nhiều nhiều các kiến trúc khác nhau được đề xuất trong giai đoạn này, ta có thể nhắc đến kiến trúc tiếp nhận (Inception) là một trong các kiến trúc sâu nhất của thời kỳ này cho phép giảm hơn nhữa tỷ lệ lỗi xuống 6,7%. Một trong các đặc trưng chính của kiến trúc này là các module tiếp nhận (Inceptive Module) áp dụng nhiều bộ lọc kenel có các kích thước khác nhau tại từng mức của kến trúc. Để giải quyết vấn đề các gradient biến mất (Vanishing Gradient), các tác giả đưa ra các bộ phân loại phụ nối đến các lớp trung gian với kỳ vọng khuyến kích phân biệt trong các tầng thấp hơn và tăng tín hiệu gradient được truyền ngược và cung cấp chính quy hóa bổ sung. Trong quá trình suy luận các bộ phân loại này được loại bỏ hoàn toàn.

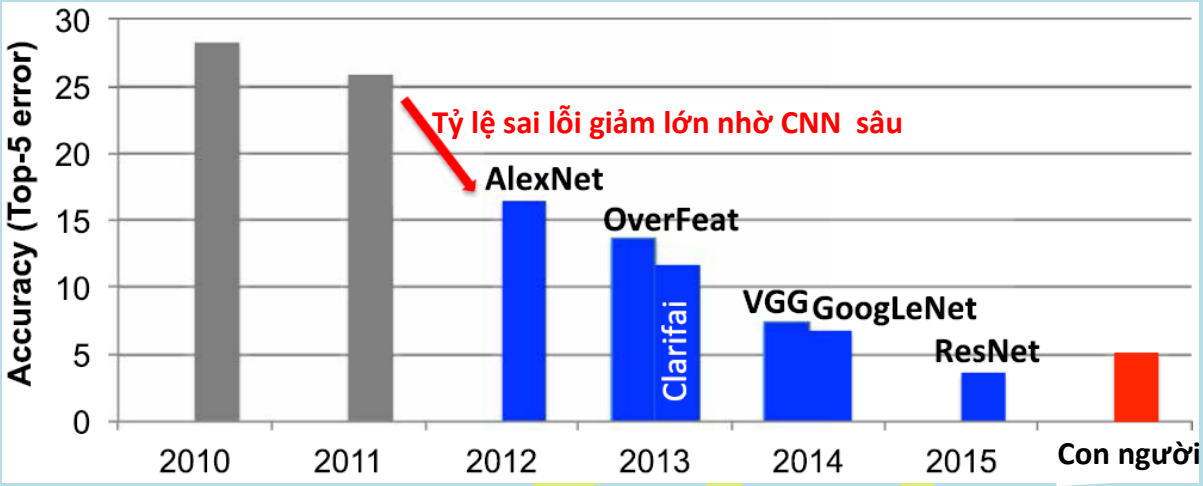
5.8. THỬ THÁCH IMAGENET CHALLANGE

Nhiều mô hình CNN đã được phát triển trong các thập kỷ qua. Mỗi mô hình này có một 'kiến trúc mạng' khác nhau về số lớp, loại lớp, hình dạng lớp (tức là kích thước bộ lọc, số kênh và bộ lọc) và kết nối giữa các lớp. Hiểu được những biến thể và xu hướng này là rất quan trọng để kết hợp tính linh hoạt phù hợp trong bất kỳ công cụ CNN hiệu quả nào.

Phần này sẽ cung cấp một cái nhìn tổng quan về các CNN phổ biến khác nhau như LeNet cũng như những CNN đã cạnh tranh và / hoặc giành chiến thắng trong Thử thách ImageNet (ImageNet Challenge) như trong hình 5.24, hầu hết các mô hình có trọng số được đào tạo trước đều có sẵn công khai để tải xuống;

Hình 5.25 cho thấy hiệu suất của các người tham gia tốt nhất trong cuộc thi ImageNet Challenge trong nhiều năm. Ta có thể thấy rằng, độ chính xác của các thuật toán ban đầu có tỷ lệ lỗi từ 25% trở lên. Năm 2012, một nhóm từ Đại học Toronto đã sử dụng các đơn vị xử lý đồ họa (GPU: Graphics Processing Unit, bộ xử lý đồ họa) do chúng có khả năng tính toán cao và cách tiếp cận mạng neuron sâu, được đặt tên là AlexNet, và đã giảm tỷ lệ lỗi khoảng 10%. Thành tích của họ đã truyền cảm hứng cho một loạt các thuật toán phong cách học sâu dẫn đến một loạt các cải tiến đều đặn. Cùng với xu hướng tiếp cận học sâu cho Thử thách ImageNet Chhallenge, đã có sự gia tăng tương ứng về số lượng người tham gia sử dụng GPU. Từ năm 2012 khi chỉ có 4 người tham gia sử dụng GPU cho đến năm 2014 khi gần như tất cả (110) người tham gia sử dụng GPU. Điều này phản ánh sự chuyển đổi gần như hoàn toàn từ các phương pháp tiếp cận thị giác máy tính truyền thống sang các phương pháp tiếp cận dựa trên học sâu cho cuộc thi. Năm 2015, chiến thắng người tham gia ImageNet Challenge, ResNet, vượt quá độ chính xác ở cấp độ con người với tỷ lệ lỗi top 5 dưới 5%. Kể từ đó, tỷ lệ lỗi đã giảm xuống dưới 3% và hiện

đang tập trung nhiều hơn vào các thành phần khó khăn hơn của cuộc thi, chẳng hạn như phát hiện và định vị đối tượng. Những thành công này rõ ràng là một yếu tố góp phần tiến tới phạm vi rộng các ứng dụng trng đó các DNN được áp dụng.



Accuracy (Top-5 error): độ chính xác (sai lỗi top-5)

Hình 5.24. Cải thiện phân loại hình ảnh được thực hiện bởi các mô hình CNN

Các mô hình CNN được tóm tắt trong Bảng 5.2 Hai kết quả cho kết quả lỗi top 5 (Top 5 Error) được báo cáo. Trong hàng đầu tiên, độ chính xác được tăng cường bằng cách sử dụng nhiều crop (mùa vụ) từ hình ảnh và một tập hợp nhiều mô hình được đào tạo (tức là CNN cần được chạy nhiều lần); những kết quả này đã được sử dụng để cạnh tranh trong Thử thách ImageNet Challenge. Hàng thứ hai báo cáo độ chính xác nếu chỉ sử dụng một mùa vụ duy nhất (tức là DNN chỉ được chạy một lần), điều này phù hợp hơn với những gì có thể được triển khai trong các ứng dụng thời gian thực và/hoặc hạn chế năng lượng.

Bảng 5.2. Tổng kết các CNN phổ biến, độ chính xác được đo trên Top-5 Error trên ImageNet

Các số đo	LeNet 5	AlexNet 5	Overfeat fast	VGG 16	GoogLeNet v1	Resnet 50
Sai số top-5	n/a	16,4	14,2	7,4	6,7	5,3
Sai số top-5 (single crop)*	n/a	19,8	17,0	8,8	10,7	7,0
Kích thước đầu vào	28×28	277×277	231×231	224×224	224×224	224×224
Số lớp CONV	2	5	5	13	57	53
Độ sâu trong các lớp CONV	2	5	5	13	22	49
Các kích thước	5	3, 5, 11	3, 5, 11	3	1, 3, 5, 7	1, 3, 7

bộ lọc						
Số kênh	1, 20	3-256	3-1024	3-512	3-832	2-2048
Số bộ lọc	20, 50	96- 384	96-1024	64-512	16-384	64-2048
Sải chân	1	1,4	1,4	1	1,2	1,2
Các trọng số	2,6k	2,3M	16M	14,7M	6,0M	23,5M
Các MAC*	283k	666M	2,67G	15,3G	1,43G	3,86G
Số lớp FC	2	3	3	3	1	1
Các kích thước bộ lọc	1, 4	1, 6	1, 6, 12	1, 7	1	1
Số kênh	50,500	256-4096	1024-4096	512-4096	1024	2048
Số bộ lọc	10, 500	1000-4096	1000-4096	1000-4096	1000	1000
Các trọng số	58k	58,6M	130M	124M	1M	2M
Các MAC	58k	58,6M	130M	124M	1M	2M
Tổng trọng số	60k	61M	146M	138M	7M	25,5M
Tổng MAC	341k	724M	2,8G	15,5G	1,43G	3,9G

MAC (Multiply-and- Accumlate): nhân và tích lũy

- *Phiên bản LetNet-5 này có 431K trọng số và yêu cầu 2,3M MAC (Multiply-and-Accumlate: nhân và tích lũy) trên một ảnh, sử dụng ReLU thay vì Sigmoid.*

Các Website của mô hình được đào tạo trước:

- *"Caffe LeNet MNIST," http://caffe.berkeleyvision.org/gathered/ examples/mnist.html.*
- *"Caffe Model Zoo," http://caffe.berkeleyvision.org/model zoo. html.*
- *"MatCNN Pretrained Models," http://www.vlfeat.org/ matCNN/pretrained/.*
- *"TensorFlow-Slim image classification library," https://github. com/tensorflow/models/tree/master/slim.*

Trong các phần dưới đây ta sẽ xét cụ thể các mô hình CNN nói trên.

5.9. LENET-5

LeNet là một trong những cách tiếp cận đầu tiên của CNN được giới thiệu vào năm 1989. Nó được thiết kế cho nhiệm vụ phân loại chữ số trong hình ảnh thang độ xám (Grayscale) có kích thước 28×28. Phiên bản nổi tiếng nhất, LeNet-5, chứa hai lớp CONV và hai lớp FC. Mỗi lớp CONV sử dụng các bộ lọc có kích thước 5×5 (1 kênh cho mỗi bộ lọc) với 6 bộ lọc ở lớp đầu tiên và 16 bộ lọc ở lớp thứ hai. Gộp trung bình 2×2 được sử dụng sau mỗi tích chập và một sigmoid được sử dụng cho tính phi tuyến tính. Tổng cộng, LeNet yêu cầu 60k trọng số và 341k nhân và tích lũy (MAC: Multiply-and-Accumlate) cho mỗi

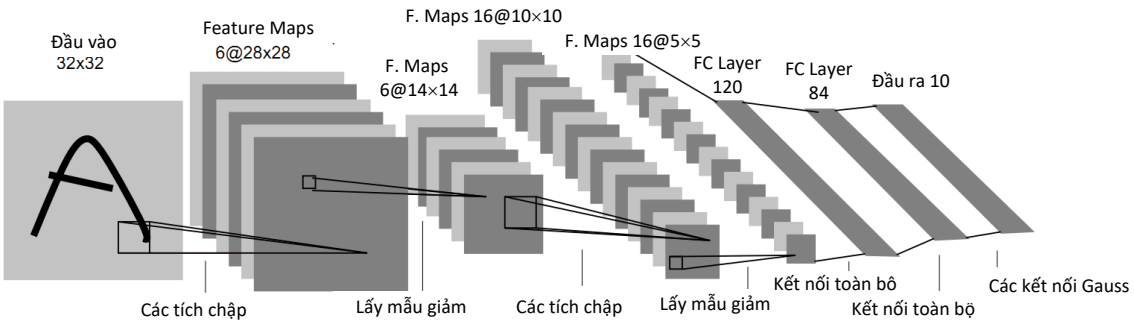
hình ảnh. LeNet đã dẫn đến thành công thương mại đầu tiên của CNN, vì nó được triển khai trong các máy ATM để nhận dạng các chữ số cho các khoản tiền gửi séc.

LeNet-5 là một CNN để phân loại tập dữ liệu MNIST. Tập dữ liệu MNIST (Modi• ed National Institute of Standards and Technology) bao gồm một tập đào tạo 60 000 ảnh các các chữ số viết tay từ 0 đến 9 và một tập kiểm ta cách biệt gồm 10. 000 chữ số viết tay. Vấn đề phân loại bao gồm phân loại các ảnh đầu vào $28 \times 28 \times 1$ vào 10 loại (hình 5.25).

Hình 5.25. Phân loại của MNIST

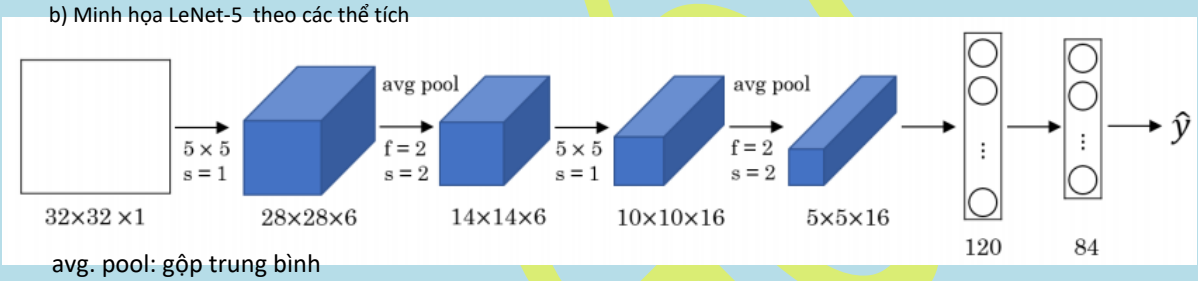
Hình 5.26 cho kiến trúc của LeNet.

a) LeNet-5 cho nhận dạng một chữ viết tay



F. Maps (Feature Maps): các bản đồ tính năng. Mỗi mặt phẳng là một bản đồ tính năng (một tập các đơn vị có các trọng số bị buộc phải giống nhau). FC Layer (Full Connected Layer): lớp kết nối toàn bộ

b) Minh họa LeNet-5 theo các thể tích



avg. pool: gộp trung bình

Hình 5.26. Kiến trúc của LeNet-5

LeNet-5 bao gồm 7 lớp (ngoại trừ đầu vào), mỗi lớp chứa các thông số có thể đào tạo được; mỗi lớp chứa một tập các bản đồ tính năng, đặc trưng của từng bản đồ tính năng đàu vào được chiết xuất bằng các bộ lọc tích chập và mỗi bản đồ tính năng có nhiều neuron

Bảng 5.2. tổng kết các phần tử trong LeNet-5

Lớp		Bản đồ tính năng	Kích thước	Kích thước bộ lọc	Sải chân	Hàm kích hoạt
Đầu vào	Ảnh	1	32×32	-	-	-
1	Tích chập	6	28×28	5×5	1	tanh
2	Gộp trung bình	6	14×14	2×2	2	Tanh
3	Tích chập	16	10×10	5×5	1	Tanh

4	Gộp trung bình	6	5×5	2×2	2	Tanh
5	FC	-	120	-	-	Tanh
6	FC	-	84	-	-	tanh
Đầu ra	FC	-	10	-	-	softmax

Từ hình 5.26 ta thấy:

- Mạng bao gồm hai tích chập, hai lớp gộp và hai lớp kết nối toàn bộ
- Cả hai sigmoid (σ) và tangent hyperbol (Tanh) được sử dụng
- Gộp trung bình được sử dụng.

Conv1 (tích chập 1):

- Kích thước thể tích đầu vào w=32
- Kích thước bộ lọc: F=5
- Đệm P=0
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{32-5+2\times 0}{1}+1=28$
- Độ sâu (số lượng bộ lọc) D=6

Vì thế kích thước của Conv1 sẽ là 28×28×6.

Pool 2 (gộp 2)

- Kích thước thể tích đầu vào 28
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục

Vì thế kích thước của Pool2 sẽ là 14×14×6.

Conv3 (tích chập 3):

- Kích thước thể tích đầu vào w=14
- Kích thước bộ lọc: F=5

BẢN GỐC

- Đệm P=0
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{14-5+2.0}{1}+1=10$
- Độ sâu (số lượng bộ lọc) D=6

Vì thế kích thước của Conv2 sẽ là $10\times10\times6$.

Pool4 (gộp 4)

- Kích thước thể tích đầu vào 10
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục

Vì thế kích thước của Pool2 sẽ là $5\times5\times6$.

FC1:

- Kích thước đầu vào là $5\times5\times6=400$
- Kích thước đầu ra là 120

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 400 đầu vào đến 120 đầu ra.

FC2:

- Kích thước đầu vào là 120
- Kích thước đầu ra là 84

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 120 đầu vào đến 84 đầu ra.

- Kích thước đầu vào là 84
- Kích thước đầu ra là một phần tử trên một giá trị phân loại có thể có, nên kích thước đầu ra sẽ là 10

Cuối cùng lớp đầu ra kết quả là một lớp FC khác.

Các trọng số có thể học

Số lượng tham số = các trọng số kết nối + số lượng các trọng số độ lệch

BẢN GỐC

$$= F \times F \times D_{\ell-1} \times D_{\ell} + D_{\ell}$$

$$= 5 \times 5 \times 1 \times 6 + 6$$

Giả sử ta có các thể tích đầu ra 28×28, số lượng các tham số có thể học là $28 \times 28 \times 156 = 122\ 304$.

Tổng số ta có 340 000 tham số trong LeNet-5. Không nhiều đối với ngày nay, nhưng rất ấn tượng trong năm 1998 khi LeNet ra đời.

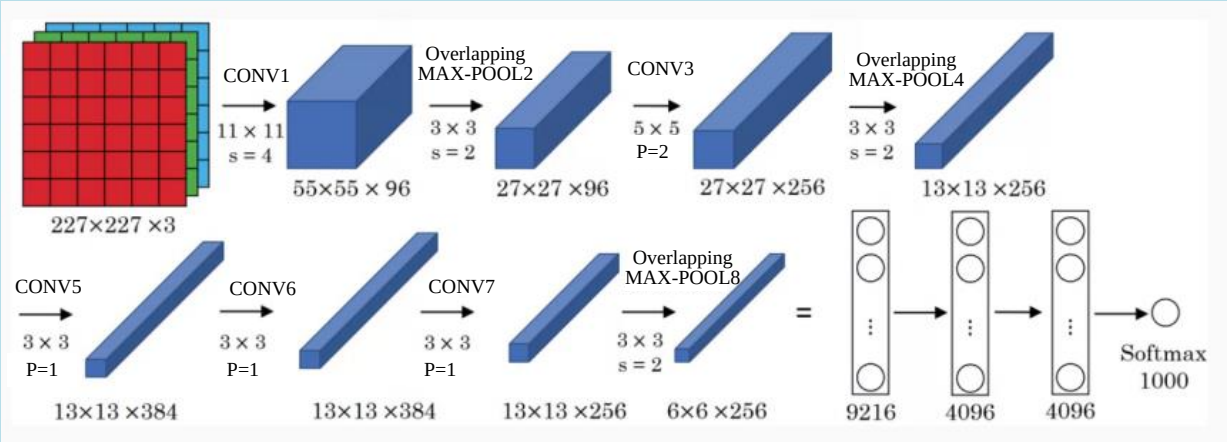
5.10. AlExNET VÀ Overfeat

5.10.1. AlexNET

AlexNet là một CNN được giới thiệu và năm 2012 bởi Krizhevsky và các cộng sự, đã chiến thăng cuộc đấu ImaeNet lọt vào top 5 sai số với biên độ sai số 16,4% so với á quân 26 % (xem hình 5.24).

Nó bao gồm năm lớp CONV và ba lớp FC. Trong mỗi lớp CONV, có 96 đến 384 bộ lọc và kích thước bộ lọc nằm trong khoảng từ 3×3 đến 11×11, với 3 đến 256 kênh mỗi kênh. Trong lớp đầu tiên, 3 kênh của bộ lọc tương ứng với các thành phần màu đỏ, xanh lá cây và xanh lam của hình ảnh đầu vào. Một phi tuyến tính ReLU được sử dụng trong mỗi lớp. Gộp tối đa 3×3 được áp dụng cho đầu ra của các lớp 1, 2 và 5. Để giảm tính toán, sải bước 4 được sử dụng ở lớp đầu tiên của mạng. AlexNet đã giới thiệu việc sử dụng LRN trong các lớp 1 và 2 trước khi gộp tối đa, mặc dù LRN không còn phổ biến trong các mô hình CNN sau này. Một yếu tố quan trọng phân biệt AlexNet với LeNet là số lượng trọng số lớn hơn nhiều và hình dạng khác nhau giữa các lớp. Để giảm số lượng trọng số và tính toán trong lớp CONV thứ hai, 96 kênh đầu ra của lớp đầu tiên được chia thành hai nhóm gồm 48 kênh đầu vào cho lớp thứ hai, sao cho các bộ lọc trong lớp thứ hai chỉ có 48 kênh. Tương tự, trọng số ở lớp thứ tư và thứ năm cũng được chia thành hai nhóm. Tổng cộng, AlexNet yêu cầu 61 triệu trọng số và 724 triệu MAC để xử lý một hình ảnh đầu vào 227×227.

Kiến trúc mạng được cho trong hình 5.27.



Hình 5.27. Kiến trúc của AlexNet

CONV1:

- Kích thước thể tích đầu vào w=224
- Kích thước bộ lọc: F=11
- Đệm P=0
- Sải chân S=4
- $\frac{W-F+2P}{S}+1=\frac{227-11+2.0}{4}+1=55$
- Độ sâu (số lượng bộ lọc) D=96

MAX-POOL2: Overlapping Max-Pool

- Kích thước thể tích đầu vào 55
- Kích thước bộ lọc 3
- Sải chân 2
- Overlapping MaxPool: (55-3)/2+1=27
- Độ sâu (số lượng bộ lọc) D=96

CONV3: Pad=2

- Kích thước thể tích đầu vào w=27
- Kích thước bộ lọc: F=5
- Đệm P=2
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{27-5+2\times 2}{1}+1=27$
- Độ sâu (số lượng bộ lọc) D=256

MAX-POOL4: Overlapping Max-Pool

- Kích thước thể tích đầu vào 27
- Kích thước bộ lọc 3
- Sải chân 2
- Overlapping MaxPool: (27-3)/2+1=13
- Độ sâu (số lượng bộ lọc) D=256

CONV5: Pad=2

- Kích thước thể tích đầu vào w=13
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{13-3+2\times1}{1}+1=13$
- Độ sâu (số lượng bộ lọc) D=38

CONV6: Pad=2

- Kích thước thể tích đầu vào w=13
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{13-3+2\times1}{1}+1=13$
- Độ sâu (số lượng bộ lọc) D=384

CONV7: Pad=2

- Kích thước thể tích đầu vào w=13
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{13-3+2\times1}{1}+1=13$
- Độ sâu (số lượng bộ lọc) D=256

MAX-POOL8: Overlapping Max-Pool

- Kích thước thể tích đầu vào 13 (13×13)

BẢN GỐC

- Kích thước bộ lọc 3 (3×3)
- Sải chân 2
- Overlapping MaxPool: (13-3)/2+1=6
- Độ sâu (số lượng bộ lọc) D=256

FC1:

- Kích thước đầu vào là 6×6×256=9216
- Kích thước đầu ra là 4096

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 400 đầu vào đến 120 đầu ra.

FC2:

- Kích thước đầu vào là 4096
- Kích thước đầu ra là 4096

FC3:

- Kích thước đầu vào là 4096
- Kích thước đầu ra là 1000

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 120 đầu vào đến 84 đầu ra.

- Kích thước đầu vào là 84
- Kích thước đầu ra là một phần tử trên một giá trị phân loại có thể có, nên kích thước đầu ra sẽ là 10

So sánh với LeNet:

- Hàm ReLu (Rectified Linear Unit) được sử dụng, được cho rằng cho phép hội tụ 6 lần lớn hơn sigmoid và tanh
- Cấu hình topo giống như LeNet, nhưng lớn hơn
- Sử dụng đệm để giữ nguyên kích thước
- Sử dụng gộp cực đại
- Khoảng 60 triệu tham số
- Nhiều GPU được sử dụng trong quá trình học và vì thế các lớp được tách đôi trong mỗi bước

5.10.2. Overfeat

BẢN GỐC

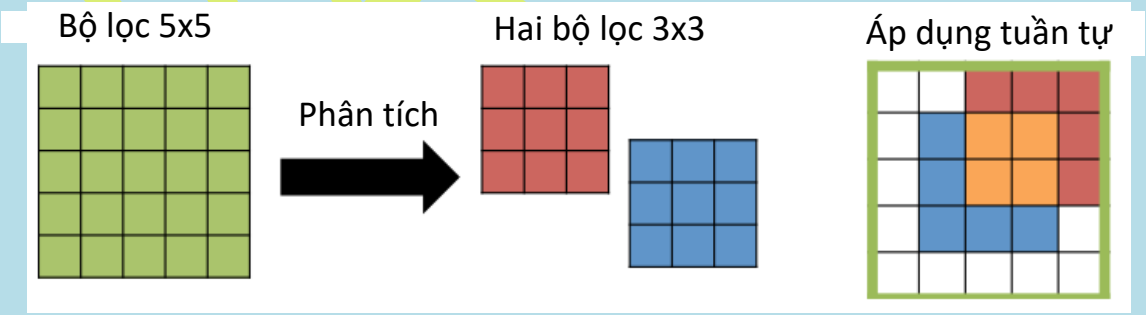
Có kiến trúc rất giống với AlexNet với năm lớp CONV và ba lớp FC. Sự khác biệt chính là số lượng bộ lọc được tăng lên cho các lớp 3 (384 đến 512), 4 (384 đến 1024) và 5 (256 đến 1024), lớp 2 không được chia thành hai nhóm, lớp đầu tiên được kết nối đầy đủ chỉ có 3072 kênh thay vì 4096 và kích thước đầu vào là 231×231 thay vì 227×227. Kết quả là, số lượng trọng số tăng lên 146 triệu và số lượng MAC tăng lên 2,8G cho mỗi hình ảnh. Overfeat có hai mô hình khác nhau: nhanh (được mô tả ở đây) và chính xác. Mô hình chính xác được sử dụng trong Thử thách ImageNet cho tỷ lệ lỗi top 5 thấp hơn 0,65% so với mô hình **Overfeat fast** với chi phí 1,9× MAC nhiều hơn.

5.11. VGG-16 VÀ GoogLeNet

5.11.1. VGG-16

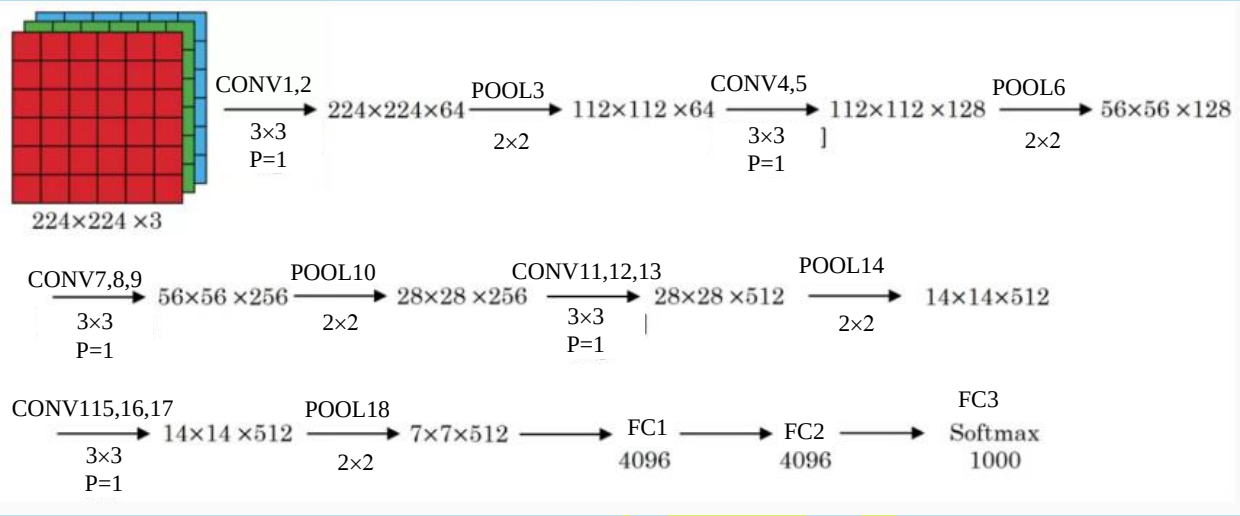
VGG được đề xuất bởi Simonyan và Zisserman. Tồn tại rất nhiều phiên bản của mạng neuron này, VGG-16 là một trong số chúng.

VGG-16 đi sâu hơn đến 16 lớp bao gồm 13 lớp CONV và 3 lớp FC. Để cân bằng chi phí đi sâu hơn, các bộ lọc lớn hơn (ví dụ: 5×5) được xây dựng từ nhiều bộ lọc nhỏ hơn (ví dụ: 3×3), có trọng số ít hơn, để đạt được các trường tiếp nhận (receptive fields) tương tự như trong Hình 5.28. Kết quả là, tất cả các lớp CONV có cùng kích thước bộ lọc là 3×3. Tổng cộng, VGG-16 yêu cầu 138M trọng số và 15,5G MAC để xử lý một hình ảnh đầu vào 224×225. VGG có hai mẫu khác nhau: VGG-16 (được mô tả ở đây) và VGG-19. VGG-19 cho tỷ lệ lỗi top 5 thấp hơn 0,1% so với VGG-16 với chi phí 1,27× MAC nhiều hơn.



Hình 5.28. Cấu trúc hỗ trợ một bộ lọc 5×5 từ các bộ lọc 3×3 trong VGG-16

Hình 5.29 cho thấy kiên trúc của mạng VGG-16.



Hình 5.28. Kiến trúc của mạng VGG-16

CONV1,2: Kích thước bộ lọc [3×3]

- Kích thước thể tích đầu vào w=224
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{224-3+2\times1}{1}+1=224$
- Độ sâu (số lượng bộ lọc) D=64

MAX-POOL3: kích thước bộ lọc [2×2]

- Kích thước thể tích đầu vào 224
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục
- Vì thế kích thước đầu ra sẽ là 112×112×64

CONV4,5: Kích thước bộ lọc [3×3]

- Kích thước thể tích đầu vào w=112
- Kích thước bộ lọc: F=3
- Đệm P=1

- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{112-3+2\times1}{1}+1=112$
- Độ sâu (số lượng bộ lọc) D=128

MAX-POOL6: kích thước bộ lọc [2×2]

- Kích thước thể tích đầu vào 112
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục
- Vì thế kích thước đầu ra sẽ là 56×56×128

CONV7,8,9: Kích thước bộ lọc [3×3]

- Kích thước thể tích đầu vào W=56
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{56-3+2\times1}{1}+1=56$
- Độ sâu (số lượng bộ lọc) D=256

MAX-POOL10: kích thước bộ lọc [2×2]

- Kích thước thể tích đầu vào 56
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục
- Vì thế kích thước đầu ra sẽ là 28×28×256

CONV15,16,17: Kích thước bộ lọc [3×3]

- Kích thước thể tích đầu vào W=28
- Kích thước bộ lọc: F=3
- Đệm P=1

BẢN GỐC

- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{26-3+2\times1}{1}+1=28$
- Độ sâu (số lượng bộ lọc) D=512

MAX-POOL14: kích thước bộ lọc [2×2]

- Kích thước thể tích đầu vào 28
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục
- Vì thế kích thước đầu ra sẽ là 14×14×512

CONV15,16,17: Kích thước bộ lọc [3×3]

- Kích thước thể tích đầu vào W=14
- Kích thước bộ lọc: F=3
- Đệm P=1
- Sải chân S=1
- $\frac{W-F+2P}{S}+1=\frac{14-3+2\times1}{1}+1=14$
- Độ sâu (số lượng bộ lọc) D=512

MAX-POOL14: kích thước bộ lọc [2×2]

- Kích thước thể tích đầu vào 14
- Kích thước bộ lọc 2
- Sải chân 2
- Đối với từng bộ 2×2 pixels ta nhận được một pixel duy nhất, bằng cách chia kích thước thể tích cho hai trên từng trục
- Vì thế kích thước đầu ra sẽ là 7×7×512

FC1:

- Kích thước đầu vào là 7×7×512=25088
- Kích thước đầu ra là 4096

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 25088 đầu vào đến 4096 đầu ra.

BẢN GỐC

FC2:

- Kích thước đầu vào là 4096
- Kích thước đầu ra là 1000

Vì thế lớp kết nối đầy đủ sẽ kết nối tất cả 4096 đầu vào đến 1000 đầu ra theo hàm Softmax.

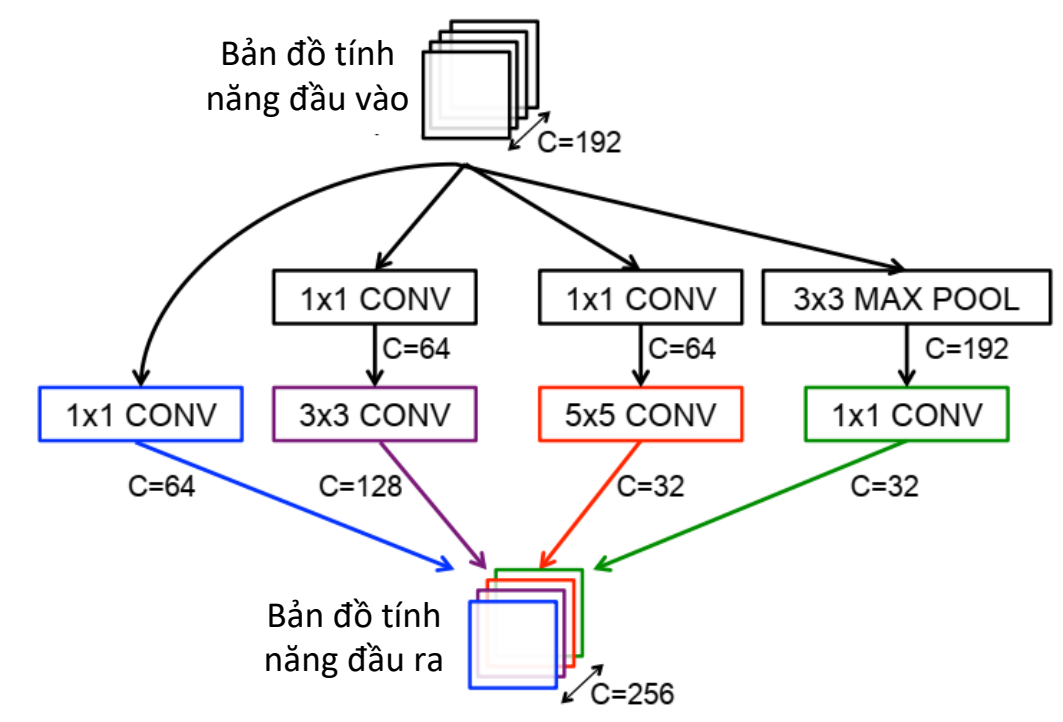
Các đặc điểm của kiến trúc VGG-16 trong hình 5.28:

- Được thiết kế theo các khối mô-dun
  - CONVi,j,k ký hiệu cho ba mạng tích chập i, j,k (độ sâu 64)
  - POOL- giả thiết là gộp cực đại nếu không có đặc tả khác
  - Hàm kích hoạt giả hiết là ReLU nếu không có đặc tả khác
- CONV luôn luôn là bộ lọc 3×3v[í sải chân s=1 và đệm P=1
- POOL luôn là bộ lọc 2×2, với sải chân S=2
- Khoảng 138 triệu tham số
- Thiết kế theo mẫu sau: mọi lớp kết tiếp theo tăng gấp đôi số lượng kênh (chiều sâu) và giảm một nửa độ rộng và độ cao
- Thiết kế theo mẫu: mỗi lớp sau nhân đôi số kênh (độ sâu) và giảm độ một nửa rộng,độ cao.

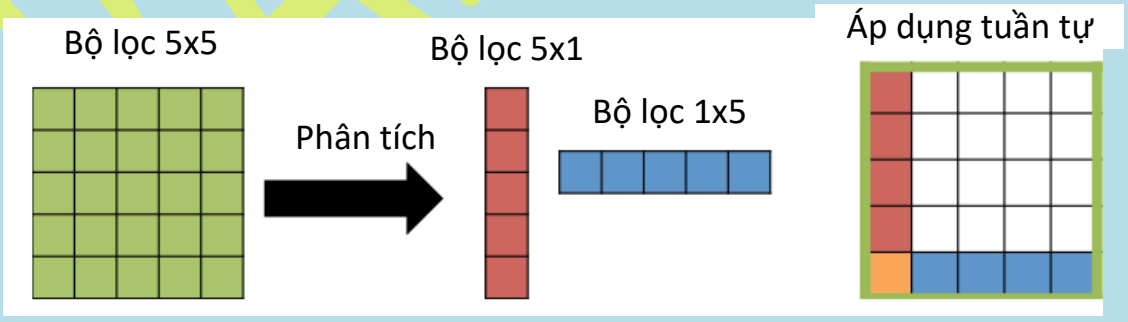
5.11.2. GoogLeNet

GoogLeNet thậm chí còn đi sâu hơn với 22 lớp. Nó giới thiệu một mô-đun khởi động (Inception Module), được hiển thị trên Hình 5.29, bao gồm các kết nối song song, trong khi trước đây chỉ có một kết nối nối tiếp duy nhất. Các bộ lọc có kích thước khác nhau (tức là 1×1, 3×3, 5×5), cùng với gộp tối đa 3×3, được sử dụng cho mỗi kết nối song song và các đầu ra của chúng được móc nối cho đầu ra mô-đun. Sử dụng nhiều kích thước bộ lọc có tác dụng xử lý đầu vào ở nhiều kích thước /tỷ lệ (scale). Để cải thiện tốc độ đào tạo, GoogLeNet được thiết kế sao cho trọng số và kích hoạt, được lưu trữ để lan truyền ngược trong quá trình đào tạo, tất cả đều có thể phù hợp với bộ nhớ GPU. Để giảm số lượng trọng số, bộ lọc 1×1 được áp dụng như một 'nút thắt cổ chai' để giảm số lượng kênh cho mỗi bộ lọc. 22 lớp bao gồm ba lớp CONV, tiếp theo là 9 lớp khởi đầu (Inception Layer), (mỗi lớp sâu hai lớp CONV) và một lớp FC. Kể từ khi được giới thiệu vào năm 2014, GoogleNet (còn được gọi là Inception)

có nhiều phiên bản: v1 (mô tả ở đây), v3 và v4. Inception-v3 phân tách các tích chập bằng cách sử dụng các bộ lọc 1-D nhỏ hơn như trong Hình 5.30 để giảm số lượng MAC và trọng số để đi sâu hơn đến 42 lớp. Kết hợp với chuẩn hóa hàng loạt, v3 đạt được sai số top 5 thấp hơn 3% so với v1 với mức tăng 2,5× tính toán. Inception-v4 sử dụng các kết nối dư, được mô tả trong phần tiếp theo, để giảm 0,4% lỗi.



Hình 5.29. Mô-đun khởi đầu từ GoogleNet với độ dài kênh ví dụ. Lưu ý rằng mỗi lớp CONV được theo sau bởi một ReLU (không vẽ)



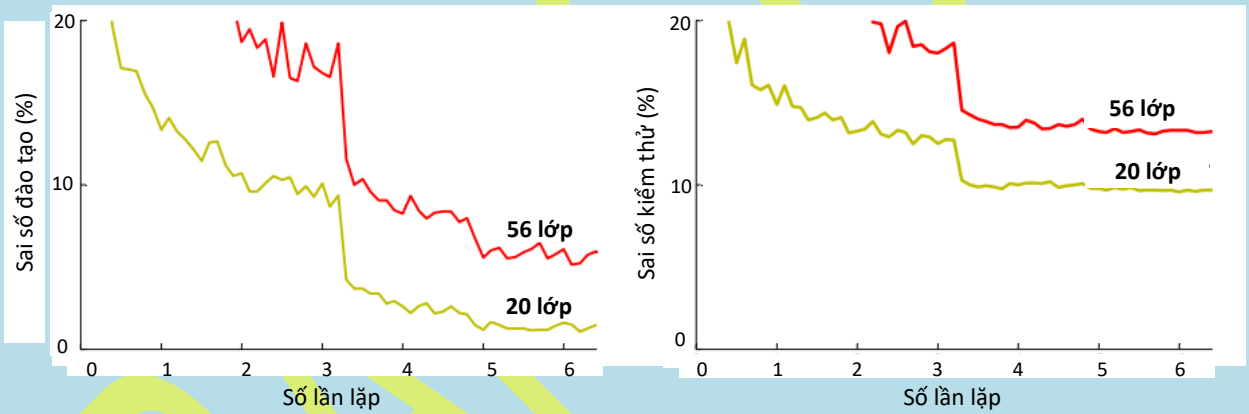
Hình 5.24. Cấu trúc hỗ trợ môt bộ lọc 5x 5 hỗ trợ từ các bộ lọc 1x5 và 5x1 được sử dụng trong GoogleNet/Incepstion v3 và v4

5.12. MẠNG HỌC DƯ SÂU ResNET

5.12.1. Tổng quan

Mạng dư sâu (ResNet) là một kiểu của kiến trúc mạng neuron tích chập (CNN) để giải quyết vấn đề giảm hiệu suất trong những mô hình rất sâu bằng cách sử dụng các khối dư với các kết nối bỏ qua (Skip Connnection hay Drop Connection). Các kết nối bỏ qua này bỏ qua một hay nhiều lớp cho phép mạng học các ánh xạ "dư" thay vì ánh xạ trực tiếp, làm cho việc đào tạo các mạng sâu dễ hơn và cải thiện hiệu suất bằng cách ngăn chặn biến mất gradient.

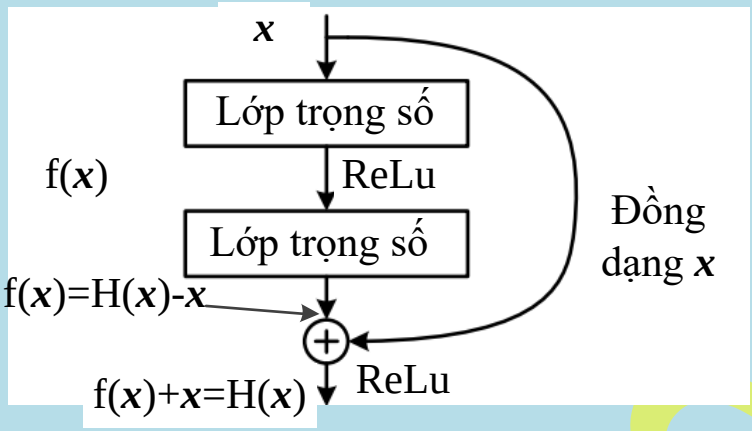
Khi các mạng sâu có thể bắt đầu hội tụ, bộc lộ một vấn đề giảm cấp: cùng với tăng độ sâu mạng, độ chính xác trở nên bão hòa và giảm cấp rất nhanh. Không theo dự đoán, giảm cấp này không bị gây ra bởi quá khớp (Overfitting) và việc bổ sung thêm các lớp cho phù hợp với một mô hình sâu dẫn đến sai số đào tạo cao hơn. Hình 5.31 cho thấy một thí dụ điển hình.



Hình 5.31. Sai số đào tạo (trái) và sai số kiểm tra (phải) trên các mạng CIFAR-10 với 20 lớp và 50 lớp. Mạng sâu hơn có sai số đào tạo và sai số kiểm tra cao hơn

Nguyên lý của học dư như sau. Ta ký hiệu ánh xạ cơ bản được mong muốn là H(x), giả thiết các lớp các lớp phi tuyến tính phù hợp với một ánh xạ khác: F(x):= H(x)-x. Ánh xạ gốc được viết lại thành F(x)+x. Hình thức hóa F(x)+x dễ ràng được thực hiện bằng các mạng neuron nối thẳng với các "kết nối tắt" (Shortcut Connection) (hình 5.32), Các kết nối tắt là các kết nối bỏ qua môt hay nhiều lớp. Trong trường hợp này, các kết nối tắt chỉ đơn giản thực hiện ánh xạ đồng nhất (Identity Mapping) và đầu ra của chúng cộng với đầu ra của các lớp được xếp chồng (hình 5.32). Các kết nối tắt không bổ sung thêm thông số cũng như sự phức tạp tính toán. Toàn bộ mạng vẫn có thể được đào tạo đầu cuối- đầu cuối bằng giảm gradient ngẫu nhiên(SGD: Stochastic Gradient Descent) với

truyền lan ngược và có thể được thực hiện dễ ràng với sử dụng các thư viện chung (ví dụ Caff) mà không cần thay đổi bộ giải (Solver).



Hình 5.32. Học dư: khối cơ sở

5.12.2. Học dư sâu

5.12.2.1. Học dư

Ta coi H(***x***) là một ánh xạ cơ bản thích hợp cho một số lớp được xếp chồng (không nhất thiết phải toàn bộ mạng), với ***x*** ký hiệu cho các đầu vào đầu tiên của các lớp này. Giả thiết rằng nhiều lớp phi tuyến tính có thể xấp xỉ hóa tiệm cận các hàm phức tạp, thì điều này tương đương với giả thiết rằng chúng có thể xấp xỉ hóa tiệm cận các hàm dư, nghĩa là, f(***x***)-***x*** (giả thiết rằng đầu vào và đầu ra có cùng kích thước). Vì thế thay vì kỳ vọng các lớp được xếp chồng xấp xỉ hóa H(***x***), ta tường minh để các lớp này xấp xỉ hóa hàm dư: f(***x***)=H(***x***)-x. Mặc dù cả hai dạng đều phải có khả năng xấp xỉ hóa tiệm cận các hàm mong muốn (theo giả thiết) nhưng độ dễ học có thể khác nhau.

Việc diễn đạt lại lại này được thúc đẩy bởi hiện tượng phản trực giác của vấn đề giảm cấp (hình 5.31, trái). Như đã bàn luận ở trên, nếu có thể cấu trúc các lớp được bổ sung như là ánh xạ các đồng dạng, mô hình sâu hơn phải có sai số đào tạo không lớn hơn đối tác nông hơn. Vấn đề giảm cấp gợi ý rằng bộ giải (Solver) có thể gặp các khó khăn khi xấp xỉ hóa các ánh xạ đồng nhất bởi nhiều lớp phi tuyến tính. Với diễn đạt lại học máy dư, nếu các ánh xạ đồng nhất tối ưu, bộ giải có thể chỉ đơn giản ép buộc các trọng số của nhiều lớp phi tuyến đến không để đạt được các ánh xạ đồng nhất.

5.12..2. Ánh xạ đồng nhất bằng nối tắt (Shortcut Connection)

Ta áp dụng học dư cho các chồng ít lớp. Một khối cơ sở như được thể hiện trong hình 5.32. Ta xét một khối cơ sở được định nghĩa như sau:

$$\boldsymbol{y} = \mathrm{f}\left(\boldsymbol{x}, \{\boldsymbol{W}_i\}\right) + \boldsymbol{x} \qquad (5.3)$$

Trong đó $\boldsymbol{x}$ và $\boldsymbol{y}$ là các vectơ đầu vào và đầu ra của các lớp được xét. Hàm $\mathrm{f}(\boldsymbol{x},\{\boldsymbol{W}_i\})$ thể hiện ánh xạ dư cần được học. Đối với thí dụ trong hình 5.32, ta có hai lớp, $\mathrm{f} = W_2\sigma\left(\boldsymbol{W}_1\boldsymbol{x}\right)$, trong đó σ ký hiệu cho ReLU và để đơn giản độ lệch được bỏ qua. Phép toán $\mathrm{f}(\boldsymbol{x})+\boldsymbol{x}$ được thực hiện bởi một nối tắt và cộng theo phần tử. Ta áp dụng phi tuyến tính thư hai sau cộng (nghĩa là $\sigma(\boldsymbol{y})$, xem hình 5.31).

Các nối tắt trong phương trình (5.3) không đưa ra thêm bất kỳ tham số nào cũng như sự phức tạp tính toán. Điều này không chỉ hấp dẫn trong thực tế mà còn quan trọng khi ta so sánh các mạng thông thường và các mạng dư. So sánh các mạng thông thường và mạng dư, ta thấy chúng đồng thời có cùng số lượng các tham số, độ sâu, độ rộng và chi phí tính toán (ngoại trừ cộng theo phần tử không đáng kể).

Các kích thước của $\boldsymbol{x}$ và f phải như nhau trong phương trình (5.3). Nếu không như vậy, (trường hợp thay đổi các kênh đầu vào/đầu ra), ta có thể thực hiện chiếu hình tuyến tính $\boldsymbol{W}_s$ bởi các kết nối tắt để khớp các kích thước.:

$$\boldsymbol{y} = \mathrm{f}\left(\boldsymbol{x}, \{\boldsymbol{W}_i\}\right) + \boldsymbol{W}_s\boldsymbol{x} \qquad (5.4)$$

Ta cũng có thể sử dụng ma trận vuông $\boldsymbol{W}_s$ trong phương trình (5.4). Nhưng thực nghiệm cho thấy rằng, ánh xạ đồng nhất đủ để giải quyết vấn đề giảm cấp và nó kinh tế. Vì thế W_s chỉ được sử dụng khi khớp các kích thước.

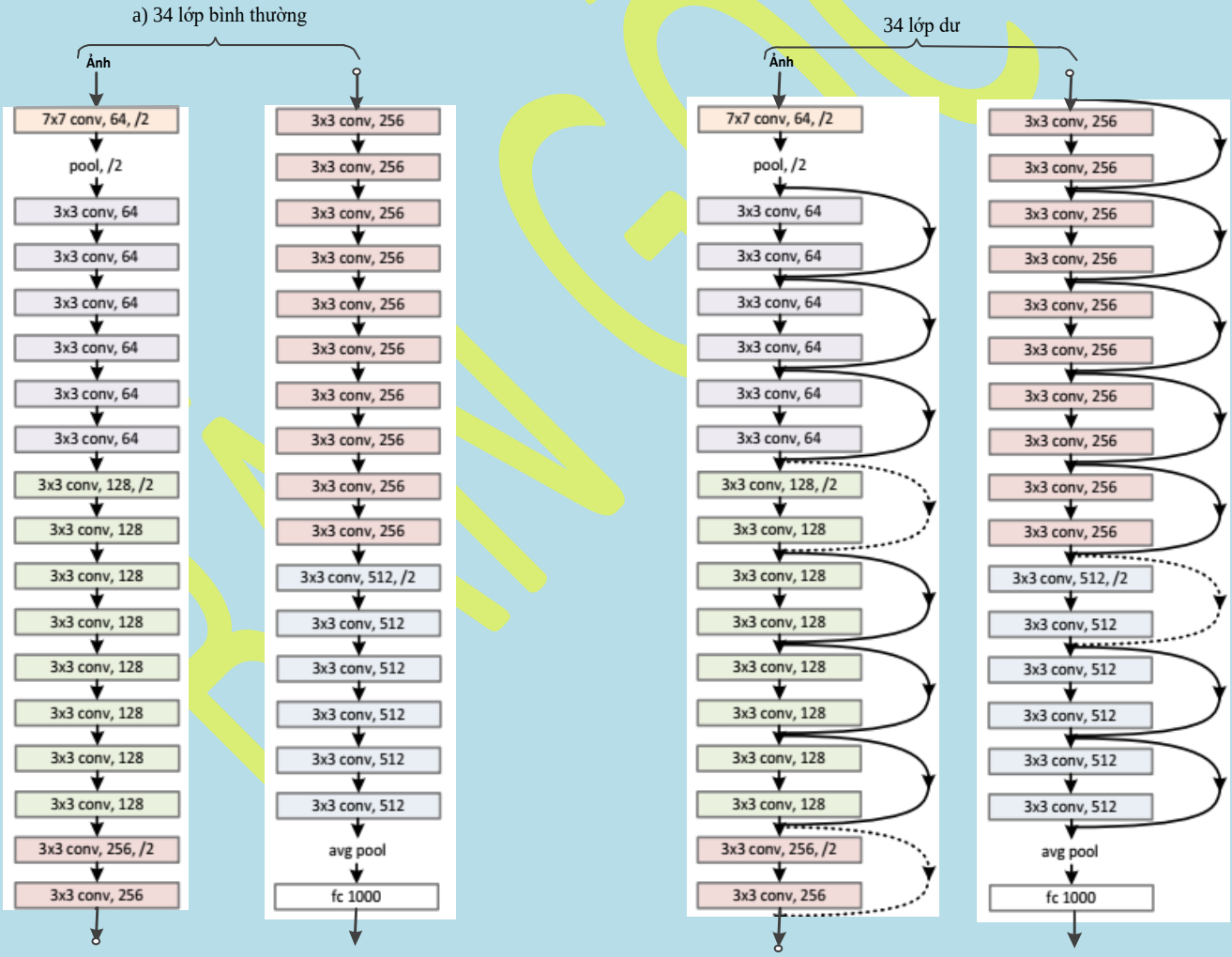
Dạng hàm dư $\mathrm{f}(\boldsymbol{x})$ là linh hoạt. Các thực nghiệm thường được sử dụng f với hai hoặc ba lớp (hình 5.33), trong khi có thể nhiều lớp hơn. Tuy nhieen nếu f chỉ có một lớp duy nhất, ta sẽ không thấy được các lợi ích.

Cũng lưu ý rằng, để đơn giản các ký hiệu ở trên dành cho các lớp kết nối toàn bộ, chung vấn có thể được áp dụng cho các lớp tích chập. Hảm $\mathrm{f}\left(\boldsymbol{x}, \{\boldsymbol{W}_i\}\right)$ cũng có thể trình bày cho nhiều lớp tích chập. Cộng theo phần tử đươc thực hiện trên hai bản đồ tính năng, theo từng kênh.

5.12.3. Kiến trúc mạng đơn giản

Để làm thí dụ, hai mô hình cho ImageNet nư sau.

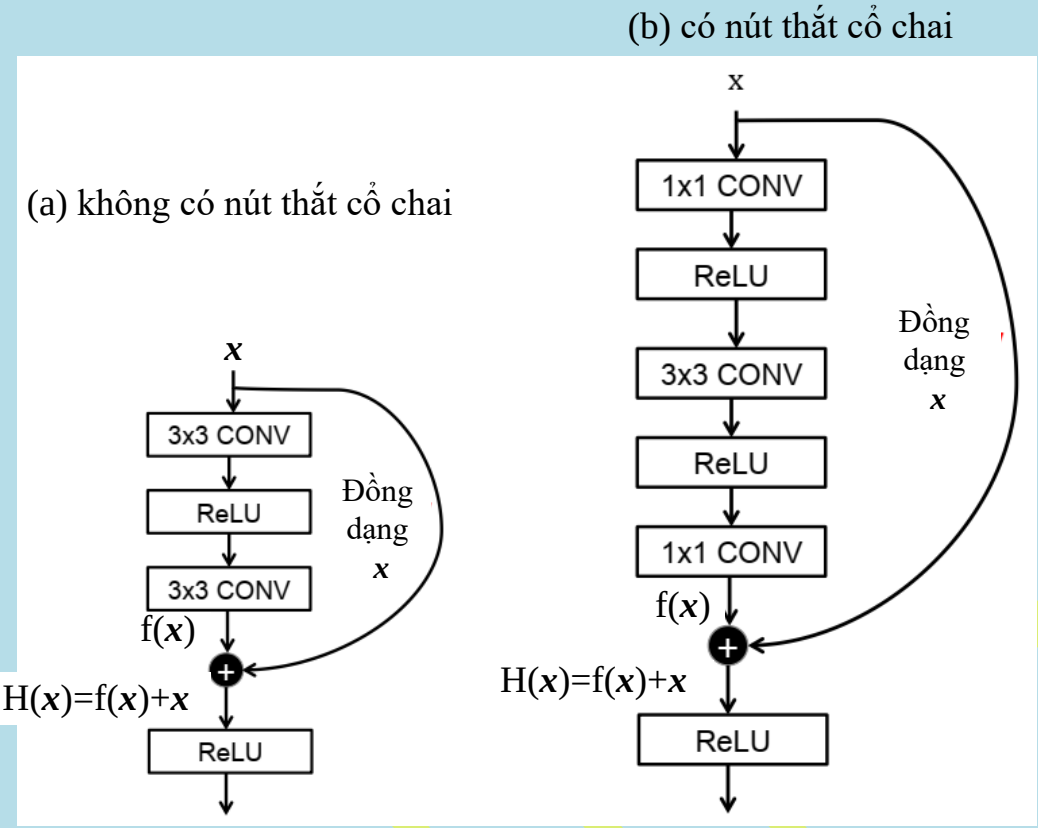
Mạng đơn giản được xây dựng dựa trên chiết lý của các mạng VGG. Hầu hết các lớp tích chập có các bộ lọc 3×3 và tuân theo hai quy tắc thiết kế đơn giản: (i) đối với cùng một kích thước bản đồ tính năng, các lớp có cùng một số lượng bộ lọc và (ii) nếu nếu kích thước bản đồ tính năng giảm một nửa, số lượng bộ lọc được nhân đôi để bảo tồn sự phức tạp thời gian trên một lớp. Lấy mẫu giảm trực tếp bởi các lớp tích chập có sải chân bằng 2. Mạng kết thúc bằng lớp gộp trung bình và lớp kết nối toàn bộ 1000 đường. Tổng số các lớp được đánh trong số là 34 trong hình 5.33 (trái).



Hình 5.33. (trái): mạng bình thường với 34 lớp tham số. (phải): mạng dư với 34 lớp tham số. Đường đứt nét biểu thị tăng kích thước.

ResNet là DNN đầu tiên trong ImageNet Challenge vượt quá độ chính xác ở cấp độ con người với tỷ lệ lỗi top 5 dưới 5%. Một trong những thách thức với mạng sâu là gradient (độ dốc) biến mất trong quá trình đào tạo: khi lỗi lan truyền ngược qua mạng, gradient sẽ thu hẹp, điều này ảnh hưởng đến khả năng cập nhật trọng số trong các lớp trước đó cho các mạng rất sâu. Mạng dư giới thiệu một mô-đun 'phím tắt' hay kết nối bỏ qua (‘shortcut’ module) có chứa kết nối đồng dạng (Identity) sao cho các lớp trọng số (tức là các lớp CONV) có thể được bỏ qua như trong Hình 5.34. Thay vì học hàm f($\boldsymbol{x}$) cho các lớp trọng số, mô-đun phím tắt học ánh xạ dư (f($\boldsymbol{x}$) = H($\boldsymbol{x}$) - $\boldsymbol{x}$). Ban đầu, f($\boldsymbol{x}$) bằng không và kết nối đồng dạng được thực hiện; sau đó dần dần trong quá trình tập luyện, kết nối thẳng (Forward) thực tế thông qua lớp trọng số được sử dụng. Điều này tương tự như các mạng LSTM được sử dụng cho dữ liệu tuần tự. ResNet cũng sử dụng cách tiếp cận 'nút thắt cổ chai' bằng cách sử dụng bộ lọc 1×1 để giảm số lượng tham số trọng số. Kết quả là, hai lớp trong mô-đun phím tắt được thay thế bằng ba lớp (1×1, 3×3, 1×1) trong đó 1×1 giảm và sau đó tăng (khôi phục) số lượng trọng số. ResNet-50 bao gồm một lớp CONV, tiếp theo là 16 lớp phím tắt (Shortcut Layer) (mỗi lớp sâu ba lớp CONV) và một lớp FC; nó yêu cầu 25,5 triệu trọng số và 3.9G MAC cho mỗi hình ảnh. Có nhiều phiên bản khác nhau của ResNet với nhiều độ sâu (ví dụ: không có nút cổ chai: 18, 34; với nút cổ chai: 50, 101, 152). ResNet với 152 lớp là người chiến thắng trong Thử thách ImageNet yêu cầu MAC 11.3G và trọng lượng 60M. So với ResNet-50, nó giảm sai số top 5 khoảng 1% với chi phí là 2,9× MAC nhiều hơn và 2,5× trọng số hơn.

BẢN NHÁP



Hình 5.34. Mô-đun phím tắt từ ResNet. Lưu ý rằng ReLU theo sau lớp CONV cuối cùng trong đường tắt là sau khi cộng

Một số xu hướng có thể được quan sát thấy trong các DNN phổ biến được biểu thị trong Bảng 5.2. Tăng độ sâu của mạng có xu hướng mang lại độ chính xác cao hơn. Bằng cách điều khiển số lượng trọng số, một mạng sâu hơn có thể hỗ trợ một phạm vi rộng hơn của các hàm phi tuyến tính tách biệt hơn và cũng cung cấp nhiều cấp độ phân cấp hơn trong biểu diễn đã học (Learned Representation). Số lượng hình dạng bộ lọc tiếp tục thay đổi giữa các lớp, do đó tính linh hoạt vẫn rất quan trọng. Hơn nữa, hầu hết các tính toán đã được đặt trên các lớp CONV thay vì các lớp FC. Ngoài ra, số lượng trọng số trong các lớp FC được giảm và trong hầu hết các mạng gần đây (kể từ GoogLeNet), các lớp CONV cũng chiếm ưu thế về trọng số. Do đó, trọng tâm của việc triển khai phần cứng nên là giải quyết hiệu quả của các lớp CONV, điều này trong nhiều lĩnh vực ngày càng quan trọng.

513. QUÁ TRÌNH PHÁT TRIỂN DNN VÀ CÁC ỨNG DỤNG

5.13.1. Quá trình phát triển DNN

Mặc dù các mạng neuron đã được đề xuất trong những năm 1940, ứng dụng thực tiễn đầu tiên sử dụng các neuron số vẫn chư xuất hiện cho đến cuối thập niên 1980 với mạng LeNet cho nhận dạng chữ số viết tay. Các hệ thống này được sử dụng rộng rãi cho các cây ATM để nhận dạng các chữ số trên các tờ séc. Tuy nhiên, đầu thập niên 2010 đã tháy bùng nổ các ứng dụng dựa trên DNN với các tính năng nooit bật, như hệ tống nhận dạng giọng nói trong năm 2011 và hệ thống AlexNet cho nhận dạng ảnh trong năm 2012 v.v.

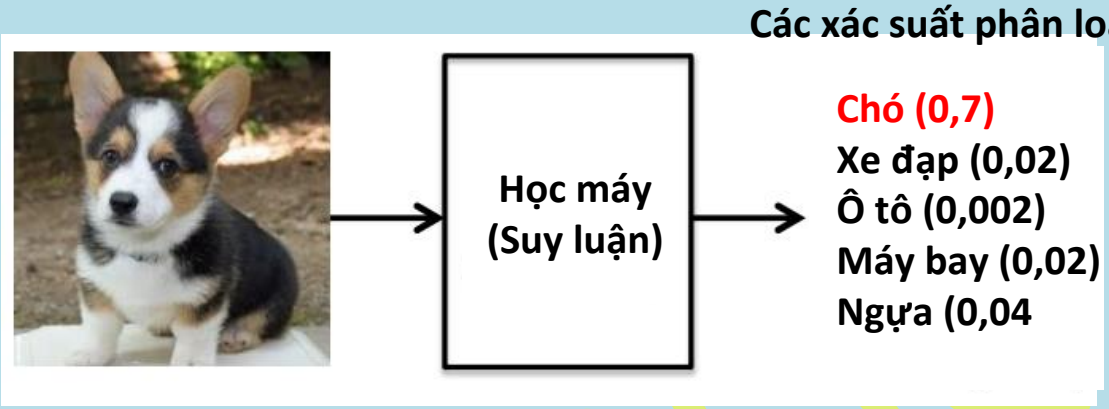
Có thể tin rằng thành công của học máy trong những năm đầu của thập niên 2010 là sự phối hợp của ba nhân tố. Nhân tố thứ nhất là khối lượng thông tin có sẵn để dào tạo. Chẳng hạn Facebook nhận được trên 350 triệu ảnh mỗi ngày. Walmart tạo ra 2,5 Petabyte (10 tỷ byte) dữ liệu người dùng hàng giờ và Youtube có 300 giờ video được tải lên mỗi phút. Kết quả là các nhà cung cấp đám mây và nhiều hoạt động kinh doanh khác có một khối lượng khổng lồ dữ liệu để đào tạo cho các giải thuật của mình.

Nhân tố thứ hai là khối lượng dung lượng tính toán. Các tiến bộ của thiết bị bán dẫn và kiến trúc máy tính không ngừng tăng để đảm bảo dung lượng tính toán tăng và có vẻ như ta đã vượt qua ngưỡng, nơi mà khối lượng lớn tính toán tổng được đánh trọng số trong các DNN cần thiết cho cả suy luận và đào tạo có thể được thực hiện trong một khoảng thời gian hợp lý.

Các thành công của các ứng dụng DNN ban đầu này đã mở ra các cổng xả lũ cho phát triển giải thuật. Nó cũng cảm hứng cho sự phát triển một số các bộ khung phần mềm (nguồn mở rộng) tạo điều kiện thuận lợi hơn cho các nhà nghiên cứu và thực hành để khám phá và sử dụng các DNN. Kết hợp các nỗ lực này đóng góp cho nhân tố thứ ba, phát triển các kỹ thuật giải thuật cải thiện đáng kể độ chính xác của ứng dụng và mở rộng các lĩnh vực mà các DNN đang được áp dụng cho chúng.

Thí dụ về các thành công tuyệt vời trong học máy được minh họa với thử thánh ImageNet Challenge. Thử thách này là một cuộc đua bao gồm một số thành phần khác nhau. Một trong số các thành phần này là nhiệm vụ phân loại ảnh, trong đó các giải thuật được cho trước một ảnh và chúng phải nhận dạng cái gì có trong ảnh này như được thể hiện trên hình 5.20. Tập đào tạo bao gồm 1,2 tỷ ảnh, mỗi ảnh được gán một nhãn với

1000 thể loại đối tượng mà ảnh này chứa. Đối với giai đoạn đánh giá, giải thuật này phải nhận dạng các đối tượng chính xác trong tập dữ liệu kiểm thử của các ảnh đối tượng nào chưa được nó nhìn thấy trước đây.



Hình 5.20. Ví dụ về một nhiệm vụ phân loại hình ảnh. Nền tảng học máy lấy hình ảnh và cho ra điểm số độ tin cậy (Confidence Score) cho một tập các lớp được xác định trước

5.13.2. Các ứng dụng của DNN

Nhiều ứng dụng có thể được hưởng lợi từ các DNN khác nhau, từ đa phương tiện đến không gian y tế. Trong phần sẽ cung cấp các ví dụ về các lĩnh vực mà DNN hiện đang tác động và nêu bật các lĩnh vực mới nổi mà DNN hy vọng sẽ tạo ra tác động trong tương lai.

Hinh ảnh và video. Video được cho là dữ liệu lớn nhất. Nó chiếm hơn 70% lưu lượng truy cập Internet ngày nay. Ví dụ, hơn 800 triệu giờ video được thu thập hàng ngày trên toàn thế giới để giám sát video. Thị giác máy tính (Computer Vision) là cần thiết để trích xuất thông tin có ý nghĩa từ video. DNN đã cải thiện đáng kể độ chính xác của nhiều tác vụ thị giác máy tính như phân loại hình ảnh, bản địa hóa và phát hiện đối tượng, phân đoạn hình ảnh và nhận dạng hành động.

Lời nói và ngôn ngữ. Các DNN đã cải thiện đáng kể độ chính xác của nhận dạng giọng nói cũng như nhiều các tác vụ liên quan như dịch máy , xử lý ngôn ngữ tự nhiên và tạo âm thanh.

Y tế. Các DNN đã đóng một vai trò quan trọng trong bộ gen để hiểu rõ hơn về di truyền của các bệnh như tự kỷ, ung thư và teo cơ cột sống. Chúng cũng đã được sử dụng trong hình ảnh y tế để phát hiện ung thư da, ung thư não và ung thư vú.

Chơi game. Gần đây, nhiều thử thách AI lớn liên quan đến chơi trò chơi đã được vượt qua bằng cách sử dụng DNN. Những thành công này cũng đòi hỏi những đổi mới trong kỹ thuật đào tạo và nhiều trong số chúng dựa vào học tăng cường. DNN đã vượt qua độ chính xác của con người khi chơi Atari cũng như Go, nơi việc tìm kiếm toàn diện tất cả các khả năng là không khả thi do số lượng nước đi có thể xảy ra khổng lồ không thể tưởng tượng được.

Robot. DNN đã thành công trong lĩnh vực các nhiệm vụ robot như nắm bằng cánh tay robot, lập kế hoạch chuyển động cho robot mặt đất, điều hướng trực quan, điều khiển để ổn định quadcopter (máy bay bốn cánh quạt) và chiến lược lái xe cho xe tự hành.

DNN đã được sử dụng rộng rãi trong các ứng dụng đa phương tiện ngày nay (ví dụ: thị giác máy tính, nhận dạng giọng nói). Trong tương lai, ta có thể hy vọng rằng DNN có thể sẽ đóng một vai trò ngày càng quan trọng trong lĩnh vực y tế và robot, như đã nói ở trên, cũng như tài chính (ví dụ: giao dịch, dự báo năng lượng và đánh giá rủi ro), cấu trúc khung (ví dụ: an toàn cấu trúc và kiểm soát giao thông), dự báo thời tiết và phát hiện sự kiện. Vô số lĩnh vực ứng dụng đặt ra những thách thức mới đối với việc xử lý hiệu quả DNN; khi này các giải pháp phải thích ứng và có thể mở rộng để xử lý các dạng DNN mới và đa dạng mà các ứng dụng này có thể sử dụng.

5.13.3. Nhúng so với đám mây

Các ứng dụng và khía cạnh khác nhau của xử lý DNN (tức là đào tạo so với suy luận) có nhu cầu tính toán khác nhau. Cụ thể, việc đào tạo thường yêu cầu một bộ dữ liệu lớn và tài nguyên tính toán đáng kể cho nhiều lần lặp lại cập nhật trọng số. Trong nhiều trường hợp, việc đào tạo mô hình DNN vẫn mất vài giờ đến nhiều ngày và do đó thường được thực hiện trên đám mây. Mặt khác, suy luận có thể xảy ra trên đám mây hoặc ở biên (ví dụ: IoT hoặc thiết bị di động).

Trong nhiều ứng dụng, thường mong muốn xử lý suy luận DNN gần cảm biến. Ví dụ, trong các ứng dụng thị giác máy tính, chẳng hạn như đo thời gian chờ đợi trong các cửa hàng hoặc dự đoán mô hình lưu lượng truy cập, nên trích xuất thông tin có ý nghĩa từ video ngay tại cảm biến hình ảnh thay vì trên đám mây để giảm chi phí truyền thông. Đối với các ứng dụng khác như xe tự hành, điều hướng máy bay không người lái và robot, xử lý tại chỗ là mong muốn vì độ trễ và rủi ro bảo mật của việc dựa vào đám mây là quá cao.

Tuy nhiên, video liên quan đến một lượng lớn dữ liệu đòi hỏi xử lý phức tạp về mặt tính toán; nên phần cứng chi phí thấp để phân tích video là một thách thức nhưng rất quan trọng để cho phép các ứng dụng này. Nhận dạng giọng nói cho phép chúng ta tương tác liền mạch với các thiết bị điện tử, chẳng hạn như điện thoại thông minh. Mặc dù hiện tại

hầu hết quá trình xử lý cho các ứng dụng như Apple Siri và dịch vụ thoại Amazon Alexa là trên đám mây, nhưng vẫn nên thực hiện nhận dạng trên chính thiết bị để giảm độ trễ và phụ thuộc vào kết nối, đồng thời cải thiện quyền riêng tư và bảo mật.

Nhiều nền tảng nhúng thực hiện suy luận DNN có giới hạn nghiêm ngặt về mức tiêu thụ năng lượng, tính toán và chi phí bộ nhớ; do đó, việc xử lý hiệu quả các DNN đã trở nên quan trọng hàng đầu trong điều kiện những ràng buộc này. Các phần dưới đây sẽ tập trung vào các yêu cầu tính toán để suy luận hơn là đào tạo

5.14. CÁC TÀI NGUYÊN ĐỂ PHÁT TRIỂN DNN

Một trong số các nhân tố then chốt cho phép phát triển nhanh các DNN là tập các tài nguyên phát triển được tạo sẵn cho cộng đồng và công nghiệp. Các tài nguyên này cũng là chìa khóa để phát triển các bộ tăng tốc DNN bằng cách cung cấp các mô tả đặc trưng khối lượng công việc tạo điều kiện để tìm ra sự cân bằng giữa độ phức tạp của mô hình và độc chính xác. Phần này sẽ mô tả các tài nguyên để những người quan tâm đến lĩnh vực này có thể nhanh chóng bắt đầu.

5.14.1. Các bộ khung phần mềm

Đối với trường hợp phát triển DNN và cho phép chia sẻ các mạng đã được đào tạo, một số bộ khung học sâu đã được phát triển từ các nguồn khác nhau. Các thư viện nguồn mở này chứa các thư viện phần mềm cho các DNN. Caffe đã được cung cấp năm 2014 từ UC Berkeley. Nó hỗ trợ C, C^{++} , Python và Matlab. Tensorflow được phát hành năm 2015 bởi Google và hỗ trợ C^{++}, Python, nó cũng hỗ trợ nhiều GPU và CPU, và có tính linh hoạt hơn Caffe, với tính toán được biểu diễn như các graph dòng dữ liệu để quản lý các tensor (các mảng đa chiều). Một bộ khung phổ biến khác là Torch được phát triển bởi Facebook và NYU, hỗ trợ C,C^{++} và Lua. Ngoài ra còn có một số bộ khung khác như Theano, MXNet, CNTK. Cũng có các thư viện mức cao hơn chạy trên đỉnh các bộ khung nói trên để cung cấp trải nghiệm phổ quát và phát triển nhanh hơn. Một thí dụ trong số các thư viện này là Keras được viết bởi Python và hỗ trợ Tensorflow, CNTK và Theano.

Sự tồn tại của các bộ khung này không chỉ tiện lợi cho các nhà nghiên cứu DNN và các nhà thiết kế ứng dụng, mà chúng còn cực kỳ quý giá cho kỹ thuật hiệu suất cao hay các công cụ tính toán DNN hiệu quả. Cụ thể là, do các khung sử dụng nhiều một tập các phép toán ngyuên thủy, chẳng hạn xử lý một lớp CONV, Chúng có thể kết hợp phần mềm và các bộ tăng tốc phần cứng. Sự tăng tốc này trong suốt đối với người dùng bộ khung. Vì vậy, chẳng hạn, hầu hết các khung có thể sử dụng thư viện DNN của Nvidia để thực hiện nhanh trên các GPU của Nvidia. Tương tự, kết hợp trong suốt các bộ tăng tốc dành riêng có thể đạt được như đã làm với Eyeriss chip.

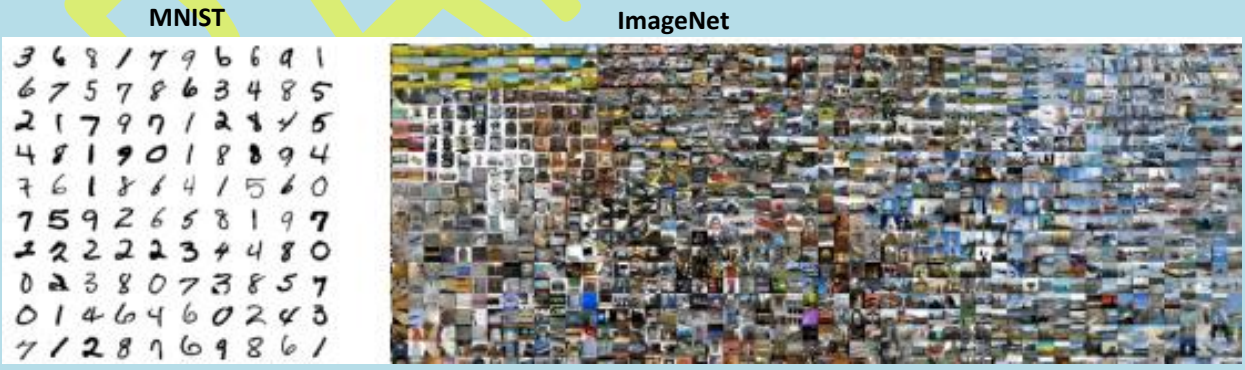
Cuối cùng các bộ khung này là một nguồn giá trị của khối lượng công việc đối với các nhà thiết kế phần cứng. Chúng có thể được sử dụng để thúc đẩy các thiết kế thí nghiệm cho các khối lượng công việc khác nhau, cho lập hồ sơ (lược tả) các khối lượng công việc khác nhau hay tìm ra cân bằng phần cứng phần mềm.

5.14.2. Các mô hình

Các mô hình được đào tạo trước có thể được tải xuống từ các website khác nhau cho các bộ khung khác nhau. Cần lưu ý rằng ngay cả đối với cùng một DNN (ví dụ: AlexNet), độ chính xác của các mô hình này có thể thay đổi khoảng 1% đến 2% tùy thuộc vào việc mô hình được đào tạo như thế nào và vì thế các kết quả không chính xác khớp với công bố ban đầu.

5.14.3. Các tập dữ liệu phổ biến cho phân loại (Classification)

Điều quan trong là cân nhắc độ khó của nhiệm vụ khi so sánh các mô hình DNN khác nhau. Chẳng hạn, nhiệm vụ phân loại các chữ số viết tay từ bộ dữ liệu MNIST đơn giản hơn nhiều so với phân loại một đối tượng trong số 1000 loại như yêu cầu trong tập dữ liệu ImageNet như trên hình 5.27. Dự kiến rằng kích thước của các DNN (ví dụ số lượng các trọng số) và số lượng các MAC (Multiply-and-Accumlate: nhân và tích lũy) sẽ lớn hơn đối với nhiệm vụ khó hơn so với nhiệm vụ dễ hơn và vì thế đòi hỏi năng lượng lớn hơn và có thông lượng thấp hơn. Chẳng hạn, LeNet-5 được thiết kế cho phân loại chữ số, trong khi AlexNet , VGG-16, GoogLeNet và ResNet được thiết kế cho phân loại ảnh với 1000 loại.



Hình 5.22. MNIST (10 loại, 60k đào tạo, 10k kiểm tra so với ImageNet (1000 loại, 1,3M đào tạo, 10k kiểm tra)

Có nhiều nhiệm vụ AI đi cùng với các tập dữ liệu được cung cấp có sẵn công khai để đánh giá độ chính xác của một DNN cho trước. Các tập dữ liệu công khai rất quan trọng để so sánh độ chính xác của các phương pháp khác nhau. Nhiệm vụ phổ biến và đơn giản nhất là phân loại ảnh, liên quan đến cho trước một ảnh và chọn ra 1 trong số N loại mà ảnh này có khả năng lớn nhất thuộc về loại này. Không có định vị hay phát hiện.

MNIST là một tập dữ liệu được sử dụng rộng rãi để phân loại chữ số (được đưa ra năm 1998). Nó bao gồm các ảnh thang độ xám 28×28 pixel các chữ số viết tay. Có 10 loại (cho 10 chữ số) , 60 000 ảnh đào tạo và 10 000 ảnh kiểm ttra. LeNet-5 có thể đạt độ chính xác 99,79% bằng cách sử dụng chính quy hóa với dropconnect (loại bỏ kết nối), trong đó mạng đặt một phần các trọng số vào không (nói một cách khác loại bỏ một phần các trọng số). Hiện này MNIST được xem là tập dữ liệu khá dễ.

CIFAR-10 là một tập dữ liệu gồm các ảnh mầu 32×32 pixel của các đối tượng khác nhau, được phát hành năm 2009. CIFAR là một tập con của tập dữ liệu 80 triệu ảnh nhỏ (Tiny Image). CIFAR-10 được tạo thành từ 10 ảnh loại trừ lẫn nhau. Có 50 000 ảnh đào tạo (5000 ảnh trên một loại) và 10 000 ảnh kiểm thử (1000 ảnh trên một loại). Mạng niềm tin sâu tích chập (CDBN: Convolutional Deep Belief Network) hai lớp đã có thể đạt độ chính xác 68.84% trên CIFAR-10 khi nó lần đầu được được đưa ra.Từ đó đến nay độ chính xác đã tăng đến 96,53% với sử dụng gộp cực đại từng phần (Fractional Max Pooling).

ImageNet là tập dữ liệu ảnh kích thước lớn (Large Scale Image Dataset), lần đầu tiên được đưa ra năm 2012. Nó chứa các ảnh mầut 256×256 pixel với 1000 loại. Các loại này được định nghĩa với sử dụng WordNet làm sương sống (Backbone) để xử lý ý nghĩa của một từ khó hiểu và kết hợp các từ đồng nghĩa với nhau vào một đối tượng. Nói một cách khác, có một hệ thống phân cấp cho các thể loại ImageNet. 1000 loại được chọn sao cho không có chồng lấn trong hệ thống phân cấp của ImageNet. Tập dữ liệu ImageNet chứa nhiều loại ảnh hạt mịn bao gồm 120 giống chó khác nhau. Có 1,3 triệu ảnh đào tạo (732 đến 13000 trên một loại), 100 000 ảnh kiểm tra (100 trên một loại) và 50 000 ảnh xác thực (Validation Image) (50 trên một loại).

Độ chính xác của thử thách Image Challenge đã được báo khi sử dụng hai chỉ số đo: sai lỗi Top-5 (Top-5 Error) và Top-1 (Top-1 Error). Sai lỗi Top-5 có nghĩa là nếu bất kỳ thể loại nào trong số các thể loại ghi một trong năm điểm số cao nhất (Top Five Scoring Categories), nó được tính là một phân loại đúng. Top-1 đòi hỏi rằng thể loại này ghi điểm số cao nhất để là thể loại đúng. Năm 2012, người chiến thắng Image Challenge (AlexNet) đã có thể đạt được độ chính xác 83,6% đối với Top-5 (tốt hơn đáng kể so với

73,8% của vị trí thứ hai năm đó không sử dụng DNN), nó đạt 61,9% top-1 của tập xác thực. năm 2017, độ chính xác cao nhất là 99,7% cho top-5.

Tổng kết các tập dữ liệu phân loại ảnh khác nhau, rõ ràng rằng MNIST là tập dữ liệu kha dễ, trong khi ImageNet là một thách thức với vùng phủ các loại rộng hơn. Vì thế liên quan đến đánh giá độ chính xác của một DNN cho trước, cần xem xét tập dữ liệu mà trên đó độ chính xác được đo.

5.14.4. Các tập dữ liệu cho các nhiệm vụ khác

Vì độ chính xác của các DNN hiện tại đang thực hiện tốt hơn độ chính xác mức độ con người đối với các nhiệm vụ phân loại ảnh, Image Challenge bắt đầu tập trung lên các nhiệm vụ khó hơn như định vị đối tượng đơn (Single-Object Localization) và phát hiện đối tượng (Object Detection). Đối với định vị đối tượng đơn, đối tượng mục tiêu phải được định vị và được phân loại (trong số 1000 loại). DNN cho ra các thể loại top-5 (top-5 category) và các vị trí hộp gới hạn top-5 (top-5 bounding box). Sẽ không có phạt đối với nhận dạng một đối tượng trong ảnh nhưng không được bao gồm trong chân lý cơ bản. Đối với phát hiện đối tượng, tất cả các đối tượng trong ảnh phải được định vị và được phân loại (trong số 200 loại). Hộp giới hạn cho tất cả các đối tượng trong các thể loại này phải được đánh nhãn. Các đối tượng không được đánh nhãn phải bị phạt như là các phát hiện trùng lặp.

Ngoài ImageNet, còn có các tập dữ liệu ảnh phổ biến khác cho các nhiệm thụ thị giác máy tính. Để phát hiện đối tượng, có tập dữ liệu PASCAL VOC (2005-2012) chứa 11k ảnh đại diện 20 loại (27k các thực thể đối tượng, 7k trong số này có phân đoạn chi tiết). Đối với phát hiện đối tượng, phân đoạn và nhận dạng trong bối cảnh, có tập dữ liệu MS COCO với 2,5 M thực thể được đánh nhãn trong 328 ảnh (91 thể loại đối tượng)); so với ImageNet, COCO cũng có nhiều thực thể được đánh nhãn hơn cho trên mỗi ảnh để hỗ trợ tiềm tàng thông tin bối cảnh.

Gần đây, thậm chí các tập dữ liệu kích thước lớn hơn đã được cung cấp. Chẳng hạn, tập dữ liệu Google và Open Image với 9M ảnh bao phủ 6000 thể loại. Còn có cả tập dữ liệu Youtube với 8M video (0,5M giờ video) bao phủ 4800 loại. Google cũng phát hành một tập dữ liệu audio gồm 632 loại sự kiện audio và một sưu tập 2M clip âm thanh 10 giây được gán nhãn của con người (Human-Labeled 10-second Sound Clip). Các tập dữ liệu này sẽ luôn luôn quan trọng vì các DNN trở nên sâu hơn với nhiều tham số trọng số hơn để đào tạo.

Không nghi ngờ, cả hai các tập dữ liệu lớn hơn và các tập dữ liệu cho các lĩnh vực mới sẽ phụ vụ như là các tài nguyên quan trọng cho phân tích và khám phá hiệu quả của các cộng cụ DNN tương lai.

5.15. TỔNG KẾT

Sử dụng các mạng neuron sâu (DNN) bùng nổ trong những năm gần đây. Hiện nay chúng được sử dụng rộng rã cho nhiều ứng dụng AI như: thị giác máy tính, nhận dạn giọng nói và robot và có thể đem lại độ chính xác tuyệt vời như vậy thì cái giá phải trả lag độ phức tạp tính toán cao. Vì thế, các kỹ thuật cho phép xử lý cảu các mạng neuron sâu cải thiện hiệu quả năng lượng và thông lượng mà không tổn hại độ chính xác cùng với phần cứng có hiệu quả chi phí là tối quan trong để mở rộng DNN trong cả các mạng hiện tại và tương lai.

Để tạo lập môt hệ thống xử lý DNN hiệu quả cần bắt đầu từ việc hiểu biến các ứng dụng hiện thời và tương lai cùng với các tính toán đặc thù cần cho cả hai cũng như tiềm năng phát triển của các tính toán này. Các ứng dụng, các giải thuật liên quan và dữ liệu đầu cần được sử dụng đang thay đổi rất nhanh. Vì thế dự đoán các xu thê này để xác định mức độ linh hoat cần để xử lý các tính toán thế hệ sau trở thành một phân quan trọng của mọi đồ án thiết kế.

Chương 6

CÁC MẠNG NEURON HỒI QUY (RNN)

6.1. MỞ ĐẦU

Mạng neuron hồi quy (RNN: Recurrent Neuron Network) là một mạng neuron chuyên biệt với kết nối phản hồi để xử lý dữ liệu lần lượt theo thời gian (Sequential Data) hay các chuỗi dữ liệu trong đó đầu ra nhận được được nối ngược lại mạng như là đầu vào cùng với đầu vào mới tại từng bước thời gian. Kết nối phản hồi cho phép mạng neuron nhắc lại dữ liệu quá khứ khi xử lý đầu vào tiếp theo. Xử lý này có thể được định nghĩa như là một xử lý hồi quy và vì thế kiến trúc cũng được gọi là một mạng neuron hồi quy. Trong chương này ta sẽ tổng kết sáu kiến trúc RNN phổ biến nhất và các biến thể của

chúng và nêu bật ưu nhược điểm của từng kiến trúc. Ta cũng bàn luận về các giải pháp và các kỹ thuật thực tế để đào tạo các một hình RNN, bao gồm các kết nối bỏ qua (Skip Connection) khác nhau và cắt xén gradient (Gradient Clipping) . Cuối cùng, ta nêu bật bốn ứng dụng mô hình ngôn ngữ phổ biến của các mô hình RNN: phân loại văn bản, tổng kết, phiên dịch máy và chuyển đổi ảnh sang ảnh; bằng cách đó giải thích nghiên cứu có ảnh hưởng lớn trong từng lĩnh vực.

6.2. CÁC KẾN TRÚC RNN PHỔ BIẾN

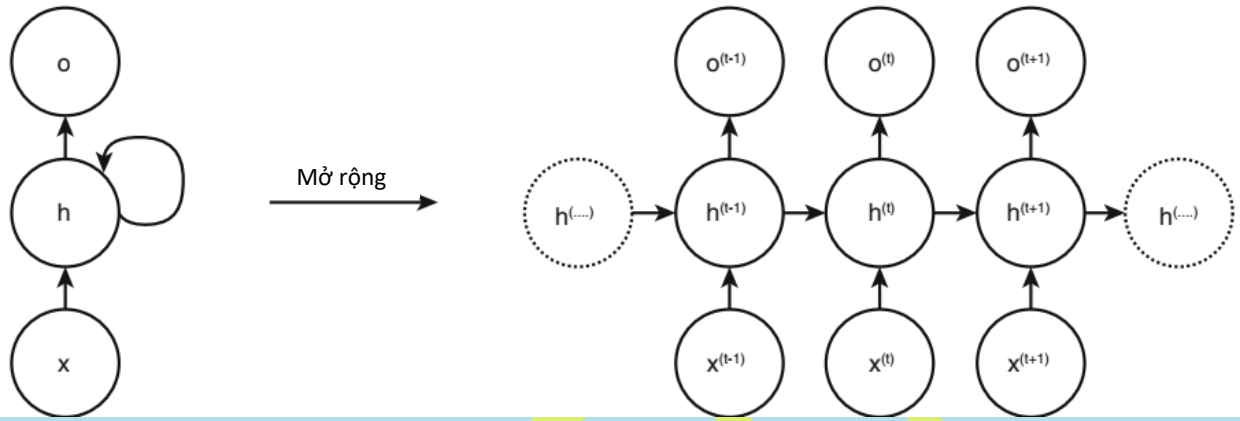
Ngoài kiến trúc RNN đơn giản (SimpleRNN), nhiều biến thể đã được đề suất để xử lý các trường hợp sử dụng khác nhau. Trong phần này ta sẽ xét một số kiến trúc RNN phổ biến như LSTM, GRU, RNN song phương (Bidirectional RNN), RNN sâu (Deep RNN) và các mô hình được quan tâm cùng với bàn luận về ưu và khuyết của chúng.

6.2.1. SimpleRNN

Kiến trúc SimpleRNNS, còn được gọi là SimpleRNN, chứa một mạng neuron đơn giản cùng với một kết nối phản hồi. Nó có khả năng xử lý dữ liệu lần lượt theo thời gian có độ dài khả biến do chia sẻ tham số để tổng quát hóa mô hình để xử lý các chuỗi có chiều dài khả biến. Không giống như các mạng neuron nối thẳng , chúng có các trọng số tách biệt cho từng tính năng đầu vào, RNN chia sẻ cùng các trọng số trên một số bước thời gian. Trong RNN, đầu ra của một bước thời gian hiện tại phụ thuộc vào đầu vào của các bước thời gian trước đó và nhận được bởi cùng một quy tắc cập nhật, quy tắc được sử dụng đê nhận được các đầu ra trước đó. Ta sẽ thấy, RNN có thể được mở rộng vào hai graph , trong đó các trọng số được chia sẽ trên các bước thời gian. RNN hoạt động trên một đầu vào $\boldsymbol{x}^{(t)}$ với chỉ số thời gian t từ 1 đến τ được minh họa trong hình 6.1. Chỉ số bước thời gian t không nhất thết đề cập đến chuyển dịch thời gian trong thế giới thực; nó có thể đề cập đến một phần của chuỗi.

Các chu kỳ của graph tính toán trình bày ảnh hưởng của giá trị quá khứ của biến lên bước thời gian hiện tại. Graph tính toán có cấu trúc lặp, nó mở rộng tính toán hồi quy của của RNN, tương ứng với một chuỗi sự kiện. Nó cho thấy các dòng chảy thông tin, đi thẳng theo thời gian của tính toán đầu ra và các mất mát cùng với đi ngược khi tính toán các gradient. Graph tính toán được mở rộng được thể hiện trong hình 6.1. Phương trình tương ứng với graph tính toán là $\boldsymbol{h}^{(t)} = f\ \boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{W}$, trong đó $\boldsymbol{h}$ là trạng thái ẩn (Hidđen

State) của mạng, ***x*** là đầu vào, t là bước thời gian và ***W*** ký hiệu cho các trọng số của các kết nối mạng bao gồm các trọng số kết nối đầu vào đến ẩn (Input-to-Hidden) , ẩn đến ẩn (Hidden-to-Hidden) và ẩn đến đầu ra (Hidden-to-Ouput).



Hình 6.1. (Trái) Sơ đồ mạch của SimpleRNN với đầu vào *x* được kết hợp vào trạng thái ẩn *h* với kết nối phản hồi và một đầu ra (o). (Phải) Cùng một mạng SimpleRNN được thể hiện như là một graph tính toán với các nút tại mọi bước thời gian

6.2.2. Nguyên lý đào tạo

Đào tạo được thực hiện bằng cách tính toán gradient của hàm mất mát liên quan đến các tham số tham gia và truyền lan thẳng từ trái sang phải graph được mở ra theo thời gian, được tiếp sau bởi truyền lan ngược chuyển động từ phải sang trái. Tính toán gradient như vậy là một hoạt động tốn kém do không thể giảm thời gian chạy bằng cách xử lý song song bởi vì truyền lan thẳng về bản chất xảy ra liên tục. Các trạng thái được tính toán khi truyền lan thẳng được lưu cho đến khi chúng được tái sử dụng trong truyền lan ngược. Giải thuật truyền lan ngược được áp dụng cho RNN được gọi là truyền lan ngược xuyên thời gian (BPTT: Back-Propagation Through Time).

Các hoạt động tính toán dưới đây dược thực hiện trong RNN trong quá trình truyền lan thẳng để tính toán ầu ra và mất mát:

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{x}^{(t)}$$
$$\boldsymbol{h}^{(t)} = \tanh\ \boldsymbol{a}^{(t)}$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)}$$
$$\hat{\boldsymbol{y}}^{(t)} = \sigma\ \boldsymbol{o}^{(t)}$$

Trong đó $\boldsymbol{b}$ và $\boldsymbol{c}$ là các độ lệch và $\boldsymbol{U}$,$\boldsymbol{V}$ và $\boldsymbol{W}$ là ma trận trọng số cho các kết nối đầu vào đến ẩn, kết nối ẩn đến đầu ra, các kết nối ẩn đến ẩn và σ là một hàm sigmoid. Toàn bộ mất mát cho một chuỗi các giá trị $\boldsymbol{x}$ và các giá trị $\boldsymbol{y}$ tương ứng nhận được bằng cách cộng các mất mát trên tất cả các bước:

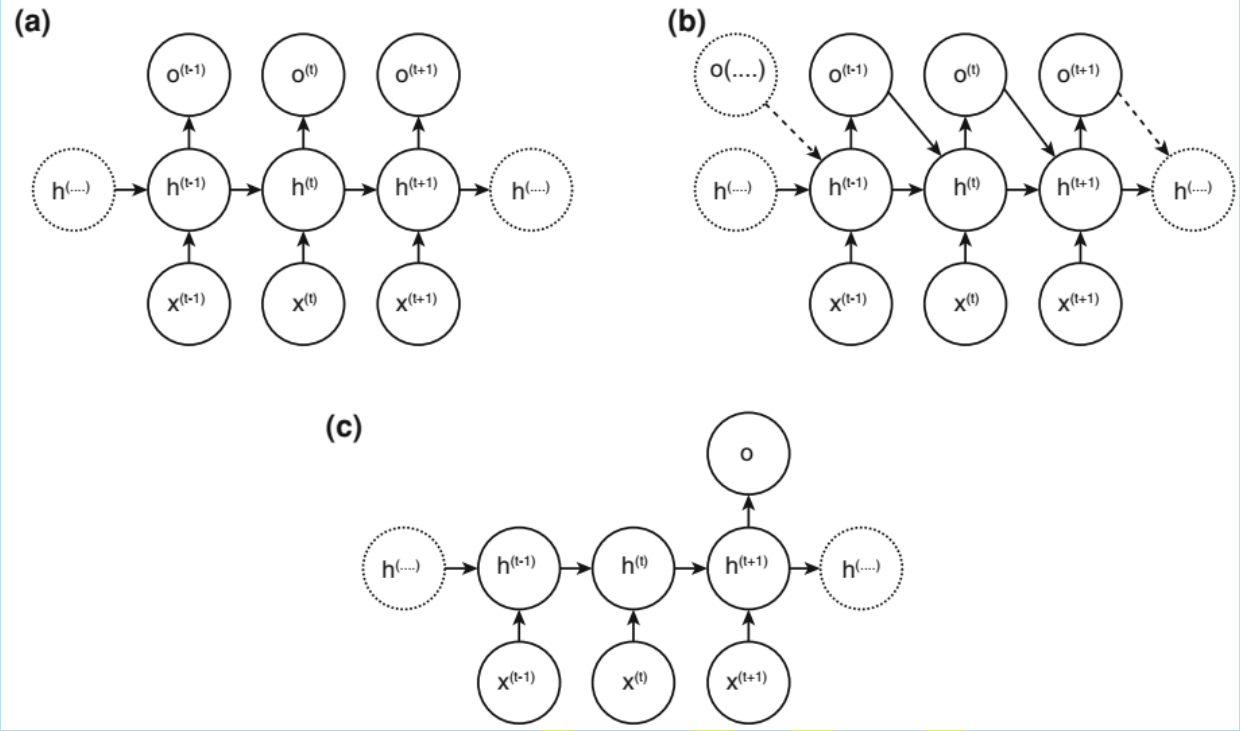
$$\sum_{t=1}^{\tau} \mathrm{L}^{(t)} = \mathrm{L}\ \ (\boldsymbol{x}^{(t)}, \ldots\ldots, \boldsymbol{x}^{(\tau)}), (\boldsymbol{y}^{(t)}, \ldots\ldots, \boldsymbol{y}^{(\tau)})$$

Để giảm thiểu mất mát, gradient của hàm mất mát được tính toán theo các tham số liên quan đến nó. Các tham số liên lết với các nút của graph tính toán là $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{x}^{(t)}$, $\boldsymbol{h}^{(t)}$, $\boldsymbol{o}^{(t)}$ và $\mathrm{L}^{(t)}$. Đầu ra $\boldsymbol{o}^{(t)}$ là đối số của hàm sofmax để nhận được vectơ $\hat{y}$ của các xác suất trên đầu ra. Trong quá trình truyền lan ngược, gradient cho từng nút được tính toán hồi quy bắt đầu bằng nút trước mất mát cuối cùng. Sa đó nó được lặp ngược hướng theo thời gian để truyền lan các gradient xuyên thời gian. tanh là một lựa chọn phổ biến cho một hàm kích hoạt và nó có xu hướng loại bỏ vấn đề gradient biến mất bằng cách duy trì giá trị khác không lâu hơn thông qua quá trình truyền lan ngược.

6.2.3. Các biến thể của SimpleRNN dựa trên chia sẻ tham số

Các biến thể của SimpleRNN có thể được thiết kế phụ thuộc và phong cách mở rộng của graph và chia sẻ tham số:

- Kết nối các đơn vị ẩn. RNN tạo ra các đầu ra tại mọi bước thời gian và các tham số được chuyển giữa các đơn vị ẩn (hình 6.2a). Điều này tương ứng với SimpleRNN tiêu chuẩn được trình bày ở trên và được sử dụng rộng rãi;
- Kết nối giữa các đầu ra và các đơn vị. RNN tạo ra các đầu ra tại mọi bước thời gian và các tham số được chuyển từ một đầu ra tại một bước thời gian cụ thể đến đơn vị ẩn tại thời gian tiếp theo (hình 6.2b);
- Đầu vào đến một đầu ra theo trình tự. RNN tạo ra một đầu ra tại đầu cuối sau khi đọc toàn bộ chuỗi và có kết nối giữa các đơn bị ẩn tại mọi bước thời gian (hình 6.2c).



Hình 6.2. Các kiểu kiến trúc SimpleRNN dựa trên chia sẻ tham số: (a) SimpleRNN với các kết nối giữa các đơn bị ẩn, ((b) SimpleRNN với các kết nối từ đầu ra đến các đơn vị ẩn, (c) SimpleRNN với các kết nối giữa các đơn vị ẩn, trong đó đọc toàn bộ chuỗi và tạo ra một đầu ra

6.2.4. Các biến thể của cấu trúc SimpleRNN dựa trên các đầu vào và các đầu ra

Các biến thể khác nhau cũng tồn tại dựa trên số lượng các đầu vào và các đầu ra:

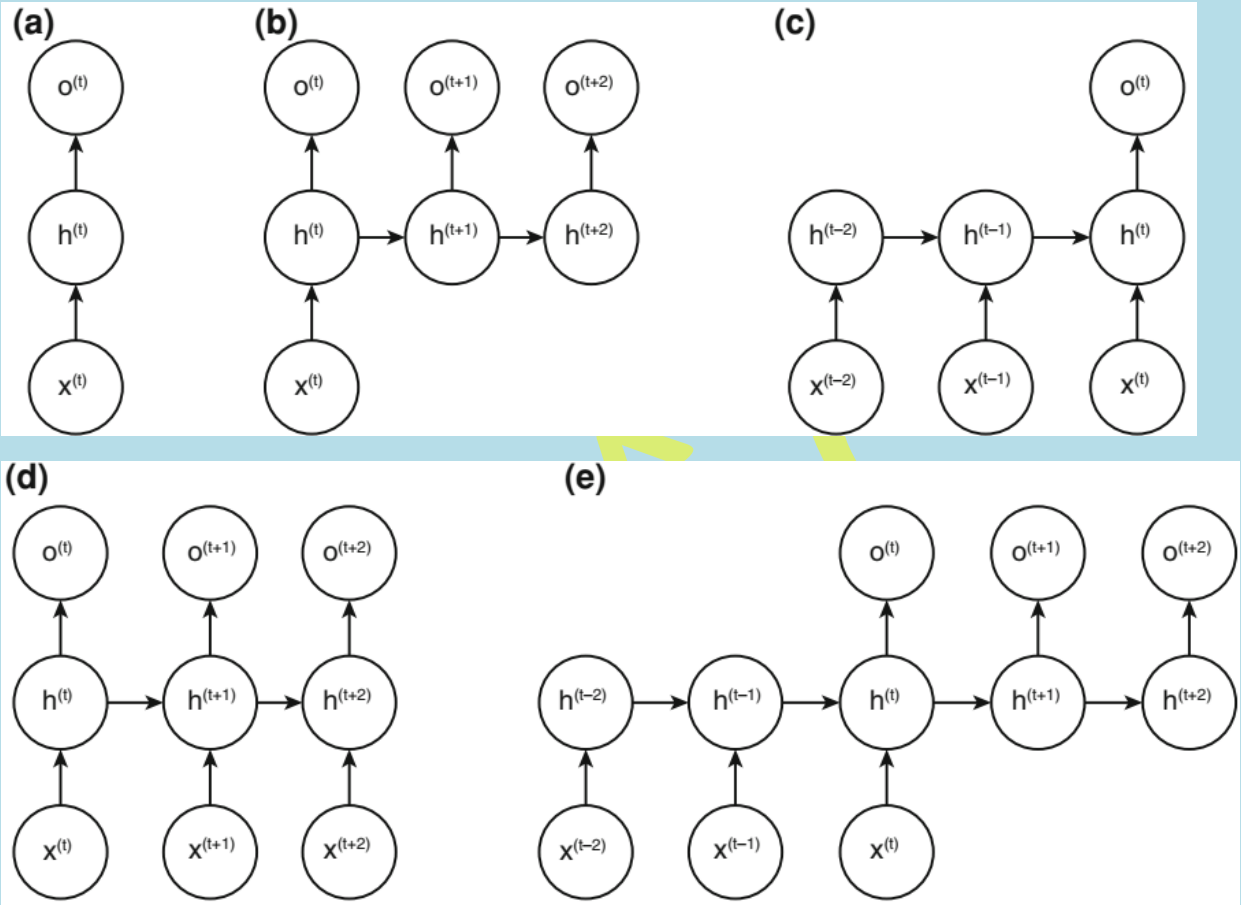
- Một đối một. RNN truyền thống có ánh xạ đầu vào đến đầu ra một đối một tại từng bước thời gian như được thể hiện trong hình 6.3a;
- Một đối nhiều. RNN một đối nhiều có một đầu vào tại một bước thời gian, tại đây nó tạo ra một chuỗi các đầu ra tại các bước thời gian liên tiếp như được thể hiện trong hình 6.3b. Kiểu cấu trúc RNN này thường được sử dụng cho mô tả hình ảnh (Image Captioning);
- Nhiều đối một. RNN nhiều đối một có nhiều đầu vào và một đầu ra tại mọi bước thời gian được thể hiện trong hình 6.3c. Kiểu kiến trúc RNN này được sử dụng cho phân loại văn bản;
- Nhiều đối nhiều. Kiến trúc RNN nhiều đối nhiều có thể được thiết kế theo hai cách. Cách thứ nhất, RNN nhận đầu vào và cho đầu ra tương ứng như minh họa

trong hình 6.3d. Kiểu RNN này được sử dụng cho nhận dạng thực thể có tên (Named Entity Recognition). Cách thứ hai, RNN nhận đầu vào tại từng bước thời gian và RNN cho đầu ra tại bước thời gian tiếp theo phụ thuộc vào toàn bộ chuỗi đầu vào như minh họa trong hình 6.3c. Kiểu kiến trúc RNN này được sử dụng phổ biến trong phiên dịch máy.

6.2.5. Các thách thức của các phụ thuộc dài hạn trong SImpleRNN

SimpleRNN làm việc tốt với các phụ thuộc ngắn hạn, nhưng khi nó chuyển đến các phụ thuộc dài hạn, nó không nhớ được thông tin dài hạn. Vấn đề này xuất hiện vì sự biến mất gradient hay bùng nổ gradient. Khi các gradient truyền lan trên nhiều tầng, nó có xu thế biến mất hầu hết thời gian hoặc đôi khi bùng nổ. Khó khăn này xuất hiện vì các trọng số giảm theo cấp số nhân được gán cho các tương tác dài hạn so với các tương tác ngắn hạn. Phải mất nhiều thời gian để học các phụ thuộc dài hạn vì các tín hiệu từ các phụ thuộc này có xu thế bị che dấu bởi các thăng giáng xất hiện từ các sự phụ thuộc ngắn hạn.

BẢN GỐC



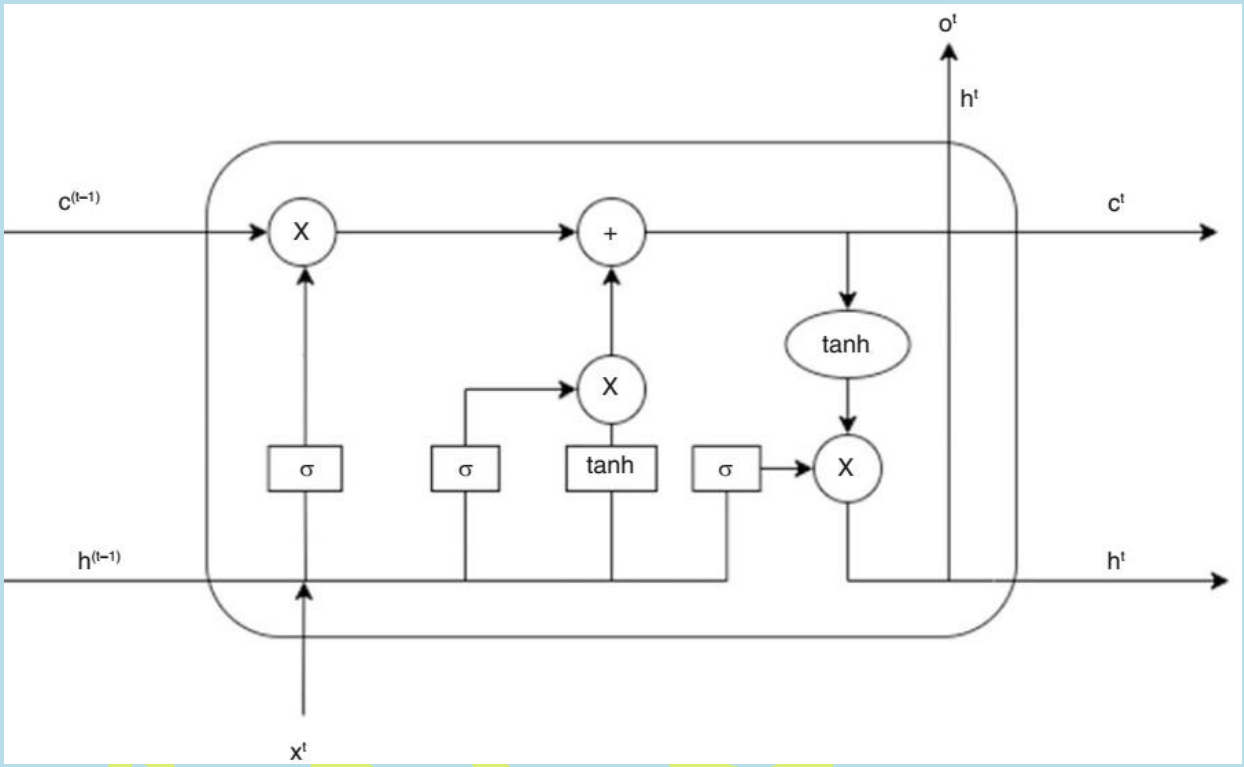
Hình 6.3. (a) RNN một đối một, (b) RNN một đối nhiều, (c) RNN nhiều đối một, (d) RNN nhiều đối nhiều. x trình bày đầu vào và o trình bày đầu ra

6.3. BỘ NHỚ DÀI NGẮN HẠN (LSTM) VÀ ĐƠN VỊ HỒI QUY CÓ CỔNG (GRU)

6.3.1. Bộ nhớ dài ngắn hạn (LSTM)

Để giải quyết vấn đề phụ thuộc dài hạn, mạng neuron hồi quy có cổng (Gated RNN) được đề xuất. Bộ nhớ dài ngắn hạn (LSTM: Long Short Term Memory) là một kiểu Gated RNN được đề xuất năm 1997. Vì tính chất nhớ được các sự phụ thuộc dài hạn, LSTM đã trở thành một mô hình thành công trong nhiều ứng dụng như nhận biết giọng nói, dịch máy, mô tả ảnh (Image Captioning), v.v. LSTM có một vòng lặp tự thân (Self Loop) bên trong bổ sung cho hồi quy bên ngoài của RNN. Các gradient trong vòng trong có thể chảy trong thời gian dài hơn và được đặt điều kiện theo ngữ cảnh thay vì bị cố định. Trong từng tế bào, đầu vào và đầu ra như nhau vì đây là của RNN gốc, nhưng có

một hệ thống các đơn vị mở cổng (Gating Unit) để điều khiển dòng chảy thông tin. Hình 6.4 cho thấy dòng chảy thông tin cùng với các đơn vị mở cổng.



Hình 6.4.Bộ nhớ dài ngắn hạn với trạng thái tế bào c^t, trạng thái ẩn h^t, đầu vào x^t và đầu ra o^t

Có ba cổng trong LSTM- cổng đầu vào bên ngoài (Exxternal Input Gate, cổng quên (Forget Gate) và cổng đầu ra (Output gate). Cổng quên tại thời điển t và trạng thái s_i f_i^t quyết định thông tin nào bị loại bỏ khỏi trạng thái tế bào. Cổng này điều khiển vòng lặp tự thân bằng cách đặt trọng số bằng 0 hoặc 1 thông qua hàm sigmoid σ. Khi giá trị này gần bằng 1, thông tin quá khứ được giữ lại và nếu giá trị này gần bằng 0, thông tin này bị loại. Sau cổng quên, trạng thái bên trong s_i^t được cập nhật. Tính toán cổng đầu vào bên ngoài giống như tính toán cổng quên bằng một hàm sigmoid để nhận được một giá trị giữa 0 và 1 nhưng bằng các tham số của chính nó. Cổng ngoài của LSTM cũng có một đơn vị sigmoid để quyết định có cho ra giá trị hay đóng không cho giá trị h_i^t đi qua cổng đầu ra q_i^t.

$$f_i^{(t)} = \sigma\left(\sum_j \boldsymbol{U}_{i,j}^f \boldsymbol{x}_j^t + \sum_j \boldsymbol{W}_{i,j}^f \boldsymbol{h}_j^{(t-1)} + \boldsymbol{b}_i^f \right)$$

$$s_i^{(t)} = f_i^t s_i^{(t-1)} + g_i^t \sigma\left(b_i + \sum_j \boldsymbol{U}_{i,j} \boldsymbol{x}_j^t + \sum_j \boldsymbol{W}_{i,j} \boldsymbol{h}_j^{(t-1)} \right)$$

$$g_i^t = \sigma\left(\boldsymbol{b}_i^g + \sum_j \boldsymbol{U}_{i,j}^g \boldsymbol{x}_j^t + \sum_j \boldsymbol{W}_{i,j}^g \boldsymbol{h}_j^{(t-1)} \right)$$

$$h_i^t = \tanh\ s_i^t\ \ \boldsymbol{q}_i^t$$

$$\boldsymbol{q}_i^t = \sigma\left(\boldsymbol{b}_i^o + \sum_j \boldsymbol{U}_{i,j}^o \boldsymbol{x}_j^t + \sum_j \boldsymbol{W}_{i,j}^o \boldsymbol{h}_j^{(t-1)} \right)$$

$\boldsymbol{x}^t$ là vectơ đầu vào tại thời điểm t, $\boldsymbol{h}^{(t)}$ và vectơ lớp ẩn , $\boldsymbol{b}_i$ ký hiệu cho các độ lệch (Bias), $\boldsymbol{U}_i$ và $\boldsymbol{W}_i$ thể hiện các trọng số đầu vào và các trọng số hồi quy.

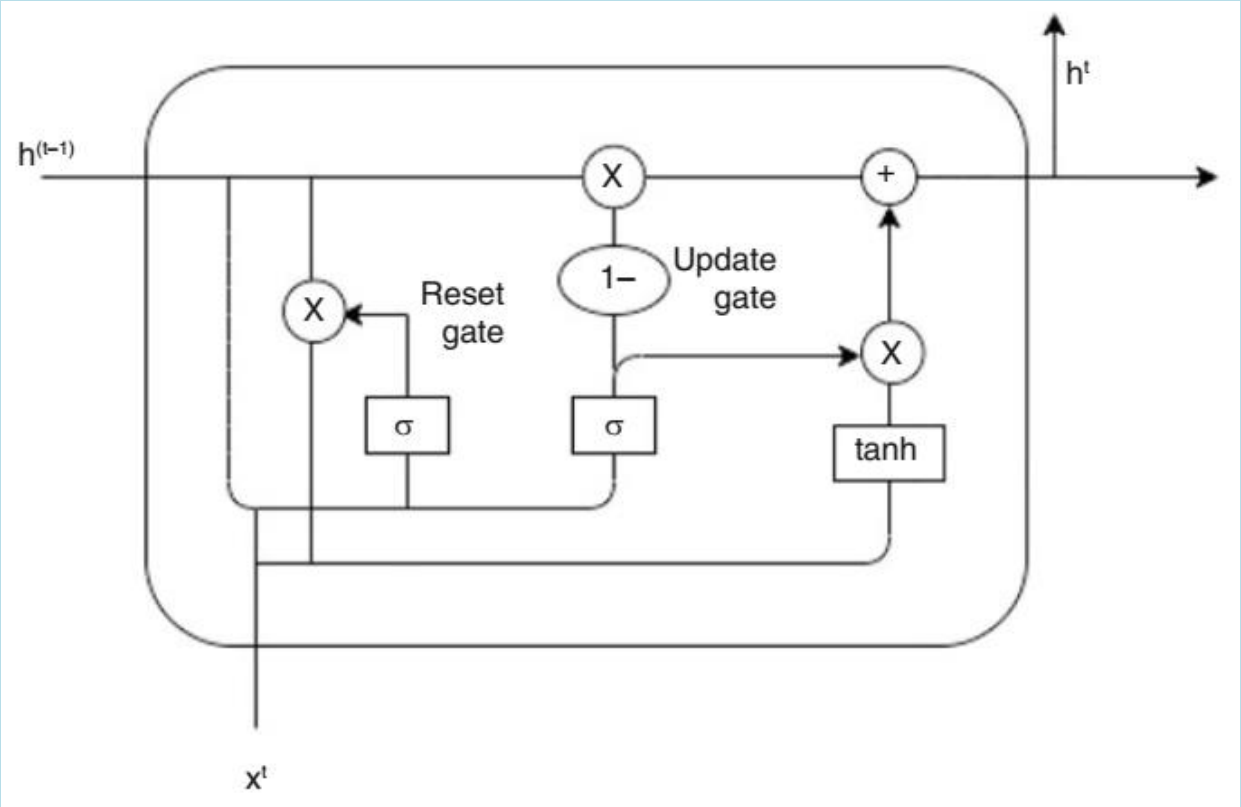
Khi giá trị của cổng khởi động lại (Reset Gate) bằn bằng 0, giá trị trạng thái ẩn trước đó bị loại và khởi động lại bằng giá trị hiện thời. Điều này cho phép trạng thái ẩn quên thông tin quá khứ vì nó không thích hợp cho tương lai. Cổng cập nhật (Update Gate) quyết định cần chuyển đi bao nhiêu thông tin qua khứ thích hợp cho tương lai.

Tính chất của cổng cập nhật chuyển tiếp thông tin quá khứ cho phép nó nhớ được các phụ thuộc dài hạn . Đối với các phụ thuộc ngắn hạn, cổng khởi động lại sẽ thường xuyên hoạt động để khởi động lại bằng giá trị hiện thời và loại bỏ các giá trị trước đó, trong khi đó, đối với các phụ thuộc dài hạn, cổng cập nhật sẽ thường xuyên hoạt động để chuyển tiếp thông tin trước đó.

6.3.2. Đơn vị hồi quy có cổng (GRU)

Trong LSTM, thời gian tính toán lớn do có nhiều tham số tham gia trong qua trình truyền lan ngược. Để giảm thời gian tính toán , đơn vị hồi quy có cổng (GRU: Gated Recurrent Unit) được đề xuất vào năm 2014 bởi Cho và các đồng tác giả khác có ít cổng hơn LSTM. Chức năng của GRU tương tự như chức năng của LSTM nhưng với một cấu trúc được cải biên. Hình 6.5 cho thấy sơ đồ biểu diễn GRU. Giống như LSTM, GRU cũng giải quyết vấn đề gradient biến mất và bùng nổ bằng cách ghi lại các phụ thuộc dài hạn với sự hỗ trợ của các đơn vị mở cổng. Có hai cổng trong GRU, cổng khởi động lại (Reset Gate) và cổng cập nhật (Update Gate). Cổng khởi động lại quyết định cần quên bao nhiêu

thông tin quá khứ và cổng cập nhật quyết định bao nhiêu thông tin qua khứ cần chuyển tiếp.



Hình 6.5. Mạng neuron hồi quy có cổng (GRU) với đầu vào x_i^t và đơn vị ẩn h_i^t

Tính toán tại cổng khởi động lại r_i^t và cổng cập nhật u_i^t, cũng như một trạng thái ẩn h_i^t tại thời điểm t, có thể được trình bày như sau:

$$r_i^t = \sigma\left(b_i^t + \sum_j U_{i,j}^r x_j^t + \sum_j W_{i,j}^r h_j^{(t)} \right)$$

$$u_i^t = \sigma\left(b_i^u + \sum_j U_{i,j}^u x_j^t + \sum_j W_{i,j}^u h_j^{(t)} \right)$$

$$h_i^t = u_i^{(t-1)} h_i^{(t-1)} + \left(1 - u_i\right) \times \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t-1)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)} \right)$$

Trong đó $\boldsymbol{b}_i$ ký hiệu cho các độ lêch, $\boldsymbol{U}_i$ và $\boldsymbol{W}_i$ ký hiệu cho các trọng số ban đầu và hồi quy.

Khi giá trị của cổng khởi động lại gần bằng 0, giá trị trạng thái ẩn trước đó bị xóa và khởi động lại bằng giá trị hiện thời. Điều này cho phép trạng thái ẩn quên thông tin quá khứ không thích hợp cho tương lai. Cổng cập nhật quyết định chuyển tiếp bao nhiêu thông tin quá khứ liên quan cho tương lai.

Tính chất của cổng chuyển tiếp thông tin qua khứ cho phép nó nhớ các phụ thuộc dài hạn. Đối với các phụ thuộc ngắn hạn, cổng khởi động lại sẽ thường xuyên hoạt động bằng các giá trị hiện thời và loại bỏ các giá trị trước đó, trong khi đó, đối với các phụ thuộc dài hạn, cổng cập nhật sẽ luôn luôn hoạt động để chuyển tiếp thông tin trước đó.

6.3.3. Lợi điểm của LSTM và GRU so với SimpleRNN

LSTM và GRU đều có thể xử lý vấn đề gradient biến mất của SimpleRNN với sự hỗ trợ của các đơn vị mở cổng (các đơn vị đóng mở). LSTM và GRU có tính năng cộng cho phép chúng giữ lại thông tin quá khứ bằng cách cộng thông tin quá khứ liên quan đến trạng thái hiện thời. Tính chất cộng này làm cho nó có thể nhớ được một tính năng đặc thù trong đầu vào trong thời gian dài hơn. Trong SimpleRNN, thông tin quá khứ mất sự liên quan của nó khi thông tin đầu vào được nhìn thấy. Trong LSTM và GRU, mọi tính năng quan trọng không bị viết đè bởi thông tin mới. Thay vào đó, nó được cộng với thông tin mới.

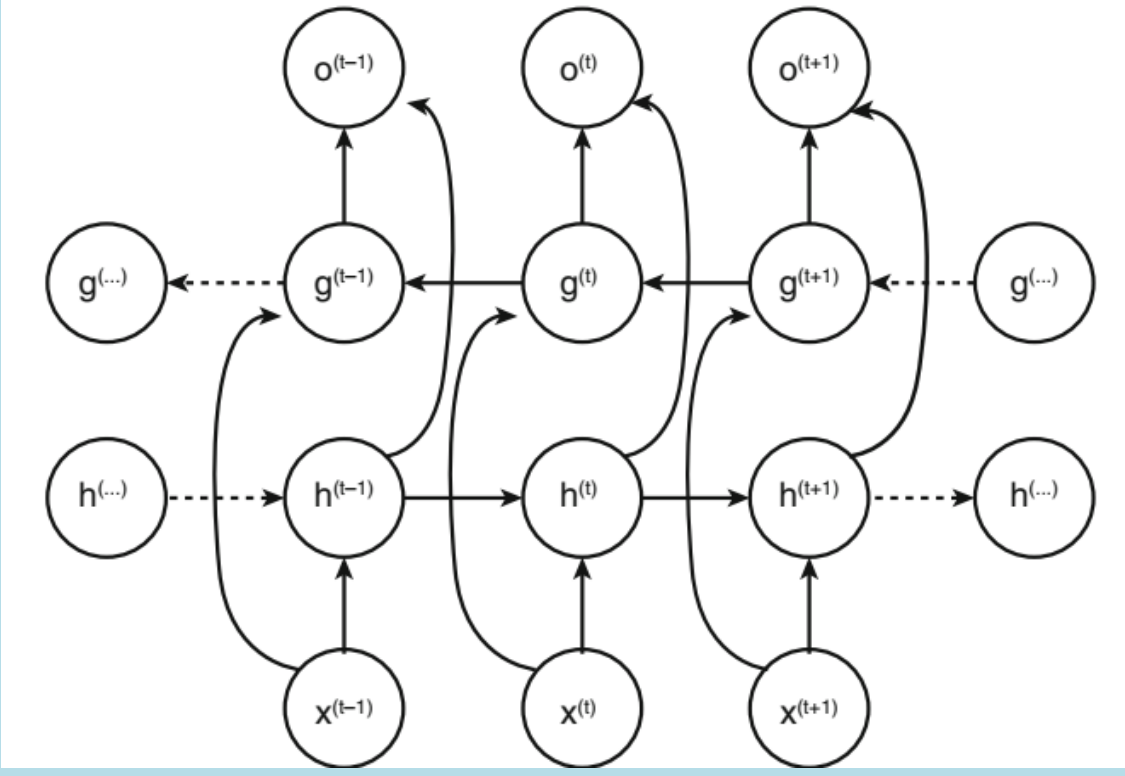
6.3.4. Các khác biệt giữa LSTM và GRU

Có một vài khác biệt giữa LTSM và GRU liên quan đến cơ chế mở cổng, cơ chế này đến lượt mình dẫn đến các khác biệt được quan sát thấy trong nội dung được tạo sinh. Trong đơn vị LSTM, khối lượng nội dung nhớ cần cho sử dụng của các đơn vị khác của mạng được điều chỉnh bởi cổng đầu ra, trong khi đó trong GRU, toàn bộ nội dung được để lộ cho các đơn vị khác. Một điểm khác biệt khác là LSTM tính toán nội dung nhớ mới không có điều khiển khối lượng dòng chảy thông tin trạng thái trước đó. Thay vào đó, nó điều khiển nội dung nhớ mới sẽ được cộng vào mạng. Trái lại, GRU điều khiển dòng thông tin quá khứ khi tính toán ứng cử mới mà không có điều khiển kích hoạt ứng cử.

6.4. RNN HAI CHIỀU (BRNN)

Trong simpleRNN, đầu ra của một trạng thái t chỉ phụ thuộc vào thông tin quá khứ $\boldsymbol{x}^{(1)}$, ……, $\boldsymbol{x}^{(t-1)}$ và đầu vào hiện tại $x^{(t)}$. Tuy nhiên, đối với nhiều ứng dụng chuỗi vào chuỗi (Seq2Seq: Sequence to Sequence), trạng thái đầu ra hiện thời phụ thuộc và toàn bộ thông tin chuỗi. Chẳng hạn, trong phên dịch ngôn ngữ, diễn giải đúng từ hiện thời phụ thuộc vào các từ quá khứ cũng như các từ tiếp theo. Để khắc phục hạn chế này của SimpleRNN, RNN hai chiều (BiRNN: Bidirectionnal RNN) được đề xuất bởi Schuster và Paliwal vào năm 1997.

Các BRNN kết hợp một RNN chuyển dịch về phía trước theo thời gian, bắt đầu từ khởi đầu chuỗi, với một RNN khác chuyển dịch ngược xuyên thời gian, bắt đầu từ cuối chuỗi. Hình 6.6 minh họa một BiRNN với $\boldsymbol{h}^{(t)}$ trạng thái của phần RNN (sub-RNN) chuyển dịch về phía trước và $\boldsymbol{g}^{(t)}$ trạng thái của t phần RNN dịch chuyển ngược chiều theo thời gian. Đầu ra của phần RNN chuyển động về phía trước không được nối đến đầu vào của phần RNN chyển động ngược chiều và ngược lại. Đầu ra $o^{(t)}$ phụ thuộc cả vào dữ liệu chuỗi quá khứ và dữ liệu chuỗi tương lai, nhưng nhạy cảm với các giá trị đầu vào xung quanh t.

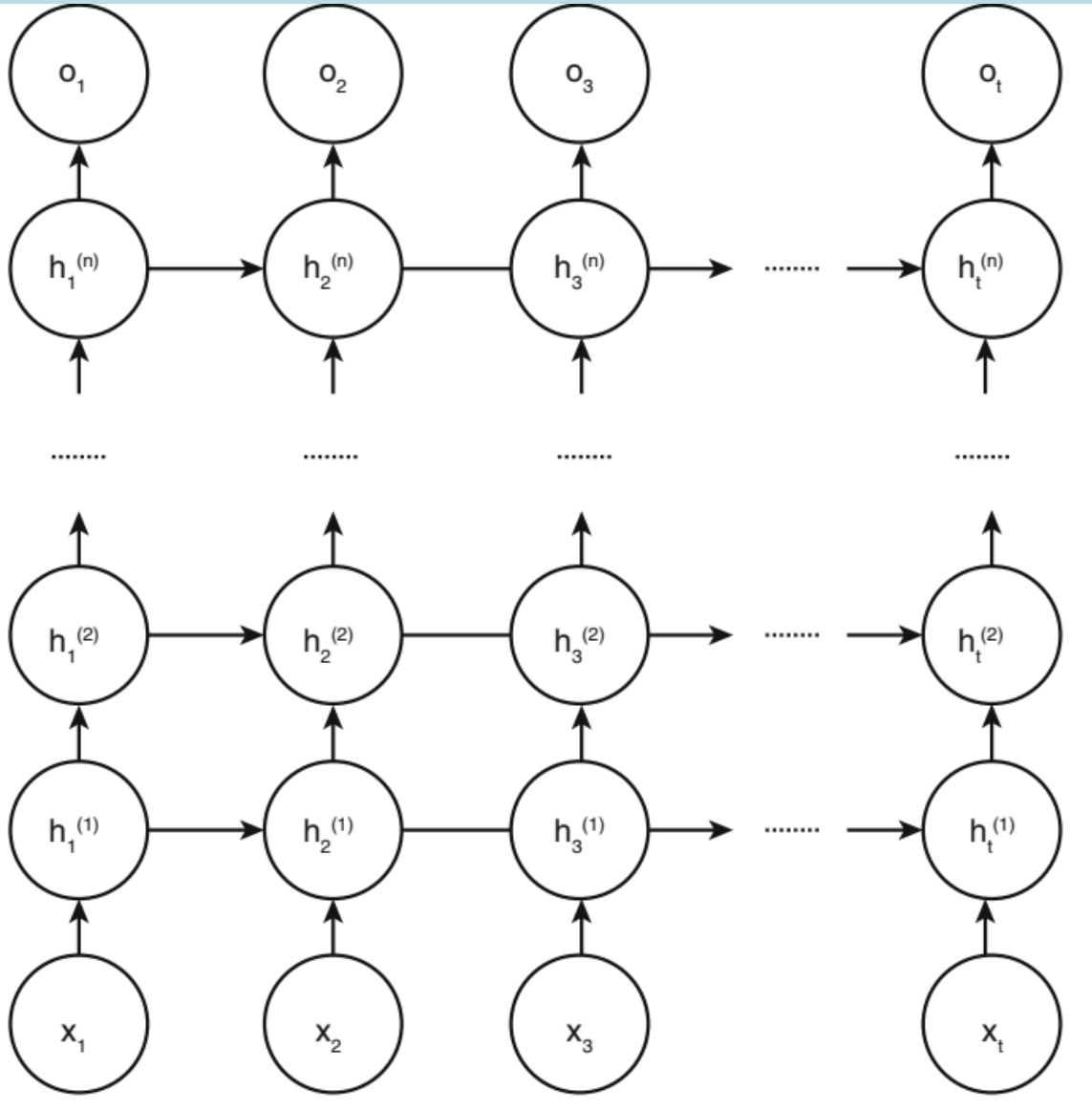


Hình 6.6. RRN hai chiều với phần RNN chuyển động thẳng có tạng thái ẩn $\boldsymbol{h}^{(t)}$ và phần RNN chuyển động ngược có trạng thái ần $\boldsymbol{g}^{(t)}$

6.5. RNN SÂU

Các RNN sâu (Deep RNN) hiệu quả hơn các đối tác tương đương của chúng và với một số giả thuyết, RNN sâu được đề xuất bởi Pascanu cùng với các đồng sự vào năm 2014. Trong RNN "nông", nói chung có ba khối tính toán các tham số: trạng thái đầu vào, trạng thái ẩn và trạng thái đầu ra. Các khối này được liên kết với một ma trận duy nhất tương ứng với một biến đổi nông. Biến đổi này có thể được trình bày một perceptron đa lớp (MLP) của một lớp đơn. Trong RNN sâu, ta có thể phân tách trạng thái của RNN vào nhiều lớp. Hình 6.7 cho thấy tổng quan một RNN sâu với các MLP đa lớp. Tuy nhiên, các kiểu khác nhau của một RNN sâu có thể được xét tách biệt như đầu vào- ẩn, ẩn- ẩn và ẩn- đầu ra. Lớp thấp hơn trong phân cấp này có thể biến đổi đầu vào thành một trình bày tương ứng cho mức cao hơn của trạng thái ẩn. Trong trạng thái ẩn- ẩn , ta có thể cấu trúc nó bằng một trạng thái ẩn trước đó và một đầu vào mới. Điều này đưa vào một phi tuyến bổ sung trong kiến trúc dẫn đến dễ ràng hơn thích nghi nhanh chóng với các chế độ thhy đổi của đầu vào. Việc đưa RNN sâu vào trạng thái ẩn - đầu ra làm cho lớp này nhỏ gọn hơn, cho phép hỗ trợ tóm tắt các đầu vào trước đó và hỗ trợ dự đoán dễ ràng đầu ra. Vì MLP sâu trong kiến trúc RNN, học trở nên chậm và tối ưu bị khó.

BẢN GỐC



Hình 6.7. Mạng neuron hồi quy sâu

6.6. BỘ MÃ HÓA-BỘ GIẢI MÃ (ENCODER-DECODER)

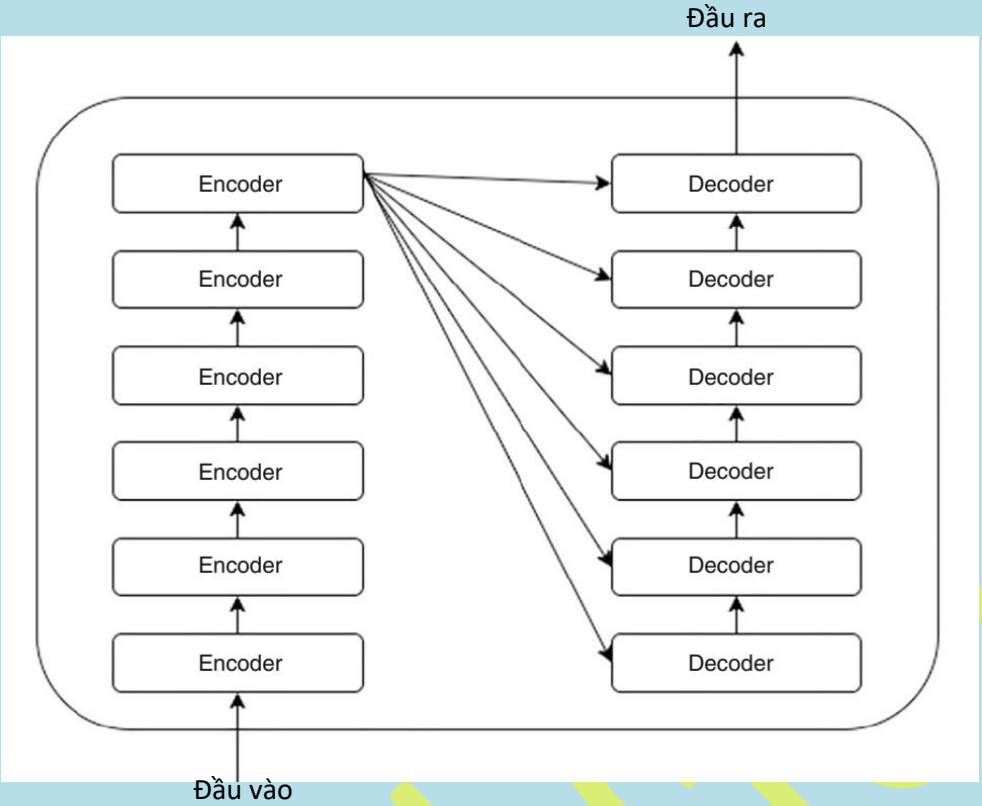
Kiến trúc bộ mã hóa-bộ giải mã được Cho và các đồng sự đề xuất (2014) để ánh xạ một chuỗi đầu vào có độ dài khả biến vào một chuỗi đầu ra có độ dài khả biến. Vì thế, nó còn được gọi là kiến trúc chuỗi vào chuỗi (Seq2Seq: Sequence to Sequence) . Trước khi đưa ra bộ mã hóa-bộ giải mã, đã có các môt hình RNN được sử dụng cho Seq2Seq, nhưng chúng có các hạn chế vì các chuỗi đầu và và các chuỗi đầu ra không có cùng độ dài. Bộ mã hóa-bộ giải mã được sử dụng để xử lý các vấn đề Seq2Seq có độ dài khả biến chẳng

hạn phiên dịch máy hay nhận dạng giọng nói, tại đây độ dài của các chuỗi đầu vào và đầu ra không giống nhau trong hầu hết các trường hợp. Bộ mà hóa và bộ giải mã cả hai đều là các RNN, trong đó RNN mã hóa mã hóa toàn bộ đầu vào $\boldsymbol{x} = x_1, \ldots\ldots, x_{n_x}$ vào một vectơ ngữ cảnh $\boldsymbol{c}$ và các đầu ra của vectơ ngữ cảnh $\boldsymbol{c}$ này được đưa đến một đầu vào của RNN giải mã. RNN giải mã tạo ra một chuỗi đầu ra $\boldsymbol{y} = y_1, \ldots\ldots, y_{n_y}$. Trong mô hình mã hóa-giải mã, độ dài đầu vào n_x và độ dài đầu ra n_y có thể khác nhau không giống như các RNN trước đây. Không nhất thiết số lượng các lớp ẩn trong bộ mã hóa và bộ giải mã phải như nhau. Hạn chế của kiến trúc này là không thể tổng kết chính xác một chuỗi dài nếu vectơ ngữ cảnh quá ngắn. Vấn đề này được giải quyết bởi Bandanau và các cộng sự (2015) bằng cách làm cho vectơ ngữ cảnh trở thành một chuỗi có độ dài khả biến bằng một cơ chế chú ý (Attention Mechanism) bổ sung.

6.7. CÁC MÔ HÌNH CHÚ Ý (ATTENTION MODELS) HAY CÁC BỘ BIẾN ĐỔI (TRANSFOMERS)

Vì cơ chế học tuần tự, vectơ ngữ cảnh được tạo bởi bộ mã hóa tập trung nhiều hơn lên phần sau của chuỗi thay vì phần đầu. Một mở rộng của mô hình bộ mã hóa-bộ giải mã được đề xuất bởi Bandanau và các cộng sự cho phiên dịch máy trong đó mô hình tạo ra từng từ dựa trên thông tin liên quan nhất trong câu nguồn và các từ được tạo trước đó. Không giống như mô hình bộ mã hóa-bộ giải mã trước đây khi mà toàn bộ chuỗi đầu vào được mã hóa vào một vectơ ngữ cảnh duy nhất, mô hình bộ mã hóa-giải mã được mở rộng này học để đưa ra chú ý cho các từ liên quan có mặt trong chuỗi nguồn bất kể vị trí trong chuỗi bằng cách mã hóa chuỗi đầu vào vào các chuỗi vectơ và chọn một cách có chọn lọc khi giải mã từng từ. Cơ chế này chú ý đến thông tin liên qua đến từng từ và được gọi là cơ chế chú ý (Attention Mechanism).

Mặc dù mô hình này giải quyết vấn đề cho các vectơ ngữ cảnh có độ dài cố định, vấn đề giải mã trình tự vẫn tồn tại. Để giải mã chuỗi trong khoảng thời gian ngắn hơn bằng cách đưa vào cấu trúc song song, nhóm Google Brain, Ashish và cộng sự, đề suất chú ý tự thân (Self Attention). Họ phát kiến mô mình bộ biến đổi (Transformer Model) dựa trên cơ chế chú ý tự thân được thiết kế để giảm thời gian tính toán. Nó tính toán trình bày của một chuỗi liên quan đến các phần khác nhau của cùng một chuỗi. Cơ chế chú ý tự thân được nhúng trong mô hình Transformer. Mô hình Transformer có một ngăn xếp sáu lớp giống nhau, mỗi lớp cho mã hóa và giải mã chuỗi được minh họa trong hình 6.8.



Hình 6.8. Transformer với sáu lớp của các bộ mã hóa và sáu lớp của các bộ giải mã

Mỗi lớp của bộ mã hóa và bộ giải mã có các lớp con (Sub-Layer) bao gồm cơ chế chú ý tự thân nhiều đầu (Multi-head Self Attention Mechanism) và các lớp được kết nối hoàn toàn theo vị trí. Có một kết nối dư xung quan hai lớp con và chuẩn hóa tiếp sau. Ngoài hai lớp con này, còn có một lớp thứ ba trong bộ giải mã thực hiện chú ý nhiều đầu (Multi-head Attention) trên đầu ra của ngăn xếp bộ mã hóa. Trong bộ giải mã, chũ ý nhiều đầu được che giấu để ngăn ngừa vị trí gây chú ý phần sau của chuỗi. Điều này đảm bảo dự đoán đối với một vị tri p chỉ phụ thuộc vào các vị trí nhỏ hơn p trong chuỗi. Có thể mô tả hàm chú ý như là ánh xạ các cặp truy vấn (query) và giá trị khóa (Key-value) vào một đầu ra. Tất cả các tham số tham gia vào tính toán đều là các vectơ. Để tính toán đầu ra, phép toán tích vô hướng được thực hiện trên truy vấn và tất cả các khóa, và chia từng khóa bởi $\sqrt{d_k}$ (trong đó d_k là kích thước của các khóa). Cuối cùng, softmax được áp dụng để nhận được các trọng số trên các giá trị. Tính toán hàm chú ý được trình bày bằng phương trình sau:

$$\text{Attention}\ \ \boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\ \ = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)V$$

trong đó ***Q***, ***K*** và ***V*** là các ma trân tương ứng với truy vấn, các khóa và các giá trị. Cụ thể hơn về phạm vi bao phủ sâu của Transformer được trình bày trong chương 8.

6.8. CÁC MẸO VÀ THỦ THUẬT

Như đã nói trước đây, các gradient biến mất và gradient bùng nổ là các lo ngại phổ biến nhất, khi nó chuyển đến đào tạo đúng đắn các mô hình RNN. Thách thức cơ bản này xẩy ra từ thực tế là các RNN có thể được mở rộng một cách tự nhiên để cho phép các kết nối hồi quy thực hiện các tính toán nối thẳng, gây hậu quả trong một RNN có số lượng các lớp và số lượng các phần tử trong chuỗi như nhau. Hai vấn đề chính xuất hiện từ hậu quả này là:

- Vấn đề gradient biến mất (Vanishing Gradient). Khó khăn khi học các phụ thuộc dài hạn trong các chuỗi do vấn đề biến mất gradient. Hậu quả là, một dự đoán mô hình có khả năng xẩy ra trong tương lai sẽ hầu như không bị ảnh hưởng của các lớp trước.
- Vấn đề bùng nổ gradient (Exploding Gradient). Bổ sung nhiều hơn các lớp vào mạng sẽ khuếch đại ảnh hưởng của các gradient lớn làm tăng rủy do học lệch (Learning Derailment) do các thay đổi đáng kể các trọng số mạng có thể được thực hiện tại từng bước, có tiềm năng làm cho các gradient bùng nổ theo cấp số nhân. Thực tế, các trọng số gần lớp đầu vào hơn sẽ nhận được các cập nhật lớn hơn so với các trọng số gần các lớp đầu ra hơn. Và mạng có thể trở nên không ổn định để học các liên quan giưã các sự kiện cách xa tạm thời.

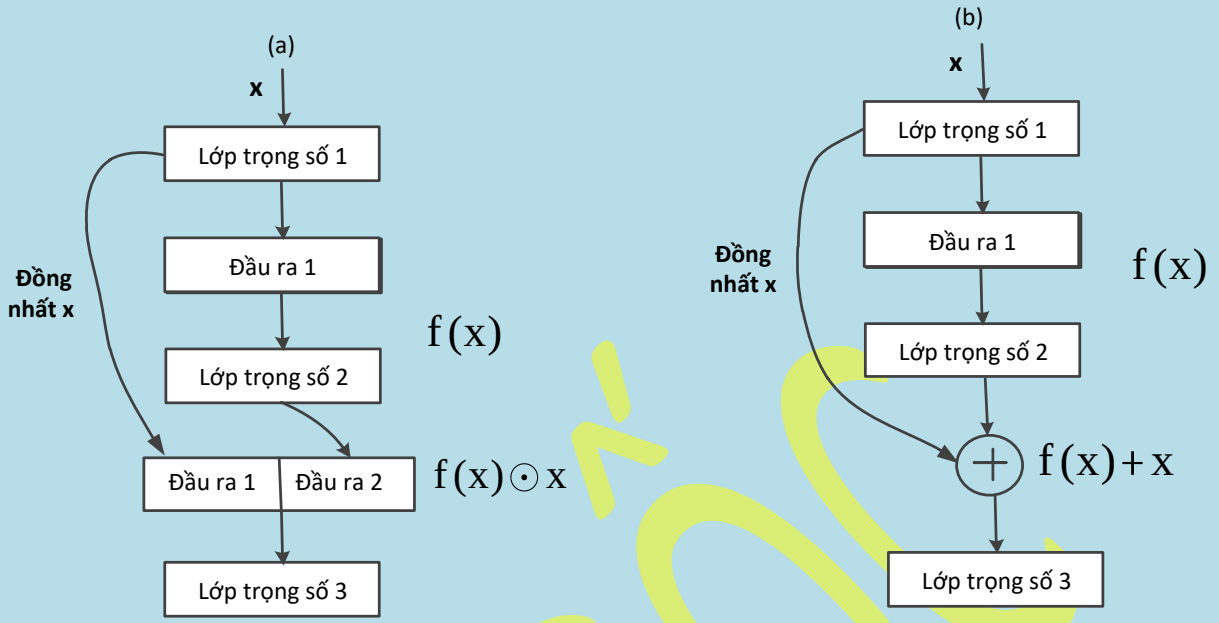
Để khắc phục các hạn chế này, chúng ta cần tao ra các giải pháp sao cho mô hình RNN có thể làm việc tại các thang thời gian khác nhau, với một số đoạn hoạt động trên các thang đo thời gian hạt mịn, xử lý các chi tiết nhỏ và một số khác hoạt động trên các thang thời gian thô, truyền hiệu quả thông tin thừ quá khứ xa đến hiện tại. Trong phần này ta sẽ xét một số chiến lược phổ biến để xử lý các vấn đề này.

6.8.1. Kết nối bỏ qua (Skip Connection)

Luyện tập nhảy qua các lớp cho phép đơn giản hiệu quả mạng bằng cách sử dụng ít hơn các lớp được kết nối trực tiếp trong các tầng đào tạo đầu. Điều này tăng tốc đào tạo bằng cách giảm nhẹ ảnh hưởng của các gradient biến mất, vì các lớp phải truyền qua sẽ ít hơn. Vì mạng học không gian tính năng trong giai đoạn đào tạo, nó sẽ từng bước khôi phục lại các lớp bị bỏ qua. Lin và các cộng sự đã đề xuất sử dụng các kết nối bỏ qua như vậy dựa

trên ý tưởng kết hợp các trễ (Delay) vào các mạng neuron truyền thẳng (Feeeforward Neural Network) của Lang và các cộng sự. Về mặt khái niệm, các kết nối bỏ qua là một module tiêu chuẩn trong các kiến trúc sâu và thường được đề cập là các mạng dư (Residual Network). Chúng chịu trách nhiệm bỏ qua các lớp trong mạng neuron và cung cấp đầu ra của một lớp như là đầu và của các lới tiếp theo. Kỹ thuật này được sử dụng để cho phép các gradient trực tiếp đi qua mạng mạng mà không cần đưa qua các hàm kích hoạt phi tuyến tính và thực nghiệm chứng minh các bước bổ sung này thường có lợi cho hội tụ mô hình. Các kết nối bỏ qua có thể được sử dụng thông qua lớp không theo thứ tự theo hai cách cơ bản trong các mạng neuron:

- Các kết nối bỏ qua cộng (Adđitive Skip Connection). Trong kiểu thiết kế này, dữ liệu từ các lớp trước được truyền tải đến các lớp sâu hơn thông qua cộng ma trận, dẫn đến truyền lan ngược được thực hiện thông qua cộng (hình 6.9b). Thủ tục này không đòi hỏi bất kỳ tham số bổ sung nào vì đầu ra từ lớp trước được cộng trước đến lớp này. Một trong các kỹ thuật phổ biến nhất được sử dụng trong kiểu kiến trúc này là xếp chồng các khối dư bỏ qua với nhau và một sử dụng một hàm đồng nhất (Identity Function) để bảo tồn gradient. Lõi của khái niệm này là sử dụng một cộng vectơ vào truyền lan ngược thông qua hàm đồng nhất. Sau đó, đơn giản nhân gradient này với một và giá trị của nó được bàn tồn trong các lớp trước.
- Các kết nối bỏ qua móc nối (Concatenative Skip Connection). Một cách khác để thiết lập các kết nối bỏ qua là móc nối các bản đồ tính năng trước đó. Mục đích của móc nối là tận dụng các đặc trưng nhận được trong các lớp trước để làm sâu hơn các lớp. Ngoài ra, các kết nối bỏ qua móc nối cung cấp một chiến lược xen kẽ (Alternate Strategy) để đảm bảo khả năng tái sử dụng cùng một kích thước từ các lớp trước mà không cần học các bản đồ sao chép. Hình 6.9(a) minh họa thí dụ sơ đồ để thấy kiến trúc này trông giống như thế nào. Khái niệm đầu tiên của kiến trúc này cho phép các lớp tiếp theo tái sử dụng các biểu diễn trung gian, cho phép chúng duy trì nhiều thông tin hơn và tăng cường hiệu suất phụ thuộc dài hạn.



Hình 6.9. Các kiến trúc kết nối bỏ qua: (a) móc nối đầu ra của lớp trước và kết nối bỏ qua; (b) cộng đầu ra của lớp trước và kết nối bỏ qua

6.8.2. Các đơn vị dò rỉ (Leaky Units)

Một trong những thách thức lớn khi đào tạo RNN là nắm bắt các phụ thuộc dài hạn và truyền tải thông tin hiệu quả từ quá khứ xa đến hiện tại. Một phương pháp hiệu quả để có được thang thời gian thô là sử dụng các đơn vị rò rỉ là các đơn vị ẩn với các kết nối tự thân tuyến tính (Linear Self Connection) và trọng số trên các kết nối gần bằng một. Trong RNN rò rỉ, các đơn vị ẩn có thể truy cập các giá trị từ các trạng thái trước đó và có thể được sử dụng để có được các biểu diễn thời gian. Công thức $\boldsymbol{h}_t = \alpha \times \boldsymbol{h}_{t-1} + (1 - \alpha) \times \boldsymbol{h}_t$ thể hiện quy tắc cập nhật trạng thái của một đơn vị rò rỉ, trong đó $\alpha \in (0, 1)$ là một ví dụ về kết nối tự thân tuyến tính từ $\boldsymbol{h}_{t-1}$ đến $\boldsymbol{h}_t$ và là một tham số cần được học trong giai đoạn đào tạo. Về cơ bản, α kiểm soát dòng thông tin trong trạng thái. Khi α gần bằng 1, trạng thái gần như không thay đổi và thông tin về quá khứ được lưu giữ trong thời gian dài, còn khi α gần bằng 0, thông tin về quá khứ nhanh chóng bị loại bỏ và trạng thái phần lớn được thay thế bằng trạng thái mới $\boldsymbol{h}_t$.

6.8.3. Cắt các gradient (Clipping Gradient)

Cắt gradient là một kỹ thuật nhằm khắc phục vấn đề gradient bùng nổ trong huấn luyện RNN, bằng cách giới hạn các chuẩn gradient (eleement-wise: theo từng phần tử) ở một giá trị ngưỡng tối thiểu hoặc tối đa được xác định trước, khi các gradient bùng nổ bị cắt

và quá trình tối ưu hóa bắt đầu hội tụ về điểm cực tiểu. Cắt gradient có thể được sử dụng theo hai cách cơ bản:

- **Cắt theo giá trị (Clipping by Value).** Bằng cách sử dụng kỹ thuật này, tai xác định giá trị cắt tối thiểu và giá trị cắt tối đa. Nếu gradient vượt quá giá trị ngưỡng, ta cắt gradient xuống ngưỡng tối đa. Nếu gradient nhỏ hơn giới hạn dưới của ngưỡng, ta cắt gradient xuống ngưỡng tối thiểu.
- **Cắt theo chuẩn (Cuting by Norm).** Ý tưởng đằng sau kỹ thuật này rất giống với cắt theo giá trị. Điểm khác biệt chính là chúng ta cắt các gradient bằng cách nhân vectơ đơn vị của gradient với ngưỡng. Gradient descent sẽ có thể hoạt động bình thường ngay cả khi mô hình mất mát không đều vì các cập nhật trọng số cũng sẽ được điều chỉnh lại tỷ lệ. Điều này làm giảm đáng kể khả năng tràn hoặc thiếu dữ liệu của mô hình.

6.8. CÁC ỨNG DỤNG CỦA RNN TRONG LẬP MÔ HÌNH NGÔN NGỮ

Lập mô hình ngôn ngữ (Languge Modelling) là quá trình học các biểu diễn vectơ có ý nghĩa cho ngôn ngữ hoặc văn bản bằng cách sử dụng thông tin chuỗi và thường được đào tạo để dự đoán mã thông báo (Token) hoặc từ tiếp theo cho chuỗi mã thông báo hoặc từ đầu vào. Bengio et al. đã đề xuất một bộ khung cho mô hình hóa ngôn ngữ dựa trên mạng neuron. Kiến trúc RNN đặc biệt phù hợp để xử lý ngôn ngữ tự nhiên trôi chảy do bản chất tuần tự của nó. Như được mô tả bởi Mikolov và các cộng sự, RNN có thể học cách nén toàn bộ chuỗi trái ngược với mạng neuron truyền thẳng chỉ nén một mục đầu vào duy nhất. Mô hình hóa ngôn ngữ có thể là một nhiệm vụ độc lập hoặc là một phần của đường ống xử lý ngôn ngữ với nhiệm vụ dự đoán hoặc phân loại hạ lưu. Trong phần này, chúng ta sẽ xét các ứng dụng của RNN cho các nhiệm vụ xử lý ngôn ngữ khác nhau.

6.8.1. Phân loại văn bản (Text Classification)

Nhiều ứng dụng thực tế thú vị liên quan đến dữ liệu ngôn ngữ có thể được mô hình hóa dưới dạng phân loại văn bản. Ví dụ bao gồm phân loại cảm xúc, nhận dạng chủ đề hoặc tác giả, và phát hiện thư rác với các ứng dụng từ tiếp thị đến trả lời truy vấn. Nhìn chung, các mô hình phân loại văn bản bao gồm một số lớp RNN để xử lý văn bản đầu vào tuần tự. Việc nhúng dữ liệu đầu vào học được bởi các lớp này sau đó được xử lý thông qua các lớp phân loại khác nhau để dự đoán nhãn lớp cuối cùng. Kiến trúc RNN nhiều đối một (Many-to-One) thường được sử dụng để phân loại văn bản.

Là một cải tiến kỹ thuật gần đây, RNN đã được kết hợp với mạng neuron tích chập (CNN), do đó kết hợp các điểm mạnh của hai kiến trúc, để xử lý dữ liệu văn bản cho các tác vụ phân loại. LSTM là kiến trúc RNN phổ biến để xử lý dữ liệu văn bản do khả năng theo dõi các mẫu trên các chuỗi dài, trong khi CNN có khả năng học các mẫu không gian

từ dữ liệu có hai hoặc nhiều chiều. LSTM tích chập (C-LSTM) kết hợp hai kiến trúc này để tạo thành một kiến trúc mạnh mẽ có thể học các mẫu cấp cụm từ cục bộ cũng như các mẫu cấp câu toàn cục. Trong khi CNN có thể học các đặc điểm cục bộ và bất biến vị trí và RNN giỏi trong việc học các mẫu toàn cục, một biến thể khác của RNN đã được đề xuất để đưa việc học đặc điểm cục bộ bất biến vị trí vào RNN. Biến thể này được gọi là RNN bị ngắt kết nối (DRNN: Disconneted RNN)). Dòng thông tin giữa các mã thông báo (Token) hoặc các từ ở lớp ẩn bị giới hạn bởi một siêu tham số gọi là *kích thước cửa sổ*, cho phép nhà phát triển chọn *độ rộng* của *ngữ cảnh* cần xem xét khi xử lý văn bản. Kiến trúc này đã cho thấy hiệu suất tốt hơn cả RNN và CNN trên một số nhiệm vụ phân loại văn bản.

6.8.2. Tóm tắt văn bản (Text Summarization)

Các phương pháp tóm tắt văn bản có thể được phân loại thành (1) tóm tắt trích xuất (Extractive Summarization) và (2) tóm tắt trừu tượng (Abstractive Summarization). Phương pháp thứ nhất dựa trên việc lựa chọn hoặc trích xuất các câu sẽ là một phần của bản tóm tắt, trong khi phương pháp thứ hai tạo ra văn bản mới để xây dựng bản tóm tắt. Kiến trúc RNN đã được sử dụng cho cả hai loại kỹ thuật tóm tắt.

6.8.2.1. Tóm tắt văn bản trích xuất

Các bộ khung tóm tắt trích xuất sử dụng RNN nhiều-đến-một làm bộ phân loại để phân biệt các câu nên là một phần của bản tóm tắt. Ví dụ, kiến trúc RNN hai lớp, trong đó một lớp xử lý các từ trong một câu và lớp kia xử lý nhiều câu theo trình tự. Mô hình tạo ra các nhãn cấp câu cho biết câu đó có nên là một phần của bản tóm tắt hay không, do đó tạo ra một bản tóm tắt trích xuất của tài liệu đầu vào. Xu và cộng sự đã trình bày một mô hình tóm tắt trích xuất phức tạp hơn không chỉ trích xuất các câu để trở thành một phần của bản tóm tắt mà còn đề xuất nén cú pháp (Syntactic Compression) có thể có cho các câu đó. Kiến trúc được đề xuất của họ là sự kết hợp giữa CNN và LSTM hai chiều, trong khi bộ phân loại neuron đánh giá các khả năng nén cú pháp trong ngữ cảnh của câu cũng như ngữ cảnh rộng hơn của tài liệu.

6.8.2.2. Tóm tắt văn bản trừu tượng

Các bộ khung tóm tắt trừu tượng kỳ vọng RNN sẽ xử lý văn bản đầu vào và tạo ra một chuỗi văn bản mới, là bản tóm tắt của văn bản đầu vào, sử dụng hiệu quả RNN nhiều-nhiều (Many-to-Many) làm mô hình tạo văn bản. Mặc dù việc đạt được độ chính xác ngữ pháp cơ bản tương đối dễ dàng so với các bộ tóm tắt trích xuất (Exractive Summarizer) khi các câu đúng được chọn từ tài liệu để tạo bản tóm tắt, nhưng đây vẫn là một thách thức lớn đối với các bộ tóm tắt trừu tượng (Abstractive Summarizer). Độ chính xác ngữ pháp phụ thuộc vào chất lượng của mô-đun tạo văn bản. Độ chính xác ngữ pháp của các bộ tóm tắt văn bản trừu tượng đã được cải thiện gần đây nhờ những phát triển trong xử lý

văn bản theo ngữ cảnh, mô hình hóa ngôn ngữ, cũng như khả năng tính toán để xử lý khối lượng lớn văn bản.

Xử lý các token/từ hiếm là mối quan tâm chính đối với các trình tóm tắt trừu tượng hiện đại. Ví dụ, các danh từ riêng như tên người và địa điểm cụ thể ít xuất hiện hơn trong văn bản; tuy nhiên, các bản tóm tắt được tạo ra sẽ không đầy đủ và khó hiểu nếu các token đó bị bỏ qua. Nallapati và cộng sự đã đề xuất một giải pháp mới bao gồm các lớp GRU-RNN với cơ chế chú ý bằng cách đưa bộ giải mã chuyển mạch vào kiến trúc bộ tóm tắt trừu tượng của họ, trong đó mô-đun tạo văn bản có một chuyển mạch cho phép mô-đun lựa chọn giữa hai tùy chọn: (1) tạo một từ từ vốn từ vựng và (2) trỏ đến một trong các từ trong văn bản đầu vào. Mô hình của họ có khả năng xử lý các token hiếm bằng cách trỏ đến vị trí của chúng trong văn bản gốc. Họ cũng sử dụng *thủ thuật từ vựng lớn* (large vocabulary trick) để giới hạn từ vựng của mô-đun tạo sinh (Generrator Module) chỉ đến các token của văn bản nguồn và sau đó thêm các token thường xuyên vào tập từ vựng cho đến khi kích thước của nó đạt đến một ngưỡng nhất định. Thủ thuật này hữu ích trong việc giới hạn kích thước của mạng.

Tóm tắt có thông tin cấu trúc tiềm ẩn, nghĩa là chúng truyền tải thông tin theo một số cấu trúc ngôn ngữ nhất định như "Chuyện gì đã xảy ra" hoặc "Ai-Hành động-Cái gì". Li và cộng sự đã trình bày một bộ giải mã tạo sinh hồi quy dựa trên bộ tự mã hóa biến thiên (VAE: Variational Auto-Encoder). VAE là một mô hình tạo sinh có tính đến các biến tiềm ẩn, nhưng về bản chất không mang tính tuần tự. Với các phụ thuộc lịch sử trong không gian tiềm ẩn, nó có thể được chuyển đổi thành một mô hình tuần tự, trong đó đầu ra tạo sinh có tính đến lịch sử của các biến tiềm ẩn, do đó tạo ra một bản tóm tắt theo các cấu trúc tiềm ẩn.

6.8.2.3. Dịch máy (Machine Translation)

Các mô hình dịch máy neuron (NMT: Neural Machine Translation) được đào tạo để xử lý chuỗi văn bản đầu vào và tạo ra chuỗi đầu ra là bản dịch của chuỗi đầu vào sang một ngôn ngữ khác. Như đã đề cập trong Phần 6.6 , dịch máy là một ví dụ điển hình về việc chuyển đổi một chuỗi này sang chuỗi khác bằng kiến trúc bộ mã hóa-giải mã trong đó độ dài của cả hai chuỗi có thể khác nhau. Vào năm 2014, kiến trúc bộ mã hóa-giải mã dựa trên RNN nhiều-nhiều đã được đề xuất trong đó một RNN mã hóa chuỗi văn bản đầu vào thành biểu diễn vectơ có độ dài cố định, trong khi một RNN khác giải mã vectơ có độ dài cố định thành chuỗi đích đã dịch. Cả hai RNN đều được đào tạo chung để tối đa hóa xác suất có điều kiện của chuỗi đích cho chuỗi đầu vào. Sau đó, mô hình hóa dựa trên sự chú ý đã được thêm vào kiến trúc bộ mã hóa-giải mã vanilla cho dịch máy. Luong và cộng sự đã thảo luận về hai loại cơ chế chú ý trong công trình của họ về dịch máy nearon (NMT: Neural Machine Translation) : (i) chú ý toàn cục (Global Attention) và (ii) chú ý cục bộ (Local Attention). Trong chú ý toàn cục, một vectơ ngữ cảnh toàn cục được ước tính bằng cách học căn chỉnh độ dài biến và điểm chú ý cho tất cả các từ nguồn. Trong mô hình chú ý cục bộ, mô hình dự đoán một vị trí căn chỉnh duy nhất cho từ mục tiêu hiện tại và sau đó tính toán một vectơ ngữ cảnh cục bộ từ chú ý được dự đoán cho các từ nguồn

trong một cửa sổ nhỏ của vị trí căn chỉnh. Các thí nghiệm của họ cho thấy hiệu suất dịch thuật được cải thiện đáng kể so với các mô hình không có chú ý. Cơ chế chú ý cục bộ có ưu điểm là ít tốn kém hơn về mặt tính toán so với cơ chế chú ý toàn cục.

6.8.2.4. Dịch hình ảnh sang văn bản

Các mô hình dịch hình ảnh sang văn bản được kỳ vọng sẽ chuyển đổi dữ liệu trực quan (tức là hình ảnh) thành dữ liệu văn bản (tức là từ). Nhìn chung, dữ liệu hình ảnh đầu vào được truyền qua một số lớp tích chập để tạo ra biểu diễn dày đặc của dữ liệu trực quan. Sau đó, biểu diễn nhúng của dữ liệu trực quan được đưa vào mạng neuron nhân tạo (RNN) để tạo ra một chuỗi văn bản. Kiến trúc RNN nhiều-một (multi-to-one) rất phổ biến cho nhiệm vụ này.

Năm 2015, Karpathy và cộng sự đã trình bày công trình có tầm ảnh hưởng của họ lên huấn luyện mạng neuron tích chập vùng (RCNN: Region Convolutional Neuron Network) để tạo ra các vectơ biểu diễn cho các vùng ảnh và RNN hai chiều (BiRNN) để tạo ra các vectơ biểu diễn cho mô tả ảnh (Image Caption) tương ứng khi căn chỉnh ngữ nghĩa giữa chúng. Họ cũng đề xuất một RNN đa phương thức mới để tạo ra mô tả được căn chỉnh ngữ nghĩa với ảnh đầu vào. Các vùng ảnh được chọn dựa trên đầu ra được xếp hạng của một mạng neuron tích chập phát hiện đối tượng (CNN).

Xu và cộng sự đã đề xuất một khuôn khổ dựa trên sự chú ý để tạo mô tả hình ảnh lấy cảm hứng từ các mô hình dịch máy. Họ đã sử dụng các biểu diễn hình ảnh được tạo bởi các lớp tích chập thấp hơn từ mô hình CNN thay vì lớp được kết nối đầy đủ cuối cùng và sử dụng LSTM để tạo các từ dựa trên trạng thái ẩn, từ được tạo cuối cùng và vectơ ngữ cảnh. Họ định nghĩa vectơ ngữ cảnh là một biểu diễn động của hình ảnh được tạo ra bằng cách áp dụng cơ chế chú ý lên các vectơ biểu diễn hình ảnh từ các lớp tích chập thấp hơn của CNN. Cơ chế chú ý cho phép mô hình chọn động vùng để tập trung vào trong khi tạo ra một từ cho chú thích hình ảnh. Một lợi thế bổ sung của phương pháp tiếp cận của họ là trực quan hóa tiêu điểm của mô hình để tạo ra từng từ. Các thí nghiệm trực quan hóa trực giác của họ cho thấy mô hình của họ đã tập trung vào phần bên phải của hình ảnh trong khi tạo ra từng từ quan trọng.

Những công trình có tầm ảnh hưởng như vậy trong lĩnh vực mô tả hình ảnh tự động dựa trên các biểu diễn hình ảnh được tạo ra bởi CNN được thiết kế để phát hiện đối tượng. Một số mô hình mô tả được đề xuất gần đây đã tìm cách thay đổi xu hướng này. Biten và cộng sự đã đề xuất một mô hình mô tả cho hình ảnh được sử dụng để minh họa cho các bài viết mới. LSTM tạo mô tả của họ tính đến cả các đặc điểm hình ảnh do CNN tạo ra và các nhúng ngữ nghĩa vào văn bản của các bài viết mới tương ứng để tạo ra một mẫu mô tả. Mẫu này chứa các khoảng trống cho tên của các thực thể như tổ chức và địa điểm. Những vị trí này được điền vào bằng cơ chế chú ý trên văn bản của bài viết tương ứng.

6.8.2.5. **ChatBot**

ChatBots là công cụ trò chuyện tự động đã trở nên phổ biến rộng rãi trong thương mại điện tử và được sử dụng như trợ lý cá nhân kỹ thuật số như Siri của Apple và Alexa của Amazon. ChatBots là một ứng dụng lý tưởng cho các mô hình RNN vì các cuộc trò chuyện với ChatBots đại diện cho dữ liệu tuần tự. Các câu hỏi và câu trả lời trong một cuộc trò chuyện nên dựa trên các lần lặp lại trước đó của các câu hỏi và câu trả lời trong cuộc trò chuyện đó, cũng như các mẫu trình tự được học từ các cuộc trò chuyện khác trong tập dữ liệu.

Gần đây, ChatBot đã tìm thấy ứng dụng trong sàng lọc và can thiệp cho các rối loạn sức khỏe tâm thần như rối loạn phổ tự kỷ (ASD: Autism Spectrum Disorter)). Zhong và cộng sự đã thiết kế một ChatBot tiếng Trung sử dụng LSTM hai chiều trong khuôn khổ trình tự-trình tự cho thấy tiềm năng lớn để can thiệp qua trung gian hội thoại cho trẻ em mắc ASD Họ đã sử dụng 400.000 câu được chọn từ lịch sử trò chuyện liên quan đến trẻ em trong nhiều trường hợp. Rakib và cộng sự đã phát triển mô hình chuỗi sang chuỗi (Sep2Seq: Sequence to Sequence) dựa trên Bi-LSTM để thiết kế một ChatBot phản hồi một cách đồng cảm với bệnh nhân tâm thần. Một khảo sát chi tiết về ChatBot y tế được trình đã được nghiên cứu. Khảo sát này bao gồm các tài liệu tham khảo về ChatBot được xây dựng bằng các kỹ thuật NLP (Natural Language Processing: xử lý ngôn ngữ tự nhiên), biểu đồ kiến thức cũng như RNN hiện đại cho nhiều ứng dụng bao gồm chẩn đoán, tìm kiếm thông qua cơ sở dữ liệu y tế, đối thoại với bệnh nhân, v.v.

6.9. KẾT LUẬN

Do tính chất tuần tự của kiến trúc, RNN được ứng dụng cho các bài toán thứ tự hoặc thời gian, chẳng hạn như dịch ngôn ngữ, tóm tắt văn bản và mô tả (hay chú thích) hình ảnh, và được tích hợp vào các ứng dụng phổ biến như Siri, tìm kiếm bằng giọng nói và Google Dịch. Ngoài ra, chúng cũng thường được sử dụng để phân tích dữ liệu theo chiều dọc trong các ứng dụng y tế (ví dụ, các trường hợp có các quan sát lặp lại tại các thời điểm khác nhau cho mỗi bệnh nhân trong một tập dữ liệu). Mặc dù nghiên cứu về RNN vẫn đang trong quá trình phát triển và các kiến trúc mới đang được đề xuất, chương này tóm tắt các nguyên tắc cơ bản của RNN, bao gồm các kiến trúc truyền thống khác nhau, chiến lược đào tạo và các công trình có tầm ảnh hưởng. Nó có thể đóng vai trò là bước đệm để khám phá các mô hình tuần tự sử dụng RNN và cung cấp các gợi ý tham khảo.

Chương 7

CÁC MẠNG ĐỐI NGHỊCH TẠO SINH VÀ CÁC MÔ HÌNH TẠO SINH KHÁC

7.1. Tổng quan

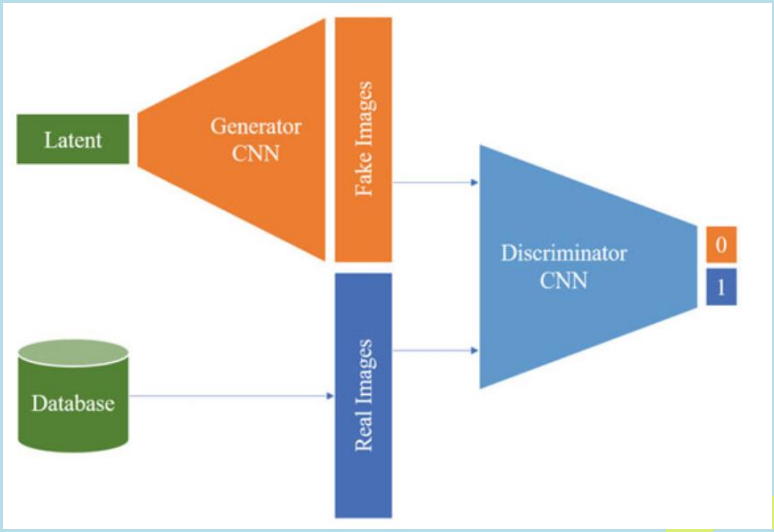
Các mạng tạo sinh (Generator Network) khác biệt căn bản về mục đích và mô hình của chúng so với các CNN cho phân loại, phân đoạn hay phát hiện đối tượng. Ban đầu nó không được dành cho một công cụ phân tích ảnh mà tạo ra các ảnh trông tự nhiên. Mẫu hình đào tạo đối nghịch (Adversarial Training) đã được đề xuất để ổn định các phương pháp tạo sinh và đã tỏ ra rất thành công mặc dù hoàn toàn không từ nỗ lực ban đầu.

Chương này cung cấp một giới thiệu cơ bản về động lực đối với các mạng đối nghịch tạo sinh (GAN: Generative Adversarial Network) và theo dõi con đường thành công của chúng bởi trừu tượng hóa nhiệm vụ cơ sở, cơ chế hoạt động và tìm ra khó khăn của các cách tiếp cận thực tế ban đầu. Chương này cũng cho thấy các phương pháp cho đào tạo ổn định hơn cũng như các dấu hiệu điển hình đối với hội tụ kém và nguyên nhân của chúng.

Mặc dù chương này tập trung lên các mạng GAN nghĩa là cho tạo ảnh và phân tích ảnh, mẫu hình đào tạo đối nghịch bản thân nó không đặc thù cho các hình ảnh mà còn tổng quát hóa cho các nhiệm vụ trong phân tích ảnh. Thí dụ về các cấu trúc cho phân đoạn ngữa nghĩa hình ảnh và phát hiện dị thường cũng được công nhận rộng rãi, trước khi các GAN tương phản (Contrasting GAN) cùng với các cách tiếp cận tạo sinh khác xuất hiện. Điều này cho ta một cách nhìn theo ngữ cảnh về các hạn chế cũng như các ưu điểm của GAN.

Các mạng tạo sinh đối nghịch là một kiểu của kiến trúc mạng neuron, trong đó một bộ phận của mạng tạo ra lời giải cho một nhiệm vụ và một bộ phận khác so sánh và đánh giá các lời giải được tạo với các giải pháp trước đó. Trong khi thoạt nhìn điều này cho thấy không khác nhiều so với bất kỳ hàm mất mát, điểm khác biệt cơ bản ở đây là nó còn so sánh lời giải được tạo với tiêu chuẩn vàng (Gold Standard). Hàm mất mát là tĩnh, nhưng bộ phận mạng “phán xét” hay “phân biệt” (Discriminator) có khả năng đào tạo (hình 7.1).

BẢN GỐC



Latent: tiềm ẩn (các ảnh chưa biết)
Database: cơ sở dữ liệu
Generator CNN: CNN của bộ tạo sinh
Fake Images: ảnh giả
Real Images: các ảnh thực
CNN của bộ phân biệt

Hình 7.1. Thiết lập GAN cơ bản cho tạo ảnh bao gồm bộ tạo sinh và một mạng phân biệt; ở đây là các CNN

Điều này có nghĩa là nó có thể được đào tạo để phân biệt các lới giải được tạo ra với các lời giải thực và cho đến khi nó thành công trong nhiệm vụ của mình, một tín hiệu cho bộ phân tạo sinh có thể được phát sinh. Đây là lý do của ghi chú đối nghịch trong cái tên GAN. Phần phân biệt được đào tạo để đạt đến các lời giải có vẻ giống thực nhất làm cho chúng đối nghịch với các mục tiêu của mình.

Các mạng tạo sinh đối nghịch ngày nay là một trong số các công cụ mạnh nhất để tạo ra các ảnh trong tự nhiên từ nhiều lĩnh vực. Trong khi chúng được tạo ra trong bối cảnh tạo ảnh, thì công bố ban đầu mô tả ý tưởng chung cách để hai mạng học bằng cách cạnh tranh không phụ thuộc vào ứng dụng. Ý tưởng mấu chốt này có thể được áp dụng cho các nhiệm vụ tạo sinh bên ngoài việc tạo ảnh bao gồm cả tạo văn bản, tạo âm nhạc và nhiều thứ khác.

Quan tâm nghiên cứu mạng tạo sinh đối nghịch bùng nổ nhiều năm sau công bố đầu tiên đề xuất mẫu hình đào tạo đối nghịch. Nhìn vào số lượng các ngiên cứu trên web về đề tài "các mạng đối nghịch tạo sinh" cho thấy mối quan tâm về đề tài này đã phát triển nhanh như thế nào, nhưng cũng bắt đầu hạn chót trong những năm gần đây. Các tác giả từ năm 2014 đã đưa tất cả các vấn đề và chương trình khung GAN cho phép cơ chế đào tạo mạng mẽ này cho đa dạng các nhiệm vụ bao gồm cả phân tích ảnh. Điều này ban đầu gây ngạc nhiên, vì không tồn tại sự giống nhau trực tiếp giữa một nhiệm vụ tạo sinh và, chẳng hạn, một nhiệm vụ phân đoạn hoặc phát hiện. Tuy vậy, như được chứng minh bởi sự thành công trong các ứng dụng này, phương pháp đào tạo đối nghịch có thể được áp dụng với nhiều lợi ích. Rõ ràng rằng thời hạn chót trong mốt quan tâm có thể ở mức độ nào đó có

thể được coi là do yêu cầu của các thực hành tốt nhất và các thực hiện đã được chứng minh, trong khi đó mối qua tâm khoa học gần đây đã dịch chuyển đến các phương pháp tiếp sau.

7.2. CÁC MÔ HÌNH TẠO SINH

7.2.1. Tổng quan

Ta khó nắm bắt được các xử lý tạo sinh bằng các tính toán. Bản chất và mục đích của chúng là tạo ra một thứ gì đó có nghĩa từ một thứ gì đó ít có nghĩa hơn (thậm chí ngẫu nhiên). Vì thế câu hỏi đầu tiên là làm thế nào điều này thậm chí có thể đối với một chương trình của máy tính vì, theo bản năng, sự tạo ra đòi hỏi một tinh thần mạnh mẽ - goại là sáng tạo, sử dụng thuật ngữ mà con người thường có xu hướng dùng đề liên kết với điều này. Để đưa ra một số thuật ngữ và các khái niệm cơ sở mà ta sẽ dùng trong phần còn lại của chương này, một số nhận xét về tính sáng tạo của con người sẽ cung cấp thông tin để người đọc hiểu rõ hơn.

Thực tế là, các hành động của con người sáng tạo là vốn có bởi các khái nệm của chúng ta về thế giới nhận được từ học hỏi và trải nghiệm thông qua nhiều phương tiện cảm nhận mà chúng ta có và bởi các phương tiện biểu cảm (các công cụ, dụng cụ, …) với chúng ta có thể hình thành việc sáng tạo một thứ gì đó. Điều này đúng với vúi mọi loại hành động sáng tạo, bao gồm viết văn, vẽ, điêu khắc gỗ hay mọi nghệ thuật khác và tương tự đối với cả lập trình máy tính. Biểu diễn trung gian có hạn của chúng ta về thế giới bên ngoài đóng khung phạm vị sáng tạo của chúng ta.

Các quá trình tạo sinh được học, được lập trình và được tính toán rất thoải mái tạo ra đầu ra. Chúng có hoặc một cơ chế có sẵn (in-built) hoặc cách thức để thu nhận cơ chế này, thể hiện các công cụ theo đó sáng tạo có thể, cũng như một mô hình thế giới định nghĩa đầu ra. Thực tế là, một quá trình tạo sinh dựa trên CNN sử dụng các tích chập như là công cụ có sẵn và bằng công cụ này nó được thiết kế để tạo các đầu ra giống như ảnh. Các lớp tích chập, nếu không được định nghĩa trước, sẽ biểu diễn một tập các phép toán được định nghĩa bởi một quá trình đào tạovà bị hạn chế trong sự biểu cảm, bởi các vật liệu đào tạo – bởi một phần của thế giới được thể hiện. Điều này dẫn ta đến lưu ý chính là làm cách nào nắm bắt được các thay đổi của "phần thế giới này" và cách thức làm cho một mạng neuron thể hiện thế giới một phần này. Điều cần lưy ý tại thời điểm này là không phải đối với các nghệ sĩ sáng tạo cũng như không phải đối với các mạng neuron, khả năng sáng tạo (hoặc tái sáng tạo) các kết quả có sức thuyết phục ngụ ý hiểu được cách thức mà các khuôn mẫu (trong thế giới thực) xuất hiện. Bất cứ thứ gì mang tính

thuyết có tính tạo sinh không ngụ ý hiểu được tự nhiên. Vì thế các GAN không thể giải thích các phần của tự nhiên mà chúng có thể tạo ra.

7.2.2. Ngôn ngữ của các mô hình tạo sinh: các phân phối, ước tính mật độ và các bộ ước tính

Hiểu biết các nguyên lý của các mô hình tạo sinh đòi hỏi kiến thức về các phân phối. Lý do là vì – như đã đề cập trong phần trước- "một phần thế giới" thực tế là một cái gì đó có thể được nghĩ như là một phân phối trong một không gian tham số. Nếu ta muốn mô tả một bộ phận của thế giới theo cách tương tác máy tính, ta phải định nghĩa các tham số mô tả. Để mô tả người, ta phải tìm ra các đặc trưng họ bằng các phép đo đơn giản như độ tuổi, chiều cao, trọng lượng, mầu tóc và mắt, và nhiều hơn nữa. Ta có thể bổ sung thêm huyết áp, nhịp tim, khối lượng cơ bắp, độ bền cực đại và nhiều hơn nữa. Từng tham số này có thể được thu thập riêng cho dân số thế giới và ta sẽ nhận được một bức tranh về tham số này được"phân phối" trên toàn thế giới như thế nào. Ngoài ra các tham số này sẽ liên quan đến nhau, chẳng hạn tuổi và độ bền cực đại. Tồn tại vô số các quan hệ như vậy, trong số chúng phần lớn có lẽ là chưa thể biết. Các quan hệ qua lại này được gọi là phân phối đồng thời (Joint Distribution). Nếu biết được phân phối đồng thời, ta có thể "tạo nên" một kết hợp tham số hợp lý có vẻ đáng tin về một con người không tồn lại.

Bây giờ ta sẽ chính thức hóa các ý tưởng này.

7.2.2.1. Các phân phối

Một phân phối mô tả tần suất của các quan sát cụ thể khi theo dõi một quá trình ngẫu nhiên, Vẽ số lượng xẩy ra trên một trục của các quan sát có thể có tạo nên một biểu đồ tần suất (Histogram). Nếu ta có thể sắp đặc các quan sát có thể có trên một thang liên tục, ta có thể nhận thấy rằng các quan sát tạo thành cụm trong các vùng nhất định và ta nói chúng tạo nên một "mật độ" hay chúng "dây đặc" tại đó. Vì thế, khi tìm cách mô tả các mật độ này ở đâu trong không gian tham số, thì điều này liên kết với mong muốn tái hiện lại hay lấy mẫu từ các phân phối, giống như ta muốn làm điều đó để tạo ra các thực thể từ một miền.

Đôi khi hình dạng của phân phối tuân theo một công thức giải tích, chẳng hạn, phân phối chuẩn. Nếu ta có thể cho trước một mô tả dạng đóng như vậy của phân phối, chẳng hạn, phân phối chuẩn, thì phân phối này sẽ tổng quát hóa dạng biểu đồ của các quan sát và khiến nó có thể dễ ràng tạo ra các quan sát mới chỉ đơn giản bằng cách lấy mẫu từ phân phối này. Khi các quan sát của ta tuân theo một phân phối chuẩn, thì có nghĩa là ta kỳ vọng quan sát các thực thể thường xuyên hơn xung quanh trung bình của phân phối chuẩn này hơn là hướng đến các đoạn cuối. Ngoài ra liệch chuẩn (Standarrd Deviation) sẽ

định lượng các quan sát gần đến đến trung bình nhiều hơn như thế nào so với các quan sát của các đoạn cuối. Ta mô tả các quan sát của ta bằng một mô tả tham số của mật độ được quan sát.

Trong phần còn lại của phần này, thay vì cung cấp định nghĩa toán học chặt chẽ, mô tả toán học của các phân phối và ước tính mật độ (xác suất), ta sẽ giới thệu các khái niệm cơ sở và các thuật ngữ theo cách trực giác được tóm lược trong hộp 1.

Hộp 1: Các phân phối xác suất (Probality Distribution): thuật ngữ

Một số thuật ngữ chung liên quan đến các phân phối có các diễn giải trực giác được cho dưới đây. Giả sử a là một sự kiện từ phân phối xác suất A, được viết là $a \sim A$, và $b \sim B$ là một sự kiện từ phân phối xác suất khác.

Trong y tế, A có thể là phân phối của các căn bệnh thần kinh có thể có và B là phân phối của tất cả các biến thể có thể có của hành vi hút thuốc.

Xác suất có điều kiện (Conditional Probality): P(A|B)

Xác suất có điều kiện của $a \sim A$, chẳng hạn một đột quỵ có thể phụ thuộc lịch sử hút thuốc cụ thể của một người được mô tả bởi $b \sim B$. Mật độ phân phối có điều kiện được viết là p(a|b) cho các thực thể cụ thể hay P(A|B) , nếu nói về toàn bộ các phân phối xác suất A và B.

Xác suất kết hợp (Joint Probability)

Xác suất khi thấy được các sự thực thể hóa cùng nhau của A và B được gọi là xác suất đồng thời. Đáng chú ý là, nếu được mở rộng, điều này dẫn đến một bảng lớn các xác suất liên kết từng $a \sim A$ (đột quỵ, sa sút trí tuệ, bệnh Parkison …) với từng $b \sim B$ có thể có (người hút thuốc không thường xuyên, người hút thuốc thường xuyên, người không hút thuốc…).

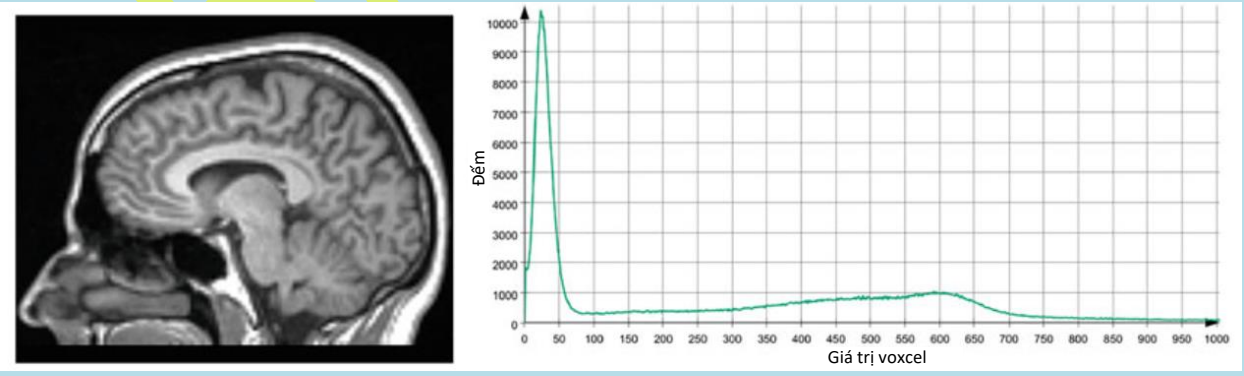
Xác suất biên (Marginal Probability)

Xác suất biên của A và B (được ký hiệu P(A) và P(B)) là các xác suất của từng kết cục có thể có trên (và độc lập) tất cả các các kết cục có thể có của các phân phối khác. Chẳng hạn, là xác suất nhìn thấy những người không hút thuốc trên tất cả các bệnh thần kinh hoặc nhìn thấy một bệnh cụ thể bất chấp trạng thái hút. Người ta nói rằng xác suất của một phân phối được biên hóa trên các phân phối xác suất khác.

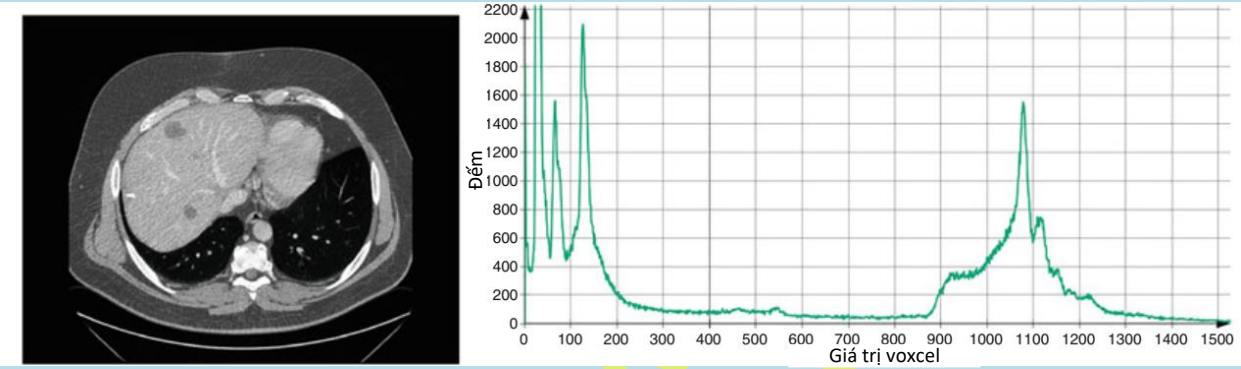
7.2.2.2. Ước tính mật độ (Density Estimattion)

Dưới đây ta giả thiết rằng các quan sát của ta được tạo bởi một hàm hay quá trình mà ta không biết và không thể đoán được từ một sắp xếp của các quan sát này. Ta xét một thí dụ thực tiễn, các ảnh từ CT và máy quét MRI được tạo ra từ một hàm như vậy. Đặc biệt là, ta ít quan tâm hơn về tính khó kiểm soát của vật lý hình ảnh, mà quan tâm hơn về vẻ ngoài của cơ thể người. Vật lý ảnh (Imaging Physics) có thể được lập mô hình bằng các phân tích đến một sai số nhất định. Nhưng hình dạng bên ngoài và cấu trúc bên trong cơ thể con người và các cơ quan chức năng của nó phụ thuộc rất lớn vào các nhân tổ chịu ảnh hưởng tương hỗ. Một số nhân tố này có thể biết và thậm chí có thể được mô hình., nhưng nhiều nhân tố thì không. Cụ thể, tính độc lập của các nhân tố phải được giả thiết là khó kiểm soát. Điều mà ta có thể tích lũy là các dữ liệu đo được cung cấp thông tin về cơ thể người, hình dạng của nó, và hoạt động của nó. Trong các thí nghiệm tưởng tượng dưới đây chúng ta sẽ khám phá cách đơn giản lập mô hình phân phối và tìm cách tạo ra các hình ảnh.

Bước đầu tiên là nghiên cứu phân phối các giá trị độ xám xám (Gray Value) hay nói một các khác đánh giá mật độ của các giá trị này. Cách cơ sở nhất để ước tính một mật độ là vẽ một biểu đồ (Histogramm). Giả sử giá trị trên trục x là giá trị độ xám của ảnh y khoa cần xém xét (trong CT nó được biểu diễn theo các đơn vị HU (Hounsfield Unit) và trong MRI nó được biểu diễn theo các đơn vị bất kỳ), (CT: Computed Topography: chụp cắt lớp), (MRI: Magnetic Resonance Image: chụp cộng hưởng từ). Hai hình vẽ cho thấy các biểu đồ của một MRI đầu người (hình 7.2) và CT vùng bụng (hình 7.3). Trong khi MRI não bộ gợi ý ba vùng "lồi" của biểu đồ tại các giá trị 25, 450 và 600, thì CT vùng bụng không phù hợp với mô tả như vậy.



Hình 7.2. Chụp MRI não bộ (trái) và biểu đồ các giá **trị xám cho một lát cắt của MRI**



Hình 7.3. Chụp CT vùng bụng (trái và biểu đồ các giá **trị độ xám cho một lát cắt của CT vùng bụng**

Trong bước tiếp theo, ta muốn mô tả các biểu đồ thông qua các hàm giải tích để dễ chấp nhận cho các mục đích tính toán. Nghĩa là, ta nhắm đến một mô tả giải tích cho các quan sát. Cực đại hóa kỳ vọng (EM: Expectation Maximization, xem hộp 2) là một giải thuật thích hợp cho nhiệm vụ này. EM cho phép ta thực hiện ước tính khả năng cực đại (MLE: Maximum Likelihood Estimation) với sự có mặt của các biến không được quan sát thấy (tiềm ẩn) và các dữ liệu không hoàn chỉnh- đây là giả thiết mặc định khi xử lý với dữ liệu thực. Ước tính khả năng cực đại là quá trình tìm các tham số của một phân phối phù hợp với độ chính xác nhất với phân phối của các quan sát. Trong MLE, điều này đạt được bằng cách điều chỉnh các tham số được điều khiển bởi một số đo sai số (Error Metric) chỉ thị mức độ gần phù hợp; nói một cách khác đây là một giải thuật tối ưu hóa tham số.

Hộp 2: Cực đại hóa kỳ vọng (EM: Expectation Maximization)- thí dụ

Tập trung lên ước tính mật độ từ dữ liệu MRI của chúng ta để tối ưu hóa các tham số cho một số lượng cố định các phân phối Gauss để đạt tới phù hợp gần nhất có thể với hình dạng thực nghiệm của biểu đồ.

Trong dữ liệu của ta, ta quan sát thấy "các vùng lồi" của biểu đồ. Bằng phân tích hình ảnh ta có thể xác định rằng các bộ phận nào đó được chụp bởi MRI đã dẫn đến các vùng lồi trong biểu đồ, vì chúng từ các vật liệu khác nhau và tạo ra các cường độ tín hiệu khác nhau. Tuy nhiên, điều này không biết đối với EM, nên được gọi là các biến "tiềm ẩn".

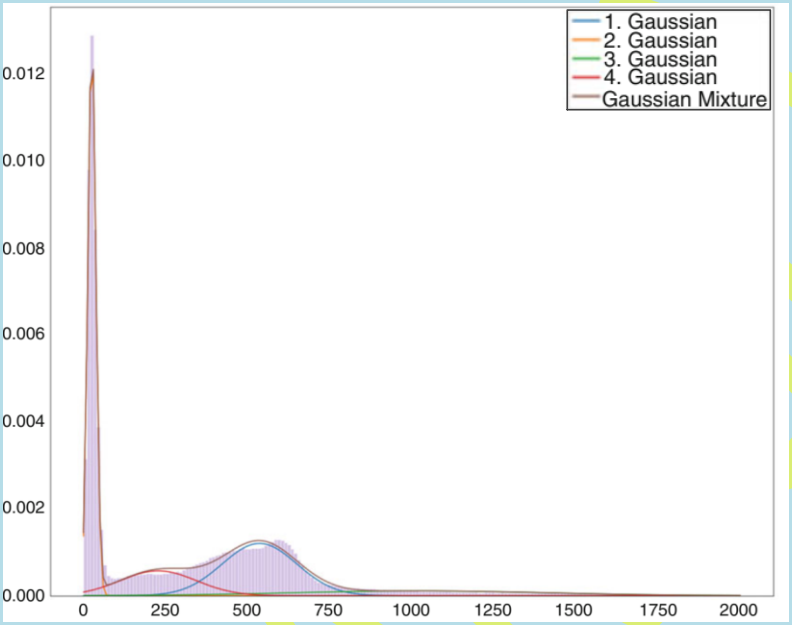
Giải thuật EM có hai phần: bước kỳ vọng và bước cực đại hóa. Với sự đơn giản các chi tiết lớn, chúng có thể được phác thảo như sau:

Kỳ vọng (Expectation)

Lấy từng điểm (hay một số điểm mẫu) của phân phối và ước tính kỳ vọng đến giá trị kỳ vọng của phân phối được tham số hóa để ấn định cho nó. Tìm ra ấn định này là bước xử

lý với các biến "ẩn tàng" (Latent) của các quan sát.
Cực đại hóa (Maximization)
Lặp lại trên tất cả các phân phối được tham số hóa, điều chỉnh các tham số của chúng để phù hợp với các điểm được ấn định tốt nhất có thể.
Quá trình này được lặp lại nhiều lần cho đến khi không thể cải thiện được nữa một sai số phù hợp (Fitting Error)

Hình 7.4 cho thấy một hỗn hợp các phân phối Gauss phù hợp với dữ liệu giá trị voxel của MRI não bộ như đã thấy trước đây.



Gaussian: phân bố Gaus
Gausian Mixture: Hỗn hợp phân bố Gauss

Hình 7.4. Mô hình hỗn hợp Gauss (GMM" Gaussian Mixture Model) để phù hợp với dữ liệu của MRI não bộ mà ta đã thấy ở dạng biểu đồ (Histogram) trong hình 7.3.

Thật hấp dẫn lập mô hình các quan sát thậm chí phức tạp hơn bằng cách pha trộn các phân phối giải tích đơn giản (ví dụ: các mô hình hỗn hợp Gauss (GMM: Gaussian Miixxture Model)), nhưng nhìn chung điều này sẽ khó giải quyết vì hai lý do sau. Thứ nhất, các phân phối đồng thời (Jiont Distribution) sẽ có rất nhiều cực đại pha trộn và vì thế đòi hỏi một số lượng lớn các phân phối cơ sở để làm phù hợp Thậm chí các phân phối chuẩn cơ sở trong không gian đa chiều (High Dimentional Space) không còn là các hàm có hai tham số (μ, σ), mà là với một vectơ trung bình và một ma trận hiệp phương sai. Thứ hai, không đơn giản lấy mẫu từ các phân phối đồng thời đa chiều, trong khi một số

phương pháp, trong số chúng các phương pháp chuỗi Markov Monte Carlo (Markove Chain Monte Carlo Method), cho phép lấy mẫu chúng, thì các phương pháp số này lại có độ phức tạp tính toán cao đến mức dẫn đến việc sử dụng chúng khó khăn trong ngữ cảnh ước tính tham số mạng neuron sâu.

Chúng ta sẽ nghiên cứu một số giải pháp khác. Về nguyên lý, có các các tiệp cận khác nhau cho ước tính mật độ (phân phối): ước tính phân phối trực tiếp, xấp xỉ hóa phân phối hay thậm trí gián tiếp hơn bằng cách sử dụng một phân phối hỗ trợ đơn giản được tạo ra giống nhất có thể với phân phối chưa biết bằng một hàm ánh xạ. Ta sẽ xét chi tiết hơn các phương pháp lập mô hình tạo sinh.

7.2.2.3. Các bộ ước tính và giá trị được kỳ vọng

Giả thiết ta đã tìm được các giá trị trung bình và các phương sai chuẩn phù hợp cho ba phân phối chuẩn cùng nhau xấp xỉ hóa hình dạng của ước tính mật độ dữ liệu MRI đáp ứng sự hài lòng của ta. Kết hợp các phân phối chuẩn (Gaus) như vậy được gọi là mô hình hỗn hợp Gauss (GMM) và lấy mẫu từ một GMM sẽ đơn giản. Như vậy ta có thể lấy mẫu các pixel đơn nhất theo mọi số lượng và theo thời gian ta sẽ lấy mẫu chúng sao cho ước tính mật độ của chúng hay đồ thị (Histogram) sẽ trông giống như mật độ mà ta đã bắt đầu.

Tuy nhiên, nếu ta muốn tạo ra một ảnh MRI của não bộ bằng cách sử dụng một quá trình lấy mẫu từ sự biểu diễn GMM dạng đóng, ta sẽ nhận thấy một vấn đề rất quan trọng đã không được tôn trọng trong cách tiếp cận của ta. Ta bắt đầu bằng một lát cắt 512×512 các voxel (điểm ảnh ba chiều) và vì thế vẽ ngẫu nhiên số lượng theo yêu cầu các giá trị voxel xung quanh phân phối này. Tuy nhiên điều này không tạo được một ảnh giống với ảnh MRI của não bộ mà gần giống như tạp âm ngẫu nhiên, vì ta đã không lập mô hình quan hệ không gian các giá trị độ xám với nhau. Vì phần lớn các voxel của MRI não bộ độc lập với nhau, nên vẽ một voxel mới từ phân phối này phải phụ thuộc vào các vị trí không gian và các giá trị độ xám của tất cả các voxel được vẽ trước đó. Các voxel lân cận sẽ có khả năng Likelihood) các giá trị xám tương tự cao hơn cao hơn các voxel ở xa. Quan trọng hơn, bên dưới các sự không phụ thuộc này là qua trình tạo ảnh: các giá trị của ảnh được quan sát thấy trong một MRI não bộ bắt nguồn từ mô (Tissue) thực tế - và đó là thứ định nghĩa sự độc lập. Điều này có nghĩa là giải phẫu học bộ não tự nó phản ánh gián tiếp trong các quy tắc mô tả sự phụ thuộc các giá trị xám với nhau.

Đối với quá trình mô hình hóa, điều này có nghĩa là ta không thể tranh luận về các giá trị voxel đơn lẻ và sự khả năng của chúng, mà ta cần tiếp cận quá trình tạo sinh tho cách khác. Một ý tưởng cho quá trình tạo sinh đã được ngụ ý trong mô tả trước đây: chọn một

vị trí ngẫu nhiên của ảnh cần tạo và dự đoán giá trị độ xám dựa trên tất cả các giá trị voxel hiện có. Giả sử thực hiện mô hình hỗn hợp, điều này dẫn đến nhiều phân phối cần được ước tính một cách khó hiểu, vì đối với từng vị trí "voxel tiếp theo" có thể có, mọi tổ hợp có thể có của các số lượng voxel đã có và các vị trí đều cần được xem xét. Ta sẽ thấy trong phần 7.5.1 về các mô hình khuếch tán (Diffusion Model) cách thức vẫn có thể làm cho phương pháp tổng quát tạo ảnh này làm việc.

Một cách tiếp cận tiếp theo khác để tạo ảnh cũng đã được thử, trong đó các pixel được tạo ra theo một thứ tự được quy định bắt đầu từ trên cùng bên trái và quét từng dòng qua các cột. Một lần nữa, hiểu biết về các pixel được tạo được ghi nhớ và được sử dụng để dự đoán voxel tiếp theo. Mạng neuron hồi quy pixel (Pixel RNN) chia sẻ ý thưởng chung của nó từ các công trình xử lý văn bản.

Cuối cùng, một cách tiếp cận trực tiếp để tạo ảnh có thể được chính thức hóa bằng cách thể hiện hay xấp xỉ hóa phân phối đồng thời đầy đủ của tất cả các voxel trên một phân phối hữu hình và lấy mẫu tất cả các voxel cùng một lúc từ phân phối này. Phân phối đồng thời đầy đủ trong cách tiếp cận này giữ nguyên ẩn tàng và ta sử dụng một phân phối thay thế. Đây thực sự là cách tiếp cận được thực hiện trong các GAN, mặc dù không phải theo một cách đơn giản.

Thực hiện một số cách tiếp cận đơn giản dựa trên khả năng xẩy ra (Likelihoood) hàm ý các khó khăn để nó hoạt động trở nên rõ ràng. Xét một ảnh MRI tại phân phối đồng thời của 512×512 voxels (một lát cắt của MRI não bộ) tại đây ta xấp xỉ hóa phân phối giá trị độ xám của một voxel bằng một GMM có sáu tham số, kết quả ta được $512\times512\times6=1$ 572 864 tham số. Về mặt khái niệm, sự biểu dễn này trải rộng một không gian 1 572 864 chiều, trong đó mỗi một lớp cắt MRI của não bộ sẽ là một điểm dữ liệu. Xem lại các biểu đồ CT mà MRI trong các hình vẽ trước đây, ta thấy các đường liên tục của các mật độ vì ta đã thu thập tất cả các voxel của toàn bộ một ảnh y khoa, chúng có số lượng nhiều triệu. Hơn nữa, ta chỉ xét một chiều (một kích thước) duy nhất trong số vào khoảng 1,5 triệu chiều. Tìm kiếm mật độ trong không gian lát cắt MRI não bộ với 1.572. 864 chiều được cho bởi tất cả các lát cắt MRI não bộ được thu thập là một nhiệm vụ khó khăn mà mọi giải thuật tạo sinh phải giải quyết. Trong không gian vô cùng lớn này, các lát cắt MRI não bộ "sống" trong một vùng rất nhỏ, vô cùng khó để tìm được nó. Ta nói, các ảnh chiếm một đa tạp ít chiều (Low dimentional Manifold) trong không gian đa chiều.

Ta xét một công thức khả năng cực đại:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \; E_{\mathrm{x}\sim \boldsymbol{P}_{\mathrm{data}}} \log \boldsymbol{Q}_{\theta} \;\; \boldsymbol{x} \mid \theta \qquad (7.1)$$

Trong đó $\boldsymbol{P}_{\text{data}}$ là một phân phối dữ liệu chưa biết và Q_θ là phân phối được tạo bởi mô hình được tham số hóa bởi $\boldsymbol{\theta}$, $\boldsymbol{\theta}$ có thể là , chẳng hạn, các trọng số và các độ lệch (Bias) của một mạng neuron sâu. Nới một cách khác, kết quả ước tính khả năng cực đại là các tham số $\hat{\boldsymbol{\theta}}$ sao cho tích của hai thành phần (trong đó chỉ có thành phần thứ hai phụ thuộc vào lựa cnọn $\boldsymbol{\theta}$) là cực đại. Thành phần thứ nhất là kỳ vọng $\boldsymbol{x}$ liên quan đến phân phối dữ liệu thật. Thành phần thứ hai là (log của) xác suất có điều kiện (khả năng) nhìn thấy thí dụ $\boldsymbol{x}$ được cho bởi $\boldsymbol{\theta}$ dưới mô hình $\boldsymbol{Q}_{\boldsymbol{\theta}}$. Vì vậy, cực đại hóa hàm khả năng nghĩa là cực đại hóa xác suất $\boldsymbol{x}$ được nhìn thấy trong $\boldsymbol{Q}_{\boldsymbol{\theta}}$, là trường hợp mà $\boldsymbol{Q}$ thích hợp tốt nhất có thể với $\boldsymbol{P}$ khi cho trước dạng tham số của $\boldsymbol{Q}$.

7.2.2.4. Lấy mẫu từ các phân phối

Khi một phân phối là mô hình các giá trị được quan sát xẩy ra như thế nào, thì lấy mẫu từ phân phối này là một quá trình tạo sinh các giá trị mới đã có thể được quan sát với xác suất giống như quan sát giá trị này trong thực tế. Có hai cách lấy mẫu từ các phân phối: tạo ra một số ngẫu nhiên từ mộ phân phối đều (Unifform Distribution) (đây là điều mà một bộ tạo sinh ngẫu nhiên luôn làm phía dưới) và đưa nó số này đến qua hàm mật độ tích lũy đảo (ICDF: Inverse Cumulative Density Function) để tích hợp hàm mật độ xác suất (PDF: Probability Density Function). Điều này chỉ đạt được nếu CDF được cho ở dạng đóng. Nếu không, cách lấy mẫu thứ hai có thể được áp dụng được gọi là lấy mẫu chấp thuận (hay từ chối). Nếu f là PDF, hai hàm $\boldsymbol{x}$ và y được rút ra từ phân phối đều. $\boldsymbol{x}$ ngẫu nhiên được tiếp nhận nếu $f(\boldsymbol{x})>y$ trái lại bị từ chối.

Trong trường hợp sử dụng, như ta đã thấy, không chỉ liên quan đến các phân phối đa chiều (đa biến) mà còn đến các liên kết của chúng (các phân phối đồng thời), các phân phối này không có dạng đóng. Trong các kịch bản này vẫn có thể lấy mẫu, bằng cách sử dụng lấy mẫu Monte Carlo chuỗi Markov (MCMC: Markov Chain Monte Carlo), là một khung được sử dụng để lấy mẫu từ chối cùng với các cơ chế bổ sung để tăng tính hiệu quả. Trong khi MCMC có các thuộc tính có lợi, thì nó rất đòi hỏi nhiều tính toán đối với các phân phối đồng thời phức tạp dẫn đến nhiều khó khăn trong ngữ cảnh lấy mẫu từ các phân phối mà ta sẽ đối mặt trong lĩnh vực phân tích và tạo ảnh.

Vì thế tại thời điểm này ta đối mặt hai vấn đề: ta khó mà hy vọng có thể ước tính mật độ và thậm chí nếu có thể, ta có thể gần như không lấy mẫu từ chúng.

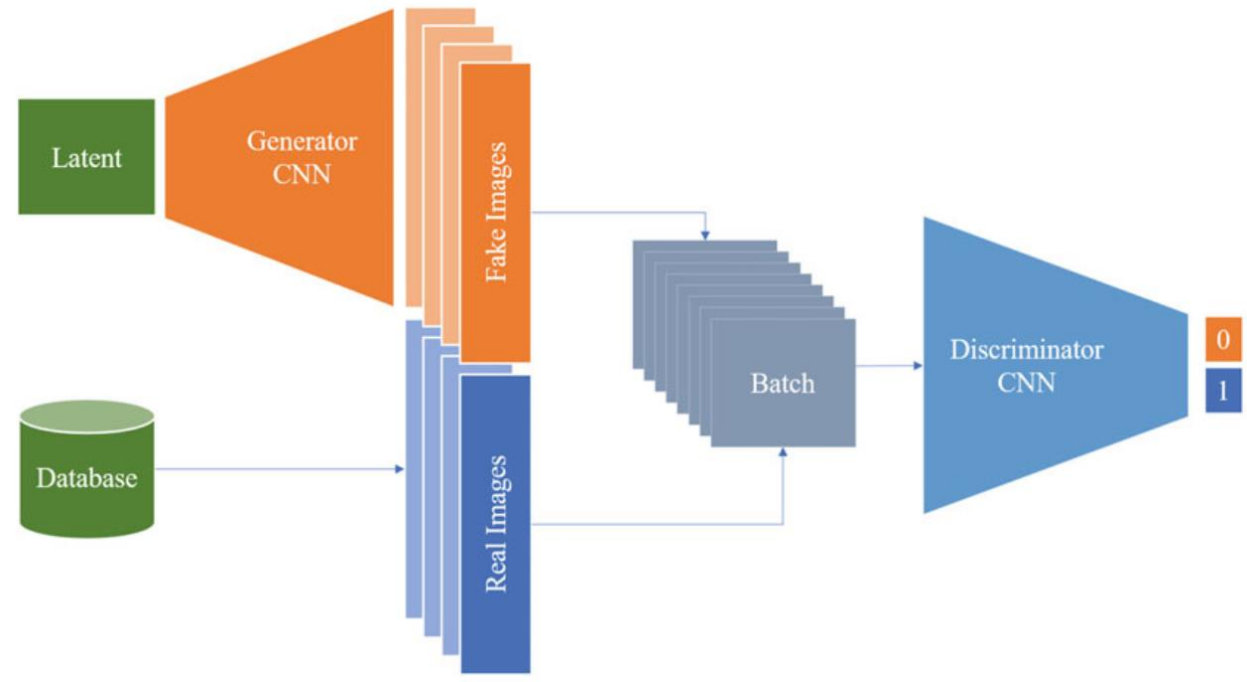
7.3. MẠNG ĐỐI NGHICH TẠO SINH (GAN)

7.3.1. Tổng quan

Các mạng đối nghịch tạo sinh (GAN: Generative Adversarial Network)) là một kỹ thuật được sử dụng trong cả học máy với bán giám sát (Semi Supervised) và học máy không có giám sát. Chúng đạt được điều này thông qua lập mô hình các phân phối dữ liệu đa chiều ẩn tàng. Chúng đặc trưng bởi đào tạo một cặp mạng cạnh tranh với nhau. Lõi của mô hình đào tạo đối nghịch là ý tưởng tạo ra hai người chơi thi đấu trong một trò chơi tối thiểu tối đa (Minimax). Trong các trò chơi này cả hai người chơi đều truy nhập đến các biến số giống nhau nưng có mục đích đối ngược, để họ sẽ thao túng các biến này theo các hướng khác nhau. Một sự giống nhau chung là suy nghĩ về một mạng, một kẻ làm giả (Forrger) và một mạng khác là chuyên gia. Kẻ làm giả, trong các tài liệu về GAN được biết đến là bộ tạo sinh (Generator), tạo ra các giả mạo nhắm đến tạo ra các ảnh có tính thực tế. Chuyên gia được biết đến như là các bộ phân biệt (Discriminator), nhận được cả các giả mạo và các ảnh thực (có nguồn gốc rõ ràng) và nhắm đến nói cho chúng về sự khác biệt.

Hình 7.6 cho thấy sơ đồ của một mạng GAN được xây dựng trên các mạng CNN. Ta thấy phần tạo sinh được thể hiện mầu da cam, tại đây các số ngẫu nhiên được rút ra từ không gian tiềm ẩn (Latent Space) và , một đối một, được chuyển đổi vào một tập các "ảnh giả" bởi mạng tạo sinh , trong hình vẽ được thực hiện bởi một mạng CNN. Đồng thời, từ cơ sở dữ liệu của các ảnh thực, một số các thí dụ phù hợp (Matching Number of Examples) được rút ra. Các ảnh thực và ảnh giả được dàn xếp vào một lô ảnh (Batch of Images) và được đưa đến một bộ phân biệt (Discriminator). Phía phải, CNN của bộ phân biệt được thể hiện trong mẫu xanh. Nó nhận lô các ảnh thực và ảo và quyết định đối với từng lô, nếu nó xuất hiện thực (cho ra giá trị gần với "1") hoặc giả ("0").

BẢN GỐC



Latent: tiềm ẩn, Generator CNN: ạng neuron tích chập của bộ tạo sinh, Fake Images: các ảnh giả, Real Images: các ảnh thật, Batch: lô; Discriminator CNN: mạng neurron của bộ phân biệt

Hình 7.5. Sơ đồ của một mạng GAN trên cơ sở CNN. Bộ tạo sinh (mầu da cam) tạo ra các ảnh giả dựa trên các số ngẫu nhiên được rút ra từ một không gian tiềm ẩn (Latent Space). Các số này cùng với một mẫu ngẫu nhên của các ảnh thật được đưa đến bộ phân biệt (mầu xanh, phải). Bộ phân biệt nhìn vào các ảnh thực/giả và cố gắng ấn định nhãn đúng ("0" cho giả và "1" cho thực)

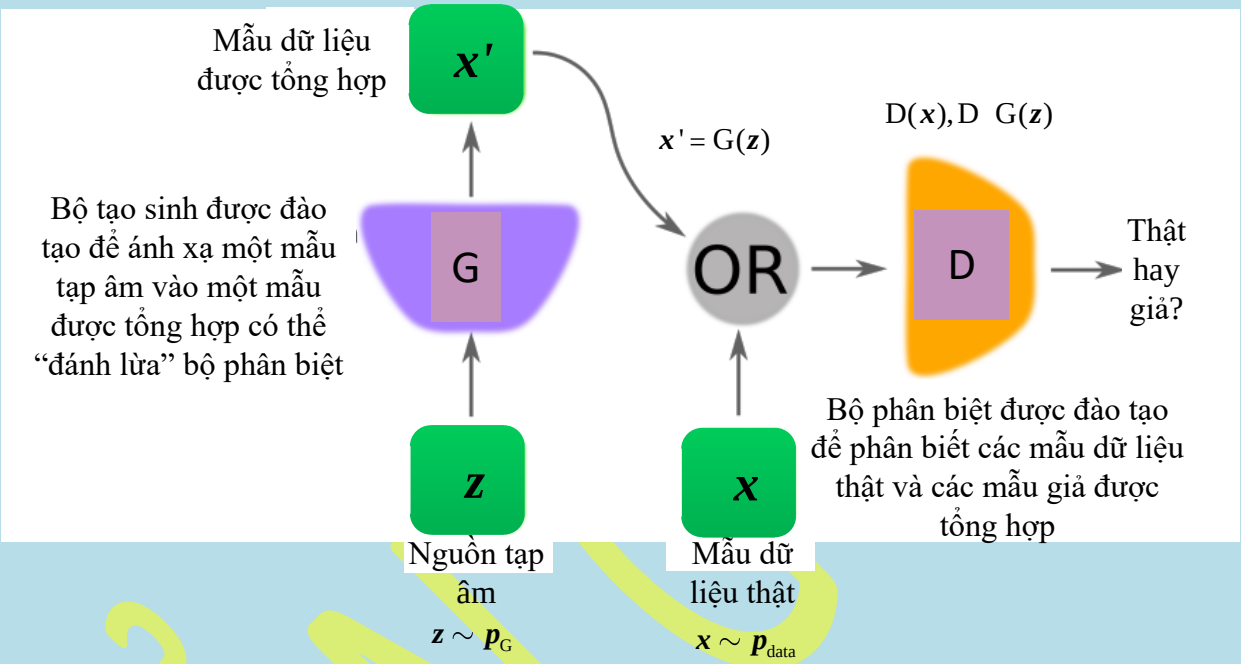
Ta đưa ra một số viết tắt để ký hiệu các phần tử của GAN. Ta ký hiệu các mạng tạo sinh và phân biệt là G và D (hình 7.6). Mục tiêu đào tạo GAN là một trò chơi giữa bộ tạo sinh và bộ phân biệt, tại đây cả hai đều ảnh hưởng lên hàm mất mát L, nhưng theo các chiều đối nghịch, ta có thể viết điều này như sau:

$$\min_G \max_D L(G, D),$$

Với hàm mục tiêu của GAN

$$L(G, D) = E_{\boldsymbol{x}\sim \boldsymbol{p}_{\text{data}}} \log D(\boldsymbol{x}) + E_{\boldsymbol{z}\sim \boldsymbol{p}_G} 1 - \log D(G(\boldsymbol{z})) \qquad (7.2)$$

D sẽ cố gắng cực đại hóa L bằng cách cực đại hóa xác suất ấn định các nhãn đúng cho các thí dụ thực và và các thí dụ được tạo sinh: đây là trường hợp khi D($\boldsymbol{x}$)=1, bằng cách cực đại hóa phần tử thứ nhất và khi D(G($\boldsymbol{x}$))=0, bằng cách cực đại hóa phần tử thứ hai. Thay vì đó, bộ tạo sinh lại sẽ cố gắng tạo ra các mẫu có vẻ thực mà bộ phân biệt gán nhãn "1", tương ứng với cực liểu hóa log[1-D(G($\boldsymbol{x}$))].



Hình 7.6. Trên hình này, hai mô hình được học trong quá trình đào tạo cho GAN, là bộ phân biệt (D) và bộ tạo sinh (G). Chúng thường được thực hiện bằng các mạng neuron, nhưng chúng có thể được thực hiện bởi một dạng bất kỳ của hệ thống khả phân biệt ánh xạ dữ liệu từ một không gian vào một không gian khác.

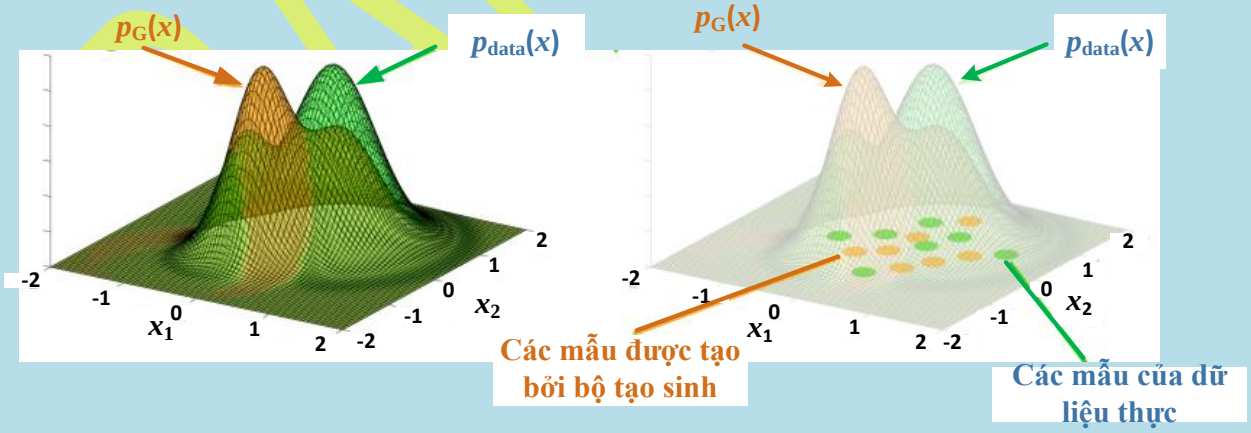
Các mô hình tạo sinh học để ghi lại phân phối thống kê của dữ liệu cho phép ta tổng hợp các mẫu từ phân phối đã học. Ngoài việc tổng hợp các mẫu mới để sử dụng cho các nhiệm vụ sau đó như biên tập ảnh ngữ nghĩa (Semantic Image), tăng cường dữ liêu và chuyển giao phong cách (Style Transfer), ta còn quan tâm đến sử dụng các biểu diễn mà các mô hình này học cho các nhiệm vụ như phân loại và truy xất ảnh ảnh (Image Retrieval).

Đôi khi ta thường đề cập đến các lớp kết nối toàn bộ hoặc tích chập của các mạng sâu; đây là các tổng quát hóa của các perceptron hay các ngân hàng bộ lọc không gian với hậu xử lý phi tuyến tính. Trong mọi trường hợp, các trọng số của mạng được học thông qua truyền lan ngược.

Tín hiệu sai số được tính toán từ số lượng các ấn định đúng mà bộ quyết định có thể thực thiến trên lô các ảnh được tạo sinh và các ảnh thực. Sau đó, cả hai bộ tạo sinh và bộ phân biệt có thể cập nhật tham số của chúng dựa trên tín hiệu của cùng một sai số. Quan trọng là, bộ tạo sinh phải nhắm đến cực đại hóa sai số, vì điều này biểu hiện rằng nó thành công đánh lừa bộ phân biệt trong việc nhận ảnh giả là ảnh thật, trong khi đó các trọng số của bộ phân biệt được cập nhật để giảm thiểu cùng một sai số này để biểu thị thành công của nó trong việc chỉ ra khác biệt của các thí dụ thực và ảo.

Các tài liệu về GAN nói chung xử lý các vectơ đa chiều và thường biểu diễn các vectơ này trong một không gian xác xuất bằng chữ in nghiêng (ví dụ không gian tiềm ẩn là *Z*). Trong lĩnh vực xử lý tín hiệu, thường biểu diễn các vectơ bằng các ký hiệu chữ thường đậm nét và ta sẽ tiếp nhận quy ước này để nhấn mạnh bản chất đa chiều của các biến. Tương tự, ta cùng thường đề cập $\boldsymbol{p}_{\text{data}}(\boldsymbol{x})$ để biểu diễn hàm mật độ xác suất trên một vectơ $\boldsymbol{x}$ nằm trong $\mathbb{R}^{|\mathbf{x}|}$. Ta sẽ sử dụng $\boldsymbol{p}_{\text{G}}(x)$ để ký hiệu phân phối của các vectơ được tạo bởi mạng tạo sinh (Generator Network) của GAN. Ta sử dụng các chữ hoa G và D để ký hiệu các mạng tạo sinh và phân biệt. Cả hai mạng có các trọng số $\boldsymbol{\Theta}_{\text{D}}$ và $\boldsymbol{\Theta}_{\text{G}}$ được học qua quá trình tối ưu, trong quá trình đào tạo.

Hình 7.6 cho thấy trong quá trình đào tạo, bộ tạo sinh được khuyến khích tạo ra một phân phối của các mẫu, $\boldsymbol{p}_{\text{g}}(\boldsymbol{x})$ phù hợp với phân phối của các mẫu dữ liệu thực, $\boldsymbol{p}_{\text{data}}(\boldsymbol{x})$. Đối với một GAN được tham số hóa và được đào tạo, các phân phối này sẽ gần như lý tưởng. Biểu diễn được thể hiện bởi các GAN là ghi lại cáchông số đã học (các trọng số) của bộ tạo sinh và bộ phân biệt.



Hình 7.6 cho thấy trong quá trình đào tạo, bộ tạo sinh được khuyến khích tạo ra một phân phối của các mẫu, $p_{\text{G}}(x)$ phù hợp với phân phối của các mẫu dữ liệu thực, $p_{\text{data}}(x)$. Đối với một GAN được tham số hóa và được đào tạo, các phân phối này sẽ

gần như lý tưởng. Biểu diễn được thể hiện bởi các GAN là ghi lại cáchông số đã học (các trọng số) của bộ tạo sinh và bộ phân biệt

Với các hệ thống học sâu, đào tạo đòi hỏi rằng ta phải có một hàm mục tiêu rõ ràng. Tuân theo các ký hiệu thường dùng, ta sử dụng $L_G\ \Theta_G;\Theta_D$ và $L_D\ \Theta_D;\Theta_G$ để đề cập đến các hàm mục tiêu của bộ tạo sinh và bộ phân biệt. Lựa chọn các ký hiệu này nhắc nhở ta rằng hai ký hiệu này có nghĩa là đồng phụ thuộc của các tập tham số đang phát triển Θ_D và Θ_G khi chúng được cập nhật lặp. Cuối cùng, lưu ý rằng các gradient đa kích thước được sử dụng trong các cập nhật. Ta sử dụng ∇_{Θ_G} ký hiệu cho thuật toán gradient liên quan đến các trọng số cuả các tham số của bộ tạo sinh bà ∇_{Θ_G} ký hiệu cho các trọng số liên quan đến các tham số của bộ phân biệt. Các gradient kỳ vọng được biểu thị bởi ký hiệu $E\nabla\bullet$.

Vấn đề trung tâm của xử lý tín hiệu và thống kê là ước tính mật độ: để nhận được một biểu diễn, ẩn tàng hay tường minh, tham số hay không tham số, của dữ liệu thể giới thực. Đây là động lực then chối đằng sau GAN. Trong các tài liệu về GAN, thuật ngữ bộ phân phối tạo sinh dữ liệu (Data Generrative Distributor) hay hàm khối xác suất (Probability Mass Function) thường được đề cập là mật độ xác suất cơ bản (Underlying Probability Density) hay hàm khối xác suất của dữ liệu quan sát. Các GAN học thông qua tính toán một loại giống nhau nào đó giữa phân phối của một mô hình ứng cử và phân phối tương ứng với số liệu thực.

Ước tính mật độ sẽ gặp nhiều vấn đề với nhiễu thị giác bao gồm nhận dạng ảnh , phát hiện và nhận dạng đối tượng thị giác, theo dõi đối tượng và đăng ký đối tượng. Về mặt nguyên tắc thông qua định lý Bayes, tất cả các vấn đề của thị giác máy tính có thể được xử lý thông qua ước tính các hàm mật độ có điều kiện, gián tiếp trong dạng một mô hình học phân phối đồng thời của các biến được quan tâm và dữ liệu được quan sát. Khó khăn mà ta phải đối mặt là khó xây dựng các hàm khả quan (Likelihood Function) cho dữ liệu ảnh của thế giới thực đa kích thước. Trong khi các GAN không cung cấp tường minh một cách thức ước tính các hàm mật độ, cho cặp bộ có dung lượng phù hợp, thì tạo sinh ghi lại một cách ẩn tàng phân phối của dữ liệu.

7.4. CÁC KIẾN TRÚC GAN

7.4.1. Kiến trúc GAN cơ sở và GAN kết nối toàn bộ

Các kiến trúc GAN đầu tiên sử dụng các mạng neuron kết nối toàn bộ cho cả bộ tạo sinh và bộ phân biệt. Kiểu kiến trúc này đã được áp dụng cho các tập dữ liệu khá đơn giản cụ thể là MNIST (các chữ số được viết bằng tay), CIFAR-10 (các ảnh tự nhiên) và tập dữ liệu khuôn mặt Toronoto (TFD (Toronto Face Dataset)

7.4.2. Các GAN tích chập

Chuyển từ các mạng kết nối toàn bộ đến các mạng tế bào thần kich tích chập là một mở rộng tự nhiên, giả sử đã cho trước các mạng CNN phù hợp tốt nất đối với dữ liệu ảnh. Các thử nghiệm đầu tiên được tiến hành trên CIFAR-10 cho thấy rằng đào tạo các mạng của bộ tạo sinh và bộ phân biệt bằng cách sử dụng các CNN với cùng mức độ dung lượng và công suất biểu diễn đã khó khăn hơn khi sử dụng các mạng cho học máy được giám sát.

Kim tự tháp Lamplace của các mạng đối nghịch (LAP-GAN: Lamplacian Pyramid of Adversatial-GAN) đưa ra một giải pháp cho vấn đề này bằng cách tạo ra quá trình tạo sinh bằng cách sử dụng đa đa kích thước/ đa tỷ lệ ảnh (Multiple Scale): ảnh thực cơ bản chính mình được phân chia vào kim tự tháp Lamplace, một cách có điều kiện, GAN có điều kiện được đào tạo để tạo ra từng lớp khi đã cho lớp trước.

Ngoài ra Radford và các tác giả khác đã đề xuất một họ các kiến trúc được gọi là DCGAN (Deep Convolution GAN: GAN tích chập sâu") cho phép đào tạo cả hai mạng tích chập của bộ tạo sinh và bộ phân biệt. DRGAN sử dụng các tích chập theo sải chân và tích chập theo sải chân một phần (Fractionally-Strided Convolution) để cho phép các thuật toán lấy mẫu giảm và lấy mẫu tăng không gian được học trong quá trình đào tạo. Các toán tử này xử lý thay đổi tốc độ và vị trí lấy mẫu, là yêu cầu then chốt để ánh xạ không gian ảnh đến không gian ảnh tiềm ẩn kích thước thấp hơn có thể và từ không gian ảnh đến bộ phân biệt.

Một mở rộng đối với tổng hợp các ảnh 2D, Wu và đồng nghiệp đã trình bày các GAN có thể tổng hợp các mẫu 3D bằng cách sử dụng các tích chập thể tích (Volumetric Convolution). Các tác giả này đã tổng hợp các đối tượng mới như các ghế, bàn và các ô tô. Họ cũng đã trình bày một phương pháp để ánh xạ từ các ảnh 2D sang các phiên bản 3D của các đối tượng được khắc họa trong các ảnh này.

7.4.3. Các GAN có điều kiện

Biểu diễn GAN có điều kiện (cGAN: Convolutional GAN) khá đơn giản. Nhắc lại hàm giá trị (Mục tiêu đào tạo) của các GAN từ phương trình (7.2):

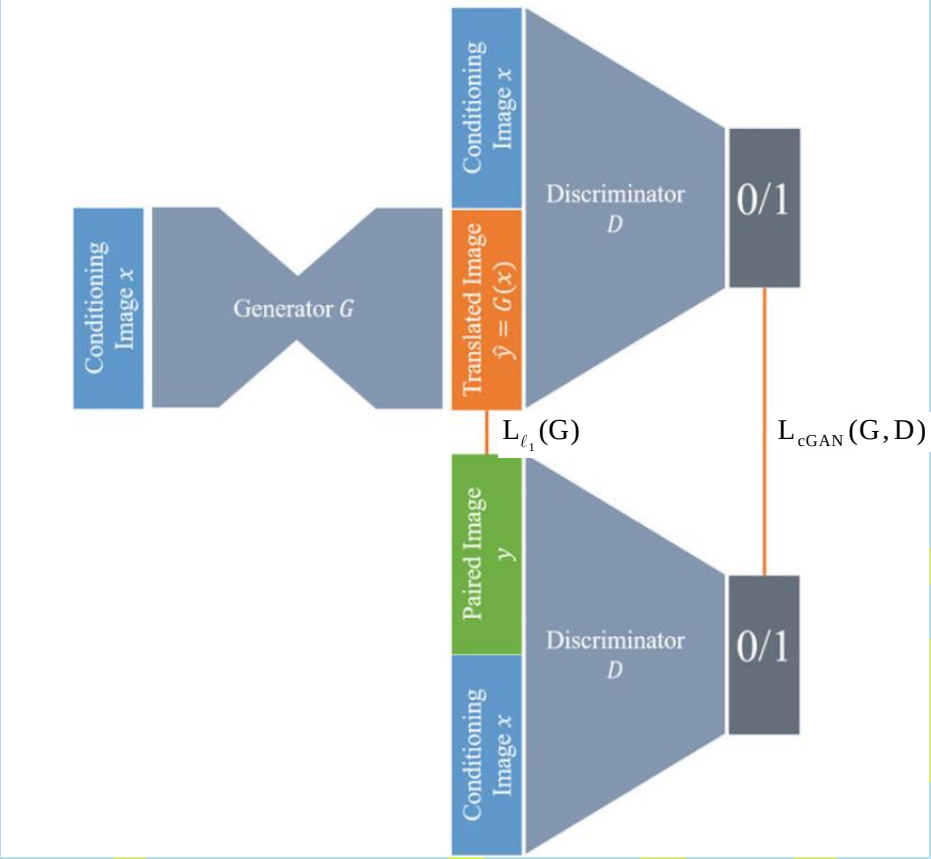
$$\mathrm{L}(\mathrm{G},\mathrm{D}) = E_{\boldsymbol{x}\sim \boldsymbol{p}_{\text{data}}} \log \mathrm{D}(\boldsymbol{x}) + \mathrm{E}_{\boldsymbol{z}\sim \boldsymbol{p}_{\mathrm{G}}} 1 - \log \mathrm{D}(\mathrm{G}(\boldsymbol{x}))$$

Bây giờ ta muốn đặt điều kiện cho bộ tạo sinh theo một hiểu biết nào đó hay đầu vào. Kết quả, cả hai bộ tạo sinh G và bộ phân biệt D sẽ nhận được một đầu vào "lập điều kiện" bổ sung, ta gọi nó là "*y*". Đây có thể là một nhãn loại (Class Label) nhưng cũng có thể là bất kỳ một thông tin liên kết khác. Rất phổ biến, đầu vào bổ sung sẽ là một ảnh, chẳng hạn, đối với ứng dụng biến đổi ảnh (chẳng hạn, biến đổi ảnh từ một phương thức chụp ảnh (Image Modality) này vào một phương thức khác , chẳng hạn, MRI và CT). Kết quả là hàm mục tiêu của cGAN như sau:

$$\mathrm{L}_{\mathrm{cGAN}}(\mathrm{G},\mathrm{D}) = E_{\boldsymbol{x}\sim \boldsymbol{p}_{\text{data}}} \log \mathrm{D}(\mathrm{x} \mid \mathrm{y}) + E_{\boldsymbol{z}\sim \boldsymbol{p}_{\mathrm{G}}} 1 - \log \mathrm{D}(\mathrm{G}(\boldsymbol{z} \mid y))$$

Hình 7.13 cho thấy một kiến trúc có thể có để áp dụng một cGAN cho chuyển đổi ảnh sang ảnh. Trong hình này, đầu vào đặt điều kiện là ảnh đích mà mạng đào tạo phải có thể tạo ra dựa trên một ảnh đầu vào nào đó. Mạng tạo sinh vì thế là U-Net. Mạng phân biệt có thể được thực hiện, chẳng hạn, bởi một mạng phân loại. Mạng này luôn nhận được hai đầu vào: ảnh đặt điều kiện (***x*** trong hình 7.7) và hoặc đầu ra được tạo sinh $\hat{y}$ hoặc đầu ra thực y.

BẢN GỐC

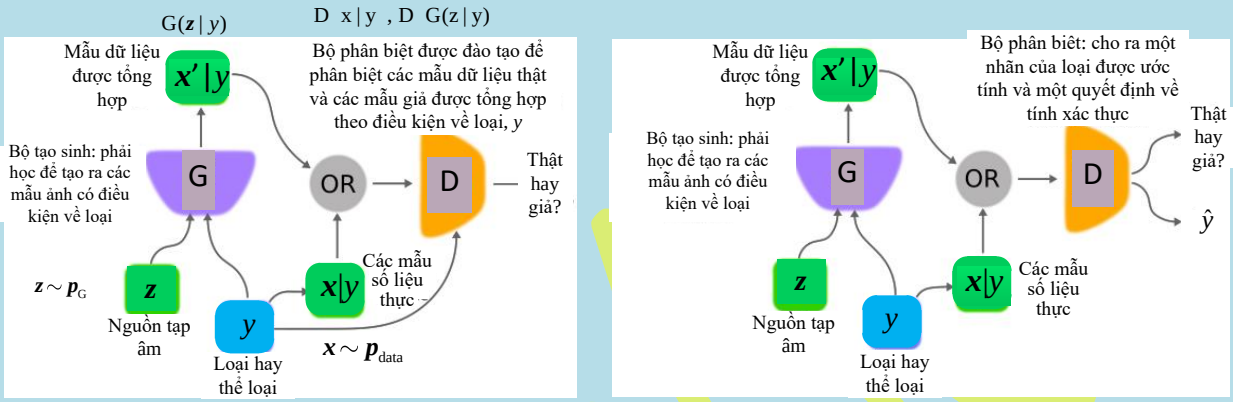


Conditioning Image ***x***: ảnh lập điều kiện ***x***, Generator: bộ tạo sinh, Translated Image: ảnh được chuyển đổi, Conditioning Image: ảnh lập điều kiện, Paired Image: ảnh cặp đôi

Hình 7.7. Một kiến trúc có thể có của cGAN. Phía trái: mạng tạo sinh nhận một ảnh cơ sở x như là đầu vào và tạo ra một ảnh được chuyển đổi $\hat{y}$. Bộ phân biệt nhận được hoặc cặp ảnh này hoặc một cặp thực x,y. Mất mát tái cấu trúc của bộ tái sinh (thường là mất mát ℓ_1) được tính toán giữa y và $\hat{y}$

Mirza và các tác giả khác đã mở rộng bộ khung (Framework) của GAN (2D) đến thiết lập GAN có điều kiện bằng cách làm cho cả hai mạng của bộ tạo sinh và của bộ phân biệt trở thành mạng có điều kiện về loại (Class-Conditional) (hình 7.8). GAN có điều kiện có ưu điểm là có khả năng cung cấp các biểu diễn tốt hơn cho tạo sinh dữ liệu đa mode. Ta có thể rút ra được sự giống nhau giữa các GAN có điều kiện và InfoGAN. InfoGan phân chia nguồn tạp âm vào một nguồn không thể nén (Incompressible Source) và một mã tiềm ẩn (Latent Code) để phát hiện các nhân tố tiềm thay đổi tiềm ẩn bằng cách cực đại hóa thông tin tương hỗ giữa mã tiềm ẩn và đầu ra của bộ tạo sinh. Mã tiềm ẩn có thể được sử dụng để khám phá các loại đối tượng theo cách học không có giám sát thuần túy,

mặc dù không hoàn toàn cần thiết mã tiềm ẩn phải là thể loại. Các biểu diễn được học bởi InfoGAN tỏ ra có nghĩa về mặt ngữ nghĩa, khi xử lý các nhân tố gắn kết chặt chẽ với nhau một cách phức tạp về vẻ bề ngoài của ảnh bao gồm các thay đổi trong tư thế, chiếu sáng và nội dung cảm xúc của ảnh khuôn mặt.



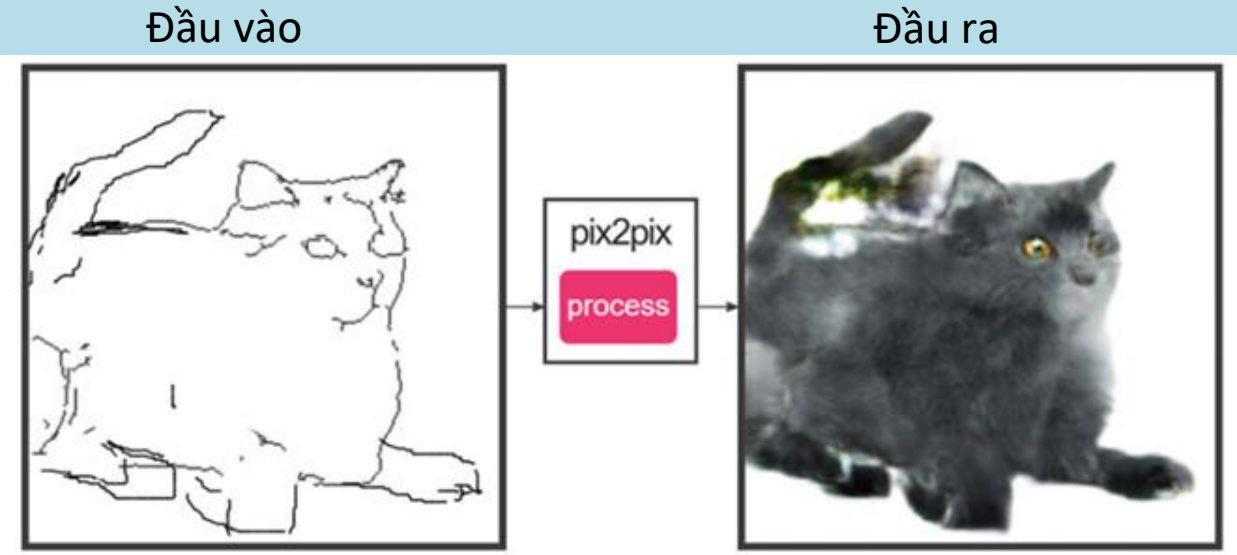
Hình 7.8. Hình trái, GAN có điều kiện, được đề xuất bởi Mirza và một số tác giả, thực hiện tổng hợp ảnh có điều liện về loại (Class-Conditional Image Synthesis); bộ phân biệt thực hiện phân biệt có điều kiện về loại giữa ảnh thực và ảnh giả. Hình phải, InfoGAN, mặt khác, có một bộ phân biệt ước tính cả nhãn của loại (Class Label)

Công trình của Isola và các cộng sự đã đưa ra một thành phần bổ mất mát bổ sung lên bộ tao sinh để đo khoảng cách ℓ_1 giữa ảnh được tạo sinh và ảnh thực cơ bản (Ground Truth Image), như sau:

$$\mathrm{L}_{\ell_1}(\mathrm{G}) = \mathrm{E}_{\mathrm{x,y,z}} \left\| y - \mathrm{G}(\boldsymbol{x}, \boldsymbol{z}) \right\|_1$$

Trong đó $\|\bullet\|$ là chuẩn của ℓ_1.

cGAN, cụ thể là trong cấu hình U-Net phục vụ như là một mạng tạo sinh đã được các họa sĩ và các nhà khoa học nhanh chóng tiếp nhận nhờ thực hiện pix2pix không mất tiền. Một thí dụ được tạo ra bằng pix2pix được cho trên hình 7.9, trong đó cGAN được đào tạo để tạo ra các ảnh từ các hình vẽ nét đơn.



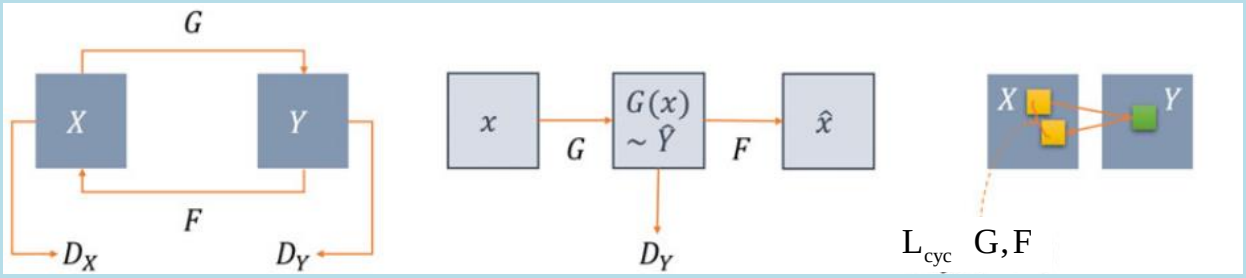
Hình 7.9. Demo thí nghiệm đầu vào đầu ra pix2pix

7.4.4. GAN chu kỳ (CycleGAN)

Trong khi cGAN đòi hỏi dữ liệu cặp đôi cho đầu vào tiêu chuẩn vàng và đầu vào lập điều kiện, thì điều này thường gập khó trong nhiều trường hợp vì thế việc phát triển GAN chu kỳ (CycleGAN) đặt một cột mốc vì nó làm giảm yêu cầu này và cho phép đào tạo mạng chuyển đổi ảnh vào ảnh (Image-Image Traslation) không cần các mẫu đầu vào cặp đôi.

Ý tưởng cơ sở của kiến trúc này là đào tạo hai hàm ánh xạ giữa hai miền và thực hiện chúng theo một thứ tự nhất định sao cho đầu ra nhận được được coi là nằm trong miền gốc. Đầu ra được so sánh với đầu vào gốc và các khoảng cách ℓ_1 và ℓ_2 thiết lập một bổ sung mới vào mất mát GAN đối nghịch thông thường nếu khác. Về mặt khái niệm, điều này nhắc ta về một trong số các mục tiêu của bộ tự mã hóa: tạo ra tín hiệu đầu vào sau khi mã hóa và giải mã; chỉ khác là lần này, không có nút cổ chai mà là một không gian ảnh khả tương tác khác. Điều này có thể được sử dụng để ổn định đào tạo, vì móc nối theo thứ tự các hàm chuyển đổi ảnh , ta gọi chúng là G và F, có thể đảo ngược. Hình 7.10 cho thấy một ngữ nghĩa của toàn bộ quá trình (trái) và một thể hiện của chu kỳ, tại đây từ miền ảnh X vào Y và ngược lại (giữa).

CycleGAN áp dụng một số thành phần mất mát trong đào tạo: hai mất mát đối nghịch $L(G, D_Y)$ và $L(F, D_X)$ và hai mất mát nhất quán chu kỳ (Cycle Consistency Loss), trong số chúng một hàm $L_{cyc}(F, D_X)$ được biểu thị ngoài cùng bên phải trong hình 7.10.



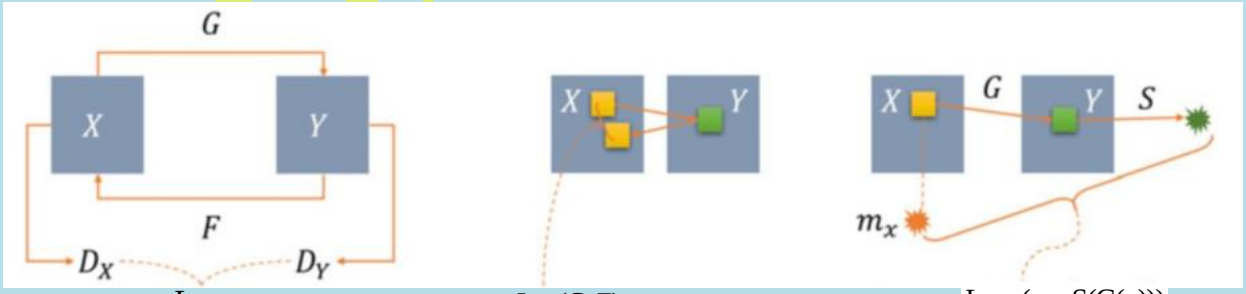
Hình 7.10. CycleGAN. Trái: các hàm chuyển đổi ảnh G và F chuyển đổi giữa hai miền. Các bộ phân biệt D_x và D_y cho ta các mất mát đối nghịch trong cả hai miền. Giữa: đối với một chuyển đổi cụ thể của ảnh *x*, chuyển đổi vào Y và ngược về ảnh X được mô tả. Phải: sau chu kỳ chuyển đổi, ảnh gốc và kết quả chuyển đổi ngược được so sánh trong mất mát nhất quán chu kỳ

Trong thiết lập mở rộng của CycleGAN với hình dạng và nhất quán chu kỳ, ba kiểu mất mát khác nhau thay cho hai mất mát gốc được kết nối trong quá trình đào tạo:

Các mất mát của GAN đối nghịch. Thành phần mất mát này giống như được định nghĩa trong phương trình (7.2).

Các mất mát nhất quán hình dạng L_{shape}. Mất mát nhất quán hình dạng (Shape Consistency Loss) là một bổ sung mới được đề xuất bởi Zhang và cộng sự. Một mất mát tương quan chéo xét đến hai phân đoạn, phân đoạn thứ nhất là phân đoạn tiêu chuẩn vàng m_x cho một $\boldsymbol{x} \in X$ và một phân đoạn được tạo ra bởi một mạng phân đoạn S được đào tạo trong miền Y và nhận được ảnh chuyển đổi $\hat{y} = G(\boldsymbol{x})$.

Hình 7.11 mô tả ba thành phần mất mát này, trong số đó hai thành phần thứ nhất đã được biết từ hình 7.10.



Hình 7.11. CycleGAN với mất mát nhất quán hình dạng (phần bên phải ngoài cùng của hình vẽ)

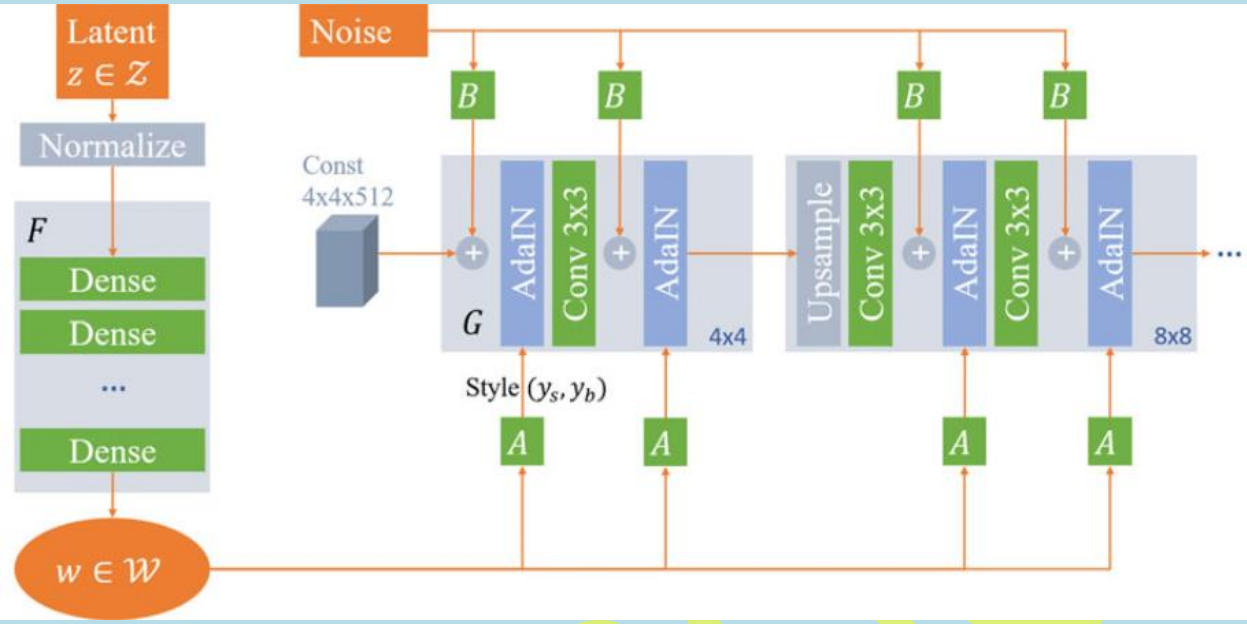
7.4.5. GAN phong cách (StyleGAN) và hậu duệ

Một trong số các GAN tổng hợp ảnh mạnh mẽ cho đến nay là hậu duệ của styleGAN, StyleGAN2, được nghiên cứu bởi Nvidia. Thay vì cách tiếp cận GAN thông thường, trong đó ảnh được tạo ra từ một vectơ được lấy ngẫu ngẫu nhiên từ không gian tiềm ẩn, họ sử dụng không gian tiềm ẩn được tạo bởi một hàm ánh xạ f mà trong cấu trúc của họ được thực hiện như là một perceptron đa lớp để ánh xạ từ một không gian Z kích thước 512 vào không gian W kích thước 512. Thay đổi lớn thứ hai là lớp được gọi là chuẩn hóa thực thể thích ứng (AdaIN: Adative Instance Normalization) thực hiện chuẩn hóa trong bình không và phương sai đơn vị của từng bản đồ tính năng, sau đó là thừa số nhân và thành phần độ lệch (Bias) cộng. Việc này phục vụ để đánh trọng số lại tầm quan trọng của các bản đồ tính năng trong một lớp. Để đảm bảo vị trí của đánh lại trọng số, thao tác này được tiếp sau bởi phi tuyến tính. Điều chnhr kích thước va độ lệch là hai phần tử của

$\boldsymbol{y} = y_s, y_b$, là kết quả của biến đổi afin có thể học (Learnable Affine Transformation) A được áp dụng cho một mẫu được rút ra từ $\boldsymbol{W}$.

Trong các thí nghiệm của mình, Karas và các cộng sự nhận thấy rằng sau các thay đổi của họ, GAN hầu như không còn phụ thuộc vao vectơ đầu vào được rút ra từ $\boldsymbol{W}$, vì thế vectơ tiềm ẩn được thay thế bằng một vectơ tĩnh đưa đến GAN. $\boldsymbol{y}$, được họ gọi là các phong cách, được giữ nguyên là kết quả từ một vectơ được lấy mẫu ngẫu nhiên từ không gian nhúng mới $\boldsymbol{W}$.

Sau cùng, tạp âm (Noise) được bổ sung vào từng lớp để cho phép GAN tạo ra thay đổi nhiêu hơn mà không cần học để tạo ra nó từ phần tử ảnh thực tế. Tạp âm, giống như vectơ tiềm ẩn, được cấp qua các bộ biến đổi B có thể học, trước khi được cộng vào các bản đồ tính năng chưa chuẩn hóa. Toàn bộ kiến trúc này được vẽ trên hình 7.12.



Dense Layer hày Full-Connected Layer: lớp kết nối toàn bộ, Style: phong cách, Upsample: tăng mẫu , Noise: tạp âm' Normalize: chuẩn hóa, Latent: không gian tiềm ẩn

Hình 7.12. Một kiến trúc styleGAN. Các lớp và các biến đổi có thể học được vẽ mầu xanh lá cây, còn AdaIN mầu xanh dương

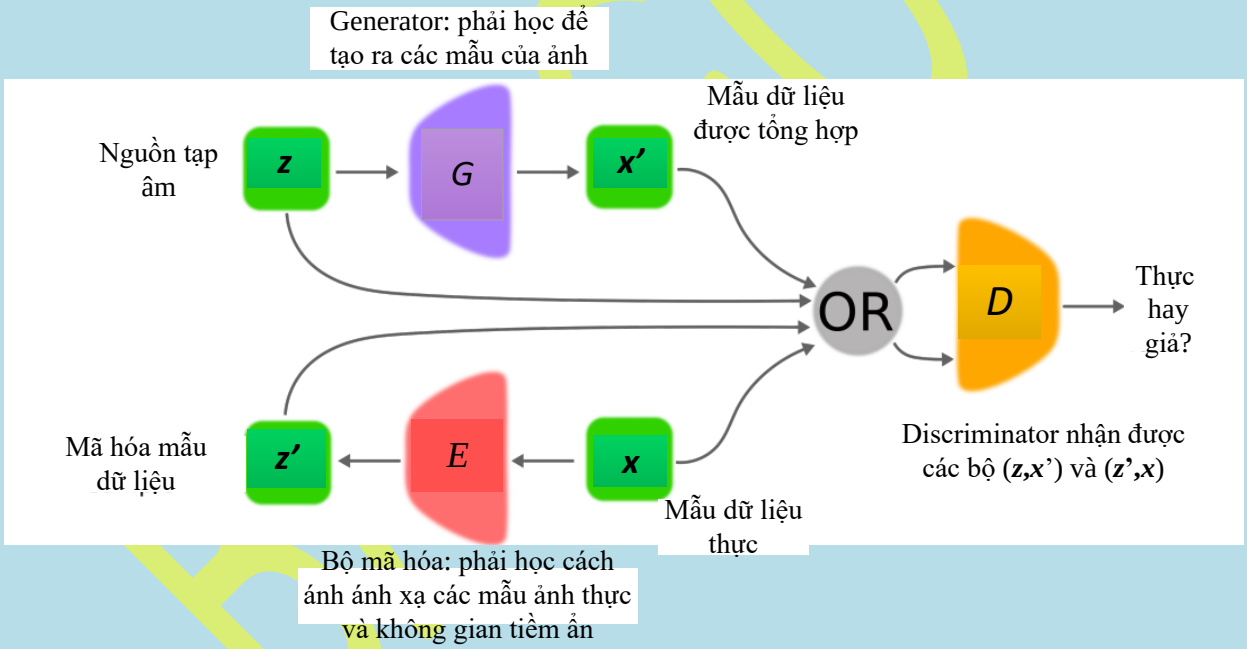
Trong thiết lập cơ sở, một mẫu được rút ra từ $\boldsymbol{W}$ và được đưa đến qua A đã được học đến từng lớp để đạt được diễn giải khác nhau của phong cách $\boldsymbol{y}=(y_s, y_b)$. Tuy nhiên, có thể thay đổi điều này, một số tác giả chỉ ra cách sử dụng một biến mẫu ngẫu nhiên w_1 trong một số khối các lớp và một mẫu khác w_2 trong phần còn lại; kết quả sẽ là một pha trộn các phong cách của cả hai các mẫu riêng biệt. Bằng cách này, các thuộc tính thô của ảnh được tạo sinh có thể bắt nguồn từ một mẫu và chi tiết mịn từ một mẫu khác. Thí dụ, áp dụng cho bộ tạo sinh khuôn mặt, điệu bộ và hình dạng khuôn mặt được xác định trong các lớp thô trước đó của mạng, còn cấu trúc tóc và kết cấu làn da là các chi tiết mịn của các lớp cuối cùng. Kết quả được mô tả trong hình 7.13.

Hình 7.13. Các ảnh được tạo ra bằng styleGAN.

7.4.5. GAN với các mô hình suy luận (GAN with Inference Models)

Trong công thức gốc, các GAN thiếu một cách thức để ánh xạ một quan sát có trước, ***x***,vào một vectơ trong không gian tiềm ẩn, trong các tài liệu điều này thường được đề cập như là một cơ chế suy luận. Một số kỹ thuật được đề xuất để đảo ngược bộ tạo sinh thành các GAN được đào tạo trước. Hai kỹ thuật được đề xuất độc lập: suy luận học được theo hướng đối nghịch (ALI: Adversarially Learned Interference) và các GAN hai chiều (BiGAN:directional GAN), cung cấp các mở rộng đơn giản nhưng hiệu quả bằng cách đưa ra một mạng suy luận trong đó các bộ phân biệt khảo sát các cặp (dữ liệu, tiềm ẩn) đồng thời.

Trong công thức này, bộ tạo sinh bao gồm hai mạng: "bộ mã hóa (mạng suy luận) và "bộ giải mã". Chúng được đào tạo đồng thời để đánh lừa bộ phân biệt. Bộ phân biệt tự mình nhận các cặp vectơ (***x, z***) (xem hình 7.14) và phải xác định cặp nào lạo thành một bộ thật sự (Genuine Tuple) bao gồm mẫu của ảnh thực và mã hóa của nó, hay một mẫu của ảnh giả và đầu vào không gian tiềm ẩn tương ứng đến bộ tạo sinh.



E (Encoder): bộ mã hóa, G/D (Generator/Decoder: bộ tạo sinh hay bộ giải mã), D (Discriminator: bộ phân biệt)

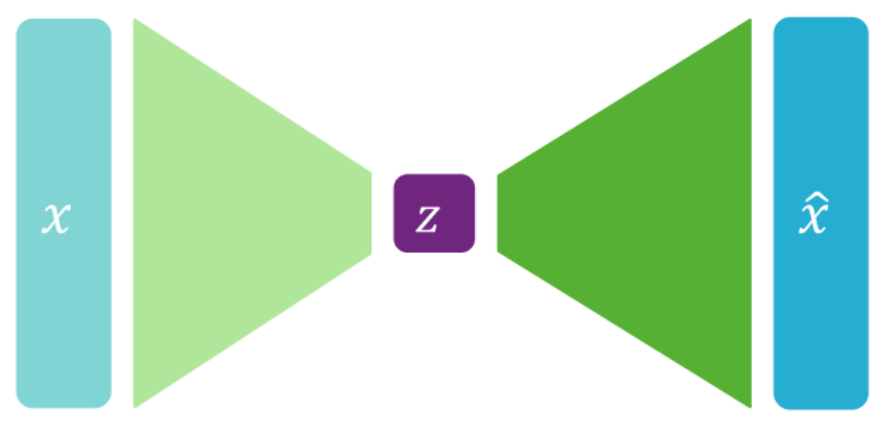
Hình 7.14. Cấu trúc của ALI/BiGAN bao gồm ba mạng. Một trong số chúng phục vụ như là bộ phân biệt, một mạng khác ánh xạ các vectơ tạp âm từ không gian tiềm ẩn vào không gian ảnh (bộ giải mã, được mô tả là là một bộ tạo sinh G trên hình vẽ) và mạng cuối cùng (bộ mã hóa, được mô tả là E) ánh xạ không gian ảnh vào không gian tiềm ẩn

Lý tưởng, trong một mô hình mã hóa-giải mã đầu ra, được đề cập như là một tái cấu trúc lại, phải giống như đầu vào. Thông thường, tính trung thực của các mẫu dữ liệu được tổng hợp bằng cách sử dụng ALI/BiGAN là kém. Có thể cải thiện thiện tính trung thực cuả các mẫu bằng một cái chi phí đối nghịch bổ sung lên phân phối các mẫu dữ liệu và các tái cấu trúc chúng.

7.4.6. Các bộ tự mã hóa đối nghịch (AAE: Adversarial Autoencoder)

7.4.6.1. Bộ tự mã hóa (Autoencoder)

Bộ tự mã hóa là một dạng của mạng neuron có thể học một trình bầy nén ((được gọi là trình bày không gian tiềm ẩn) của dữ liệu đào tạo. Ngược lại với các các perceptron nhiều lớp và CNN đã thấy cho đến nay sử dụng học có giám sát, Bộ tự mã hóa được sử dụng rộng rãi cho học máy không có giám sát với đa dạng các ứng dụng. KIến trúc của các bộ tự mã hóa bao gồm một đường dẫn co lại (Contracting Path) (được gọi là bộ mã hóa) nhằm biến đổi đầu vào vào một trình bầy có kích thước thấp hơn và một đường dẫn mở rộng (được gọi là bộ giải mã) nhằm kết cấu lại đầu vào tốt nhất có thể từ trình bầy kích thước thấp (Low Dimetion) (hình 7.15).



Hình 7.15.Nguyên lý tổng quát của một bộ tự mã hóa gảm tạp âm (Denoising Autoencoder). Nó nhắm đến học một trình bày kích thước thấp (không gian tiềm ẩn) *z* của dữ liệu đào tạo. Học được thực hiện bằng cách hướng đến cung cấp một tái kết cấu trung thực $\hat{x}$ của đầu vào *x*

Mất mát thường là mất mát ℓ_2 và khi này hàm chi phí là:

$$L(\theta,\phi) = \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - D_{\boldsymbol{\theta}} \; E_{\phi}(\boldsymbol{x}_i) \right\|_2^2 \qquad (7.2)$$

Trong đó E_ϕ là bộ mã hóa (ϕ là các tham số) và D_θ là bộ giải mã (θ là các tham số). Lưu ý là trên hình 7.15, $D_\theta \; E_\phi(\boldsymbol{x})$ được ký hiệu là $\hat{\boldsymbol{x}}$. Tổng quát hơn ta có thể viết:

$$L(\phi,\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mu_{xref}} \left[d \; \boldsymbol{x}, D_{\boldsymbol{\theta}} \; E_{\boldsymbol{\phi}}(\boldsymbol{x}) \right] \qquad (7.3)$$

Trong đó μ_{ref} là phân phối tham chuẩn để xấp xỉ hóa và d là hàm tái cấu trúc. Khi μ_{ref} là một phân phối dựa trên kinh nghiệm của tập đào tạo, và d là chuẩn ℓ_2 phương trình (7.2) tương đương phương trình (7.3).

Tồn tại rất nhiều biến thể khác nhau của các bộ tự mã hóa với mục đích để ngăn chặn các bộ tự mã hóa học hàm đồng nhất (Identity Function) và để cải thiện khả năng ghi lại thông tin quan trọng của chúng và học các biểu diễn phong phú hơn. Trong số chúng là, các bộ tự mã hóa cung cấp một phương pháp khác để đưa ra một nút thắt cổ chai thông tin mà không đòi hỏi giảm số lượng các nút tại các tính năng ẩn. Điều này được thực hiện bằng cách cấu trúc hàm mất mát sao cho nó phạt (Penalize) các kích hoạt trong một lớp. Điều này đạt được bằng cách ép buộc tính thưa (Sparsity) trong mạng và khuyến khích nó học một mã hóa và một giải mã chỉ dựa trên kích hoạt một số lượng nhỏ các tế bào thần kinh. Ép buộc tính thưa được thực hiện theo hai cách, chính quy hóa ℓ_1 trên các tham số của mạng và phân kỳ Kullback-Leibler (Kullback-Leibler Divergence), là thước đo sự khác biệt giữa hai phân phối xác suất. Ngoài ra một dạng khá phổ biến của bộ tự mã hóa là các bộ mã hóa giảm tạp âm (Denoising Autoencoder), trong đó mô hình được giao nhiệm vụ tái tạo lại đầu vào càng gần với bản gốc càng tốt, trong khi đi qua một loại qua nút thắt cổ chai thông tin nào đó (hình 7.15). Theo cách này, mô hình không thể đơn giản phát triển một ánh xạ ghi nhớ lại dữ liệu đào tạo mà là học một trường vectơ để ánh xạ đầu vào đến một tập hợp có kích thước thấp hơn. Cũng cần lưu ý rằng, trường vectơ thường hoạt động tốt trong các vùng mà tại đó mô hình đã quan sát thấy dữ liệu trong quá trình đào tạo. Trong dữ liệu nằm ngoài phân phối, sai số không chỉ lớn là và không thể chỉ được phương của phân phối thực. Ngoài ra các bộ tự mã hóa co rút (Contractive Autoencoder) là các biến thể khác của các mô hình này, chúng bổ sung thêm mất mát chính quy co rút (Conractive Regularization) vào mất mát tiêu chuẩn của bộ tự mã hóa. Một cách trực quan chúng buộc các đầu vào rất giống nhau có một mã hóa giống nhau mà cụ thể là, chúng đòi hỏi đạo hàm của các kích hoạt lớp ẩn phải nhỏ đối với các thay đổi nhỏ trong đầu vào. Sự quan sát này làm cho các mạng này khá phổ biến để phát hiện sự dị thường trong dữ liệu y tế. Có thể coi các bộ tự mã hóa giảm nhiễu là một biến thể

của bộ tự mã hóa co rút. Trong giới hạn của tạp âm Gauss, bộ tự mã hóa giảm tạp âm (hay giảm nhiễu) giúp cho sai lỗi tái cấu trúc kháng lại sự nhiễu loạn đầu vào có kích thước hữu hạn, trong khi đó các bộ tự mã hóa co rút giúp cho các tính năng được chiết suất chống lại các nhiễu loạn nhỏ đầu vào.

Phụ thuộc vào kiểu đầu vào, có thể thiết kế các kiến trúc bộ tự mã hóa khác nhau. Cụ thể là, khi đầu vào là các hình ảnh, bộ mã hóa và bộ giải mã được cấu thành kinh điển từ các khối tích chập. Bộ giải mã sử dụng, chẳng hạn, tích chập chuyển vị để thực hiện mở rộng. Cuối cùng, bổ sung các kết nối bỏ qua (Skip Connection) dẫn đến các kến trúc U-Net được sử dụng phổ biến cho các mục đích phân đoạn.

Các bộ tự mã hóa (Autoencoder) là các mạng cấu thành từ một "bộ mã hóa" và "bộ giải mã", học để ánh xạ dữ liệu vào một biểu diễn tiềm ẩn nội bộ và ra ngoài một lần nữa. Nghĩa là, chúng học một ánh xạ mặc định (thông qua bộ mã hóa) từ một không gian dữ liệu,ví dụ: các ảnh, vào một không gian hay biểu diễn tiềm ẩn và ánh xạ ngược (thông qua bộ giải mã) từ không gian tiềm ẩn vào không gian dữ liệu. Cấu trúc của hai ánh xạ này là một "tái cấu trúc", và hai ánh xạ này được đào tạo sao cho ảnh được tái cấu trúc càng giống ảnh gốc càng tốt.

Các bộ tự mã hóa gợi nhớ các ngân hàng lọc tái cấu trúc hoàn hảo được sử dụng rộng rãi trong xử lý ảnh và tín hiệu. Tuy nhiên, các bộ tự mã hóa nói chung học các ánh xa phi tuyến tính theo cả hai hướng. Ngoài ra, khi được thực hiện bằng các mạng sâu, các kiến trúc có thể có có thể được sử dụng để thực hiện các bộ tự mã hóa thì linh hoạt một các khác thường. Đào tạo có thể không bị giám sát, với truyền lan ngược được áp dụng giữa ảnh được tái cấu trúc và ảnh gốc theo thứ tự để học các tham số của các bộ mã hóa và giải mã.

7.4.6.2. Bộ tự mã hóa đối nghịch

Như đã đề xuất trước đây, ta thường muốn không gian tiềm ẩn có một tổ chức hữu ích. Ngoài ra ta có thể muốn thực hiện lấy mẫu tổ tiên theo hướng truyền thẳng (Feedforward Ancestral Sampling). Đào tạo đối nghịch cung cấp một lộ trình đề đạt được hai mục đích. Cụ thể, đào tạo đối nghịch có thể được áp dụng giữa không gian tiềm ẩn và một phân phối trước theo mong muốn (Desired Prior Distribution) trên không gian tiềm ẩn (Laten-Space GAN: GAN không gian tiềm ẩn). Điều này dẫn đến một hàm mát mát kết hợp và số đo về sự khác nhau của phân phối trước với phân phối được tạo bởi một mạng mã hóa ứng cử. Phương pháp liên quan đến bộ tự mã hóa biến thiên (Variational Autoencoder),

trong đó GAN không gian ẩn đóng vai trò thành phần phân kỳ KL (Kullback-Laibler Divergence) của hàm mất mát.

Một số tác giả đã kết hợp các bộ tự mã hóa biến thiên với đào tạo đối nghịch vào bộ khung Bayes biến thiên đối nghich (AVB: Adversarial Variational Bayes). Một số ý trưởng tương tự cũng đã được trình bày trong các tài liệu. AVB tìm cách tối ưu hóa cùng một tiêu chuẩn như các bộ tự mã hóa đối nghịch, nhưng sử dụng một đối tượng đào tạo đối nghịch thay vì phân kỳ KL.

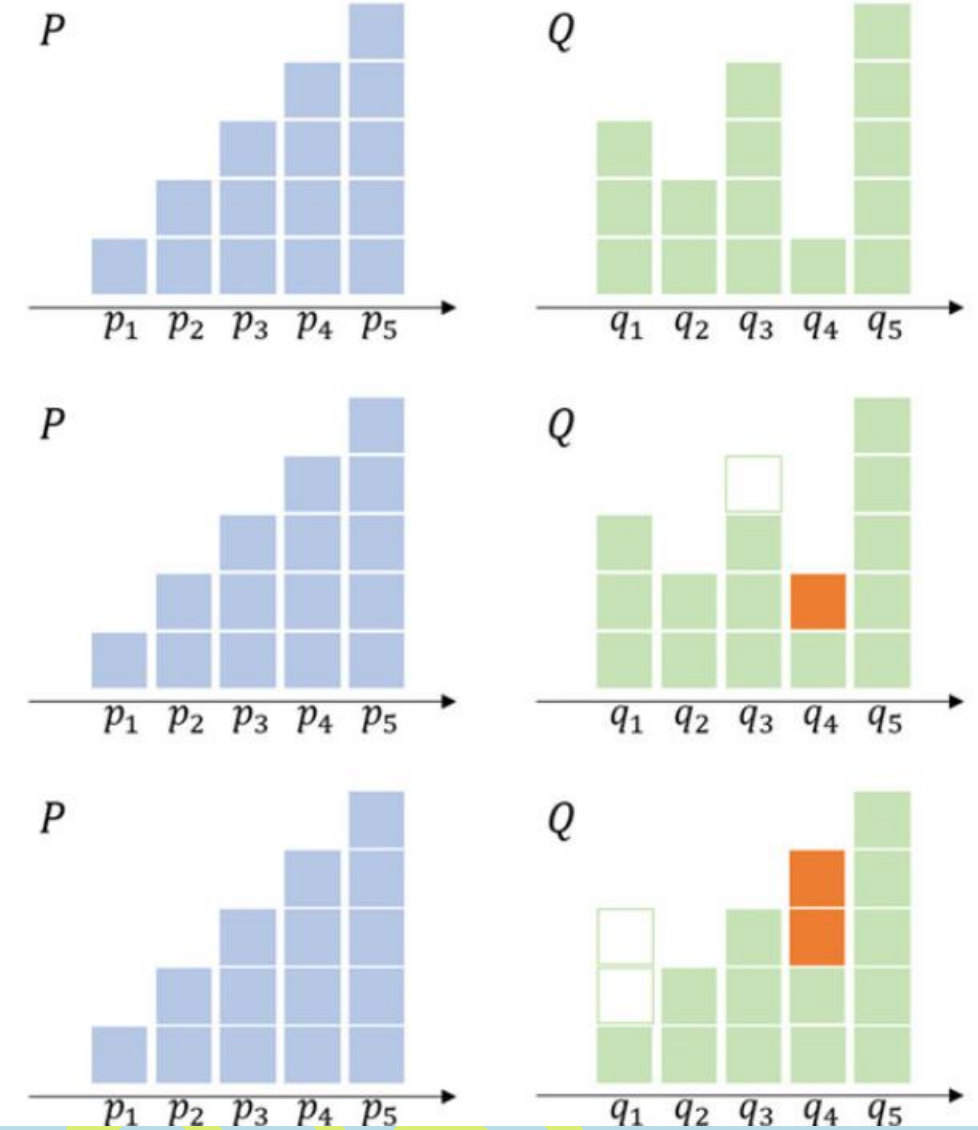
7.4.7. Wasserstein GAN

Wassterstein GAN được xây dựng dựa trên ý tưởng lý thuyết là thay đổi hàm mất mát vào khoảng cách Wassterstein sẽ dẫn đế các kết quả được cải thiện.

7.4.7.1. Khoảng cách Wassterstein

Khoảng cách Wassterstein đo khoảng cách mà với một mặt phẳng vận chuyển tối ưu, khối lượng có thể chuyển dịch từ một cấu hình này sang một cấu hình khác là bao nhiêu. Hình 7.16 cho thấy hai đống đất P và Q (phân phối xác suất rời rạc), cả hai đều có cùng một khối lượng đất tổng cộng, nhưng trong trong các trạng thái cụ thể x và y trong tất cả các trạng thái có thể có khác nhau .

Công việc được định nghĩa như là các xẻng đất đầy nhân với khoảng cách dịch chuyển của nó. Trong ba hàng đầu, xẻng đất được chuyển dịch (chỉ trong một P hoặc Q không từ đống này sang đống khác) để làm cho cấu hình giống nhau. Thứ nhất, một xẻng đất dịch chuyển qua một chồng đất (Pile) , dẫn đến cộng một vào khoảng cách Wasserstein. Sau đó hai xẻng đất dịch chuyển qua ba chồng đất (pile) dẫn đến cộng sáu vào khoảng cách Wasserstein cuối cùng D_w=7.



Hình 7.16. Một hình vuông là một xẻng đất. Vận chuyển từng xẻng đất từ chồng đất này đến chồng đất khác (Pile to Pile) tích lũy công việc được thực hiện : khoảng cách Wasserstein (chuyển dịch đất). Thí dụ cho thấy khoảng cách Wasserstein D_w=7

Lưu ý rằng, trong một kế hoạch khác, sẽ có thể dịch chuyển hai xẻng đất từ p_4 đến p_1 (trả giá 6) và một xẻng từ p_4 đến p_3, đây là kết hoạch chuyển vận đảo so với kế học trên, được thực hiện trên P, dẫn đến khoảng cách Wasserstein như nhau. Thực chất, khoảng cách Wasserstein là một khoảng cách chứ không phải là một phân kỳ, vì nó cho ta cùng một kết quả không phụ thuộc vào chiều dịch chuyển. Cũng cần lưu ý, ta một cách ẩn tàng giả thiết là P và Q chia sẻ hỗ trợ của chúng, nhưng trong trường hợp này là hỗ trợ riêng biệt, chỉ có thể cộng một thành phần hằng số, nó tăng cùng với khoảng cách giữa hai miền hỗ trợ.

7.4.7.2. Khoảng cách Wasserstein (Wasserstein Distance) của hàm mất mát

Để hình thức hóa tìm kiếm kế hoạch vận chuyển tối ưu, ta xét tất cả các phân phối chung có thể có của P và Q, hình thành tập các kế hoạch vận chuyển có thể có và ký hiệu tập này là $\Pi(P,Q)$, ngụ ý rằng đối với tất cả $\gamma \in \Pi(P,Q)$, P và Q sẽ là các phân phối biên (Marginal Distribution). Đến lượt mình, điều này có nghĩa là bởi định nghĩa $\sum_x \gamma(\boldsymbol{x},\boldsymbol{y}) = P(\boldsymbol{y})$ và $\sum_y \gamma(\boldsymbol{x},\boldsymbol{y}) = Q(\boldsymbol{x})$. Đối với một kế hoạch vận chuyển cụ thể γ, công việc giữa một trạng thái $\boldsymbol{x}$ trong P và một trạng thái $\boldsymbol{y}$ trong Q, ta quan tâm kế hoạch vận chuyển $\gamma(\boldsymbol{x},\boldsymbol{y})$. Giả sử $\|\boldsymbol{x}-\boldsymbol{y}\|$ là khoảng cách Euclid để dịch chuyển đất giữa $\boldsymbol{x}$ và $\boldsymbol{y}$ và sau đó nhân nó với mọi giá trị của γ (khối lượng đất được dịch chuyển) dẫn đến:

$$D_W(P,Q) = \inf_{\gamma \in \Pi(P,Q)} \sum_{x,y} \|\boldsymbol{x}-\boldsymbol{y}\| \gamma(\boldsymbol{x},\boldsymbol{y})$$

trong đó inf (Infinum: cận dưới) giống như min (Minimum)

Ta có thể viết lại phương trình trên như sau:

$$D_W(P,Q) = \inf_{\gamma \in \Pi(P,Q)} E_{(x,y)\sim\gamma} \|\boldsymbol{x}-\boldsymbol{y}\| \qquad (7.4)$$

Phương trình (7.4) đo cả khoảng cách của hai phân phối với hỗ trợ rời rạc và sự khác nhau của các phân phối này với hỗ trợ chồng lần hoàn hảo, vì nó bao gồm cả hai, dịch chuyển đất và khoảng các để dịch chuyển nó.

Tiếc rằng, ta không quan tâm đến bản thân kế hoạch vận chuyển γ, mà chỉ khoảng cách (của kế hoạch vận chuyển γ). Vấn đề này có thể được thể hiện trong dạng kép, cuối cùng để thực hiện khoảng cách của một hàm mất mát, ta có thể viết như sau:

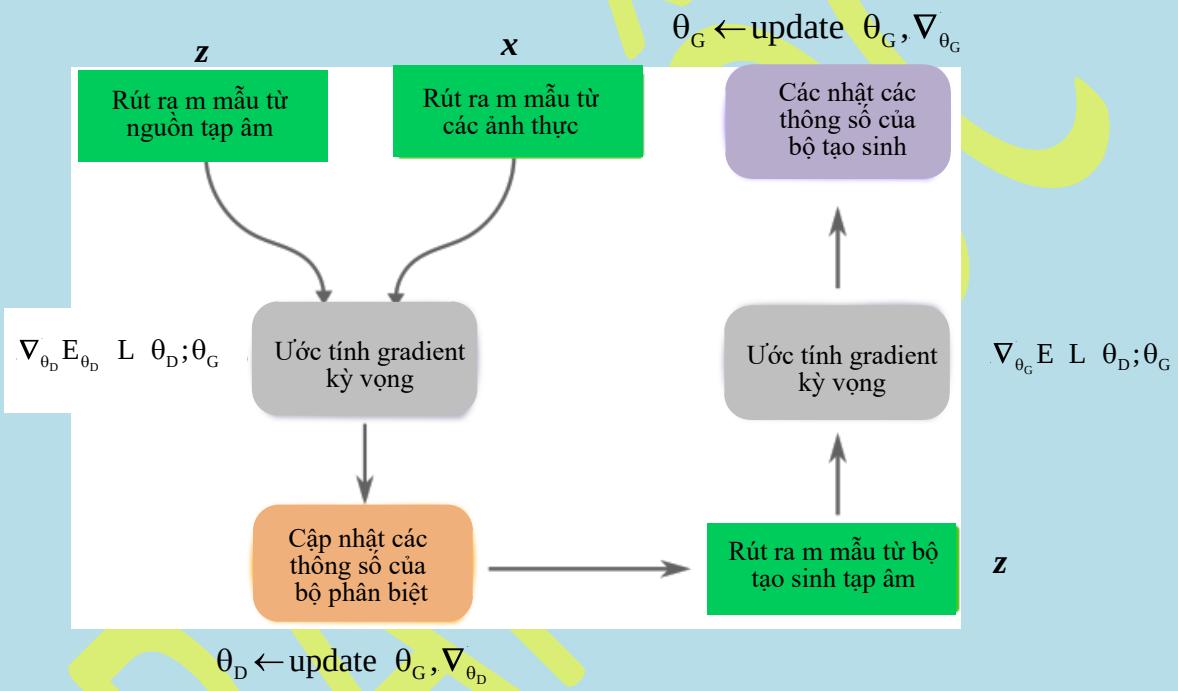
$$D_W(P,Q) = \max_{w \in W} E_{\boldsymbol{x}\sim P}\ D_w(\boldsymbol{x})\ - E_{z\sim Q}\ D_w(G_W(\boldsymbol{x})) \qquad (7.5)$$

Lưu ý rằng khác với các mất mát khác mà ta đã thấy trước đây, không còn log, vì vì, lần này, "bộ phân biệt" không còn là một mạng phân loại nữa, mạng phải học để phân loại các mẫu "thực" và các "mẫu giả" mà thay vào đó nó phục vụ như là một hàm trợ lý trống" (Blank Helper Function), hàm này trong quá trình đào tạo học ước tính khảng cách Wasserstein giữa các tập mẫu thực và giả.

7.5. ĐÀO TẠO GAN

7.5.1. Tổng quan

Đào tạo GAN bao gồm cả tìm kiếm các tham số của một bộ phân biệt để cực đại hóa phân loại của nó và tìm các tham số của một bộ tạo sinh để xáo trộn cực đại bộ phân loại. Quá trình đào tạo này được tổng kết trong hình 7.17.



Hình 7.17.Vòng đào tạo GAN chính. Các mẫu dữ liệu mới x',có thể được rút ra bởi đưa qua các mẫu nguẫu nhiên, z qua mạng tạo sinh. Gradient của bộ tạo sinh có thể được cập nhật k lần trước khi cập nhật bộ tạo sinh

Giá của đào tạo được ước tính bằng cách sử dụng hàm giá $L(G,D)$ phụ thuộc cả bô tạo sinh lẫn bộ phân biệt. Đào tạo liên quan đến giải phương trình (7.2) đã xét ở trên:

$$\min_G \max_D L(G,D),$$

Với hàm mục tiêu của GAN

$$L(G, D) = E_{\boldsymbol{x} \sim \boldsymbol{p}_{\text{data}}(\boldsymbol{x})} \log D(\boldsymbol{x}) + E_{\boldsymbol{z} \sim \boldsymbol{p}_G(\boldsymbol{x})} 1 - \log D(G(\boldsymbol{z})) \qquad (7.6)$$

Trong quá trình đào tạo, các tham số của một mô hình được cập nhật, trong khi các tham số của mô hình kia giữ nguyên. Goodfellow cùng một số tác giả đã chỉ ra rằng đối với một bộ tạo sinh cố định sẽ có một bộ phân biệt tối ưu duy nhất, $D^*(\boldsymbol{x}) = \frac{\boldsymbol{p}_{\text{data}}(\boldsymbol{x})}{\boldsymbol{p}_{\text{data}}(\boldsymbol{x}) + \boldsymbol{p}_G(\boldsymbol{x})}$. Họ cũng chỉ ra rằng bộ tạo sinh, G, tối ưu khi $\boldsymbol{p}_G(\boldsymbol{x}) = \boldsymbol{p}_{\text{data}}(\boldsymbol{x})$, tương đương với dự đoán của bộ phân biệt tối ưu là 0,5 cho tất cả các mẫu được rút ra từ x. Nói một cách khác, bộ tạo sinh tối ưu khi bộ phân biệt, D, bị rối loạn nhất và không thể phân biệt được các mẫu thực với các mẫu giả.

Lý tưởng, bộ phân biệt được đào tạo cho đến khi tối ưu liên quan với bộ tạo sinh hiện thời; khi này, bộ tạo sinh lại được cập nhật lần nữa. Tuy nhiên, trong thực tế, bộ phân biệt không thể được đào tạo cho đến khi tối ưu, mà thay vào đó có thể chỉ được đào tạo đối với một số lượng nhỏ lần lặp và bộ tạo sinh được cập nhật đồng thời với bộ phân biệt. Ngoài ra một phương án khác là, tiêu chuẩn đào tạo không bão hòa (Non-Saturating Criterion) thường được sử dụng bằng cách sử dụng $\max_G \log D(G(\boldsymbol{x})$ thay vì $\min_G \log 1 - D(G(\boldsymbol{x}))$.

Mặc dù tồn tại các giải pháp duy nhất về lý thuyết, đào tạo GAN vẫn là một thách thức và thường không ổn định vì một số lý do. Một phương pháp để cải thiện đào tạo GAN là đánh giá "các triệu chứng" thực nghiệm (Empirical Symptom) có thể trải nghiệm trong quá trình đào tạo. Các triệu chứng này bao gồm:

- Các khó khăn để làm cho cặp các mô hình hội tu;
- Mô hình tạo sinh, "sụp đổ", tạo sinh các mẫu giống nhau cho các đầu vào khác nhau;
- Mất mát (Loss) của bộ phân biệt hội tụ nhanh vào không, tạo ra đường dẫn không tin cậy đối với các cập nhật gradient cho bộ tạo sinh.

Một số tác giả đưa đưa ra các cách tiếp cận dựa trên kinh nghiệm để giải quyết các vấn đề này.

Goodman và các tác giả khác là những người đầu tiên tìm cách giải thích tại sao GAN không ổn định. Họ quan sát thấy rằng các phương pháp giảm gradient thường được sử dụng để cập nhật cả các tham số của bộ tạo sinh và của bộ phân biệt. Điều này không

thích hợp khi giải pháp cho vấn đề tối ưu được đặt ra bởi đào tạo GAN lại tạo ra một điểm yên ngựa. Tuy nhiên, giảm gradient ngẫu nhiên thường được sử dụng để cập nhật các mạng neuron và chúng đã được phát triển tốt cho cho các môi trường lập trình học máy tạo điều kiện cho cấu trúc và cập nhật các mạng sử dụng giảm gradient ngẫu nhiên.

Mặc dù một nghiên cứu lý thuyết đầu tiên đã chỉ ra rằng bộ tạo sinh tối ưu khi $\boldsymbol{p}_{\mathrm{G}}(\boldsymbol{x}) = \boldsymbol{p}_{\mathrm{data}}(\boldsymbol{x})$, một kết quả rất đẹp với trực giác sâu sắc, các mẫu dữ liệu thực trú ngụ một đa tạp nằm trong một không gian đa chiều cảu các biểu diễn có thể có. Chẳng hạn, nếu các mẫu ảnh mầu có kích thước là N×N×3 với các giá trị pixel $\left[0, \mathbb{R}^{+}\right]^{3}$, không gian này có thể được biểu diễn, là một không gian mà ta có thể gọi gọi là $\mathbb{X}$, là một không gian $3N^2$ chiều, với mỗi chiều nhận các giá trị giữa 0 và cường độ pixel có thể đo được cực đại. Các mẫu dữ liệu với sự hỗ trợ của $\boldsymbol{p}_{\mathrm{data}}$, tuy nhiên, tạo nên đa tạp của dữ liệu thực được liên kết với một vấn đề cụ thể nào đó, thường chiếm một phần rất nhỏ của toàn bộ không gian, $\mathbb{X}$. Tương tự, các mẫu được tạo bởi bộ tạo sinh cũng nên chỉ chiếm một phần nhỏ không gian của $\mathbb{X}$.

Arjovsky và các tác giả khác cũng chỉ ra rằng hỗ trợ $p_g(x)$ và p_{data} nằm trong một không gian ít chiều hơn không gian tương ứng với $\mathbb{X}$. Hậu quả của điều này là $\boldsymbol{p}_{\mathrm{G}}(\boldsymbol{x})$ và $\boldsymbol{p}_{\mathrm{data}}(\boldsymbol{x})$ có thể không chồng lấn và vì thế tồn tại một bộ phân biệt gần như đơn giản có thể phân biệt các mẫu thực, $\boldsymbol{x} \sim \boldsymbol{p}_{\mathrm{data}}(\boldsymbol{x})$ từ các mẫu giả $\boldsymbol{x} \sim \boldsymbol{p}_{\mathrm{G}}(\boldsymbol{x})$ với độ chính xác 100%. Trong trường hợp này, sai số của bộ phân biệt nhanh chóng hội tụ vào không. Chỉ có thể cập nhật các tham số của bộ tái sinh thông qua bộ phân biệt, vì thế khi xảy ra điều này, các gradient được sử dụng để cập nhật các tham số của bộ tái sinh cũng hội tụ vào không và vì thế có thể không cần các cập nhật cho bộ tái sinh. Các giải thích của Arjovsky và các ctác giả khác lý giải cho một số các triệu chứng liên quan đến đào tạo GAN.

Goodfellow và các đồng tác giả cũng chỉ ra rằng khi D tối ưu, đào tạo G tương đương với cực tiểu hóa phân kỳ Jensen-Shannon (Jensen-Shannon Divergence) giữa $\boldsymbol{p}_{\mathrm{G}}(\boldsymbol{x})$ và $\boldsymbol{p}_{\mathrm{data}}(\boldsymbol{x})$. Nếu D không tối ưu, cập nhật có thể ít ý nghĩa hơn, hay không chính xác. Cái nhìn sâu sắc về lý thuyết này đã là động lực thúc đẩy các nghiên cứu về các hàm chi phí dựa trên các khoảng cách lựa chọn khác.

7.5.2. Các kỹ thuật đào tạo

Một trong số các cải thiện chính đầu tiên trong đào tạo GAN cho tạo sinh các ành là các kiến trúc DGAN (Dynamicall Adaptive GAN: mạng đối nghịch tạo sinh thích ứng động) được đề xuất bởi Raddfor và một số tác giả khác. Công trình này là kết quả của sự khám phá sâu rộng các kiến trúc của các CNN được sử dụng trước đó trong thị giác máy tính và kết quả dẫn đến một tập các hướng dẫn cho việc cấu trúc và đàu tạo cả bộ tạo sinh lẫn bộ phân biệt. Trong phần 7.2.2, ta đã nhắc đến tầm quan trong của tích chập theo sải chân và theo một phần sải chân, đây là các phần tử then chốt của thiết kế kiến trúc. Điều này cho phép cả hai bộ tạo sinh và bộ phân biệt học các phép toán lấy mẫu tăng và lấy mẫu giảm tốt để có thể đóng góp các cải hiện cho chất lượng tổng hợp ảnh. Đặc biệt hơn đối với đào tạo là, chuẩn hóa lô (Batch Normalization) được khuyến nghị cho sử dụng trong cả hai mạng để ổn định đào tạo trong các mô hình sâu hơn. Một đề xuất khác là cực tiểu hóa số lượng các lớp kết nối hoàn toàn được sử dụng để tăng tính khả thi của các mô hình đào tạo sâu hơn. Cuối cùng, Raddfỏd và các tác giả khác đã chỉ ra rằng sử dụng các hàm kích hoạt ReLU dò (Leaky ReLU) giữa các lớp trung gian của bộ phân biệt cho hiệu suất vượt trội so với sử dụng các ReLU bình thường (Regular ReLU)

Muộn hơn, Salimans và các tác giả khác đã đề xuất các phương pháp dựa trên kinh nghiệm hơn nữa để ổn định và đào tạo các GAN. Phương pháp thứ nhất, thích hợp tính năng (Feature Matching), thay đổi đối tượng của bộ tạo sinh để tăng khối lượng thông tin khả dụng. Cụ thể là, bộ phân biệt vẫn được đào tạo để phân biệt giữa các mẫu thực và các mẫu giả, nhưng bộ tạo sinh bây giờ được đào tạo để làm phù hợp các kích hoạt trung gian (các tính năng) của các mẫu giả của nó được kỳ vọng bởi bộ phân biệt với các kích hoạt trung gian của các mẫu thực được kỳ vọng. Thứ hai, phân biệt phù hợp mini (Mini-Match), thêm một đầu vào bổ sung cho bộ phân biệt, đây là một tính năng để mã hóa khoảng cách giữa một mẫu cho trước trong một lô mini (Mini-Batch) các mẫu khác. Mục đích là để ngăn chặn sự sụp đổ chế độ (Mode Collapse) khi bộ phân biệt có thể dễ ràng phân biệt nếu bộ tạo sinh tạo ra các đầu ra giống nhau.

Một kỹ thuật thứ ba dựa trên kinh nghiệm, trung bình hóa theo kinh nghiệm (Heuristic Avaraging), phạt các tham số mạng nếu chúng lệch khỏi trung bình chạy (Running Average) của các giá trị trước, để có thể hội tụ đến một cân bằng. Thứ tư, chuẩn hóa lô ảo (Virtual Batch Normalization), giảm sự lệ thuộc một mẫu vào các mẫu khác trong lô mini bằng cách tính toán thông kê lô để chuẩn hóa với mẫu được đặt trong một lô mini tham chuẩn được cố định tại bắt đầu đào tạo.

Cuối cùng, làm mịn nhãn một mặt (One-Sided Label Smoothing) làm cho mục tiêu đối với bộ phân biệt là 0,9 thay vì 1 bằng cách làm mịn biên giới phân loại của bộ phân loại, vì thế ngăn chặn bộ phân biệt quá tự tin dẫn đẫn cung cấp các gradient yếu cho bộ tạo sinh. Sonderby và đồng tác giả đã nâng cấp ý tưởng thách thức bộ phân biệt bằng cách

cộng thêm tạp âm cho các mẫu trước khi đưa chúng đến bộ phân biệt. Sonderby và đồng tác giả đã lập luận rằng làm mịn nhãn một mặt làm lệch bộ phân biệt tối ưu, trong khi kỹ thuật của họ, tạp âm thực thể (Instance Noise), dịch chuyển các đa tạp của các mẫu thực và giả tạo gần nhau hơn, đồng thời ngăn chặn bộ phân biệt dễ ràng tìm thấy một biên giới phân tách hoàn toàn các mẫu thực và các mẫu giả. Trong thực tế điều này có thể được thực hiện bằng cách cộng tạp âm Gauss cho cả ảnh được tổng hợp và ảnh thực bằng cách đồng thời ủ (Annealing) lệch chuẩn theo thời gian. Muộn hơn, quá trình cộng tạp âm cho các mẫu dữ liệu để ổn định đào tạo đã được chính thức chứng minh là đúng bởi Arjovsky và các đồng tác giả.

7.6. CÁC TRÌNH BÀY KHÁC

Phần một của phần này xét các diễn giải lý thuyết thông tin và các tổng quát hóa của GAN. Phần hai xét đến các hàm chi phí khác nhắm đến trực tiếp xử lý vấn đề các gradient biến mất.

6.6.1. Các tổng quát hóa hàm chi phí của GAN

Nowozin và các đồng tác giả đã chỉ ra rằng đào tạo GAN có thể được tổng quát hóa để giảm thiểu không chỉ phân kỳ Jensen-Shannon (Jensen-Shannon Disvergence), mà còn ước tính các phân kỳ f (f-Disvergence); các GAN này được đề cập như là các f-GAN. Các phân kỳ f bao gồm các số đo phân kỳ nổi tiếng chẳng hạn phân kỳ Kullback-Lejbler. Nowozin và các đồng tác giả đã chỉ ra rằng phân kỳ f có thể được xấp xỉ hóa bằng cách áp dụng liên hợp Frenchel (Frenchel Conjugate) của phân kỳ f cho các mẫu được rút ra từ phân phối của các mẫu được tạo sinh mong muốn sau khi chuyển các mẫu này qua một bộ phân tích. Chúng cung cấp một danh sách các liên hợp Frenchel cho các phân kỳ f được dùng chung cũng như cũng như các hàm kích hoạt có thể được sử dụng trong lớp cuối cùng của mạng tạo sinh tùy thuộc vào lựa chọn phân kỳ f. Sau khi rút ra được các hàm chi phí tổng quát để đào tạo bộ tạo sinh và bộ phân biệt của một f-GAN, Nowozin và các đồng tác giả quan sát thấy rằng, trong dạng thô của nó, cực đại hóa đối tượng của bộ tạo sinh có khả năng dẫn đến các gradient yếu, đặc biệt tại khởi đầu đào tạo và đã đề xuất môt hàm chi phí khác để cập nhật bộ tạo sinh ít có khả năng gây bão hòa hơn tại bắt đầu đào tao. Nghĩa là, khi bộ phân biệt được đào tạo, đạo hàm của phân kỳ f trên tỷ số của của các phân phối dữ liệu thực và giả được ước tính, trong khi đó khi bộ tái sinh được đào tạo chỉ cực tiểu hóa ước tính của phân kỳ f. Uehara và các đồng tác giả mở rổng f GAN hơn nữa, tại đây trong bước của bộ phân biệt tỷ số của các các phân phối thực và

giả được dự đoán và trong bước của bộ tạo sinh phân kỳ f được trực tiếp giảm thiểu. Goodfellow cũng đưa ra các giải pháp khác cho phân kỳ JS.

7.6.2. Các hàm chi phí khác để ngăn ngừa các gradient biến mất

Arjovsky và các đồng tác giả khác đã đề xuất WGAN, GAN với một ham chi phí khác được rút ra từ một xấp xỉ hóa khoảng cách Wassertein (Wassertein Distance). Không giống hàm chi phí gốc, WGAN cóa khả năng hơn cung cấp các gradient hữu ích cho cập nhật bộ tạo sinh. Hàm chi phí được rút ra cho WGAN dựa trên bộ phân biệt, được đề cập đến như là "critic", là một hàm liên tục k-Lipschitz; thực tế, điều này có thể được thực hiện bằng cách đơn giản cắt xén ((rút ngắn) các tham số của bộ tạo sinh. Tuy nhiên một nghiên cứu cho rằng cắt xén trọng số một cách tiêu cực giảm dung lượng của mô hình bộ phân biệt, buộc nó phải học các hàm đơn giản hơn. Gulrajani và các đồng tác giả đề xuất một phương pháp được cải thiện cho đào tạo bộ phân biệt đối với một WGAN bằng cách phạt chuẩn của các gradient liên quan đến các mẫu dữ liệu trong quá trình đào tạo thay vì thực hiện cắt xén tham số.

7.7. SO SÁNH CÁC PHƯƠNG ÁN GAN

GAN cho phép ta tổng hợp các mẫu dự liệu mới từ tạp âm, nhưng bị coi là khó đào tạo do một phần là sự biến mất của các gradient. Tất cả các mô hình GAN mà ta đã xét đòi hỏi điều chỉnh siêu tham số và lựa chọn mô hình cẩn thận để đào tạo. Tuy nhiên, có thể là các mô hình AAE và WGAN dễ đào tạo hơn. AAE khá dễ đào tạo vì áp dụng mất mát đối nghịch cho một phân phối khá đơn giản với ít chiều hơn (Lower Dimentions) so với dữ liệu ảnh. WGAN được thiết kế để dễ đào tạo hơn bằng các sử dụng một biểu diễn đối tượng đào tạo khác không bị vấn đến biến mất gradient. Cũng có thể đào tạo thành công WGAN mà không cần chuẩn hóa lô; nó cũng ít nhậy cảm hơn với lựa chọn các phi tuyến tính được sử dụng giữa các lớp.

Các mẫu được tổng hợp bằng các sử dụng GAN hay WGAN có thể thuộc bất kỳ loại nào có mặt trong dữ liệu đào tạo. Các GAN có điều kiện là một cách tiếp cận để tổng hợp các mẫu có nội dung được đặc tả bởi người dùng.

Rõ ràng, từ các kỹ thuật trực quan hóa khác nhau (Visualization Techniques) (hình 7.10) rằng tổ chức của không gian tiềm ẩn chứa một nội dung nào đó, nhưng GAN thuần túy

(Vanilla GAN) không cung cấp một mô hình suy luận cho phép các mẫu số liệu ánh xạ đến các trình bày tiềm ẩn. Cả hai ALI và BiGAN đều cung cấp cơ chế để ánh xạ dữ liệu ảnh đến một không gian tiềm ẩn (suy luận), tuy nhiên, chất lượng tái cấu trúc gợi ý rằng chúng không nhất thiết phải trung thực mã hóa và giải mã các mẫu. Một phát hiện mới nhất chỉ ra rằng ALI có thể khôi phục trung thực các mẫu dữ liệu. Tuy nhiên, mô hình này chia sẻ nhiều điểm chung với AVB và AAR. Đây là các bộ tự mã hóa, giống như các bộ tự mã hóa biến thiên (VAE: Variational Autoencoder), nơi mà không gian tiềm ẩn được chính quy hóa bằng cách sử dụng đào tạo đối nghịch thay vì phân kỳ KL giữa các mẫu được mã hóa và mẫu trước đó.

7.8. KHÔNG GIAN TIỀM ẨN (LATENT SPACE)

Các GAN xây dựng các trình bày số liệu riêng của chúng, mà qua đó chúng được đào tạo, và thực hiện để tạo ra các không gian vectơ hình học có cấu trúc cho các lĩnh vực khác nhau. Đây là một phẩm chất được chia sẻ với các mô hình khác nhau bao gồm các VAE cũng như các mô hình ngôn ngữ học, chẳng hạn word2vector. Tổng quát, lĩnh vực dữ liệu cần lập mô hình được ánh xạ vào một không gian vectơ ít chiều hơn (Fewer Dimentions) so với không gian dữ liệu, buộc mô hình phải khám phá cấu trúc quan tâm trong dữ liệu và trình bày nó hiệu quả. Không gian tiềm ẩn này là tại đầu khởi nguồn của mạng tạo sinh và dữ liệu tại mức độ được trình bày này (không gian tiềm ẩn) có thể được cấu trúc rõ ràng, chặt chẽ và có thể hỗ trợ các hoạt động ngữ nghĩa mức cao. Các thí dụ bao gồm quay các khuôn mặt từ các quỹ đạo thông qua không gian tiềm ẩn, cũng như các tương tự ảnh có hiệu ứng hay bổ sung các đặc điểm chẳng hạn "kính mắt trên mặt trần" (Bare Face).

Tất cả các mô hình GAN (thuần chất) có một bộ tạo sinh để ánh xạ dữ liệu từ không gian tiềm ẩn vào không gian được lập mô hình, nhưng nhiều mô hình GAN có một "bộ mã hóa" hỗ trợ thêm ánh xạ đảo (Inverse Mapping). Điều này trở thành một phương pháp mạng mẽ để khám phá và sử dụng không gian tiềm ẩn được cấu trúc của mạng GAN. Bằng một bộ mã hóa, các tuyển tập của các ảnh được đánh nhãn được ánh xạ vào không gian tiềm ẩn và được phân tính để phát hiện "các vectơ khái niệm" trình bày các đặc điểm mức độ cao chẳng hạn "cười" hay "đội mũ". Các vectơ này có thể được áp dụng tại các bù trừ có định cỡ trong không gian tiềm ẩn để ảnh hưởng lên hành vi của bộ tạo sinh (hình 7.18). Tương tự như sử dụng một quá trình mã hóa để lập mô hình cho phân phối của các mẫu tiềm ẩn, Gurumurthy và các đồng tác giả đề xuất mô hình hỗn hợp các phân phối Gauss và học các thành phần của hỗn hợp để cực đại hóa khả năng (Likelihood) của các mẫu dữ liệu được tạo sinh bên dưới phân phối tạo sinh dữ liệu.

Hình 7.18. Thí dụ về áp dụng một "vectơ nụ cười" bằng mô hình ALI. Phía trái của hình này là một thí dụ về một phụ nữ không cười và phía phải là một phụ nữ có nụ cười. Giá trị *z* đối với ảnh của người phụ nữ phía trái là vốn có, z_1 và đối với phía phải là z_2. Nội suy trong một vectơ kết nối z_1 với z_2 cho ta các giá trị *z* có thể được truyền qua một tạo sinh để tổng hợp các mẫu mới. Lưu ý rằng sự liên quan, vectơ dịch chuyển trong không gian tiềm ẩn quét ngang "cường độ nụ cười" trong không gian hình ảnh

7.9. CÁC ỨNG DỤNG CỦA GAN

Khám phá các ứng dụng mới để đào tạo đối nghịch các mạng sâu là một lĩnh vực hoạt động của nghiên cứu. Ta khảo sát một số ứng dụng thị giác máy tính. Các ứng dụng này được lựa chọn nhấn mạnh các cách tiếp cận khác nhau sử dụng các trình bày dựa trên GAN để chỉnh sửa ảnh (Image Manipulation), phân tích hay mô tả đặc điểm ảnh, tất nhiên không phản ánh đầy đủ phạm vi sử dụng của các GAN.

Việc sử dụng các GAN cho phân loại ảnh đặt chúng vào một ngữ cảnh học máy rộng hơn và cung cấp một đánh giá định lượng các tính năng được chiết xuất trong học không có giám sát. Tổng hợp ảnh vẫn giữ nguyên là một khả năng lõi của GAN và đặc biệt hữu ích khi ảnh được tạo sinh có thể phải chịu các ràng buộc tồn tại trước đó. Siêu phân giải (Super Resolution) đưa ra một thí dụ về cách thức mà một phương pháp hiện có thể được bổ sung một phần tử mất mát đảo nghịch để đạt được các kết quả chất lượng cao hơn. Cuối cùng, chuyển đổi ảnh sang ảnh (Image-Image Traslation) chứng minh cách thức các GAN cung cấp một giải pháp đa dụng cho một họ các nhiệm vụ đòi hỏi tự động chuyển đổi một ảnh đầu vào thành một ảnh đầu ra.

7.9.1. Phân loại và hồi quy

Sau khi hoàn thành đào tạo GAN, mạng neuron có thể được sử dụng cho các nhiệm vụ phía sau. Chẳng hạn, các đầu ra của các lớp tích chập của bộ phân biệt có thể được sử dụng làm một bộ chiết xuất tính năng (Feature Extractor) với các mô hình tuyến tính đơn giản phù hợp trên đỉnh các tính năng này khi sử dụng một số lượng khiêm tốn nhất các cặp (ảnh hay nhãn). Chất lượng của các trình bày không có giám sát trong một mạng DGAN đã được đánh giá bằng cách áp dụng một bộ phân loại L2-SLM được chuẩn hóa. Các điểm số phân loại tốt đã đạt được khi sử dụng phương pháp này cho cả các tập dữ liệu có giám sát và không có giám sát, ngay cả các dữ liệu khác biệt với dữ liệu gốc.

Có thể cải thiện chất lượng trình bày dữ liệu khi đào tạo đối nghịch bao gồm cả cùng học một cơ chế suy luận như với ALI. Một vecto trình bày được được xây dựng khi sử dụng ba lớp ẩn cuối cùng của bộ mã hóa ALI, một bộ phân loại L2-LSM tương tự, đã đạt được tỷ lệ phân loại sai thấp hơn đáng kể so với DGAN. Ngoài ra, ALI đạt được các kết quả phân loại cao khi thông tin nhãn được kết hợp vào quy trình đào tạo.

Khi nguồn dữ liệu đào tạo được đánh nhãn hạn chế, đào tạo đối nghịch có thể được sử dụng để tổng hợp nhiều mẫu đào tạo hơn. Một số tác giả sử dụng GAN để tinh chế ảnh trong khi vẫn duy trì thông tin siêu dữ liệu (Annotation Information) của họ. Bằng cách đào tạo các mô hình chỉ trên các ảnh được tinh chế bởi GAN, Shivasstava và các đồng sự đã đạt được hiệu năng rất cao cho các nhiệm vụ ước tính tư thế và ánh mắt (Pose and Gaze Estimation). Tương tự kết quả tốt cũng nhận được cho ước tính ánh măt (Gaze Estimation) và dự đoán bằng cách sử dụng kiến trúc GAN không gian-thời gian. Trong một số trường hợp, các mô hình được đào tạo trên dữ liệu được tổng hợp không tổng quát hóa tốt khi được áp dụng cho số liệu thực. Bousmalis và các đồng sự đề xuất giải quyết vấn đề này bằng cách điều chỉnh thích ứng các mẫu tổng hợp từ một miền nguồn để phù hợp với một miền đích bằng cách sử dụng đào tạo đối nghịch. Liu và các đồng sự đề xuất sử dụng nhiều GAN -một GAN cho một miền- với các trọng số ràng buộc để tổng hợp các cặp của các mẫu ảnh tương ứng từ các miền khác.

Do chất lượng các mẫu được tạo sinh khó để đánh giá định lượng trên các mô hình, các nhiệm vụ phân loại có lẽ vẫn là một công cụ định lượng quan trọng cho ước tính hiệu năng các GAN, ngay cả khi các ứng dụng mới và khác nhau trong thị giác máy tinh đang bùng nổ.

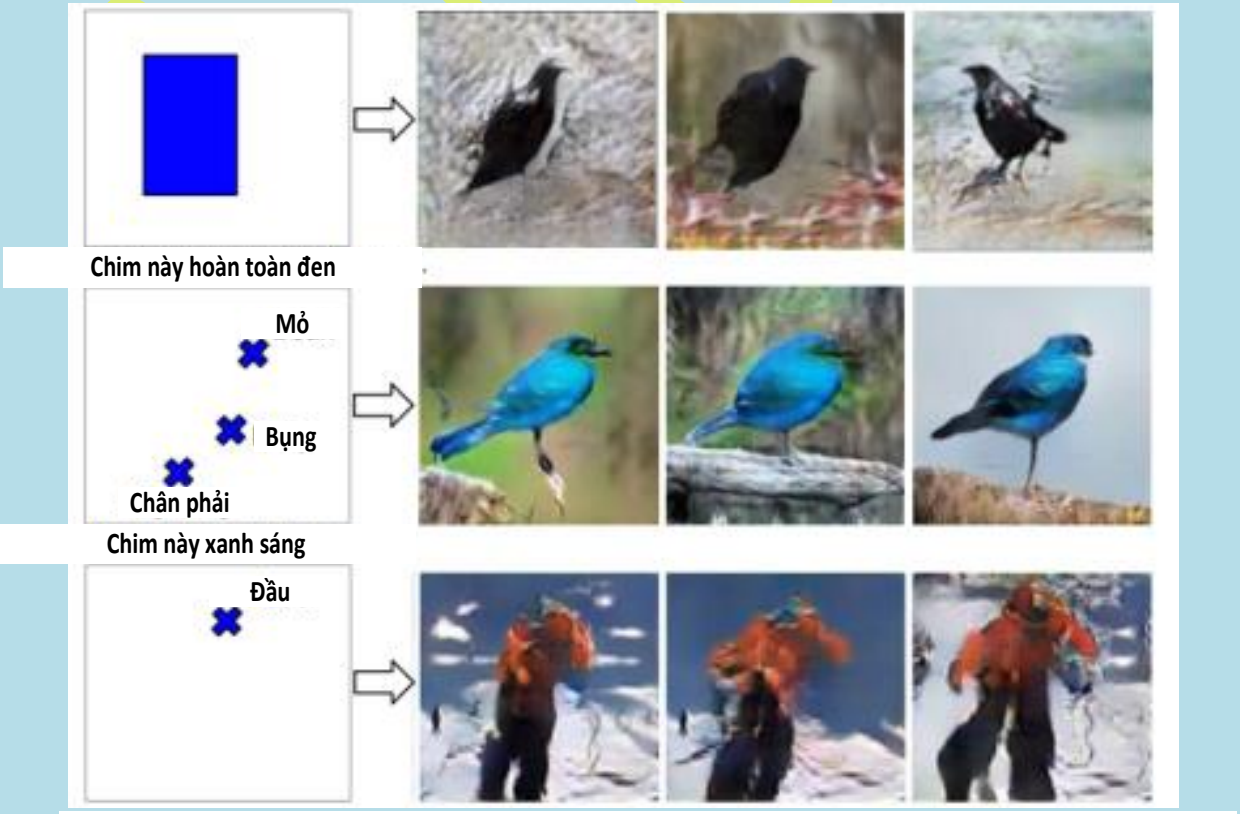
7.9.2. Tổng hợp ảnh

Rất nhiều nghiên cứu GAN tập trung lên cải thiện chất lượng ảnh và tiện ích của các khả năng tạo sinh ảnh. Mô hình LAPGAN đưa ra một nối tầng các mạng tích chập trong một bộ khung hình tháp Laplace để tạo sinh các ảnh theo cách từ thô đến mịn (Coarse to

Fine). Một các tiếp cận tương tự được Huang và các cộng sự sử dụng với các GAN hoạt động trên các trình bày trung gian thay vì các ảnh có độ phân giải thấp hơn.

LAPGAN cũng mở rộng đến phiên bản mô hình GAN tích chập, trong đó cả hai D và G nhật được thông tin nhãn bổ sung tại đầu vào; kỹ thuật chứng tỏ hữu ích hiện nay là một một thói quen phổ biến để cải thiện chất lượng ảnh. Ý tưởng này của GAN có điều kiện sau đó đã được mở rộng để kết hợp ngôn ngữ tự nhiên. Chẳng hạn, Reed và các cộng sự đã sử dụng một kiến trúc GAN để tổng hợp ảnh từ các mô tả văn bản trong đó ta có thể mô tả như là đảo ngược quy trình tạo chú thích (Reverse Captioning) hay chuyển đổi văn bản thành hình ảnh (Text to Image). Chẳng hạn cho trước một chú thích văn bản (Text Caption) của một con chim như là "trắng với một ít đen trên đầu và đôi cánh và một cái mỏ vàng dài", một GAN được đào tạo có thể tạo ra một số ảnh có thể có phù hợp với mô tả này.

Ngoài ra để đặt điều kiện cho các mô tả văn bản, mạng cái gì-ở đâu đối nghịch tạo sinh (GAWWN: Generative Adversarial What-Where Network) đặt điều kiện về vị trí của ảnh. Hệ thống GAWWN hỗ trợ một giao diện tương tác trong đó các ảnh lớn có thể được xây dựng từng bước với các mô tả văn bản của các phần và các hộp giới hạn (hình 7.19).



Hình 7.19.Các thí dụ tổng hợp ảnh sử dụng mạng cái gì-ở đâu đối nghịch tạo sinh (GAWWN) . Các ảnh trong GAWWN được đặt điều kiện trên cả hai các mô tả văn bản và các vị trí ảnh được đặc tả hoặc bởi điểm chính hay hộp giới hạn

Các GAN có điều kiện không chỉ cho phép ta tổng hợp các mẫu mới với các thuộc tính đặc thù, chùng còn cho phép ta phát triển các công cụ cho các ảnh biên tập theo trực giác-thí dụ biên tập kiểu tóc của một người trong một ảnh làm cho họ đeo kinh hay làm cho họ trong trẻ hơn.

7.9.3. Chuyển đổi ảnh vào ảnh

Các mạng đảo nghịch có điều kiện cũng phù hợp tốt cho chuyển đổi một ảnh đầu vào vào một ảnh đầu ra, đây là kỹ thuật cho một chủ đề lặp lại (Recurring Theme) trong đồ họa máy tính, xử lý ảnh, và thị giác máy tính. Pix2pix cung cấp một giải pháp đa năng cho họ các vấn đề này. Ngoài việc học ánh xạ từ ảnh đầu vào vào ảnh đầu ra, mô hình pix2pix cũng cấu trúc một hàm mất mát để đào tạo ánh xạ này. Mô hình này đã chứng minh các kết quả hiệu quả cho các vấn đề khác nhau của thị giác máy tính, theo đó thị giác máy tính trước đây đòi hỏi máy móc tách biệt bao gồm phân đoạn ngữ nghĩa, tạo sinh các bản đồ từ các photo thực và tô mầu các ảnh đen trắng. Wang và các cộng sự trình bày một ý tưởng tương tự khi sử dụng các GAN để tổng hợp trước tiên các bản đồ pháp tuyến bề mặt (Surface Normal Map) (giống như các bản đồ độ sâu) và sau đó ánh xạ chúng vào các cảnh tự nhiên.

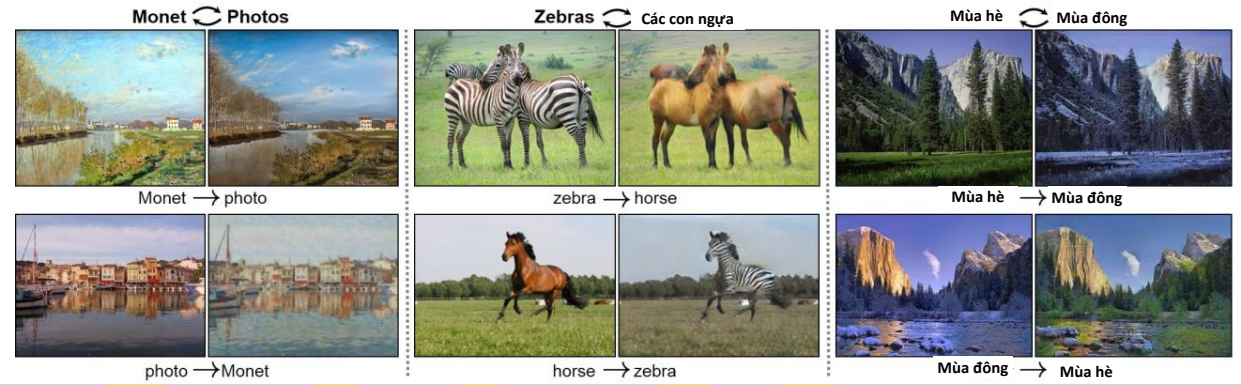
CycleGAN mở rộng công trình này bằng cách đưa ra một mất mát nhất quán tuần hoàn (Cycle Consistency Loss) nhằm bảo tồn ảnh gốc sau một chu kỳ chuyển đổi và chuyển đổi đảo ngược. Trong phương pháp này, việc làm phù hợp cặp ảnh không cần thiết đối với đào tạo. Điều này làm cho việc chuẩn bị dữ liệu đơn giản hơn nhiều và mở ra kỹ thuật cho họ các ứng dụng lớn hơn. Chẳng hạn, chuyển giao phong cách nghệ thuật (Artistic Style Transfer) kết xuất các ảnh theo phong cách của các họa sĩ, chẳng hạn Picasso hay Monet, đơn giảm bằng cách được đào tạo trên một tuyển tập không cặp đôi các bức tranh và các ảnh tự nhiên (hình 7.19).

7.9.4. Siêu phân giải (Super Resulation)

Siêu phân giải cho phép tạo ra một ảnh phân giải cao từ một ảnh phân giải thấp bằng cách mô hình được đào tạo suy luận các chi tiết giống như ảnh thật (Photo Realistic) khi lấy mẫu tăng. Mô hình SRGAN (Super-Resolution GAN: GAN siêu phân giải) mở rộng các

kết quả đạt được trước đây bằng cách bổ sung thêm một phần tử mất mát đối nghịch để ràng buộc các ảnh nằm trong đa tạp các ảnh tự nhiên.

Bộ tạo sinh SRGAN được lập điều kiện trên một ảnh phân giải thấp và suy luận các ảnh tự nhiên giống như ảnh thật với các thừa số tăng độ phân giải 4× (4×up-scaling factor). Không như hầu hết các ứng dụng của GAN, mất mát đối nghịch là một phần tử của một hàm mất mát lớn hơn bao gồm cả mất mát nhận thức từ bộ phân loại được đào tạo trước và mất mát chính quy hóa khích lệ các ảnh kết hợp không gian (Spacially Coherent Images). Trong ngữ cảnh này, mất mát đối nghịch ràng buộc giải pháp tổng thể đến đa tạp các ảnh tự nhiên và tạo ra các giải pháp có sức thuyết phục hơn về mặt nhận thức.



Hình 7.19. Mô hình CycleGAN học biến đổi ảnh vào ảnh giữa hai tuyển tập ảnh không có thứ tự. Hình vẽ cho thấy các thí dụ ánh xạ ảnh hai chiều: ánh xạ các bức tranh của Monet vào các ảnh phong cảnh, các con ngựa vằn vào các con ngựa và mùa hè vào mùa đông

Tùy biến các ứng dụng học sâu có thể xảy ra thường xuyên bởi sự sẵn có của các tập dữ liệu đào tạo được lựa chọn liên quan. Tuy nhiên, SRGAN tùy biến đến các miền đặc thù, vì các cặp ảnh đào tạo mới có thể được cấu trúc dễ ràng bằng cách lấy mẫu giảm (Down-Sampling) một ngữ liệu (Corpus) của các ảnh phân giải cao. Đây là một sự suy xét quan trọng trong thực tế, vì các các chi tiết giống như ảnh thật được suy luận do GAN tạo sinh sẽ thay đổi phụ thuộc vào miền các ảnh được sử dụng trong tập đào tạo.

Chương 8

CÁC BỘ BIẾN ĐỔI VÀ CÁC BỘ BIẾN ĐỔI THỊ GIÁC (TRANSFORMERS AND VISION TRANSFORMERS)

Bộ biến đổi ban đầu được giới thiệu cho các nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP: Natural Language Processing), nhưng chúng nhanh chóng được hầu hết các lĩnh vực học sâu, bao gồm cả thị giác máy tính, áp dụng. Chúng đo lường mối quan hệ giữa các cặp mã thông báo đầu vào (Input Token), từ (Word) trong trường hợp chuỗi văn bản, các phần hình ảnh đối với bộ biến đổi thị giác, được gọi là sự chú ý (Attention). Chi phí tăng theo cấp số nhân với số lượng mã thông báo. Đối với phân loại hình ảnh, kiến trúc bộ biến đổi phổ biến nhất chỉ sử dụng bộ mã hóa bộ biến đổi để biến đổi các mã thông báo đầu vào khác nhau. Tuy nhiên, cũng có rất nhiều ứng dụng khác trong đó phần giải mã của kiến trúc bộ biến đổi truyền thống cũng được sử dụng. Ở đây, trước tiên chúng cơ chế chú ý được đề cập (phần 8.1) và sau đó là khối bộ biến đổi cơ bản bao gồm bộ biến đổi thị giác (phần 8.2). Tiếp theo, một số cải tiến của bộ biến đổi thị giác để tính đến các tập dữ liệu nhỏ hoặc ít tính toán hơn được bàn luận (phần 8.3). Cuối cùng, các bộ biến đổi thị giác được áp dụng cho các tác vụ khác ngoài phân loại hình ảnh, chẳng hạn như phát hiện, phân đoạn, tạo và đào tạo mà không cần nhãn (phần 8.4) và các lĩnh vực khác, chẳng hạn như video hoặc đa phương thức (Multimodality) sử dụng dữ liệu văn bản hoặc âm thanh được giới thiệu (phần 8.5).

8.1. CHÚ Ý (ATTENTION)

Chú ý là một kỹ thuật trong Khoa học Máy tính mô phỏng cách não bộ tập trung vào các phần liên quan của thông tin đầu vào. Phần này sẽ giới thiệu về sự chú ý: lịch sử (tiểu phần 8.1.1), định nghĩa (tiểu phần 8.1.2), các loại và biến thể (các tiểu phần 8.1.3 và 8.1.4), và các thuộc tính của nó (tiểu phần 8.1.5).

Để hiểu chú ý là gì và tại sao nó lại hữu ích, hãy xem bài đánh giá phim sau:

Trong khi những người khác cho rằng câu chuyện này nhàm chán, tôi lại thấy nó hấp dẫn.

Bài đánh giá phim này mang tính tích cực hay tiêu cực? Phần đầu của câu không liên quan đến ý kiến của nhà phê bình, trong khi phần thứ hai gợi ý một cảm xúc tích cực với

từ "hấp dẫn". Đối với con người, câu trả lời là hiển nhiên; tuy nhiên, kiểu phân tích này không nhất thiết phải hiển nhiên đối với máy tính.

Thông thường, dữ liệu tuần tự cần được hiểu theo ngữ cảnh . Trong ngôn ngữ tự nhiên, một từ có nghĩa tùy thuộc vào vị trí của nó trong câu, so với các từ khác: ngữ cảnh của nó. Trong ví dụ của chúng ta, mặc dù từ " nhàm chán" chỉ gợi ý rằng bài đánh giá là tiêu cực, nhưng mối quan hệ ngữ cảnh của nó với các từ khác cho phép người đọc đi đến kết luận phù hợp. Trong thị giác máy tính, trong một nhiệm vụ như phát hiện đối tượng, bản chất của một pixel không thể được xác định riêng lẻ: chúng ta cần tính đến vùng lân cận nó, ngữ cảnh của nó. Vậy, làm thế nào chúng ta có thể chính thức hóa khái niệm ngữ cảnh trong dữ liệu tuần tự?

8.1.1. Lịch sử của sự chú ý

Khái niệm ngữ cảnh này là động lực đằng sau việc giới thiệu cơ chế chú ý (Attention Mechanism) vào năm 2015. Trước đó, việc dịch ngôn ngữ chủ yếu dựa vào kiến trúc bộ mã hóa-bộ giải mã: mạng neuron hồi quy (RNN) và đặc biệt là mạng bộ nhớ dài-ngắn hạn (LSTM) được sử dụng để mô hình hóa mối quan hệ giữa các từ. Cụ thể, mỗi từ của câu đầu vào được bộ mã hóa xử lý tuần tự. Tại mỗi bước, thông tin quá khứ và hiện tại được tóm tắt và mã hóa thành một vectơ có độ dài cố định. Cuối cùng, bộ mã hóa đã xử lý mọi từ và đưa ra một vectơ có độ dài cố định cuối cùng, tóm tắt tất cả thông tin đầu vào. Sau đó, vectơ cuối cùng này được giải mã và cuối cùng dịch thông tin đầu vào sang ngôn ngữ đích.

Tuy nhiên, vấn đề chính của cấu trúc này là tất cả thông tin đều được nén vào một vectơ có độ dài cố định. Do kích thước của các câu thay đổi và khi các câu dài ra, một vectơ có độ dài cố định thực sự là một nút thắt cổ chai: việc không mất thông tin trong quá trình mã hóa ngày càng trở nên khó khăn do vấn đề gradient biến mất.

Để giải quyết vấn đề này, Bahdanue và cộng sự đã đề xuất mô-đun chú ý (Attention Module) vào năm 2015. Mô-đun chú ý cho phép mô hình xem xét các phần của câu có liên quan đến việc dự đoán từ tiếp theo. Hơn nữa, điều này giúp hiểu rõ hơn mối quan hệ giữa các từ ở khoảng cách xa hơn.

8.1.2. Định nghĩa sự chú ý

Cho trước hai danh sách token (mã thông báo), $\boldsymbol{X} \in \mathbb{R}^{N \times d_x}$ và $\boldsymbol{Y} \in \mathbb{R}^{N \times d_y}$, chú ý mã hóa thông tin từ $\boldsymbol{Y}$ thành $\boldsymbol{X}$, trong đó N là độ dài của các đầu vào $\boldsymbol{X}$ và $\boldsymbol{Y}$, và d_x và d_y là các chiều hay kích thước (Dimention) tương ứng của chúng. Để làm được điều này, trước tiên chúng ta định nghĩa ba phép ánh xạ tuyến tính, ánh xạ truy vấn (Query Mapping) $\boldsymbol{W}^Q \in \mathbb{R}^{d_x \times d_q}$, ánh xạ khóa (Key Mapping) $\boldsymbol{W}^K \in \mathbb{R}^{d_x \times d_k}$ và ánh xạ giá trị (Value Mapping) $\boldsymbol{W}^V \in \mathbb{R}^{d_x \times d_v}$, trong đó d_q, d_k và d_v là các chiều nhúng mà truy vấn, khóa và giá trị sẽ được tính toán tương ứng.

Sau đó, ta định nghĩa truy vấn $\boldsymbol{Q}$, khóa $\boldsymbol{K}$ và giá trị $\boldsymbol{V}$ là:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^{Q}, \boldsymbol{K} = \boldsymbol{Y}\boldsymbol{W}^{K} \text{ và } \boldsymbol{V} = \boldsymbol{Y}\boldsymbol{W}^{V} \qquad (8.1)$$

Tiếp theo, ma trận chú ý được định nghĩa như sau:

$$\mathrm{A}\ (\boldsymbol{Q}, \boldsymbol{K}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}}\right) \qquad (8.2)$$

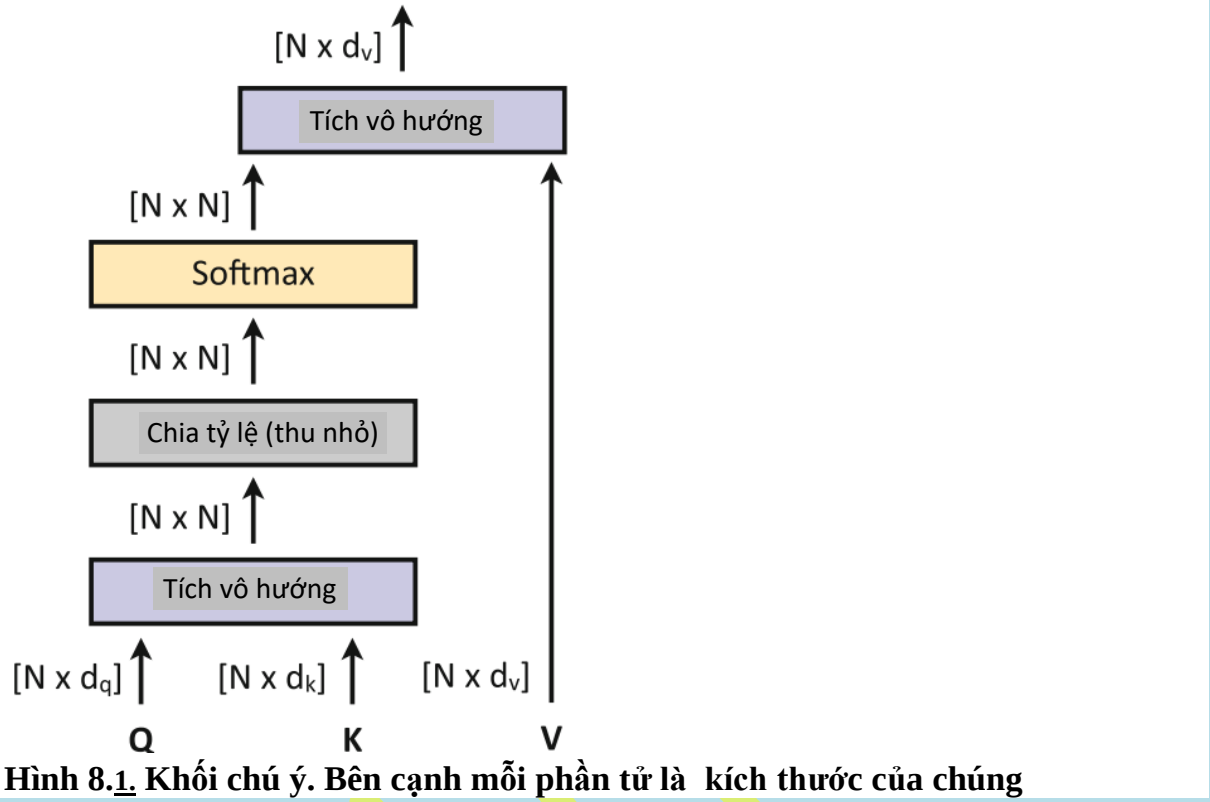
Điều này được thể hiện trong phần trái của hình 8.1. Tử số $\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ mô tả các thức mà từng phần của đầu vào trong X chú ý từng phần trong đầu ra y. Tích vô hướng này sau đó được đưa qua hàm softmax để chuẩn hóa các giá trị của nó và nhận được các giá trị dương cộng lại bằng 1. Tuy nhiên, đối với các giá trị lớn của $\mathrm{d_k}$, điều này có thể khiến softmax có độ dốc cực kỳ nhỏ, do đó nó được thu nhỏ lại theo $\sqrt{\mathrm{d_k}}$.

Ma trận N × N kết quả mã hóa mối quan hệ giữa ***X*** đối với ***Y***: nó đo lường mức độ quan trọng của một mã thông báo trong ***X*** so với một mã thông báo khác trong ***Y.***

Cuối cùng, đầu ra sự chú ý (Attention Output) được định nghĩa như sau:

$$\text{Attention}\ (\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{A}\ (\boldsymbol{Q}, \boldsymbol{K})\ \boldsymbol{V} \qquad (8.3)$$

Hình 7.1 minh họa điều này. Đầu ra của sự chú ý mã hóa thông tin của từng mã thông báo bằng cách tính đến thông tin ngữ cảnh. Do đó, thông qua các tham số có thể học (Learnable Parameters) – các truy vấn, các khóa và các giá trị-các lớp chú ý học được cách nhúng mã thông báo có tính đến mối quan hệ của chúng.



Hình 8.1. Khối chú ý. Bên cạnh mỗi phần tử là kích thước của chúng

Mối quan hệ theo ngữ cảnh

Phương trình (8.3) mã hóa các mối quan hệ ngữ cảnh như thế nào? Để trả lời câu hỏi này, hãy cùng xem xét lại việc phân tích cảm xúc của các bài đánh giá phim. Để mã hóa các mối quan hệ ngữ cảnh vào việc nhúng từ, trước tiên chúng ta cần một ma trận biểu diễn mối quan hệ giữa tất cả các từ. Để làm như vậy, cho một câu có độ dài N , chúng ta lấy từng vectơ từ và đưa nó vào hai lớp tuyến tính khác nhau, gọi một đầu ra là "truy vấn" và đầu ra kia là "khóa". Chúng ta đóng gói các truy vấn vào ma trận **Q** và các khóa vào ma trận **K** , bằng cách lấy tích của chúng ($\mathbf{QK}^T$). Kết quả là một ma trận $N \times N$ giải thích tầm quan trọng của từ thứ i (theo hàng) đối với việc hiểu từ thứ j (theo cột). Ma trận này sau đó được chia tỷ lệ (Scaled) và chuẩn hóa bằng phép chia và softmax. Tiếp theo, chúng ta đưa các vectơ từ vào một lớp tuyến tính khác, gọi đầu ra của nó là "giá trị". Chúng ta nhân hai ma trận này với nhau. Kết quả của tích của chúng là các vectơ chú ý mã hóa nghĩa của từng từ, bằng cách bao gồm cả nghĩa ngữ cảnh của chúng. Vì mỗi truy vấn, khóa và giá trị này là tham số có thể học được nên khi lớp chú ý được đào tạo, mô hình sẽ học cách mã hóa các mối quan hệ giữa các từ trong dữ liệu.

8.1.3. Các loại chú ý

Có hai loại cơ chế chú ý chiếm ưu thế: chú ý tự thân (Self Attention) và chú ý chéo (Cross Attention). Trong chú ý tự thân , các truy vấn, khóa và giá trị đến từ cùng một đầu

vào, tức là $\mathbf{X} = \mathbf{Y}$; trong chú ý chéo , các truy vấn đến từ một đầu vào khác với các vectơ khóa và giá trị, tức là $\mathbf{X} \neq \mathbf{Y}$.

8.1.3.1. Chú ý tự thân

Trong chú ý tự thân, các token (mã thông báo) của $\boldsymbol{X}$ tự chú ý đến chính chúng ($\boldsymbol{X} = \boldsymbol{Y}$). Do đó, nó được mô hình hóa như sau:

$$\text{SA}(\boldsymbol{X}) = \text{Attention}(\boldsymbol{X}\boldsymbol{W}^{Q}, \boldsymbol{X}\boldsymbol{W}^{K}, \boldsymbol{X}\boldsymbol{W}^{V}) \qquad (7.4)$$

Chú ý tự thân chính thức hóa khái niệm ngữ cảnh. Nó học các mô hình cơ bản về cách các phần của dữ liệu đầu vào tương ứng với nhau. Bằng cách thu thập thông tin từ cùng một tập hợp, với một chuỗi các token được cung cấp, một token có thể chú ý đến các token lân cận để tính toán đầu ra của nó.

8.1.3.2. Chú ý chéo

Hầu hết dữ liệu trong thế giới thực đều là đa phương thức-ví dụ, video chứa khung hình, âm thanh và phụ đề, hình ảnh có chú thích, v.v. Do đó, các mô hình có thể xử lý các loại thông tin đa phương thức như vậy đã trở nên cần thiết.

Chú ý chéo là một cơ chế chú ý được thiết kế để xử lý các đầu vào đa phương thức. Không giống như chú ý tự thân, nó trích xuất các truy vấn từ một nguồn đầu vào và các cặp khóa-giá trị từ một nguồn khác ($\boldsymbol{X} \neq \boldsymbol{Y}$). Nó trả lời câu hỏi sau: "Những phần nào của đầu vào $\boldsymbol{X}$ và đầu vào $\boldsymbol{Y}$ tương ứng với nhau?" Sự chú ý chéo (CA) được định nghĩa là:

$$\text{CA}(\boldsymbol{X}, \boldsymbol{Y}) = \text{Attention}(\boldsymbol{X}\boldsymbol{W}^{Q}, \boldsymbol{Y}\boldsymbol{W}^{K}, \boldsymbol{Y}\boldsymbol{W}^{V}) \qquad (7.5)$$

8.1.4. Biến thể của chú ý

Sự chú ý thường được sử dụng theo hai cách: (1) sự chú ý tự thân đa đầu (MSA: Multi-head Self Attention), và (2) sự chú ý đa đâu được che giấu (MMSA: Masked Multi-Head Self Attention).

Đầu chú ý (Attention Head)

Ta gọi sự chú ý là cơ chế được trình bày trong phần con 8.1.2 , tức là phép chiếu khóa-giá trị truy vấn, theo sau là sự chú ý theo tích vô hướng được chia tỷ lệ (Phương trình (8.1) và (8.2)).

Khi sử dụng mô hình dựa trên chú ý, việc chỉ dựa vào một đầu chú ý duy nhất có thể ức chế việc học. Do đó, khối chú ý đa đầu được giới thiệu.

Chú ý tự thân đa đầu (MSA)

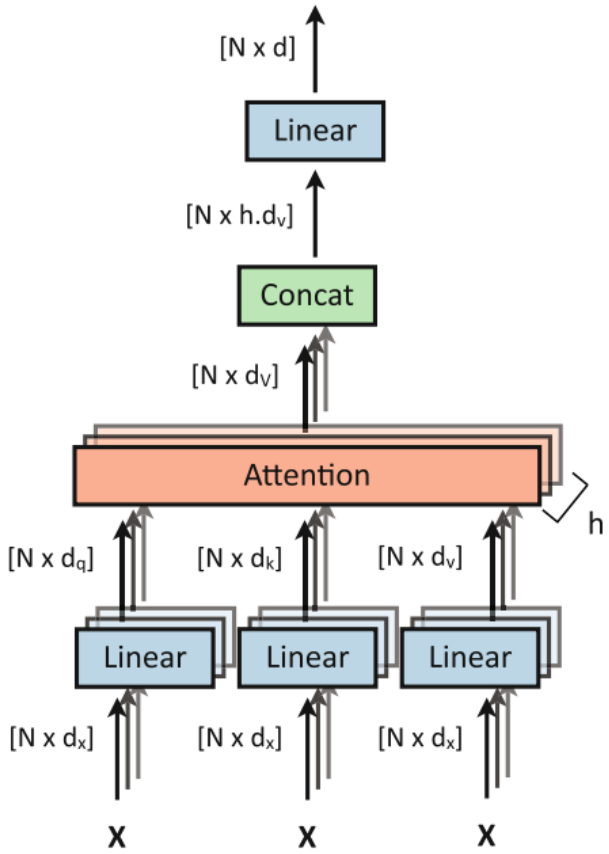
MSA được thể hiện trong Hình 8.2 và được định nghĩa như sau:

$$\text{MSA}\ (\boldsymbol{X}) = \text{Concat}((\text{head}_1\ (\boldsymbol{X}), \ldots, \text{head}_h\ (\boldsymbol{X}))\mathbf{W}^O \qquad (8.6)$$

trong đó

$$\text{Head}_i(\mathbf{X}) = \text{SA}(\mathbf{X}),\ \forall\, i \in \{1, h\},$$

Concat (Concatenation) là móc nối của h đầu chú ý và $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d}$ là ma trận chiếu (Project Matrix). Điều này có nghĩa là chiều nhúng ban đầu d_x được phân tích thành $h \times d_v$ và việc tính toán trên mỗi đầu được thực hiện độc lập. Các đầu chú ý độc lập thường được móc nối và nhân với một lớp tuyến tính để khớp với chiều đầu ra mong muốn. Kích thước đầu ra thường giống với kích thước nhúng đầu vào d . Điều này cho phép xếp chồng nhiều khối dễ dàng hơn.



Linear: tuyến tính
Concat (Concatenation): móc nối
Attention: chú ý
[a×b]: kích thước ma trận

Hình 8.2. Khối chú ý tự thân đa đầu (MSA). Trước tiên đầu vào *X* được chiếu hình vào các tham vấn (Queries), các khóa (Keys) và các giá trị (Values), sau đó được đưa qua các khối chú ý h. Đầu ra chú ý h kết quả sau đó được móc nối với nhau và

cuối cùng được chiếu hình vào một vectơ đầu ra kích thước d. Bên cạch các phần tử là kích thước của nó (kích thước ma trận)

Chú ý tự thân chéo đa đầu (MCA)

Tương tự như MSA, MCA được định nghĩa như sau:

$$\text{MCA}(\boldsymbol{X}) = \text{Concat}((\text{head}_1(\boldsymbol{X},\boldsymbol{Y}), \ldots, \text{head}_h(\boldsymbol{X},\boldsymbol{Y}))\boldsymbol{W}^O \quad (8.6)$$

trong đó

$\text{Head}_i(\mathbf{X},\mathbf{Y}) = \text{CA}(\mathbf{X},\mathbf{Y}), \ \forall i \in \{1, h\},$

Chú ý tự thân đa đầu được che dấu (MMSA)

Lớp MMSA là một biến thể khác của sự chú ý. Nó có cùng cấu trúc với khối chú ý tự thân đa đầu (MSA), nhưng tất cả các vectơ sau trong đầu ra mục tiêu đều được che dấu. Khi xử lý dữ liệu tuần tự, điều này có thể giúp việc đào tạo song song.

8.1.5. Các tính chất của chú ý

Trong khi chú ý mã hóa các mối quan hệ theo ngữ cảnh, thì nó tương đương với hoán vị , vì cơ chế này không tính đến thứ tự của dữ liệu đầu vào. Như được thể hiện trong phương trình 8.3, tất cả các phép tính chú ý đều là phép nhân ma trận và chuẩn hóa. Do đó, một đầu vào được hoán vị sẽ dẫn đến một đầu ra được hoán vị. Tuy nhiên, trên thực tế, đây có thể không phải là biểu diễn chính xác của thông tin. Ví dụ, hãy xem xét các câu "con khỉ ăn chuối" và "quả chuối ăn khỉ". Chúng có ý nghĩa riêng biệt do thứ tự của các từ. Nếu thứ tự của đầu vào là quan trọng, nhiều cơ chế khác nhau, chẳng hạn như mã hóa vị trí, được thảo luận trong tiểu mục. 8.2.1.2 , được sử dụng để nắm bắt sự tinh tế này.

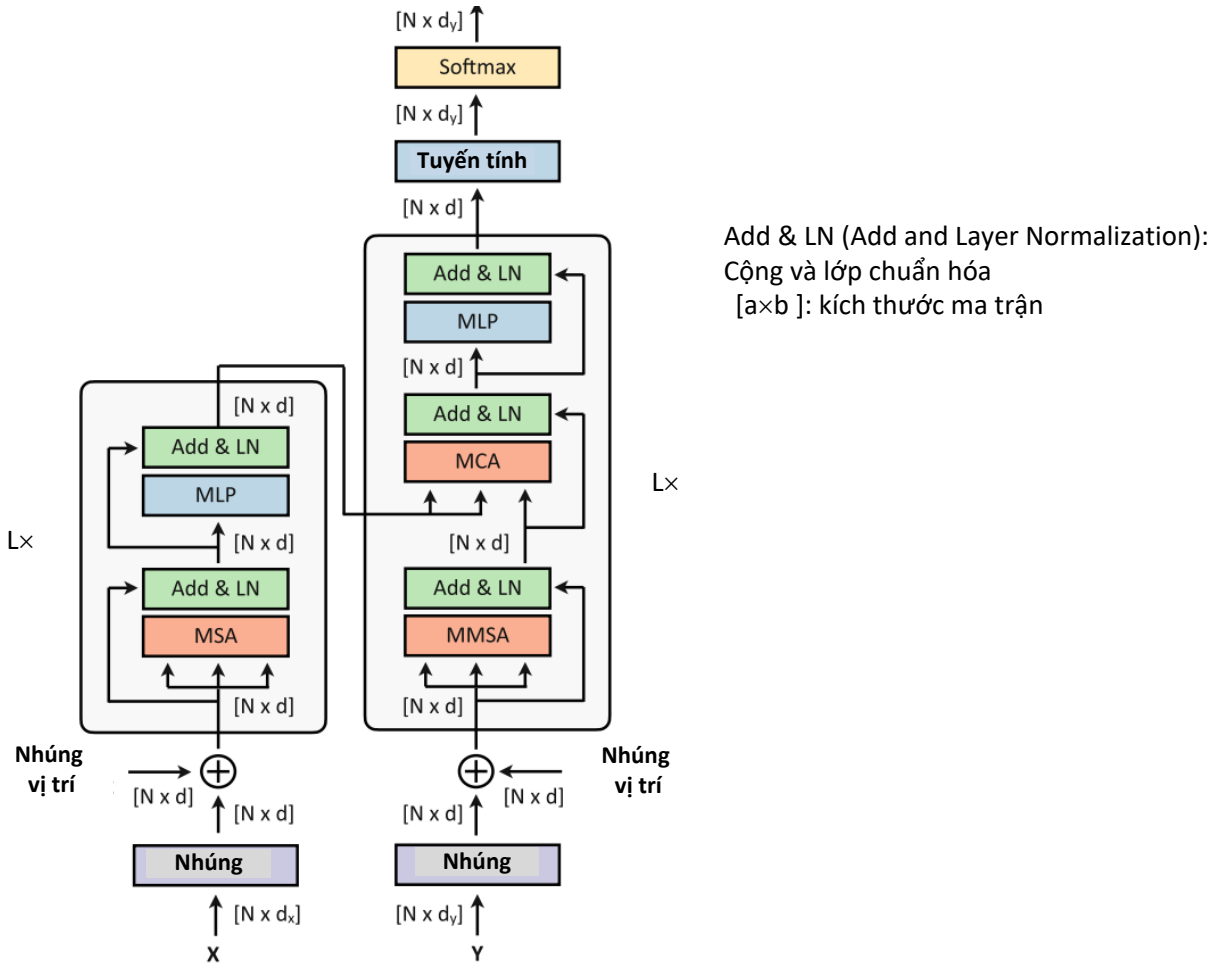
8.2. BỘ BIẾN ĐỔI THỊ GIÁC (VISION TRANSFORMER)

Kiến trúc của bộ biến đổi thị giác (ViT: Vision Transformer) là kiến trúc đầu tiên hoàn toàn dựa vào chú ý để rút ra kết nối giữa đầu vào và đầu ra. Kể từ khi ra mắt, nó đã cách mạng hóa học sâu, tạo ra những đột phá trong nhiều lĩnh vực, bao gồm xử lý ngôn ngữ tự nhiên, thị giác máy tính, hóa học và sinh học, do đó trở thành kiến trúc mặc định cho các biểu diễn học tập. Gần đây, bộ biến đổi tiêu chuẩn này đã được điều chỉnh cho các nhiệm vụ thị giác. Và một lần nữa, bộ biến đổi thị giác đã trở thành một trong những kiến trúc trung tâm trong thị giác máy tính.

Phần này trước tiên sẽ trình bày kiến trúc cơ bản của bộ biến đổi (8.2.1) và sau đó trình bày các ưu điểm của nó (8.2.2) . Cuối cùng, phần này mô tả bộ biến đổi thị giấc (Vision Transforer) (8.2.3).

8.2.1. Bộ biến đổi cơ bản

Như thể hiện trong hình 8.3 , kiến trúc bộ biến đổi tiêu chuẩn là một mô hình bô mã hóa-bộ giải mã (Encoder-Decoder). Đầu tiên, nó nhúng các mã thông báo đầu vào $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ vào một không gian tiềm ẩn, tạo ra các vectơ tiềm ẩn $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$, $\boldsymbol{Z}$ được đưa vào bộ giải mã để xuất ra $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$. Bộ mã hóa là một ngăn xếp gồm L lớp, mỗi lớp bao gồm hai khối con: các lớp chú ý tự thân đa đầu (MSA) và một perceptron đa lớp (MLP). Bộ giải mã cũng là một ngăn xếp gồm L lớp, mỗi lớp bao gồm ba khối con: chú ý tự thân đa đầu có mặt nạ (MMSA), chú ý chéo đa đầu (MCA) và MLP.



Hình 8.3. Kiến trúc của bộ biến đổi. Kiến trúc gồm một bộ mã hóa (khối bên trái) và một bộ giải mã (khối bên phải), mỗi khối có một chuỗi các khối chú ý tự thân (MSA và MMSA). Bên cạnh mỗi phần tử là kích thước của nó (kích thước ma trận)

Dưới đây, ta sẽ mô tả các phần khác nhau của kiến trúc bộ biến đổi trong hình 8.3 Đầu tiên, các mã thông báo đầu vào (Input Tokens) được chuyển đổi thành các mã thông báo nhúng (Embedđing Tokens). Sau đó, mã hóa vị trí thêm một mã thông báo vị trí vào mỗi mã thông báo nhúng để biểu thị thứ tự của các mã thông báo. Sau đó là bộ mã hóa của bộ

biến đổi (Transformer Encoder) . Bộ này bao gồm một ngăn xếp L lớp chú ý đa đầu, chuẩn hóa và các lớp MLP, mã hóa đầu vào thành một tập hợp các tính năng có ý nghĩa về mặt ngữ nghĩa. Sau đó, bộ giải mã theo sau. Bộ này bao gồm một chồng L lớp chú ý đa đầu có mặt nạ, lớp chú ý đa đầu và MLP theo sau là chuẩn hóa, bộ này giải mã các tính năng đầu vào liên quan đến các mã thông báo nhúng đầu ra. Cuối cùng, đầu ra được chiếu lên các lớp tuyến tính và softmax.

8.2.1.1. Nhúng

Bước đầu tiên của bộ biến đổi bao gồm việc chuyển đổi các mã thông báo đầu vào thành các mã thông báo nhúng, tức là các vectơ có các tính năng có ý nghĩa. Để thực hiện việc này, theo quy trình tiêu chuẩn, mỗi đầu vào được chiếu vào một không gian nhúng để thu được các mã thông báo nhúng $\boldsymbol{Z}^{e}$. Không gian nhúng được cấu trúc theo cách mà khoảng cách giữa một cặp vectơ liên quan đến độ tương đồng ngữ nghĩa của các từ liên kết với chúng. Đối với trường hợp NLP ban đầu, điều này có nghĩa là chúng ta nhận được một vectơ của mỗi từ, sao cho các vectơ gần nhau hơn có ý nghĩa tương tự nhau.

8.2.1.2. Mã hóa vị trí

Như đã trình bày trong phần con 8.1.5, cơ chế chú ý không phụ thuộc vào vị trí, nghĩa là nó không lưu trữ thông tin về vị trí của mỗi đầu vào. Tuy nhiên, trong hầu hết các trường hợp, thứ tự của các mã thông báo đầu vào là quan trọng và cần được lưu ý, chẳng hạn như thứ tự các từ trong một câu rất quan trọng vì chúng có thể thay đổi nghĩa của câu. Do đó, các nhà nghiên cứu giới thiệu mã hóa vị trí (Positoning Encoding) $\mathrm{PE} \in \mathbb{R}^{N \times d_x}$, để thêm một mã thông báo vị trí vào mỗi mã thông báo nhúng $\mathrm{PE} \in \mathbb{R}^{N \times d_x}$.

Mã hóa vị trí hình sin

Mã hóa vị trí hình sin là phương pháp mã hóa vị trí chính, mã hóa vị trí của mỗi mã thông báo bằng các sóng hình sin của nhiều tần số. Đối với mã thông báo nhúng $Z^{e} \in \mathbb{R}^{N \times d_x}$, mã hóa vị trí của nóPE $\mathrm{PE} \in \mathbb{R}^{N \times d_x}$ được định nghĩa là:

$$\mathrm{PE}\ i, 2j\ = \sin\left(\frac{i}{10000^{2j/d}}\right) \qquad (8.7)$$

$$\mathrm{PE}\ i, 2j\ = \cos\left(\frac{i}{10000^{2j/d}}\right)$$

trong đó $\forall i, j \in \left[\left|1, n\right|\right] \times \left[\left|1, d\right|\right]$

Mã hóa vị trí có thể học được

Một cách tiếp cận trực giao là để mô hình học mã hóa vị trí. Trong trường hợp này, $\mathrm{PE} \in \mathbb{R}^{N \times d_x}$ trở thành một tham số có thể học được. Tuy nhiên, điều này làm tăng yêu cầu về bộ nhớ, mà không nhất thiết mang lại sự cải thiện so với mã hóa hình sin.

Nhúng vị trí

Sau khi tính toán, **PE** mã hóa vị trí sẽ hoặc được thêm vào mã thông báo nhúng hoặc chúng được nối lại như sau:

$$Z^{pe} = Z^{e} + \mathrm{PE}$$

hoặc

$$Z^{pe} = \mathrm{Concat}\ \ Z^{e}, \mathrm{PE} \qquad (8.8)$$

trong đó Concat biểu thị móc nối vector. Lưu ý rằng móc nối có ưu điểm là không làm thay đổi thông tin chứa trong $\mathbf{Z}^{e}$, vì thông tin vị trí chỉ được thêm vào kích thước (Dimention) không sử dụng. Tuy nhiên, nó lại làm tăng kích thước đầu vào, dẫn đến yêu cầu bộ nhớ cao hơn. Thay vào đó, phép cộng lại bảo toàn cùng một kích thước đầu vào trong khi thay đổi nội dung của các token nhúng. Khi kích thước đầu vào lớn, việc thay đổi nội dung này là không đáng kể, vì hầu hết nội dung được bảo toàn. Do đó, trên thực tế, đối với kích thước lớn, việc cộng các mã hóa vị trí được ưu tiên, trong khi đối với kích thước nhỏ, việc móc nối chúng lại được ưu tiên.

8.2.1.3 Khối bộ mã hóa

Khốibộ mã hóa nhận đầu vào là các token nhúng và token vị trí, sau đó xuất các đặc trưng của đầu vào để được giải mã bởi khối giải mã. Khối bộ mã hóa bao gồm một chồng L lớp chú ý tự tự thân đa đầu (MSA) và một perceptron đa lớp (MLP). Cụ thể, các nhúng và token vị trí, $Z_x^{pe} \in \mathbb{R}^{N \times d}$, đi qua một khối chú ý tự thân đa đầu. Sau đó, một kết nối dư với chuẩn hóa lớp được triển khai. Trong bộ chuyển đổi, thao tác này được thực hiện sau mỗi lớp con. Tiếp theo, ta đưa đầu ra của nó vào một MLP và một lớp chuẩn hóa. Thao tác này được thực hiện L lần, và mỗi lần đầu ra của mỗi khối mã hóa (kích thước $N \times d$) là đầu vào của khối tiếp theo. Trong lần thứ L , đầu ra của chuẩn hóa là đầu vào của khối chú ý chéo trong bộ giải mã (phần con 8.2.1.4).

8. 2.1.4. Khối bộ giải mã

Bộ giải mã có hai đầu vào: đầu tiên, một đầu vào cấu thành các truy vấn $Q \in \mathbb{R}^{N \times d}$ của bộ mã hóa và thứ hai, đầu ra của bộ mã hóa tạo nên cặp khóa - giá trị $K, V = \mathbb{R}^{N \times d}$. Tương tự như phần con 8.2.1.1 và 8.2.1.2 , bước đầu tiên là mã hóa token đầu ra thành token nhúng đầu ra và mã token vị trí đầu ra. Các token này được đưa vào phần chính của bộ giải mã, bao gồm một chồng các lớp chú ý tự thân đa đầu (MMSA) có mặt nạ L , các lớp chú ý chéo đa đầu (MCA) và perceptron đa lớp (MLP) theo sau là các phép chuẩn hóa. Cụ thể, các token nhúng và token vị trí, $Z_y^{pe} \in \mathbb{R}^{N \times d}$, đi qua một khối MMSA. Sau đó, một kết nối dư thừa với chuẩn hóa lớp được thực hiện. Tiếp theo, một lớp MCA (tiếp theo là chuẩn hóa) ánh xạ các truy vấn đến các giá trị khóa đã mã hóa trước khi chuyển tiếp đầu ra đến một MLP. Cuối cùng, ta chiếu đầu ra của các khối giải mã L (kích thước $N \times d_y$) qua một lớp tuyến tính và lấy xác suất đầu ra thông qua một lớp softmax.

8.2.3. Ưu điểm của bộ biến đổi

Kể từ khi ra mắt, bộ biến đổi đã có tác động đáng kể đến các phương pháp học sâu.

Trong xử lý ngôn ngữ tự nhiên (NLP), trước khi có bộ biến đổi, hầu hết các kiến trúc thường dựa vào các mô-đun hồi quy, chẳng hạn như RNN và đặc biệt là LSTM. Tuy nhiên, các mô hình hồi quy xử lý đầu vào theo trình tự, nghĩa là để tính toán trạng thái hiện tại, chúng cần đầu ra của trạng thái trước đó. Điều này khiến chúng cực kỳ kém hiệu quả vì không thể song song hóa. Ngược lại, trong bộ biến đổi, mỗi đầu vào được xử lý độc lập với các đầu vào khác và bộ chú ý đa đầu có thể thực hiện nhiều phép tính chú ý cùng một lúc. Điều này làm cho bộ biến đổi có hiệu suất cao vì chúng có khả năng song song hóa cao.

Điều này không chỉ mang lại khả năng mở rộng đặc biệt, cả về độ phức tạp của mô hình và quy mô của tập dữ liệu, mà còn về tốc độ đào tạo tương đối nhanh. Đáng chú ý, các bộ biến đổi gần đây đã được đào tạo trước trên 34 tỷ mã thông báo (Token) từ tập dữ liệu C4, giúp mở rộng mô hình lên hơn 1 nghìn tỷ tham số.

Khả năng mở rộng này là lý do chính cho sức mạnh của bộ biến đổi. Mặc dù ban đầu nó được đưa vào để dịch, nhưng nó không đưa vào nhiều độ lệch quy nạp (Inductive Bias), tức là tập hợp các giả định mà người dùng đưa ra về cấu trúc của đầu vào mô hình. Khi làm như vậy, bộ biến đổi dựa vào dữ liệu để tìm hiểu cách chúng được cấu trúc. So với các đối tác có nhiều độ lệch hơn, bộ biến đổi yêu cầu nhiều dữ liệu hơn để tạo ra kết quả tương đương. Tuy nhiên, nếu có đủ lượng dữ liệu, việc thiếu độ lệch quy nạp sẽ trở thành một điểm mạnh. Bằng cách học cấu trúc của dữ liệu từ dữ liệu, bộ biến đổi có thể học tốt hơn mà không bị cản trở bởi các giả định của con người.

Trong hầu hết các tác vụ liên quan đến bộ biến đổi, mô hình trước tiên được đào tạo trước trên một tập dữ liệu lớn và sau đó được tinh chỉnh cho nhiệm vụ trong sẵn có trên một tập

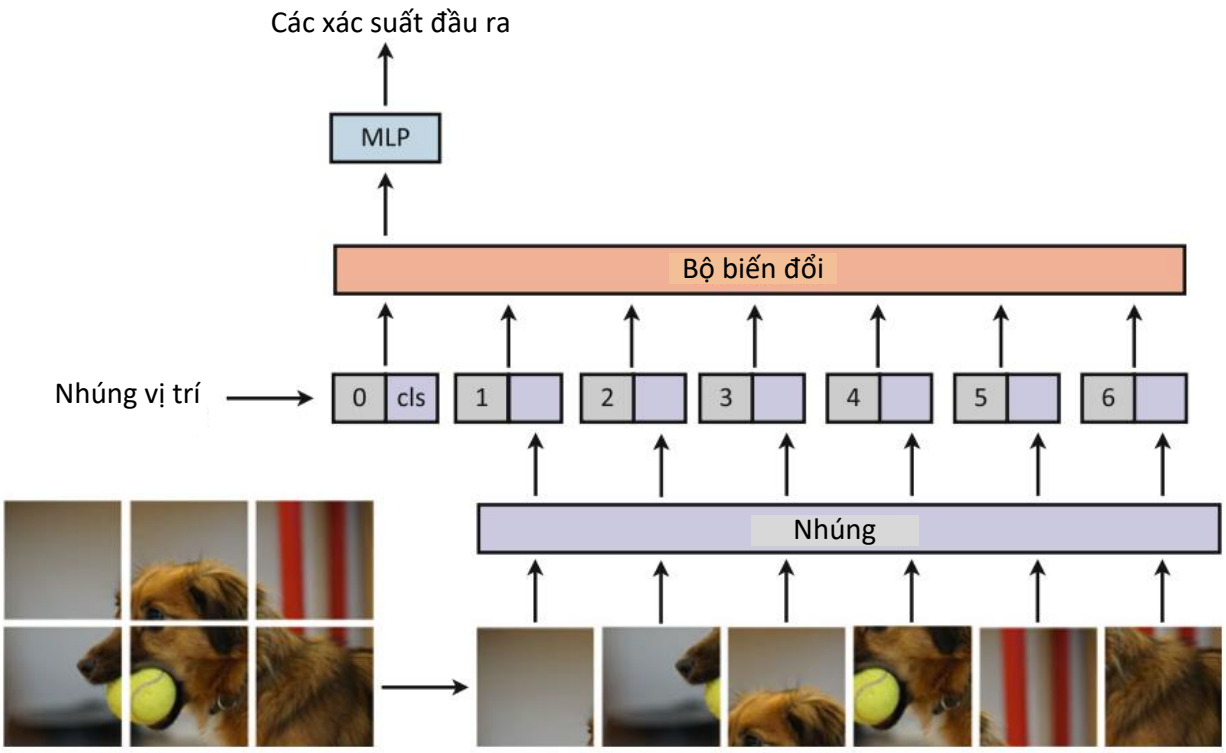
dữ liệu nhỏ hơn. Giai đoạn tiền đào tạo là cần thiết để bộ biến đổi học được cấu trúc toàn cục của phương thức đầu vào cụ thể. Để tinh chỉnh, thường thì ít dữ liệu hơn là đủ vì mô hình đã phong phú. Ví dụ, trong xử lý ngôn ngữ tự nhiên, BERT (Binary Encoder Reppresantations from Transformer: mô tả bộ mã hóa nhị phân từ bộ biến đổi) , một mô hình ngôn ngữ hiện đại, được đào tạo trước trên một tập dữ liệu dựa trên Wikipedia, với hơn 6 triệu bài viết và kho sách với hơn 10.000 cuốn sách. Sau đó, mô hình này có thể được tinh chỉnh trên các nhiệm vụ cụ thể hơn nhiều. Trong thị giác máy tính, bộ biến đổi thị giác (ViT) được đào tạo trước trên tập dữ liệu JFT-300M, chứa hơn 1 tỷ nhãn cho 300 triệu hình ảnh. Do đó, với một lượng dữ liệu đủ, bộ biến đổi đạt được kết quả chưa từng có trước đây trong nhiều lĩnh vực của học máy.

8.2.4. Bộ biến đổi thị giác (ViT:Vision Transformer)

Bộ biến đổi cung cấp một giải pháp thay thế cho CNN vốn từ lâu đã thống trị thị giác máy tính. Trước năm 2020, hầu hết các nỗ lực sử dụng bộ biến đổi cho các tác vụ thị giác vẫn phụ thuộc rất nhiều vào CNN, bằng cách sử dụng sự chú ý tự thân kết hợp với phép tích chập hoặc bằng cách giữ nguyên cấu trúc chung của CNN trong khi sử dụng sự chú ý tự thân Nguyên nhân của điều này bắt nguồn từ hai điểm yếu chính của các bộ biến đổi. Thứ nhất, độ phức tạp của thao tác chú ý cao. Vì sự chú ý là một phép toán bậc hai, số lượng tham số tăng vọt nhanh chóng khi xử lý dữ liệu trực quan, tức là hình ảnh và thậm chí còn hơn thế nữa với video. Ví dụ, trong trường hợp của ImageNet, việc nhập một hình ảnh duy nhất có $256 \times 256 = 65.536$ pixel vào một lớp chú ý sẽ quá nặng về mặt tính toán. Thứ hai, các bộ biến đổi thiếu các đọ lệch quy nạp. Vì CNN được tạo riêng cho các nhiệm vụ thị giác, nên kiến trúc của chúng bao gồm các độ lệch quy nạp không gian, như tương đương dịch thuật và tính cục bộ. Do đó, các bộ biến đổi phải được đào tạo trước trên một tập dữ liệu lớn đáng kể để đạt được hiệu suất tương tự.

Bộ biến đổi thị giác (ViT)] là phương pháp tiếp cận có hệ thống đầu tiên sử dụng trực tiếp bộ biến đổi cho các nhiệm vụ thị giác bằng cách giải quyết cả hai vấn đề đã đề cập ở trên. Nó loại bỏ hoàn toàn khái niệm tích chập, chỉ sử dụng kiến trúc dựa trên bộ biến đổi. Nhờ đó, nó đạt được trình độ tiên tiến trong nhận dạng hình ảnh trên nhiều tập dữ liệu khác nhau, bao gồm ImageNet và CIFAR-100.

Hình 8.4 minh họa kiến trúc ViT. Ảnh đầu vào trước tiên được chia thành các mảng 16×16, làm phẳng và ánh xạ đến kích thước mong đợi thông qua phép chiếu tuyến tính có thể học được. Vì kích thước ảnh được giảm xuống còn 16×16, nên độ phức tạp của cơ chế chú ý không còn là nút thắt nữa. Sau đó, ViT mã hóa thông tin vị trí và gắn một nhúng có thể học được vào phía trước của chuỗi, tương tự như mã thông báo phân loại của BERT(BERT’s Classification Token: Bidirectional Enchoder Represatations from transformer Classification Token) . Đầu ra của mã thông báo này biểu diễn toàn bộ đầu vào bộ mã hóa thông tin từ mỗi phần của đầu vào. Sau đó, chuỗi này được đưa vào một khối mã hóa, có cùng cấu trúc như trong các bộ biến đổi tiêu chuẩn. Đầu ra của mã thông báo phân loại sau đó được đưa vào một MLP để đưa ra xác suất lớp.



Cls: Classification Token

Hình 8.4. Kiến trúc biến đổi thị giác (ViT). Đầu tiên, ảnh đầu vào được chia thành các mảng (dưới cùng), được chiếu tuyến tính (nhúng), sau đó được nối với các mã thông báo nhúng vị trí. Các mã thông báo kết quả được đưa vào bộ biến đổi, và cuối cùng mã phân loại nhân được được đưa qua MPL để tính toán các xác suất đầu ra

Do thiếu độ lệch quy nạp, khi ViT chỉ được huấn luyện trên các tập dữ liệu cỡ trung bình như ImageNet, điểm số của nó thấp hơn một vài phần trăm so với công nghệ hiện đại. Do đó, mô hình đề xuất trước tiên được huấn luyện trước trên tập dữ liệu JFT-300M và sau đó được tinh chỉnh trên các tập dữ liệu nhỏ hơn, nhờ đó tăng độ chính xác lên 13%.

8.3. CÁC CẢI TIẾN SO VỚI BỘ BIẾN ĐỔI THỊ GIÁC

Trong phần này, chúng ta sẽ trình bày các phương pháp dựa trên bộ biến đổi cải tiến so với bộ biến đổi thị giác ban đầu (mục con 8.2.3) theo hai cách chính. Đầu tiên, chúng ta xét các phương pháp được đào tạo trên các tập dữ liệu nhỏ hơn, không giống như ViT yêu cầu đào tạo trước trên 300 triệu hình ảnh được gắn nhãn. Thứ hai, chúng ta xét các phần mở rộng trên ViT hiệu quả tính toán hơn ViT, vì việc đào tạo ViT có mối tương quan trực tiếp với độ phân giải hình ảnh và số lượng bản vá Number of Patchs).

8.3.1. Hiệu quả dữ liệu

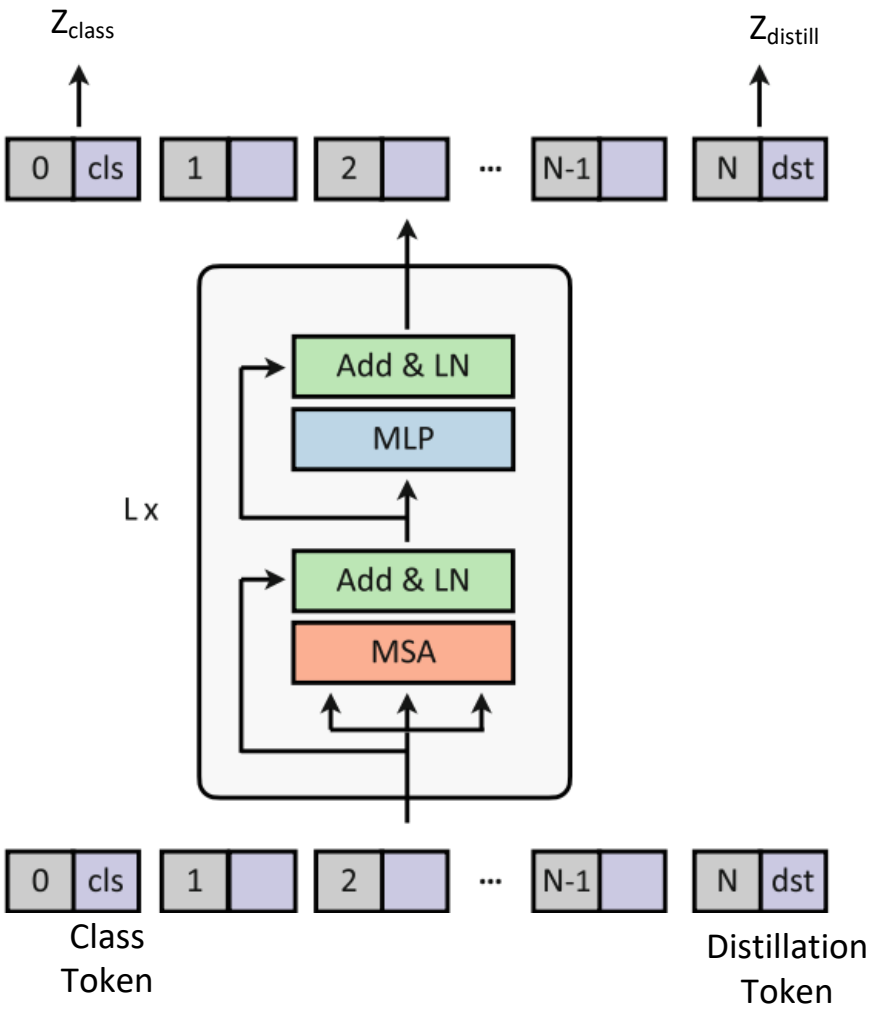
Như đã trình bày trong phần con 8.2.3 , bộ biến đổi thị giác (ViT) được đào tạo trước trên một tập dữ liệu độc quyền khổng lồ (JFT-300M) chứa 300 triệu hình ảnh được gắn nhãn. Nhu cầu này phát sinh với các bộ biến đổi vì chúng ta loại bỏ các độ lệch quy nạp khỏi kiến trúc so với các mạng dựa trên tích chập. Thật vậy, các phép tích chập chứa một vài tương đương dịch thuật (Traslation Equipvalence). ViT không được hưởng lợi từ đặc tính này và do đó phải học các độ lệch như vậy, đòi hỏi nhiều dữ liệu hơn. JFT-300M là một tập dữ liệu khổng lồ và để làm cho ViT hoạt động trong thực tế, cần phải có hiệu quả dữ liệu tốt hơn. Thật vậy, việc thu thập lượng dữ liệu đó rất tốn kém và có thể không khả thi đối với hầu hết các nhiệm vụ.

Các Bộ biến đổi hình ảnh hiệu quả dữ liệu (DeiT)

Công trình đầu tiên nhằm nâng cao hiệu quả dữ liệu là DeiT (Data-Efficient Image Transformer: Bộ biến đổ ảnh hiệu quả dữ liệu). Ý tưởng chính của DeiT là chắt lọc các độ lệch quy nạp từ CNN thành một bộ biến đổi (hình 8.5). DeiT bổ sung một mã thông báo (Token) khác hoạt động tương tự như mã thông báo loại (Classification Token). Khi huấn luyện, các nhãn chân lý cơ bản (Ground Truth Label) được sử dụng để huấn luyện mạng theo đầu ra của mã thông báo loại với mất mát entropy chéo (CE: Cross Entropy)). Tuy nhiên, đối với mạng chưng cất (Distillation Network), các nhãn đầu ra được so sánh với các nhãn được cung cấp từ một mạng giáo viên với mất mát entropy chéo. Mất mát cuối cùng cho nhiệm vụ phân loại N thể loại được định nghĩa như sau:

$$L_{\text{global}}^{\text{hardDistill}} = \frac{1}{2} L_{\text{CE}} \left(\Psi(\boldsymbol{Z}_{\text{class}}), \boldsymbol{y} \right) + L_{\text{CE}} \left(\Psi(\boldsymbol{Z}_{\text{distill}}), \boldsymbol{y}_{\text{T}} \right)$$
$$L_{\text{CE}} \left(\hat{\text{y}}, \text{y} \right) = -\frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \boldsymbol{y}_{\text{i}} \log \hat{\boldsymbol{y}}_{\text{i}} + (1 - \boldsymbol{y}_{\text{i}}) \log(1 - \hat{\boldsymbol{y}}_{\text{i}}) \quad (8.9)$$

Với Ψ là hàm softmax, $\boldsymbol{Z}_{\text{class}}$ là đầu ra của token phân loại (mã thông báo phân loại), $\boldsymbol{Z}_{\text{distill}}$ là đầu ra của token chưng cất (mã thông báo chưng cất) và y_{T} là mã dự đoán nhãn giáo viên.



Hình 8.5. Kiến trúc DeiT. Kiến trúc này có một mã thông báo bổ sung, mã thông báo chưng cất (Distillation Token). Mã thông báo này được sử dụng tương tự như mã thông báo loại (Class Token)

Mạng lưới giáo viên là mạng neuron tích chập (CNN). Ý tưởng chính là đầu chưng cất sẽ cung cấp các độ lệch quy nạp cần thiết để cải thiện hiệu quả dữ liệu của kiến trúc. Bằng cách này, DeiT đạt được hiệu suất đáng kể trên tập dữ liệu ImageNet, bằng cách huấn luyện "chỉ" trên ImageNet-1K, chứa 1,3 triệu hình ảnh.

Convit

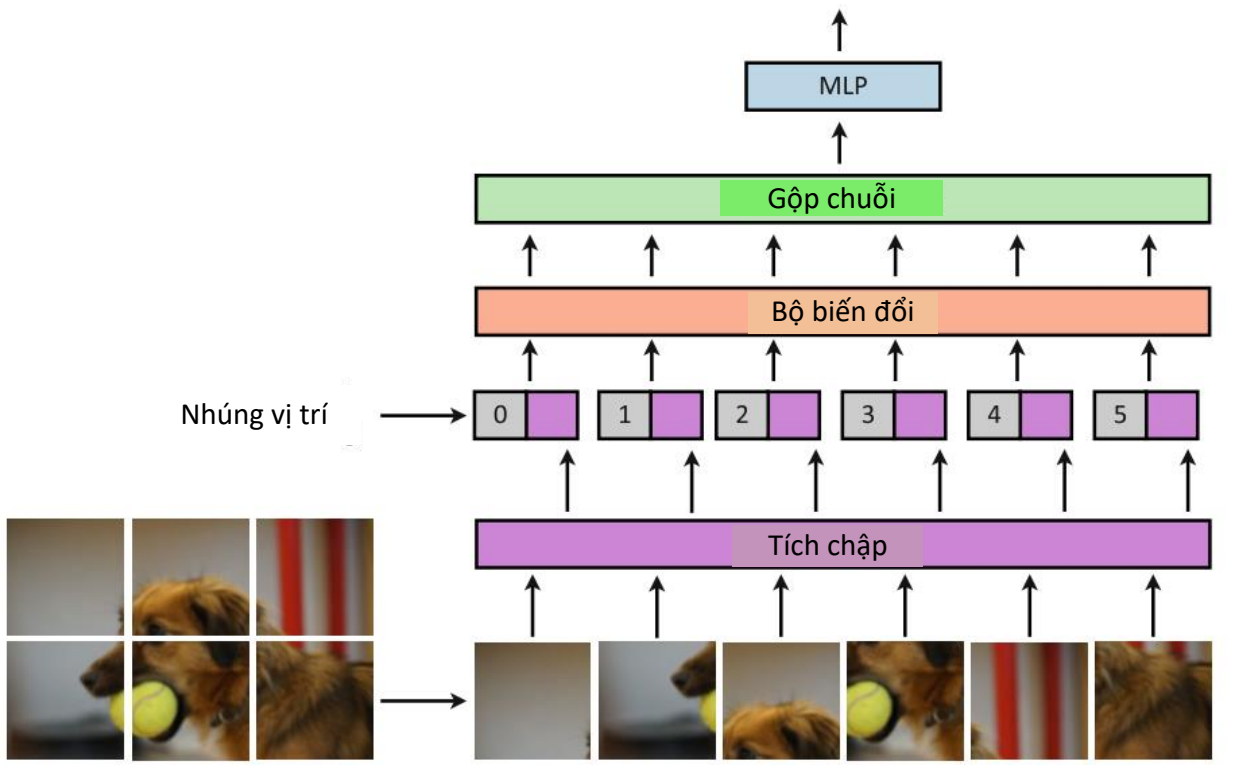
Nhược điểm chính của DeiT là nó yêu cầu một CNN được huấn luyện trước, điều này không lý tưởng, và sẽ thuận tiện hơn nếu không có yêu cầu này. CNN có ràng buộc độ lệch quy nạp cứng, có thể là một hạn chế lớn. Thực tế, nếu có đủ dữ liệu, việc học độ lệch từ dữ liệu có thể mang lại kết quả biểu diễn tốt hơn.

Convit vượt qua vấn đề này bằng cách đưa độ lệch quy nạp của CNN vào bộ biến đổi một cách mềm dẻo. Cụ thể, nếu độ lệch quy nạp hạn chế quá trình huấn luyện, bộ biến đổi có thể loại bỏ nó. Ý tưởng chính là đưa độ lệch quy nạp vào quá trình khởi tạo ViT. Do đó, trước khi bắt đầu huấn luyện, ViT tương đương với một CNN. Sau đó, mạng có thể dần dần học các độ lệch cần thiết và tách khỏi quá trình khởi tạo CNN.

Bộ biến đổi tích chập nhỏ gọn (Compact Convolutional Transformer)

DeiT và Convit đã đạt được hiệu quả dữ liệu thành công ở quy mô ImageNet. Tuy nhiên, ImageNet là một tập dữ liệu lớn với 1,3 triệu hình ảnh, trong khi hầu hết các tập dữ liệu khác đều nhỏ hơn đáng kể.

Để đạt được hiệu quả dữ liệu cao hơn, bộ biến đổi tích chập nhỏ gọn sử dụng một thao tác CNN để trích xuất các bản vá và sau đó sử dụng các bản vá này trong mạng biến đổi (hình 8.6). Bộ biến đổi tích chập nhỏ gọn đi kèm với một số sửa đổi dẫn đến những cải thiện lớn. Đầu tiên, bằng cách mã hóa các bản vá phức tạp hơn, hệ thống dựa vào các độ lệch cảm ứng tích chập ở các thang đo thấp hơn và sau đó sử dụng mạng biến đổi để loại bỏ ràng buộc cục bộ của CNN. Thứ hai, các tác giả chỉ ra rằng việc loại bỏ mã thông báo "loại" dẫn đến hiệu quả cao hơn. Cụ thể, thay vì mã thông báo loại, bộ biến đổi tích chập nhỏ gọn gộp tất cả các mã thông báo bản vá lại với nhau và phân loại trên mã thông báo gộp này. Hai sửa đổi này cho phép sử dụng các bộ biến đổi nhỏ hơn đồng thời cải thiện cả hiệu quả dữ liệu và hiệu quả tính toán. Do đó, những cải tiến này cho phép bộ biến đổi tích chập nhỏ gọn được đào tạo thành công trên các tập dữ liệu nhỏ hơn, chẳng hạn như CIFAR hoặc MNIST.



Hình 8.6 . Bộ biến đổi tích chập nhỏ gọn. Kiến trúc này có tính năng trích xuất bản vá dựa trên tích chập để tận dụng mạng biến đổi nhỏ hơn, dẫn đến hiệu quả dữ liệu cao hơn

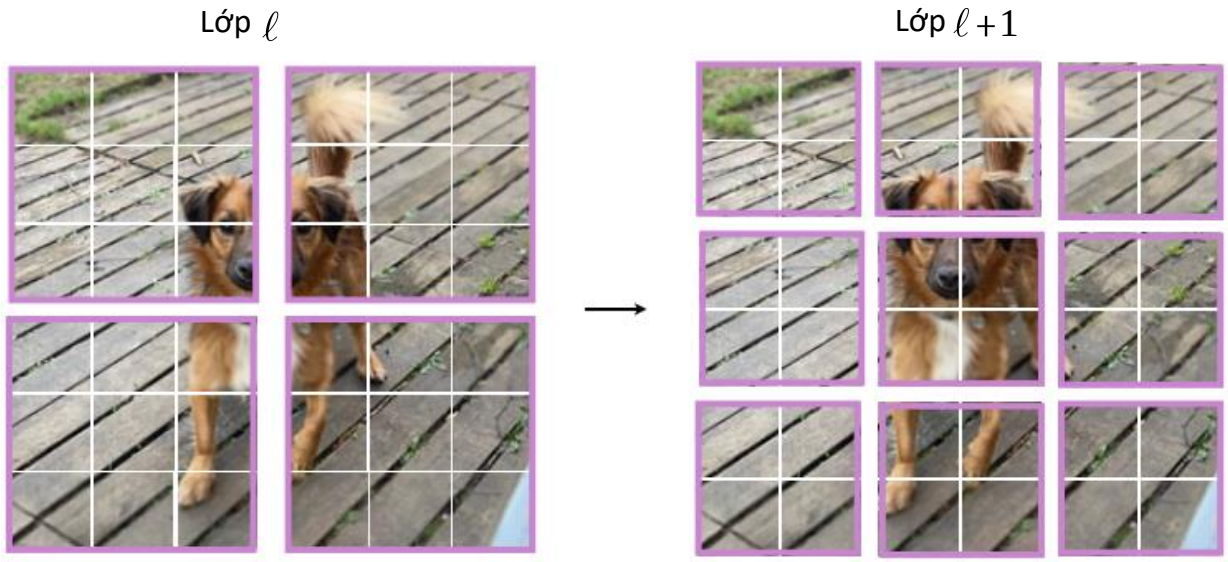
8.3.2. Hiệu quả tính toán

Kiến trúc bộ biến đổi thị giác (phần con 8. 2.3) bị ảnh hưởng bởi $O(n^2)$ độ phức tạp liên quan đến số lượng token. Khi xem xét các hình ảnh có độ phân giải nhỏ hoặc kích thước mảng lớn, đây không phải là một hạn chế; ví dụ, đối với một hình ảnh có độ phân giải 224 × 224 với các mảng 16 × 16, con số này tương đương với 196 token. Tuy nhiên, khi cần xử lý các hình ảnh lớn hơn (ví dụ: hình ảnh 3D trong hình ảnh y tế) hoặc khi xem xét các mảng nhỏ hơn, việc sử dụng và huấn luyện các mô hình như vậy trở nên khó khăn . Ví dụ, trong các nhiệm vụ như phân đoạn hoặc tạo ảnh, cần có các biểu diễn chi tiết hơn các mảng 16 × 16; do đó, việc giải quyết vấn đề này là rất quan trọng để cho phép nhiều ứng dụng hơn của bộ chuyển đổi thị giác.

Bộ biến đổi Swin

Một ý tưởng giúp bộ biến đổi hiệu quả tính toán hơn là bộ biến đổi Swin. Thay vì chú ý (Attend) mọi bản vá trong ảnh, bộ biến đổi Swin đề xuất thêm một ràng buộc cục bộ. Cụ

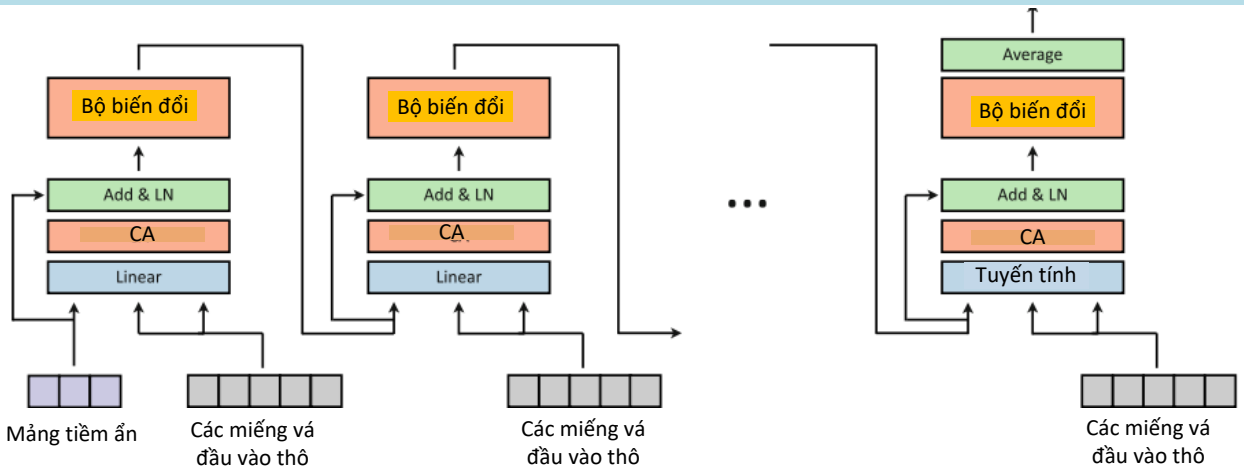
thể, các bản vá chỉ có thể chú ý các bản vá khác bị giới hạn trong cửa sổ lân cận K. Điều này khôi phục lại độ lệch quy nạp cục bộ của CNN. Để cho phép giao tiếp giữa các mảng trong toàn mạng, bộ biến đổi Swin chuyển các cửa sổ chú ý từ thao tác này sang thao tác khác (hình 8.7). Do đó, bộ biến đổi Swin là bậc hai đối với kích thước của cửa sổ K nhưng tuyến tính đối với số lượng mã thông báo n với độ phức tạp $O(nK^2)$. Tuy nhiên, trên thực tế, K nhỏ và điều này giải quyết được vấn đề phức tạp bậc hai của sự chú ý.



Hình 8.7 . Hoạt động dịch chuyển trong bộ biến đổi Swin. Giữa mỗi hoạt động chú ý, cửa sổ chú ý được dịch chuyển để mỗi bản vá có thể giao tiếp với các bản vá khác nhau so với trước đó. Điều này cho phép mạng thu thập được nhiều kiến thức toàn cục hơn cùng với độ sâu của mạng

Bộ nhận biết (Perceiver)

Một ý tưởng khác cho các bộ biến đổi hình ảnh hiệu quả tính toán hơn là thay đổi kiến trúc một cách triệt để hơn. Nếu thay vì sử dụng tự chú ý tự thân, mô hình sử dụng chú ý chéo (CA: Crooss Attention) , vấn đề độ phức tạp bậc hai liên quan đến số lượng token có thể được giải quyết. Thật vậy, việc tính toán sự chú ý chéo giữa hai tập hợp có độ dài tương ứng là m và n , có độ phức tạp O(mn). Ý tưởng này được giới thiệu trong bộ nhận thức. Ý tưởng này là có một tập hợp nhỏ hơn các biến tiềm ẩn sẽ được sử dụng như là các truy vấn và sẽ truy xuất thông tin trong tập mã thông báo hình ảnh (Image Token Set), (Hình 8.8). Vì điều này giải quyết được vấn đề phức tạp bậc hai, nó cũng loại bỏ nhu cầu sử dụng các bản vá; do đó, trong trường hợp của bộ biến đổi, mỗi pixel được ánh xạ tới một mã thông báo duy nhất.



Add&LN (Add and Normaliation: cộng và chuẩn hóa lớp;
CA (Cross Attention: chú ý chéo

Hình 8.8. Kiến trúc bộ nhận thức. Một tập hợp các mã thông báo tiềm ẩn (Latent Token) thu thập thông tin từ hình ảnh thông qua sự chú ý chéo. Sự chú ý tự thân được thực hiện giữa các mã thông báo để tinh chỉnh biểu diễn đã học. Các thao tác này tuyến tính theo số lượng

8.4. BỘ BIẾN ĐỔI THỊ GIÁC CHO CÁC NHIỆM VỤ KHÁC VỚI PHÂN LOẠI

Các phần 8.1-8.3 giới thiệu bộ biến đổi thị giác cho một ứng dụng chính: phân loại. Tuy nhiên, bộ biến đổi có thể được sử dụng cho nhiều nhiệm vụ khác ngoài phân loại .

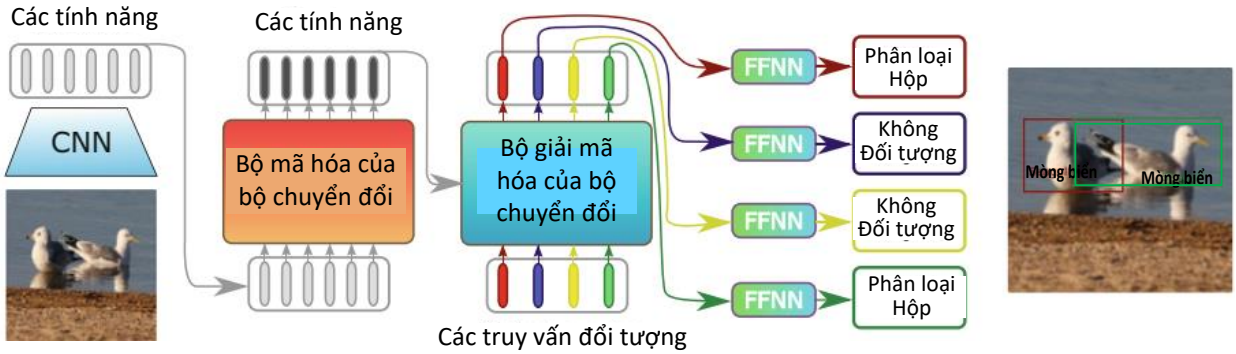
Trong phần này, chúng ta sẽ xét một số nhiệm vụ thị giác cơ bản mà bộ biến đổi có tác động lớn: phát hiện đối tượng trong hình ảnh (phần con 8.4.1), phân đoạn hình ảnh (phần con 8. 4.2), đào tạo bộ biến đổi thị giác không có nhãn (phần con 8.4.3) và tạo hình ảnh bằng mạng đối nghịch tạo sinh (GAN) (phần con 8.4.4).

8.4.1. Phát hiện đối tượng bằng các bộ biến đổi (Object Detection With Transformers)

Phát hiện là một trong những nhiệm vụ ban đầu đã được cải thiện nhờ bộ biến đổi. Phát hiện là một bài toán kết hợp giữa nhận dạng và định vị; điều này có nghĩa là một hệ thống phát hiện thành công phải vừa nhận biết được sự hiện diện của vật thể trong ảnh vừa định vị vật thể đó trong không gian. Carion và cộng sự đưa ra phương pháp đầu tiên sử dụng bộ biến đổi để phát hiện.

Bộ biến đổi phát hiện (DETR: Detection Transformer)

Đầu tiên, DETR trích xuất các biểu diễn thị giác bằng mạng tích chập (hình 8.9). Sau đó, các mã hóa được xử lý bởi một mạng biến đổi. Cuối cùng, các mã thông báo (Token) đã xử lý được cung cấp cho một bộ giải mã biến đổi. Bộ giải mã sử dụng chú ý chéo giữa một tập hợp các mã thông báo đã học và các mã thông báo hình ảnh được mã hóa bởi bộ mã hóa và xuất ra một tập hợp các mã thông báo. Mỗi mã thông báo đầu ra sau đó được truyền qua một mạng truyền thẳng để dự đoán xem một đối tượng có hiện diện trong hình ảnh hay không; nếu thực sự có, mạng cũng dự đoán loại và vị trí không gian của đối tượng, tức là tọa độ trong hình ảnh.

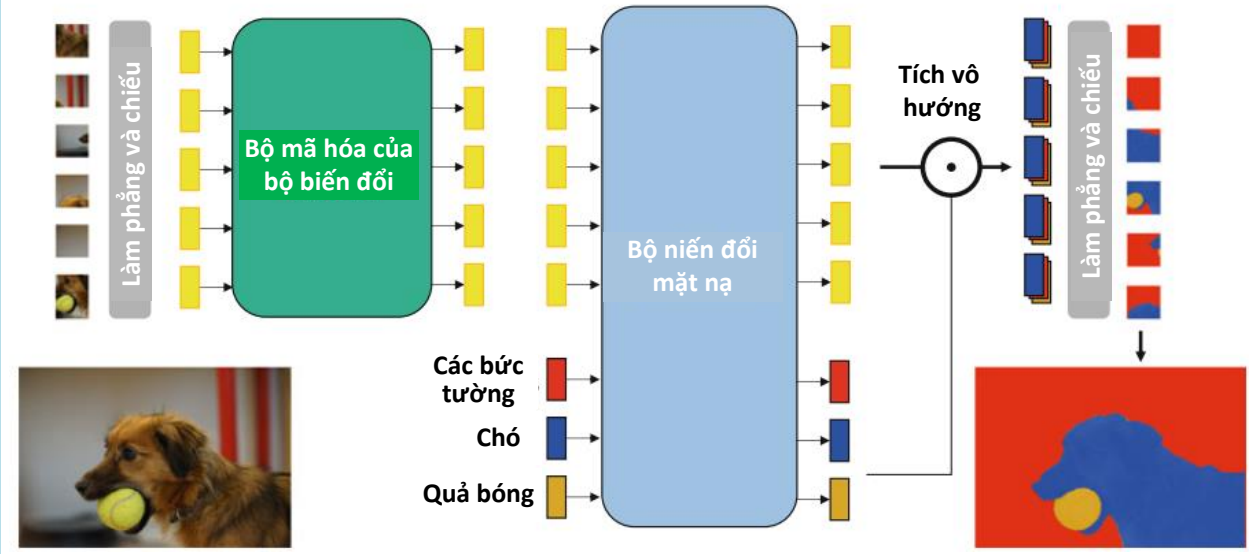


FFNN (Feed-Forward Neuron Network): mạng neuron truyền thẳng

Hình 8.9 . Kiến trúc DETR. Nó tinh chỉnh biểu diễn trực quan của CNN để trích xuất vị trí và loại đối tượng

8.4.2. Phân đoạn hình ảnh bằng các bộ biến đổi (Image Segmentation With Transformers)

Mục tiêu của phân đoạn ảnh là gán nhãn của đối tượng mà nó thuộc về cho mỗi điểm ảnh của ảnh. Bộ phân đoạn (Segmenter) là một phương pháp ViT thuần túy, tập trung vào phân đoạn ảnh. Ý tưởng là trước tiên sử dụng ViT để mã hóa ảnh. Sau đó, bộ phân đoạn học một mã thông báo cho mỗi nhãn ngữ nghĩa. Các mã thông báo bản vá đã mã hóa và các mã thông báo ngữ nghĩa sau đó được đưa vào bộ biến đổi thứ hai. Cuối cùng, bằng cách tính tích vô hướng giữa các mã thông báo ngữ nghĩa và các mã thông báo ảnh, mạng gán một nhãn cho mỗi bản vá. Hình 8.10 minh họa điều này.



Hình 8.10 . Kiến trúc phân đoạn. Đây là một phương pháp hoàn toàn dựa trên ViT để thực hiện phân đoạn ngữ nghĩa

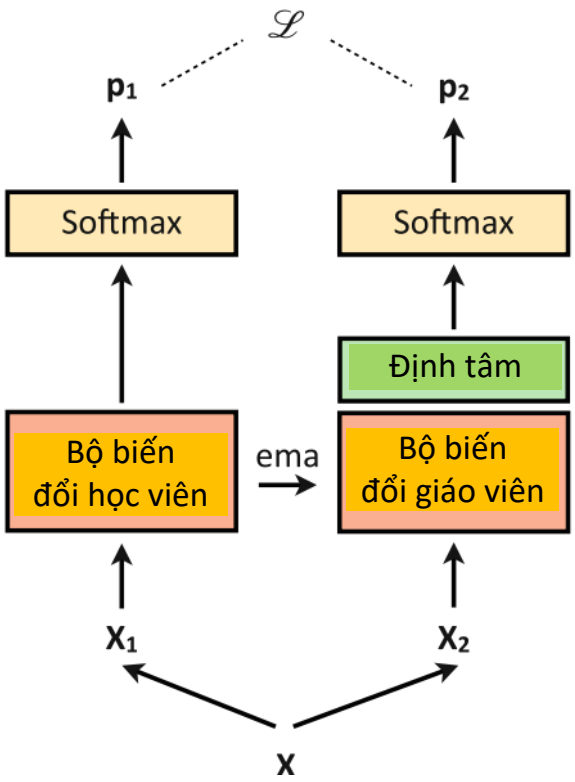
8.4.3. Đào tạo bộ biến đổi không có nhãn (Training Transformer Without Label)

Các bộ biến đổi hình ảnh ban đầu được huấn luyện cho các nhiệm vụ phân loại. Tuy nhiên, các nhiệm vụ này đòi hỏi phải truy cập vào một lượng lớn dữ liệu được gắn nhãn, vốn rất khó để có được (như đã xét trong phân con 8.3.1). Phần 8.3.1 và 8.3.2 trình bày các cách để huấn luyện ViT hiệu quả hơn. Tuy nhiên, việc có thể huấn luyện loại mạng này với dữ liệu "rẻ hơn" cũng rất thú vị. Do đó, mục tiêu của phần này là giới thiệu phương pháp học không giám sát với các bộ biến đổi, tức là huấn luyện các bộ biến đổi không có bất kỳ nhãn nào.

Tự chưng cất không cần nhãn (DINO: Self-DIstillation with NO labels)

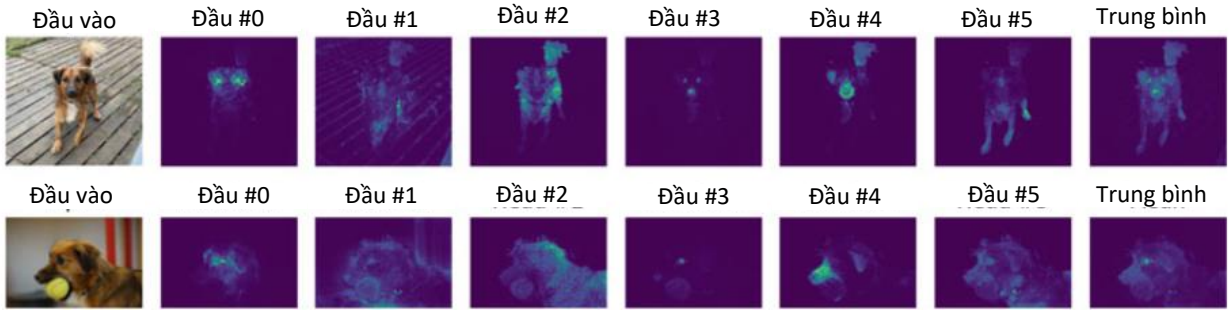
DINO là một trong những công trình đầu tiên đào tạo ViT với phương pháp học tự giám sát (Self-supervised Learning), (hình 8.11). Ý tưởng chính là có hai mô hình ViT theo mô hình giáo viên-học sinh: mô hình đầu tiên được cập nhật thông qua phương pháp giảm dần độ dốc (Gradient Descent) và mô hình thứ hai là trung bình chuyển động theo hàm mũ (Expotential Moving Paradigm) của mô hình đầu tiên. Sau đó, toàn bộ mạng DINO hai luồng được đào tạo bằng cách sử dụng hai phép tăng cường (Augmentation) của cùng một hình ảnh, mỗi phép tăng cường được truyền đến một trong hai mạng. Mục tiêu của quá trình đào tạo là khớp đầu ra giữa hai mạng, tức là bất kể phép tăng cường nào trong dữ liệu đầu vào, cả hai mạng đều phải tạo ra cùng một kết quả. Phát hiện chính của DINO là ViT có khả năng học hiểu ngữ nghĩa của hình ảnh, vì ma trận chú ý hiển thị

một số thông tin ngữ nghĩa. Hình 8.12 trực quan hóa ma trận chú ý của các đầu ViT khác nhau được đào tạo bằng DINO.



Ema (Expotential Moving Average): trung bình chuyển động theo hàm mũ
Centering: định tâm

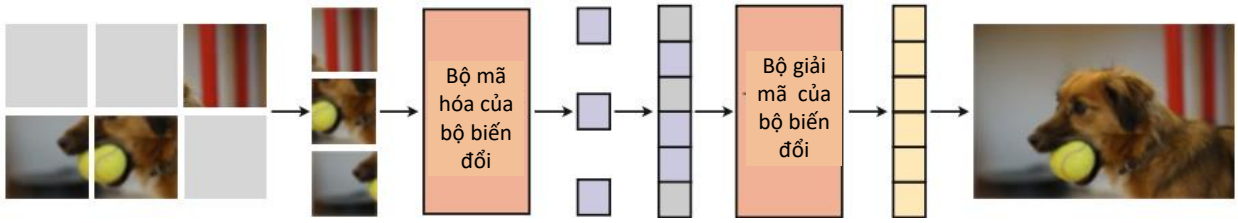
Hình 8.11. Quy trình huấn luyện DINO. Nó bao gồm việc khớp các đầu ra giữa hai mạng (p_1 và p_2) có hai phép tăng cường khác nhau (X_1 và X_2) của cùng một hình ảnh với đầu vào (X). Các tham số của mô hình giáo viên được cập nhật theo hàm mũ (ema) của các tham số học viên



Hình 8.12. Mẫu DINO. Hình ảnh hóa ma trận chú ý của các đầu ViT được huấn luyện bằng DINO. ViT khám phá cấu trúc ngữ nghĩa của hình ảnh theo cách không giám sát

Bộ mã hóa tự động được che dấu (MAE: Masked Autoencoder)

Một cách khác để huấn luyện ViT mà không cần giám sát là sử dụng kiến trúc mã hóa tự động. Bộ mã hóa tự động có mặt nạ (MAE) thực hiện việc che dấu ngẫu nhiên mã thông báo đầu vào và giao nhiệm vụ tái tạo hình ảnh gốc cho bộ giải mã. Bộ mã hóa học một biểu diễn hoạt động tốt trong một nhiệm vụ hạ nguồn (Downstream Task) nhất định. Điều này được minh họa trong hình 8.13. Một trong những quan sát chính của công trình MAE là bộ giải mã không cần phải quá tốt để bộ mã hóa đạt được hiệu suất tốt: chỉ bằng cách sử dụng một bộ giải mã nhỏ, MAE đã huấn luyện thành công ViT theo kiểu mã hóa tự động.



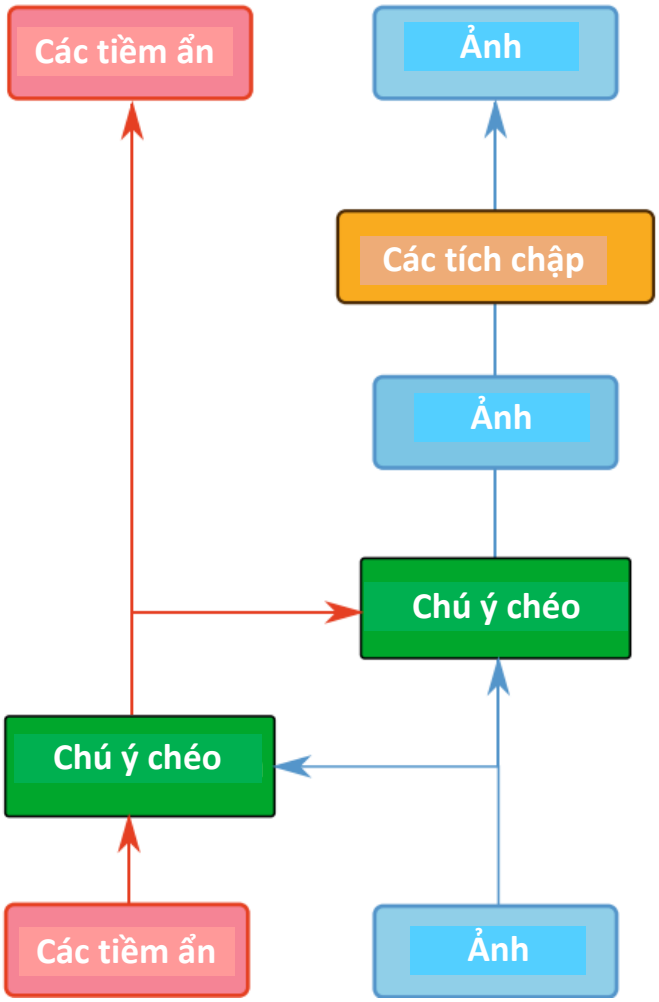
Hình 8.13. Quy trình đào tạo MAE. Sau khi che giấu một số mã thông báo của hình ảnh, các mã thông báo còn lại được đưa vào bộ mã hóa. Sau đó, bộ giải mã sẽ cố gắng tái tạo hình ảnh gốc từ biểu diễn này

8.4.4. Tạo hình ảnh bằng bằng các bộ biến đổi và chú ý (Image Generation with Transformers aqnd Attention)

Các bộ chuyển đổi thị giác và chú ý cũng đã giúp phát triển những ý tưởng mới và tạo ra kiến trúc mới cho các mô hình tạo sinh và đặc biệt là cho các mạng đối nghịch tạo sinh (GAN).

GANsformers

GANsformer là công trình tiêu biểu nhất của GAN có bộ biến đổi, vì chúng là một kiến trúc lai sử dụng cả chú ý và CNN. Kiến trúc GANsformer được minh họa trong hình 8.14. Đầu tiên, mô hình chia vectơ tiềm ẩn của GAN thành nhiều token. Sau đó, một cơ chế chú ý chéo được sử dụng để cải thiện các bản đồ tính năng được tạo ra, đồng thời, kiến trúc GANsformer lấy thông tin từ bản đồ tính năng đã tạo ra để làm giàu các token. Cơ chế này cho phép GAN có kiến thức ngữ nghĩa tốt hơn và phong phú hơn, điều này được chứng minh là hữu ích cho việc tạo ra các hình ảnh đa phương thức.



Hình 8.14. Kiến trúc GANsformer. Một tập hợp các tiềm ẩn góp phần đưa thông tin vào bản đồ tính năng CNN

StyleSwin

Một cách tiếp cận khác cho mô hình tạo sinh là sử dụng hoàn toàn kiến trúc ViT như StyleSwin. StyleSwin là một GAN tận dụng loại chú ý tương tự như bộ biến đổi Swin. Điều này cho phép tạo ra hình ảnh có độ nét cao mà không cần phải giải quyết vấn đề chi phí bậc hai.

8.5. BỘ BIẾN ĐỔI THỊ GIÁC CHO CÁC LĨNH VỰC KHÁC (VISION TRANSFORMER FOR OTHER DOMAINS)

Trong phần này, ta sẽ xét các ứng dụng của bộ biến đổi hình ảnh cho các lĩnh vực khác. Đầu tiên, phần này mô tả các bộ biến đổi đổi đa phương thức (Multimodial Transformer) hoạt động với thị giác và ngôn ngữ (phần con 8.5.1), sau đó phần này mô tả sự chú ý ở cấp độ video và các bộ chuyển đổi video (các phần con 8.5.2 và 8.5.3) và cuối cùng phần này xét các bộ biến đổi video đa phương thức hoạt động với thị giác, ngôn ngữ và âm thanh (phần con 8.5.4).

8.5.1. Bộ biến đổi đa phương thức: thị giác và Ngôn ngữ (Multimodial Transformer: Vision and Language)

Vì các bộ biến đổi đã đạt được thành công to lớn trong cả xử lý ngôn ngữ tự nhiên và thị giác máy tính, việc sử dụng chúng trong các nhiệm vụ thị giác-ngôn ngữ cũng rất đáng quan tâm. Phần này sẽ mô tả một số phương pháp đa phương thức tiêu biểu cho thị giác và ngôn ngữ: ViLBERT (phần con 8.5.1.1), DALL-E (phần con 8.5.1.3) và CLIP (phần con 7.5.1.2).

8.5.1.1. ViLBERT

BERT thị giác và ngôn ngữ (VilBERT: Vision-Language BERT) là một ví dụ về kiến trúc hợp nhất hai phương thức. Nó bao gồm hai luồng song song, mỗi luồng làm việc với một phương thức. Luồng thị giác trích xuất các hộp giới hạn từ hình ảnh thông qua mạng phát hiện đối tượng, bằng cách mã hóa vị trí của chúng. Luồng ngôn ngữ nhúng các vectơ từ (Word Vector) và trích xuất các vectơ tính năng bằng cách sử dụng khối mã hóa của bộ biến đổi cơ bản (Hình 8.3 bên trái). Hai vectơ đặc trưng kết quả này sau đó được hợp nhất với nhau bằng một loại chú ý chéo (phần con 8.1.3.2). Điều này tuân theo kiến trúc tiêu chuẩn của khối mã hóa của bộ biến đổi, trong đó các khóa và giá trị của một phương thức được truyền vào khối MCA của phương thức kia. Đầu ra của loại chú ý chéo được truyền vào một khối mã hóa của bộ biến đổi khác và hai loại này được xếp chồng nhiều lần.

Luồng ngôn ngữ được khởi tạo bằng BERT được đào tạo trên Book Corpus và Wikipedia, trong khi luồng hình ảnh được khởi tạo bằng Faster R-CNN. Ngoài việc đào tạo trước cho từng luồng, toàn bộ kiến trúc được đào tạo trước trên tập dữ liệu các mô tả khái niệm (Conceptual Captions Dataset) trên hai nhiệm vụ tiền đề (Pretex Task).

ViLBERT đã được chứng minh là mạnh mẽ cho nhiều nhiệm vụ đa phương thức. Trong bài báo gốc, ViLBERT đã được tinh chỉnh cho nhiều tác nhiệm khác nhau, bao gồm trả lời câu hỏi trực quan, suy luận trực quan, biểu thức tham chiếu và truy xuất hình ảnh dựa trên mô tả.

8.5.1.2. Kết nối văn bản và hình ảnh (CLIP)

Kết nối văn bản và hình ảnh (CLIP: Connecting Text and Images) được thiết kế để giải quyết hai vấn đề chính của các mô hình học sâu: bộ dữ liệu tốn kém và tính không linh hoạt. Trong khi hầu hết các mô hình học sâu được đào tạo trên các bộ dữ liệu có nhãn, CLIP được đào tạo trên 400 triệu cặp văn bản-hình ảnh được lấy từ Internet. Điều này làm giảm công việc phải dán nhãn thủ công hàng triệu hình ảnh cần thiết để đào tạo các mô hình học sâu mạnh mẽ. Khi các mô hình được đào tạo trên một bộ dữ liệu cụ thể, chúng cũng có xu hướng khó mở rộng sang các ứng dụng khác. Ví dụ, độ chính xác của một mô hình được đào tạo trên ImageNet thường bị giới hạn trong bộ dữ liệu riêng của nó và không thể áp dụng cho các vấn đề trong thế giới thực. Để tối ưu hóa đào tạo, các mô hình CLIP học cách thực hiện nhiều nhiệm vụ khác nhau trong quá trình đào tạo trước và nhiệm vụ này cho phép chuyển giao không cần bắn (zero-shot Transfer) đến nhiều bộ dữ liệu hiện có. Mặc dù vẫn còn một số cải tiến tiềm năng, nhưng cách tiếp cận này có tính cạnh tranh với các mô hình có giám sát được đào tạo trên các bộ dữ liệu cụ thể.

Kiến trúc và Đào tạo CLIP (CLIP Architecture and Training)

CLIP được sử dụng để đo lường độ tương đồng giữa văn bản đầu vào và hình ảnh được tạo ra từ một vectơ tiềm ẩn. Cốt lõi của phương pháp này là ý tưởng học nhận thức (Learning Perception) từ sự giám sát chứa trong ngôn ngữ tự nhiên. Các phương pháp hoạt động trên ngôn ngữ tự nhiên có thể học thụ động từ sự giám sát chứa trong khối lượng lớn văn bản trên Internet.

Với một lô N cặp (hình ảnh, văn bản), CLIP được huấn luyện để dự đoán cặp nào trong số $N \times N$ cặp (hình ảnh, văn bản) có thể có trong lô thực sự xảy ra. Để làm được điều này, CLIP học một không gian nhúng đa phương thức bằng cách huấn luyện chung một bộ mã hóa hình ảnh và một bộ mã hóa văn bản để tối đa hóa độ tương đồng cosin của các nhúng hình ảnh và văn bản của N cặp thực trong lô, đồng thời giảm thiểu độ tương đồng cosin của các nhúng của $N^2 - N$ cặp không chính xác. Mất mát entropy chéo đối xứng trên các điểm tương đồng này được tối ưu hóa.

Hai kiến trúc khác nhau đã được xem xét cho bộ mã hóa hình ảnh. Đối với kiến trúc đầu tiên, ResNet-50 được sử dụng làm kiến trúc cơ sở cho bộ mã hóa hình ảnh do được áp dụng rộng rãi và hiệu suất đã được chứng minh. Một số sửa đổi đã được thực hiện đối với phiên bản gốc của ResNet. Đối với kiến trúc thứ hai, ViT được sử dụng với một số sửa đổi nhỏ: thứ nhất, thêm một lớp chuẩn hóa bổ sung vào các bản vá kết hợp và nhúng vị trí trước bộ biến đổi và thứ hai, sử dụng một sơ đồ khởi tạo hơi khác một chút.

Bộ mã hóa văn bản là một bộ chuyển đổi chuẩn (phần con 8.2.1) với các sửa đổi về kiến trúc. Với kích thước cơ sở, CLIP sử dụng mô hình 63M tham số 12 lớp rộng 512 với tám đầu chú ý. Bộ chuyển đổi hoạt động trên biểu diễn mã hóa cặp byte viết thường (Lowcased BPE: Byte Pair Encoding) của văn bản với kích thước từ vựng là 49.152. Độ dài chuỗi tối đa được giới hạn ở 76. Chuỗi văn bản được đặt trong ngoặc với các mã

thông báo [SOS] và [EOS] (SOS: Start of Sequence: Bắt đầu chuỗi; EOS: End of Sequence: kết thúc chuỗi) , và các kích hoạt của lớp cao nhất của bộ chuyển đổi tại mã thông báo [EOS] được coi là biểu diễn tính năng của văn bản được chuẩn hóa theo lớp và sau đó được chiếu tuyến tính vào không gian nhúng đa phương thức.

8.5.1.3. DALL-E và DALL-E 2

DALL-E là một ví dụ khác về ứng dụng của các bộ biến đổi thị giác. Nó tạo ra hình ảnh từ một lời nhắc ngôn ngữ tự nhiên- ví dụ "một chiếc ghế bành hình quả bơ" và "một chú chim cánh cụt làm từ dưa hấu". Nó sử dụng mô hình chỉ giải mã, tương tự như GPT-3 . DALL-E sử dụng 12 tỷ tham số và được đào tạo trước trên mô tả khái niệm (Conceptual Caption) với hơn 3,3 triệu cặp văn bản-hình ảnh. DALL-E 2 là phiên bản nâng cấp của DALL-E, dựa trên các mô hình khuếch tán và CLIP (phần con 8.5.1.2), và cho phép hiệu suất tốt hơn với hình ảnh được tạo ra chân thực và chính xác hơn. Ngoài việc tạo ra kết quả chân thực hơn với độ phân giải tốt hơn DALL-E, DALL-E 2 còn có thể chỉnh sửa đầu ra. Thật vậy, với DALL-E 2, người ta có thể thêm hoặc bớt một phần tử một cách chân thực trong đầu ra và cũng có thể tạo ra các biến thể khác nhau của cùng một đầu ra. Hai mô hình này chứng minh rõ ràng bản chất mạnh mẽ và khả năng mở rộng của các bộ biến đổi có khả năng xử lý hiệu quả lượng dữ liệu trên quy mô web.

8.5.1.4. Chim hồng hạc (Flamingo)

Flamingo là một mô hình ngôn ngữ thị giác (VLM: Visual Language Model) xử lý nhiều nhiệm vụ đa phương thức đa phương thức dựa trên phương pháp học ít ảnh. Đây là một phiên bản cải tiến của các mô hình ngôn ngữ lớn (LLM: Large Language Model)) xử lý một phương thức thị giác bổ sung với 80B tham số.

Flamingo bao gồm ba thành phần chính: bộ mã hóa thị giác (Visual Encoder), bộ lấy mẫu lại của bộ nhận thức (Perceiver Resampler) và mô hình ngôn ngữ (Language Model). Đầu tiên, để mã hóa hình ảnh hoặc video, bộ mã hóa tích chập thị giác được đào tạo trước theo cách tương phản, sử dụng các cặp hình ảnh và văn bản. Sau đó, lấy cảm hứng từ kiến trúc bộ nhận biết (được trình bày chi tiết trong phần con 8. 1.3.2), bộ lấy mẫu lại của bộ nhận thức lấy một số lượng khả biến các tính năng thị giác được mã hóa và đưa ra một mã tiềm ẩn (Latent Code) có độ dài cố định. Cuối cùng, mã tiềm ẩn trực quan này điều kiện hóa mô hình ngôn ngữ bằng cách truy vấn các mã thông báo ngôn ngữ thông qua các khối chú ý chéo. Các khối chú ý chéo này được đan xen với các khối chế độ ngôn ngữ được đóng băng và đào tạo trước.

Toàn bộ mô hình được đào tạo bằng ba loại tập dữ liệu khác nhau mà không có chú thích (văn bản có nội dung hình ảnh từ các trang web, cặp văn bản và hình ảnh và cặp văn bản và video). Sau khi mô hình được đào tạo, nó được tinh chỉnh bằng các kỹ thuật học tập ít mẫu để giải quyết các nhiệm vụ cụ thể.

8.5.2. Chú ý video (Video Attention)

Hiểu video là một vấn đề tồn tại từ lâu, và mặc dù công nghệ thị giác máy tính đã có những tiến bộ đáng kinh ngạc, việc tìm kiếm biểu diễn video tốt nhất vẫn là một lĩnh vực nghiên cứu đang được quan tâm. Video đòi hỏi phải sử dụng phương pháp xử lý không gian-thời gian hiệu quả của các luồng RGB và thời gian để ghi lại các tương tác tầm xa, đồng thời tập trung vào các phần video quan trọng với tài nguyên tính toán tối thiểu.

Thông thường, việc hiểu video được hưởng lợi từ thị giác máy tính 2D, bằng cách áp dụng các phương pháp xử lý hình ảnh 2D vào các phương pháp không gian thời gian 3D. Và thông qua bộ biến đổi thị giác video (ViViT: Video Vision Transformer), lịch sử lặp lại. Thật vậy, với sự ra đời của các bộ biến đổi và những tiến bộ gần đây trong phân loại hình ảnh, các bộ biến đổi video xuất hiện như những người kế thừa hợp lý của CNN.

Tuy nhiên, ngoài việc xử lý video tốn kém về mặt tính toán, bộ biến đổi còn đòi hỏi rất nhiều tài nguyên tính toán. Do đó, việc phát triển các cơ chế chú ý không gian-thời gian hiệu quả là điều cần thiết.

Trong phần này, trước tiên nguyên lý chung của bộ chuyển đổi video (phần con 8.5.2.1) được xét, sau đó, chi tiết ba cơ chế chú ý khác nhau được sử dụng để biểu diễn video (phần con 8.5.2.2 , 8.5.2.3 và 8.5.2.4) được xet.

8.5.2.1. Nguyên tắc chung

Nói chung, đầu vào của bộ biến đổi video là các clip video RGB $\boldsymbol{X} \in \mathbb{R}^{F\times H\times \mathbf{W}\times 3}$, với khung F có kích thước H ×W.
Để bắt đầu, bộ biến đổi video chia đoạn video đầu vào $\boldsymbol{X}$ thành các mã thông báo ST (ST Token) $\boldsymbol{x}_i \in \mathbb{R}^K$, trong đó S và T lần lượt là số lượng mã thông báo theo chiều không gian và thời gian và K là kích thước của một mã thông báo.

Để thực hiện như vậy, phương pháp đơn giản nhất là trích xuất các bản vá 2D không chồng lấn có kích thước P × P từ mỗi khung hình, như được sử dụng trong TimeSformer. Điều này dẫn đến $S = HW/P^2$, $T = F$ và $K = P^2$.

Tuy nhiên, có những phương pháp trích xuất mã thông báo video hiệu quả và tinh tế hơn. Ví dụ, trong ViViT, các tác giả đề xuất trích xuất các khối 3D từ video (bao gồm $T \neq F$) để nắm bắt thông tin không gian-thời gian trong các mã thông báo. Trong TokenLearner, họ đề xuất một trình trích xuất mã thông báo có thể học được để chọn các phần quan trọng nhất của video.

Sau khi các mã thông báo thô $\boldsymbol{x}_i$ được trích xuất, kiến trúc của bộ biến đổi sẽ hướng tới việc ánh xạ chúng vào các vectơ nhúng d chiều $\boldsymbol{Z} \in \mathbb{R}^{ST\times d}$ dsử dụng nhúng tuyến tính $\boldsymbol{E} \in \mathbb{R}^{d\times K}$:

$$\boldsymbol{Z} = \left[\boldsymbol{z}_{\text{cls}}, \boldsymbol{E}_{\boldsymbol{x}_1}, \boldsymbol{E}_{\boldsymbol{x}_2}, \ldots, \boldsymbol{E}_{\boldsymbol{x}_{ST}}\right] + \boldsymbol{PE} \qquad (8.10)$$

trong đó $\boldsymbol{x}_{\text{cls}} \in \mathbb{R}^{\text{n}}$ là một mã thông báo phân loại mã hóa thông tin từ tất cả các mã thông báo của một mẫu duy nhất và $\boldsymbol{PE} \in \mathbb{R}^{\text{ST}\times\text{d}}$ là một nhúng vị trí (Positional Embedđing) mã hóa vị trí không gian thời gian của các mã thông báo, vì các khối chú ý tiếp theo không thay đổi theo hoán vị.

Cuối cùng, các vectơ nhúng $\boldsymbol{Z}$ đi qua một chuỗi L các lớp của bộ biến đổi. Lớp ℓ của bộ biến đổi bao gồm một loạt các khối chú ý tự thân đa đầu (MSA), chuẩn hóa lớp (LN) và perception đa lớp (MLP):

$$\begin{aligned} \boldsymbol{V}^{\ell} &= \text{MSA}\ \ \text{LN}(\boldsymbol{Z}^{\ell})\ \ + \boldsymbol{Z}^{\ell} \\ \boldsymbol{Z}^{\ell+1} &= \text{MLP}\ \ \text{LN}(\boldsymbol{V}^{\ell})\ \ + \boldsymbol{V}^{\ell} \end{aligned} \qquad (8.11)$$

Theo cách này, như được thể hiện trong hình 8.2 , bốn thành phần khác nhau trong một lớp biến đổi video được biểu thị phép chiếu truy vấn-khóa-giá trị (QKV), khối MSA, phép chiếu MSA và MLP. Đối với một lớp có h đầu, độ phức tạp của mỗi thành phần là:

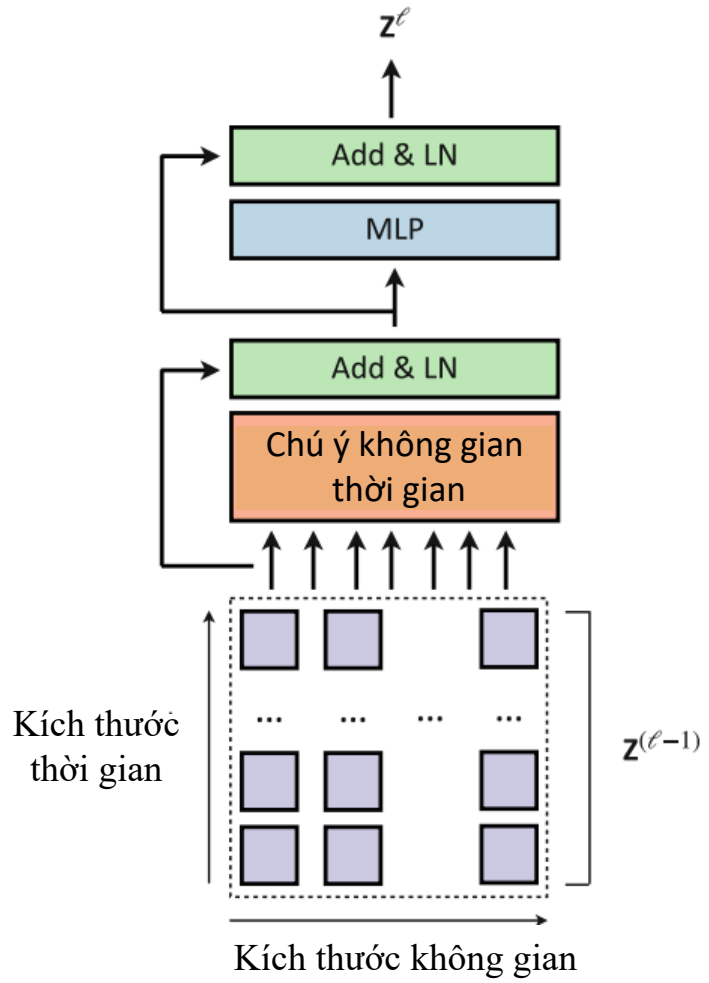
- Phép chiếu QKV: $\text{O}\ \ \text{h.}(2\text{STdd}_{\text{k}} + \text{STdd}_{\text{k}}$
- MSA: $\text{O}\ \ \text{h.S}^2\,\text{T}^2(\text{d}_{\text{k}} + \text{d}_{\text{v}})$
- MPL: $\text{O}\ \ \text{STd}^2$

Lưu ý rằng độ phức tạp của MSA là thành phần có tác động lớn nhất, với độ phức tạp bậc hai theo số lượng token. Do đó, để dễ hiểu và rõ ràng hơn, trong phần còn lại của phần này, ta coi rằng độ phức tạp toàn cục của bộ biến đổi video với L lớp bằng $\text{O}(\text{LS}^2\text{T}^2)$.

8.5.2.2. Chú ý hoàn toàn không gian-thời gian (Full Space-Time Attention)

Như được mô tả trong một số lài liệu, cơ chế chú ý không gian-thời gian đầy đủ là cơ chế chú ý không gian-thời gian cơ bản và trực tiếp nhất. Như được thể hiện trong Hình 8.15 , nó bao gồm việc tính toán sự chú ý tự thân trên tất cả các cặp mã thông báo được trích xuất.

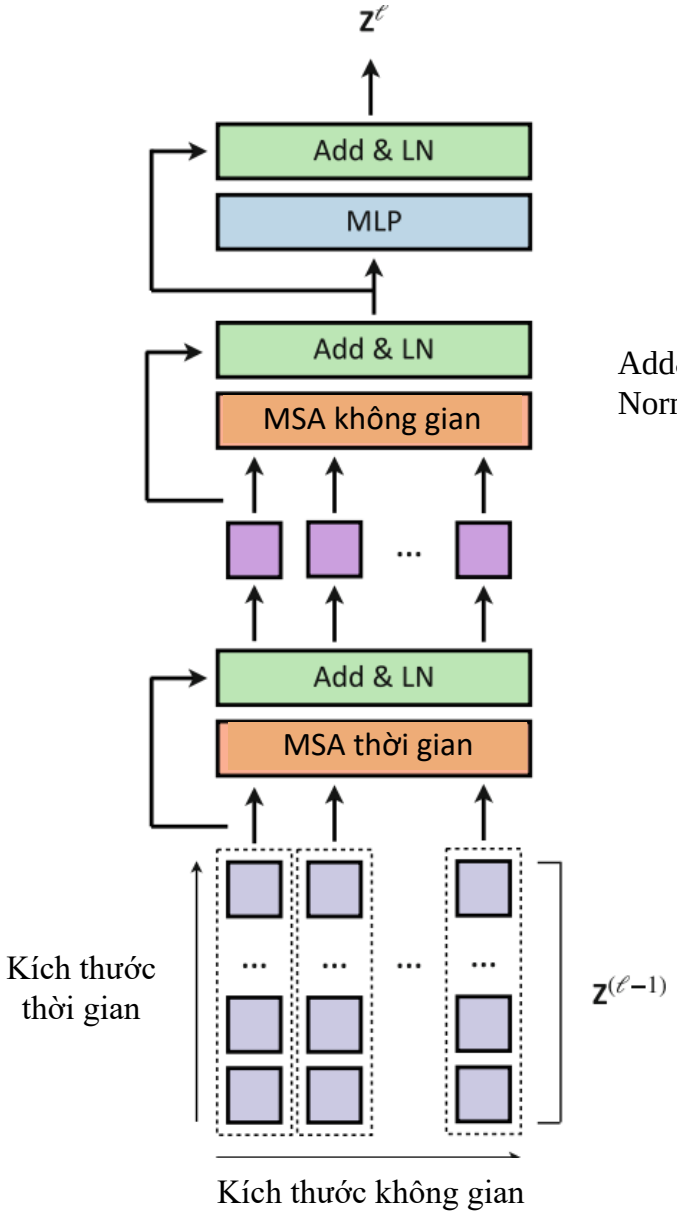
Phương pháp này dẫn đến sự phức tạp nặng nề $\text{O}(\text{LS}^2\text{T}^2)$. Độ phức tạp bậc hai này có thể nhanh chóng tiêu tốn bộ nhớ, đặc biệt đúng khi xem xét video. Do đó, việc sử dụng cơ chế chú ý không gian-thời gian đầy đủ là không thực tế.



Hình 8.15 Cơ chế chú ý không gian-thời gian đầy đủ. Các mã thông báo nhúng ở lớp $\ell-1$, $Z^{\ell-1}$ đều được đưa đồng thời qua một khối chú ý không gian-thời gian duy nhất. Cuối cùng, nhúng không gian-thời gian được truyền qua một MLP và được chuẩn hóa đến các token đầu ra của lớp tiếp theo, Z^{ℓ}

8.5.2.3. Chú ý không gian-thời gian phân chia (Divided Space-Time Attention)

Một cách thông minh hơn và hiệu quả hơn để tính toán chú ý không gian-thời gian là cơ chế chú ý không gian-thời gian phân chia , lần đầu tiên được mô tả trong [Bertasius G, Wang H, Torresani "Is space-time attention all you need for video understanding?"]. Như được minh họa trong hình 8.16, phương pháp này dựa trên việc tính toán chú ý không gian và thời gian riêng biệt trong mỗi lớp của bộ biến đổi. Thực tế, trước tiên chúng ta tính sự chú ý không gian, tức là sự chú ý tự thân trong mỗi chỉ số thời gian, và sau đó là sự chú ý thời gian, tức là sự chú ý tự thân trên tất cả các chỉ số thời gian.



Add& LN (Add and Layer Norma;ization): cộng và cuẩn hóa lớp

Hình 8.16 . Cơ chế chú ý không gian-thời gian phân chia. Các token nhúng tại lớp $\ell-1$, $Z^{\ell-1}$ trước tiên được xử lý dọc theo chiều thời gian thông qua khối MSA đầu tiên, và các token kết quả được xử lý dọc theo chiều không gian. Cuối cùng, nhúng không gian thời gian được chuyển qua một MLP và được chuẩn hóa để cho ra các token nhúng của lớp tiếp theo, Z^{ℓ}

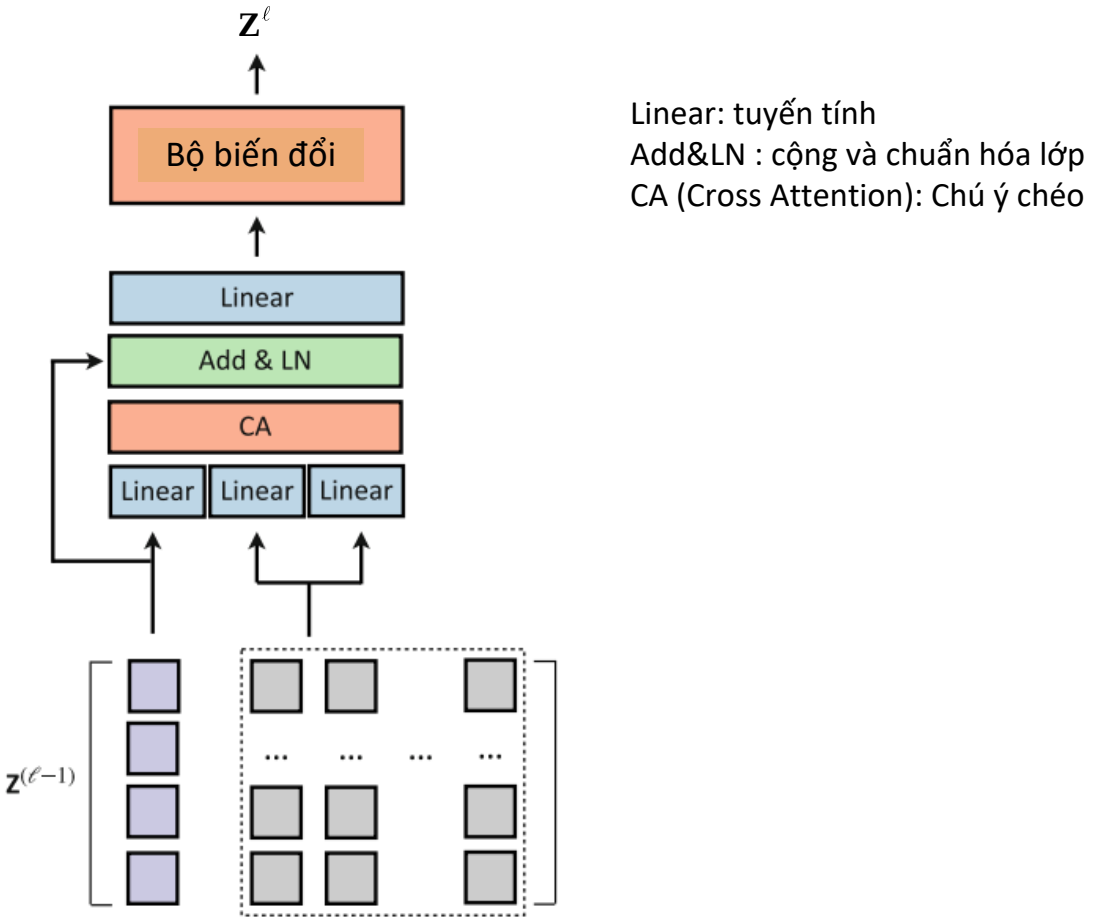
Sự phức tạp của cơ chế chú ý này là O(L ST. (S+ T))). Bằng cách tách tính toán chú ý tự than trên các chiều khác nhau, người ta có thể chế ngự được độ phức tạp bậc hai của

mô-đun MSA. Cơ chế này làm giảm đáng kể độ phức tạp của mô hình so với độ phức tạp không gian-thời gian đầy đủ. Do đó, việc sử dụng nó để xử lý video là hợp lý.

8.5.2.4. Nút thắt cổ chai do sự chú ý chéo (Cross Attention Botleneck)

Một cách tinh vi hơn nữa để giảm chi phí tính toán sự chú ý bao gồm sử dụng sự chú ý chéo làm nút thắt. Ví dụ, như được thể hiện trong hình 8.17 và được đề cập trong phần con 8.3.2 , bộ nhận thức chiếu các mã thông báo $\boldsymbol{x}_i$ đã trích xuất vào một nhúng rất ít chiều (kích thước rất thấp) thông qua một khối sự chú ý chéo được đặt trước các lớp của biến biến đổi.

Tại đây, khối chú ý chéo được đặt trước L lớp của bộ biến đổi làm giảm kích thước đầu vào từ ST xuống N , trong đó $N \ll ST$, dẫn đến độ phức tạp là O(STN). Do đó, tổng độ phức tạp của khối chú ý này là O(STN+ LN2). Nó giảm độ phức tạp so với độ phức tạp của một mô hình liên quan đến cơ chế chú ý không gian-thời gian bị chia cắt . Lưu ý rằng nó cho phép thiết kế các kiến trúc sâu, như trong bộ nhận thức, và khi đó cho phép trích xuất các tính năng cấp cao hơn.



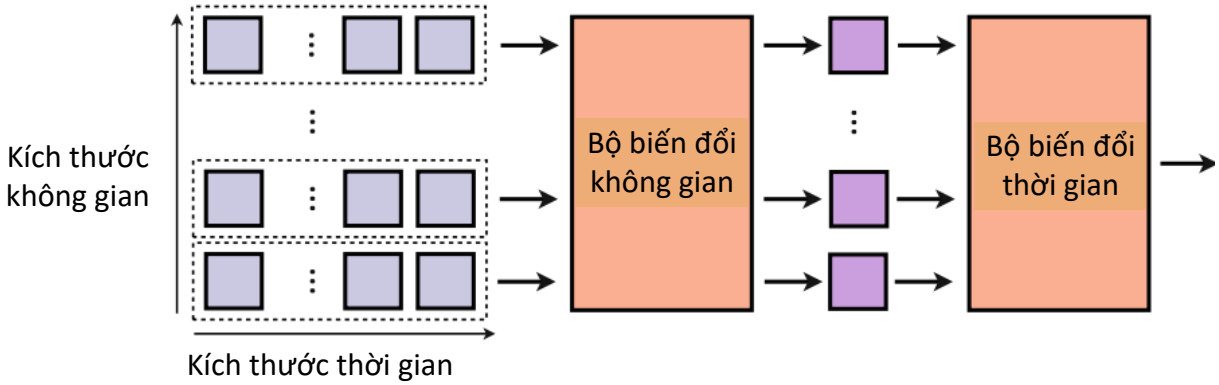
Hình 8.17 . Cơ chế thắt cổ chai chú ý. Các bản vá đầu vào thô và mã thông báo nhúng ở lớp $\ell-1$, $\mathrm{Z}^{\ell-1}$ được đưa vào khối chú ý chéo (CA), sau đó được chuẩn hóa

và chiếu. Cuối cùng, kết quả nhúng được đưa qua một bộ biến đổi để cho ra các token nhung của lớp thiếp theo, Z^{ℓ}

8.5.2.5. Bộ mã hóa phân tích nhân tử (Factorized Encoder)

Cuối cùng, kiến trúc bộ mã hóa phân tích nhân tử là hiệu quả nhất khi xét đến sự đánh đổi giữa độ phức tạp/hiệu suất.

Giống như trong sự chú ý không gian-thời gian phân chia , bộ mã hóa phân tích nhân tử hướng đến việc tính toán sự chú ý không gian và thời gian riêng biệt. Tuy nhiên, như thể hiện trong hình 8.18, thay vì trộn lẫn các token không gian-thời gian trong mỗi lớp biến đổi, ở đây, tồn tại hai bộ mã hóa riêng biệt : thứ nhất, biểu diễn của từng chỉ số thời gian được thu thập nhờ bộ mã hóa không gian với L_s lớp. Thứ hai, các token này được truyền qua bộ mã hóa thời gian với L_t (tức là $L=L_s+L_t$).



Hình 8.18. Cơ chế mã hóa phân tích nhân tử. Đầu tiên, một bộ biến đổi không gian xử lý các mã thông báo đầu vào dọc theo chiều không gian. Sau đó, một bộ biến đổi thời gian xử lý nhúng không gian kết quả dọc theo chiều thời gian

Do đó, độ phức tạp của một kiến trúc như vậy có hai thành phần chính: độ phức tạp của bộ mã hóa không gian $O\ L_sS^2$ và độ phức tạp của bộ mã hóa thời gian $O\ L_sS^2$. Nó dẫn đến sự phức tạp toàn cục là $O\ L_sS^2 + L_tT^2$. Do đó, nó dẫn đến các mô hình nhẹ ký (Lightweight Model). Tuy nhiên, vì trước tiên nó trích xuất các tính năng trên mỗi khung hình và sau đó tổng hợp chúng thành biểu diễn cuối cùng, nên nó tương ứng với cơ chế hợp nhất muộn (Late-Fussion Mechanism), đôi khi có thể là một nhược điểm vì nó không kết hợp thông tin không gian và thời gian đồng thời.

8.5.3. Bộ biến đổi video (Video Transformer)

Trong phần này, chúng tai trình bày hai kiến trúc hiện đại dựa trên bộ biến đổi (transformer) để phân loại video. Chúng ta bắt đầu bằng việc giới thiệu kiến trúc TimeSformer trong phần con 8.5.3.1 và sau đó là kiến trúc ViViT trong phần con 8. 5.3.2.

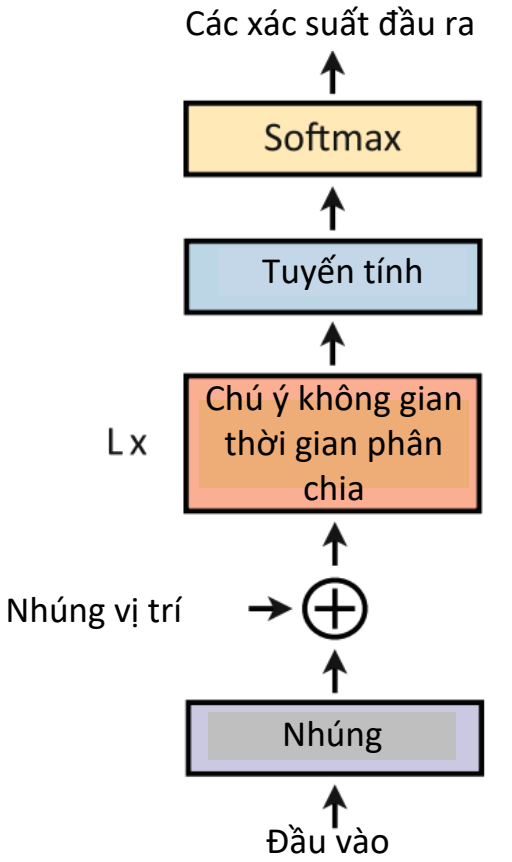
8.5.3.1. TimeSformer

TimeSformer (Time-Space Transformer: Bộ biến đổi thời gian- không gian) là một trong những kiến trúc đầu tiên chú trọng đến không gian-thời gian, tác động đến lĩnh vực phân loại video. Nó tuân theo cùng cấu trúc và nguyên tắc được mô tả trong phần con 8.5.2.1 .

Đầu tiên, nó lấy đầu vào là một video clip RGB được lấy mẫu ở tốc độ 1⁄32 và phân tích thành các bản vá 2D 16 × 16.

Như thể hiện trong hình 8.19 , kiến trúc TimeSformer dựa trên kiến trúc ViT (phần con 8. 2.3), với 12 lớp MSA 12 đầu. Tuy nhiên, giá trị gia tăng so với ViT là TimeSformer sử dụng cơ chế chú ý không gian-thời gian được phân chia (phần con 8.5.2.3). Cơ chế chú ý như vậy cho phép nắm bắt các tính năng không gian thời gian cấp cao trong khi vẫn làm chủ được sự phức tạp của mô hình. Hơn nữa, các tác giả giới thiệu ba biến thể của kiến trúc: (i) TimeSformer, phiên bản tiêu chuẩn của mô hình, hoạt động trên 8 khung hình 224 × 224; (ii) TimeSformer-L, cấu hình có độ phân giải không gian cao, hoạt động trên 16 khung hình 448 × 448; và (iii) TimeSformer-HR, thiết lập phạm vi thời gian dài, hoạt động trên 96 khung hình 224 × 224.

Cuối cùng, mã thông báo phân loại đầu cuối được nhúng qua MLP để đưa ra xác suất cho tất cả các loại video. Trong quá trình suy luận, dự đoán cuối cùng nhận được bằng cách lấy trung bình các xác suất đầu ra từ ba góc cắt không gian khác nhau của video clip đầu vào (trên cùng bên trái, giữa và dưới cùng bên phải)



Hình 8.19. Kiến trúc TimeSformer. Đầu tiên, TimeSformer chiếu dữ liệu đầu vào vào các token nhúng, sau đó được cộng lại thành các token nhúng vị trí. Các token kết quả sau đó được truyền qua L khối chú ý không gian-thời gian chia nhỏ và sau đó được chiếu tuyến tính để thu được các xác suất đầu ra

TimeSformer đạt được hiệu suất tiên tiến tương tự như CNN 3D trên các tập dữ liệu phân loại video khác nhau, chẳng hạn như Kinetics-400 và Kinetics-600. Lưu ý rằng TimeSformer được đào tạo nhanh hơn nhiều (416 giờ đào tạo so với 3840 giờ đối với kiến trúc SlowFast) và cũng hiệu quả hơn (0,59 TFLOP so với 1,97 TFLOP đối với kiến trúc SlowFast), (TFLOP: Trillion Floating Point Operration là một nghìn tỷ phép tính động mỗi giây).

8.5.3.2. ViViT

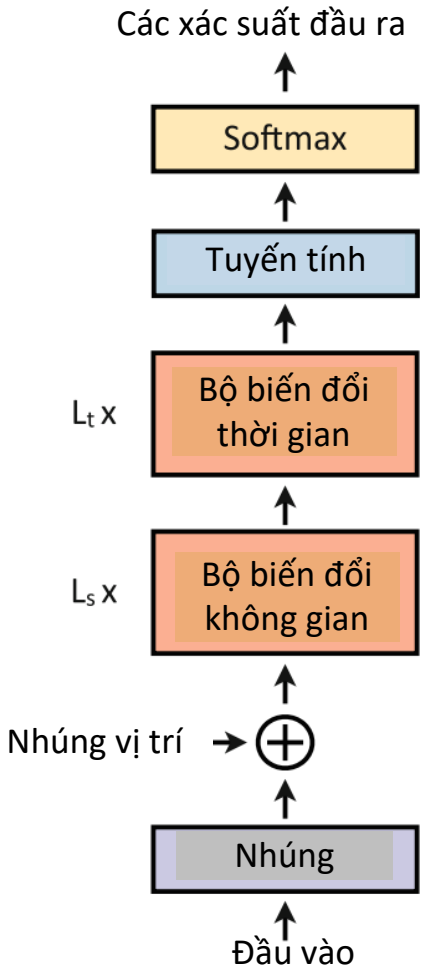
ViViT (Video Vision Transformer: bộ biến đổi thị giác video) là phần mở rộng chính của kiến trúc ViT (phần con 8.2.3) để phân loại video.

Đầu tiên, các tác giả sử dụng nhúng 16 tubelet (ống nhỏ) thay vì nhúng bản vá 2D, như đã đề cập trong tiểu phần 8. 5.2.1 . Phương pháp nhúng thay thế này nhằm mục đích nắm bắt thông tin không gian-thời gian từ bước bước teken hóa, không giống như các kiến trúc tiêu chuẩn hợp nhất thông tin không gian-thời gian từ khối chú ý đầu tiên.

Như thể hiện trong hình 8.20, kiến trúc ViViT dựa trên kiến trúc bộ mã hóa được phân tích thành nhân tử (tiểu phần 8.5.2.5) và bao gồm một bộ mã hóa không gian và một bộ mã hóa thời gian hoạt động trên các clip đầu vào với 32 khung hình 224 × 224. Bộ mã hóa không gian sử dụng một trong ba biến thể ViT làm xương sống. Đối với bộ mã hóa thời gian, số lượng lớp không ảnh hưởng nhiều đến hiệu suất, do đó, tương ứng với sự đánh đổi giữa hiệu suất/độ phức tạp, số lớp MSA được cố định ở mức 4. Các tác giả chỉ ra rằng kiến trúc như vậy đạt được hiệu suất cao trong khi giảm đáng kể độ phức tạp.

Cuối cùng, giống như trong TimeSformer (tiểu phần 8.5.3.1), ViViT đưa ra xác suất cho tất cả các loại video thông qua mã thông báo phân loại cuối cùng và tính trung bình các xác suất thu được trên ba lần cắt của mỗi clip đầu vào (trên cùng bên trái, giữa và dưới cùng bên phải).

ViViT vượt trội hơn cả CNN 3D và TimeSformer trên các tập dữ liệu Kinetics-400 và Kinetics-600. Lưu ý rằng độ phức tạp của kiến trúc này giảm đáng kể khi so sánh với các mô hình tiên tiến khác. Ví dụ, số FLOP của ViViT-L/16 × 16 × 2 là $3{,}89 \times 10^{12}$ so với $7{,}14 \times 10^{12}$ của TimeSformer-L và $7{,}14 \times 10^{12}$ của kiến trúc SlowFast.



Hình 8.20 . Kiến trúc ViViT. Đầu tiên, ViViT đầu tiên chiếu các dữ liệu đầu vào vào các mã thông báo nhúng, sau đó được cộng với các mã thông báo nhúng vị trí. Các mã thông báo kết quả trước tiên được truyền qua L_s bộ biến đổi không gian và sau đó qua L_t biến đổi thời gian. Đầu ra kết quả được hiếu tuyến tính để nhận được các xác suất đầu ra

8.5.3.4. Bộ biến đổi video đa phương thức (Multimodal Video Transformer)

Ngày nay, một trong những khoảng cách chính giữa trí tuệ nhân tạo và trí tuệ con người là khả năng xử lý tín hiệu đa phương thức và làm phong phú thêm kết quả phân tích bằng cách kết hợp các phương thức (Modality) khác nhau. Hơn nữa, cho đến gần đây, các mô hình học sâu chủ yếu tập trung vào các nhiệm vụ thị giác rất cụ thể, thường dựa trên một phương thức duy nhất, chẳng hạn như phân loại hình ảnh, phân loại âm thanh và dịch máy. Hai yếu tố này kết hợp lại đã thúc đẩy các nhà nghiên cứu tiếp nhận những thách thức đa phương thức .

Giải pháp mặc định cho các nhiệm vụ đa phương thức bao gồm việc đầu tiên tạo một mô hình riêng lẻ (hoặc mạng) cho mỗi phương thức, sau đó kết hợp các tính năng đơn phương thức thu được lại với nhau. Tuy nhiên, cách tiếp cận này không thể mô hình hóa các tương tác hoặc mối tương quan giữa các phương thức khác nhau. Tuy nhiên, sự gia tăng gần đây của chú ý (Attention) rất hứa hẹn cho các ứng dụng đa phương thức, vì chú ý hoạt động rất tốt trong việc kết hợp đa đầu vào.

Ở đây, hai cách chính để xử lý một số phương thức được trình bày:

1) **Ghép các mã thông báo từ các phương thức khác nhau thành một vectơ.**

Bộ chuyển đổi video đa phương thức (MM-ViT) kết hợp các khung RGB thô, các tính năng chuyển động và phổ đồ (Spectogram) để nhận dạng hành động video. Để thực hiện như vậy, các tác giả hợp nhất các mã thông báo từ tất cả các phương thức khác nhau thành một nhúng đầu vào đơn và truyền nó qua các lớp biến đổi. Tuy nhiên, một nhược điểm của phương pháp này là nó không phân biệt tốt một phương thức này với phương thức khác. Để khắc phục vấn đề này, các tác giả của bộ nhận thức đề xuất học một nhúng phương thức (Modality Embedding) ngoài nhúng vị trí (xem phần con 8.3.2 và 8.5.2.1). Điều này cho phép liên kết từng mã thông báo với phương thức của nó. Tuy nhiên, xét đến (i) độ phức tạp của một lớp của bộ biến đổi là bậc hai đối với số lượng mã thông báo (phần con 8.5.2.1) và (ii), với phương pháp này, số lượng mã thông báo được nhân với số lượng phương thức, điều này có thể dẫn đến chi phí tính toán tăng vọt.

2) **Khai thác chú ý chéo.**

Một số phương pháp tiếp cận hiện đại khai thác sự chú ý chéo để kết hợp nhiều phương thức, chẳng hạn như cho âm thanh và video, cho văn bản và video và cho âm thanh, văn bản và video. Điểm chung giữa tất cả các phương pháp này là chúng khai thác các thuộc tính nội tại của sự chú ý chéo bằng cách truy vấn một phương thức với một cặp khóa-giá trị từ phương thức kia. Ý tưởng này có thể dễ dàng được khái quát hóa thành nhiều hơn hai phương thức bằng cách tính toán sự chú ý chéo trên mỗi tổ hợp các phương thức.

8.6. KẾT LUẬN

Chú ý là một kỹ thuật trực quan và hiệu quả giúp xử lý các tín hiệu cục bộ và toàn cục.

Trên cơ sở này, kiến trúc chú ý thuần túy đầu tiên, bộ biến đổi của Vaswani A, Shazeer N và các cộng sự đã được thiết kế cho mục đích NLP. Lĩnh vực thị giác máy tính đã nhanh chóng điều chỉnh kiến trúc bộ biến đổi để phân loại hình ảnh bằng cách thiết kế mô hình bộ biến đổi thị giác đầu tiên: bộ biến đổi thị giác ViT (Vision Transformer).

Tuy nhiên, ngay cả khi các biến đổi một cách tự nhiên mang lại hiệu suất cao, cơ chế chú ý thô sơ vẫn là một kỹ thuật tham lam và nặng về mặt tính toán. Vì lý do này, một số dẫn xuất được cải tiến và tinh chỉnh của cơ chế chú ý đã được đề xuất.

Sau đó, nhanh chóng, nhiều nhiệm vụ khác nhau đã được chinh phục bởi các kiến trúc dựa trên bộ biến đổi, chẳng hạn như phát hiện đối tượng, phân đoạn hình ảnh, học tự giám sát và tạo hình ảnh. Ngoài ra, các kiến trúc dựa trên bộ biến đổi đặc biệt phù hợp để xử lý các nhiệm vụ đa chiều (Multidimentional Task). Điều này là do các tín hiệu đa phương thức dễ dàng được kết hợp thông qua các khối chú ý, đặc biệt là các tín hiệu thị giác và ngôn ngữ và các tín hiệu không gian thời gian cũng dễ dàng được thuần hóa.

Vì những lý do này, kiến trúc dựa trên bộ biến đổi đã cho phép nhiều lĩnh vực đạt được những tiến bộ vượt bậc trong vài năm qua. Trong tương lai, bộ biến đổi sẽ cần phải ngày càng hiệu quả hơn về mặt tính toán, ví dụ như có thể sử dụng trên điện thoại di động, và sẽ đóng một vai trò to lớn trong việc giải quyết các thách thức đa phương thức và kết nối hầu hết các lĩnh vực AI.

THUẬT NGỮ VÀ KÝ HIỆU

AAE	Adversarial Autoencoder	Các bộ tự mã hóa đối nghịch
AI	Artificial Intelligence	Trí tuệ nhân tạo
Adagrad	Adaptive Gradient	Gradient thích ứng
Adam	Adaptive Moment Estimation	Ước tính momen thích ứng
	Adversarial Training	Đào tạo đối nghịch
ANN	Artificial Neural Network	Mạng tế bào tần kinh nhân tạo
	Artistic Style Transfer	Chuyển giao phong cách nghệ thuật
	Attention	Chú ý. Một kỹ thuật trong Khoa học Máy tính mô phỏng cách não bộ tập trung vào các phần liên quan của thông tin đầu vào
	Autoencoder	Bộ tự mã hóa
	Average Pooling	Gộp trung bình
	Backpropagation	Truyền lan ngược
	Batch Normalization	Chuẩn hóa lô
BERT	Binary Encoder Reppresantations from Transformer	Mô tả bộ mã hóa nhị phân từ bộ biến đổi
	Bias	độ lệch hay thên kiến

	Bias-Variance Trade-off	Đánh đổi giữa độ lệch và phương sai
	Binary Classification	Phân loại tiuyến tính
	Between Class Scatter	Độ phân tán giữa các lớp
	Blank Helper Function	Hàm trợ lý trống
BPTT	Back-Propagation Through Time	Truyền lan ngược xuyên thời gian
	Brute Force Method	Phương pháp vét cạn hay toàn diện
CA	Cross Attention	Chú ý chéo
cGAN	Convolutional GAN	GAN có điều kiện
CDF	Cumulative Density Function	Hàm mật độ tích lũy
CIFAR-10		Tập dữ liệu các ảnh tự nhiên
CLIP	Connecting Text and Images	Kết nối văn bản và hình ảnh
	Classification	Phân loại
	Classification Token	Mã thông báo phân loại
	Compact Convolutional Transformer	Bộ biến đổi tích chập nhỏ gọn
CT	Computed Topography	Chụp cắt lớp
CycleGAN	Cycle-Generative Adversarial Network	Mạng tạo sinh đối nghịch nhất quán chu kỳ
	Conditional Probability	Xác suất có điều kiện
	Convolution	Tích chập
CNN	Convolution Neural Network	Mạng tế bào thần kinh tích chập
	Computer Vision	Thị giác máy tính
	Contrasting GAN	GAN tương phản
	Cost Function	Hàm chi phí
CE	Cross Entropy	Entropy chéo
	Cross Entropy Loss	Mất mát entropy chéo
	CycleGAN	GAN chu kỳ
	Data Analysis	Phân tích dữ liệu
	Data Generrative Distributor	bBộ phân phối tạo sinh dữ liệu
	Decision Tree	Cây quyết định
	Decision Function With Normal Distribution	Hàm quyết định với phân phối chuẩn
DGAN	Dynamicall Adaptive GAN	mạng đối nghịch tạo sinh thích ứng động
DeiT	Data-Efficient Image Transformer	Bộ biến đổ ảnh hiệu quả dữ liệu
DETR	Detection Transformer	Bộ biến đổi phát hiện
	Denoising Autoencoder	Bộ mã hóa giảm tạp âm
	Divided Space-Time Attention	Chú ý không gian-thời gian phân chia
DINO	Self-DIstillation with NO labels	Tự chưng cất không cần nhãn
	Distillation Network	Mạng chưng cất
DL	Deep Learning	Học máy sâu

BẢN GỐC

	Downsampling	Lấy mẫu giảm mẫu
DRN	Deep Residual Network	Mạng dư sâu
DNN	Deep Neural Network	Mạng tế bào thần kinh sâu
	Downstream Task	Nhiệm vụ hạ nguồn
DRL	Deep reinforcement learning	Học tăng cường sâu
	Density Estimattion	Ước tính mật độ
	Desired Prior Distribution	Phân phối trước theo mong muốn
	Diffusion Model	Mô hình khuếch tán
	Dimensionality Reduce	Giảm kích thước
	Elastic-net penalty	Phạt lưới đàn hồi
	Embedđing Token	Mã thông báo nhúng
	Error Corecting Output Code	Mã đầu ra sửa lỗi
EM	Expectation-Maximization	Tối đa hóa kỳ vọng
EOS	End of Sequence	Kết thúc chuỗi
	Extremely Randomized Trees	Các cây được ngẫu nhiên hóa cực đoan
	ε-insensitive Loss	mất mát không nhậy cảm ε
	Feature	Tính năng
	Factorized Encoder	Bộ mã hóa phân tích nhân tử
	Feature Engineeering	Kỹ thuật tính năng
	Feedforward Ancestral Sampling	Lấy mẫu tổ tiên theo hướng truyền thẳng
	Flamingo	Một mô hình ngôn ngữ thị giác
	Full Decomposition	Phân tích hoàn toàn
	Full Space-Time Attention	Chú ý hoàn toàn không gian-thời gian
GAN	Generative Adversarial Network	mạng đối nghịch tạo sinh
GAWWN	Generative Adversarial What-Where Network	Mạng cái gì-ở đâu đối nghịch tạo sinh
	GAN with Inference Models	GAN với các mô hình suy luận
	Gaussian Mixture Model	Mô hình hỗn hợp Gauss
	Generator Network	Mạng tạo sinh
	Gradient Descent	Sự giảm độ dốc
	Ground Truth Label	Nhãn chân lý cơ bản
	Heuristic Avaraging	Trung bình hóa theo kinh nghiệm
	Hidden Layer	Lớp ẩn
	Hyperplane	Siêu mặt phẳng
ifmaps	input feature maps	Các bản đồ tính năng đầu vào
	Image-Image Traslation	Chuyển đổi ảnh sang ảnh
	Image Manipulation	Chỉnh sửa ảnh
	Image Segmentation	Phân đoạn ảnh
	Inference	Suy luận
	Input Token	Mã thông báo đầu vào
ICDF	Inverse Cumulative Density	Hàm mật độ tích lũy đảo

BẢN GỐC

	Function	
	Joint Probability	Xác suất kết hợp
	Kenel Principal Component Analysis	Phân tích phần tử chính hạt nhân
	Kenel Trick	Kỹ thuật kenel
	k-means	k –trung bình
	K-Nearest Neighborg Classifier	Bộ phân loại K láng giềng gần nhất
	Kullback-Leibler Divergence	Phân kỳ Kullback-Leibler
LAPGAN	Laplace Generative Adversarial Network	Mạng tạo sinh đối nghịch Laplace
	Latent Space	Không gian tiềm ẩn
	Likelihoood Function	Hàm khả năng
LDA	Linear Discrimnant Analysis	Phân tích phân biệt tuyến tính
	Linear Kenel	Kenel tuyến tính
	Logistic Regression	Hồi quy logistic
	Loss Function	Hàm mất mát
LSTM	Long-Short Term Memory Network	Mạng bộ nhớ dài-ngắn hạn
MAC	Multiply-and-Accumlate	Nhân và tích lũy
MAE	Bộ mã hóa tự động được che dấu	Masked Autoencoder
MCA	Multi-Head Cross Attention	Chú ý tự thân chéo đa đầu
MMSA	Masked Multi-Head Self Attention	Chú ý tự thân đa đầu được che giấu
MSA	Multi-head Self Attention	Chú ý tự thân đa đầu
	Marginal Probability	Xác suất biên
	Multimodial Transformer	Bộ biến đổi đa phương thức
	Multimodal Video Transformer	Bộ biến đổi video đa phương thức
	Markove Chain Monte Carlo Method	Chuỗi Markov Monte Carlo
	Max Pooling	Gộp cực đại
MLE	Maximum Likelihood Estimation	Ước tính khả năng cực đại
MLP	Multi Layer Perceptron	Mạng perceptron đa lớp
MRI	Magnetic Resonance Image	Chụp cộng hưởng từ
MNIST	Modiffied National Standard Institute of Technology	Viện công nghệ tiêu chuẩn quốc giađược sửa đổi
	MNIST Database	Tập dữ iệu các chữ số được viết bằng tay
MSEL	Mean Squared Error Loss	Mất mát sai số bình phương trung bình
	Mode Collapse	Sụp đổ chế độ
	Multiclasss Classification	Phân loại nhiều loại
	Multiple Linear Regression	Hồi quy tuyến tính đa biến

BÁN GỐC

	Multinomial Logistic Regression	Hồi quy logistic đa thức
	Naïve Bayes	Bayes ngây thơ
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
	Nearest Neighbor	Láng giềng gần nhất
	Nearest Neighbor Classifier	Bộ phân loại láng giềng gần nhất
	Neighborhood	Vùng láng giềng
	Neuron	Tế bào thần kinh
	Objective Function	Hàm mục tiêu
	Overfitting	Quá khớp
	Perceiver	Bộ nhận biết
	Perceiver Resampler	Bộ lấy mẫu lại của bộ nhận biết
	Perceptron	Mô hình một tế bào thần kinh nhân tạo đơn lẻ
PCA	Principal Component Analysis	Phân tích phần tử chính
	Penalized Models	Các mô hình phạt
PDF	Probability Density Function	Hàm mật độ xác suất
	Positoning Encoding	Mã hóa vị trí
	Probability Distribution	Phân phối xác suất
	Probability Mass Function	Hàm khối xác suất
	Polymomial Kenel	Kenel đa thức
	Pooling	Gộp
	Prediction	Dự đoán
	Pricipal Component Analysis	Phân tích thành phần chính
	Quadratic Discriminant Analysis	Phân tích phân biệt bậc hai
RBF	Radial Basis Function	Hàm cơ sở xuyên tâm
	Random Forest Classifier	Bộ phân loại rừng ngẫu nhiên
	Real Life Metaphor	Ẩn dụ đời thực
	Receptive Field	Trường tiếp nhận
	Đảo ngược quy trình tạo chú thích (Reverse Captioning	
RNN	Recurrent Neuron Network	Mạng neuron hồi quy
RNNs	Recurrent Neuron Networks	Các mạng neuron hồi quy
	Regularization	Chính quy hóa
	Regression	Hồi quy
RL	Reinforcement Learning	Học máy tăng cường
	Representer Theorem	Định lý biểu diễn
ResNet	Residual Network	Mạng dư sâu
RMSProp	Robot Mean Square Proropogation	Lan truyền bình phương trung bình robot
	Running Average	Trung bình chạy
	Rridge regression	Hồi quy ridge
	Sigmoid Kenel	Kenel sigmoid
	Simple Linear Regression	Hồi duy tuyến tính đơn biến
	SimpleRNN	RNN đơn giản

BẢN GỐC

	Self Attention	Chú ý tự thân
	Shortcut Connection	Kết nối tắt
SL	Supervised Learning	Học máy có giám sát
	Squared Error	Sai số bình phương
SOS	Start of Sequence	Bắt đầu chuỗi
	Surface Normal Map	Bản đồ pháp tuyến bề mặt
SGD	Stochastic Gradient Descent	Giảm gradient ngẫu nhiên
SRGAN	Super-Resolution GAN	GAN siêu phân giải
	Sparse Matrix	Ma trận thưa
	StyleGAN	GAN phong cách
SVM	Support Vector Machine	Máy vectơ hỗ trợ
	Text to Image	Chuyển đổi văn bản thành hình ảnh
TFD	Toronto Face Dataset	Tập dữ liệu khuôn mặt Toronto
TimeSformer	Time-Space Transformer	Bộ biến đổi thời gian- không gian
	Toeplitz Matrix	ma trận Toeplitz
	Training	Đào tạo
	Transfer Learning	Học chuyển giao
	Transformer	Bộ biến đổi
	Transformer Encoder	Bộ mã hóa của bộ biến đổi
	Translation Equivariance	Tương biến tịnh tiến
	Tree Based Algorithm	Giải thuật dựa trên cây
	Tree Based Method	Phương pháp dựa trên cây
UL	Unsupervised Learning	Học máy không giám sát
	Validation sets for Hyperparameter tuning	Các tập xác thực để điều chỉnh siêu tham số
	Variance	Phương sai
	Variational Autoencoder	Bộ tự mã hóa biến thiên
	Video Attention	Chú ý video
	Video Transformer	Bộ biến đổi video
VilBERT	Vision-Language BERT	BERT thị giác và ngôn ngữ
VLM	Visual Language Model	Mô hình ngôn ngữ thị giác
	Virtual Batch Normalization	Chuẩn hóa lô ảo
	Visual Encoder	Bộ mã hóa thị giác
ViT	Visual Transformer	Bô biến đổi thị giác
ViViT	Video Vision Transformer	Bộ biến đổi thị giác video
	Wasserstein Distance	Khoảng cách Wasserstein
WGAN	Wasserstein GAN	Mạng đối kháng tạo sinh Wasserstein
	Within-Class Scatter	Phân tán trong cùng lớp

BẢN GỐC

TÀI LIỆU THAM KHẢO

1. Abdelfatteh Haidine , "Artificial Intelligence and Machine Learning in 5G and beyond: A Survey and Perspectives". IntechOpen, 2021
2. Krizhevsky A, Sutskever I, Hinton GE "ImageNet classification with deep convolutional neural networks". In: Advances in neural information processing systems, vol 25, pp 1097–1105, 2012
3. Olivier Colliot , "A Non-technical Introduction to Machine Learning". In: Machine Learning for Brain Disorders , 2023
4. Johann Faouzi and Olivier Colliot, "Classic Machine Learning Methods". In: Machine Learning for Brain Disorders , 2023
5. Maria Vakalopoulou and Others,, "Deep Learning: Basics and Convolutional Neural Networks (CNNs)". In: Machine Learning for Brain Disorders , 2023
6. Susmita Das , "Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research". In: Machine Learning for Brain Disorders , 2023
7. Markus Wenzel , "Generative Adversarial Networks and Other Generative Models". In: Machine Learning for Brain Disorders , 2023
8. Robin Courant and Others , "Transformers and Visual Transformers". In: Machine Learning for Brain Disorders , 2023
9. Vincent Dumoulin and Others, "Generative Adversarial Networks: An Overview". ArXiv, 2017
10. Riccardo Bnedetti, "From Artifcial Intellegence to Artificial Adversarial Networks". 2019
11. Yichuan Tang ,"Deep Learning using Linear Support Vector Machines". ArXiv, 2015
12. Bharati Vidyapeeth and Orthers ,"The Fundamentals of Machine Learning". Research Gate, 6/2023
13. Kaiming He and Orthers , "Deep Residual Learning for Image Recognition". Arxiv, 2015
14. Roger Grosse , "Lecture 10: Image Classification"
15. Goel Surbhi, "Linear and Logistic Regresion". CIS 5200: Machine Learning, 2023
16. "Lecture 2: Machine Learning". cs 221.Standford.Edu, 2019
17. Surbhi Goel , "Lecture 1: Binary Classification with Linear Predictors". CIS5200: Machine Learning, 2013
18. Kai-Wei Chang ," Lecture 2: Binary Classification". UCLA Engineeering CS6269
19. Kai-Wei Chang ,"Lecture 3: Multi-Class Classification"., UCLA Engineeering CS6501

20. Daniel Hugo Campora Perez , "A practical approach to Convolutional Neural Networks". CERN School of Computing, Mar 5th - 7th, 2019
21. Maria Vakalopoulou, "Deep learning: basics and convolutional neural networks (CNN)". HAL, 11/2023
22. CS231n Deep Learning For Computer Vision, "Linear Classification".
23. CS231n Deep Learning For Computer Vision, "Backpropagation".
24. CS231n Deep Learning For Computer Vision, "Image Classification".
25. CS231n Deep Learning For Computer Vision, "Loss Functions and Optimization"
26. CS231n Deep Learning For Computer Vision, "Convolutional Neural Networks (CNNs / ConvNets)"
27. Vivienne Sze and Orthers, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". ArXiv, 2017

BẢN GỐC

PHẦN 2

MỤC LỤC

Chương 9. Ứng dụng AI/ML cho các dịch vụ trong các hệ thống truyền thông di động không dây
9.1. Tổng quan
9.2. Lựa chọn điểm phân chia cho hoạt động AI/ML giữa thiết bị di động và mạng lõi
9.3. Các chế độ hoạt động cho AI/ML phân chia giữa đầu cuối di động và mạng
9.4. Phân phát mô hình AI/ML trong các hệ thống truền thông di động không dây
9.5. Nguyên lý học phân tán/ liên kết trên các hệ thống truyền thông di động không dây
9.6. Các kịch bản hoạt động của AI/ML được phân chia giữa các điểm cuối trong các hệ thống truyền thông di động không dây và các yêu cầu hiệu năng
9.7. Điều khiển được phân chia cho robot
9.8. Các hoạt động tính toán phân chia chuyển giao mô hình đặc thù phiên
9.9. Phân phối và chia sẻ mô hình/dữ liệu AI/ML trong hệ thống 5G
9.10. Học phân tán/ liên kết trên hệ thống 5G (Distributed/ Federated Learning Over 5G System)

Chương 10. Tích hợp AI/ML vào mạng truyền thông di động 5G
10.1. Tổng quan kiến trúc 5G dựa trên SBA (5G SBA)
10.2. Khung AI/ML chung
10.3. AI/ML trong nghiên cứu các khía cạnh hệ thống của 3GPP
10.4. Hoạt động AI/ML lớp ứng dụng trong hệ thống 5G
10.5. AI/ML cho các phương tiện truyền thông (Media) và quản lý AI/ML trong 5G
10.6. AI/ML cho mạng truy nhập vô tuyến 5G
10.7. Các trường hợp sử dụng AI/ML cho giao diện vô tuyến của 5G NR

Chương 11. Tích hợp AI/ML vào mạng truyền thông di động 6G
11.1. Lộ trình phát triển lên 6G
11.2. Tổng qua tích hợp AI và truyền thông trong 6G
11.3. AI cho mạng (AI4NET)
11.4. Mạng cho AI (NET4AI)
11.5. Các công nghệ hỗ trợ chủ chốt
11.6. AI as a Service (AIaaS)
11.7. Mô hình lớn của mạng không dây
11.8. Tổng kết

Chương 12. RAN mở (O-RAN) và hoạt động AI/ML trong O-RAN
12.1. Tổng quan
12.2. Các nguyên lý then chốt của O-RAN
12.3. RIC gần thời gian thực và điều khiển RAN
12.4. Khung RIC phi thời gian thực (Non RT-RIC) và điều phối
12.5. Các giao diện mở của O-RAN
12.6. Hoạt động AI/ML trong O-RAN
12.7. Các trường hợp sử dụng và nghiên cứu O-RAN
12.8. An ninh trong O-RAN
12.9. Tổng kết

Chương 13. Các mạng truy nhập vô tuyến dựa trên AI/ML
13.1. Tổng quan
13.2. KIến trúc tham chiếu
13.3. Các khái niệm và các khả năng then chốt của RAN AI gốc
13.4. Các thiết kế then chốt cho RAN AI gốc giai đoạn đầu
13.5. Các kiến trúc O-RAN AI gốc (AI-Native O-RAN) giai đoạn đầu
13.6. Các kiến trúc O-RAN AI gốc (AI-Native O-RAN) cho 6G
13.7. Tổng kết

BẢN GỐC

Chương 9

TÍCH HỢP AI/ML VÀO CÁC MẠNG TRUYỀN THÔNG KHÔNG DÂY DI ĐỘNG 5G

9.1. TỔNG QUAN

AI/ML đóng vai trò thay đổi sâu sắc trong các mạng truyền thông di động để tăng cường hiệu quả nhiệm vụ, tối ưu hóa lập cấu hình và quản lý mạng, dự đoán phát hiện sự cố và cải thiện từ đầu đến cuối trải nghiệm người dùng đầu cuối. 3GPP bắt đầu tích hợp các tính năng AI/ML vào hệ thống 5G từ phát hành 17. Công việc phát triển tiêu chuẩn của 3GPP chia lm ba nhóm chính: mạng truy nghập vô tuyến (RAN: Radio Access Network), các khía cạnh dịch vụ và hệ thống (SA: Service and System Aspects) và mạng lõi và thiết bị đầu cuối (CT: Core Network and Terminal). Nhóm SA chịu trách nhiệm cho tiêu chuẩn hóa các chức năng mạng lõi (CN), an ninh, các khai thác, quản trị và bảo trì (OAM: Operations, Adminisstration and Maintenance) và hỗ trợ ứng dụng từ đầu đến cuối. Nhóm RAN phát triển các tiêu chuẩn trạm gốc và giao diện vô tuyến, trong khi đó nhóm CT phát triển các giao thức thiết bị người dùng (UE) và CN cũng như tương tác với mạng ngoài.Trong 3GPP, AI/ML bắt đầu được khám phá trên nhiều lĩnh vực bao gồm trí tuệ RAN, trí tuệ OAM và việc tích hợp các khả năng AI/ML vào kiến trúc 5G tiên tiến (5G Advanced) bắt đầu từ phát hành 18.

Trong 5G và B5G, các kết hoạch đang được tiến hành để hỗ trợ AI/ML liên quan đến các trường hợp sử dụng như tính toán phân tán cho cộng tác đám mây biên, khai thác hiệu quả năng lượng tại mạng cũng như tại các UE, quản lý mạng tự quản để làm cho các mạng di động thích ứng, khả định cỡ và chống chịu tốt hơn với các yêu cầu thay đổi. Các trường hợp này có thể được phân chia theo nghĩa rộng thành:

1) **AI cho mạng (AI for Network)**. Bao gồm việc sử dụng AI để tối ưu hóa các giải thuật truyền thông, các chức năng mạng cũng như các hoạt động của mạng để cải thiện hiệu suất và trải nghiệm người dùng từ đầu đến cuối. Chẳng hạn AI/ML có thể được sử dụng để thiết kế một giải thuật quản lý tài nguyên mạng hiệu quả.
2) **Mạng cho AI (Network for AI)**. Tại đây, mạng cần cung cấp các khả năng như truyền thông đảm bảo chất lượng dịch vụ (QoS),các tài nguyên tính toán và lưu trữ để hỗ trợ các ứng dụng AI/ML. Chẳng hạn khi chạy một ứng dụng AI/ML nhận dạng ảnh trên UE, mạng cần cung cấp kết nối trễ thấp đến biên hay đám mây để hỗ trợ xử lý bổ sung.

Để hiểu được quá trình tích hợp AI và mạng truyền thông di đông 5G, trước tiên ta sẽ xét ngắn gọn kiên trúc hệ truyền thông di động 5G.

9.1. KIẾN TRÚC HỆ THỐNG TRUYỀN THÔNG DI ĐỘNG 5G

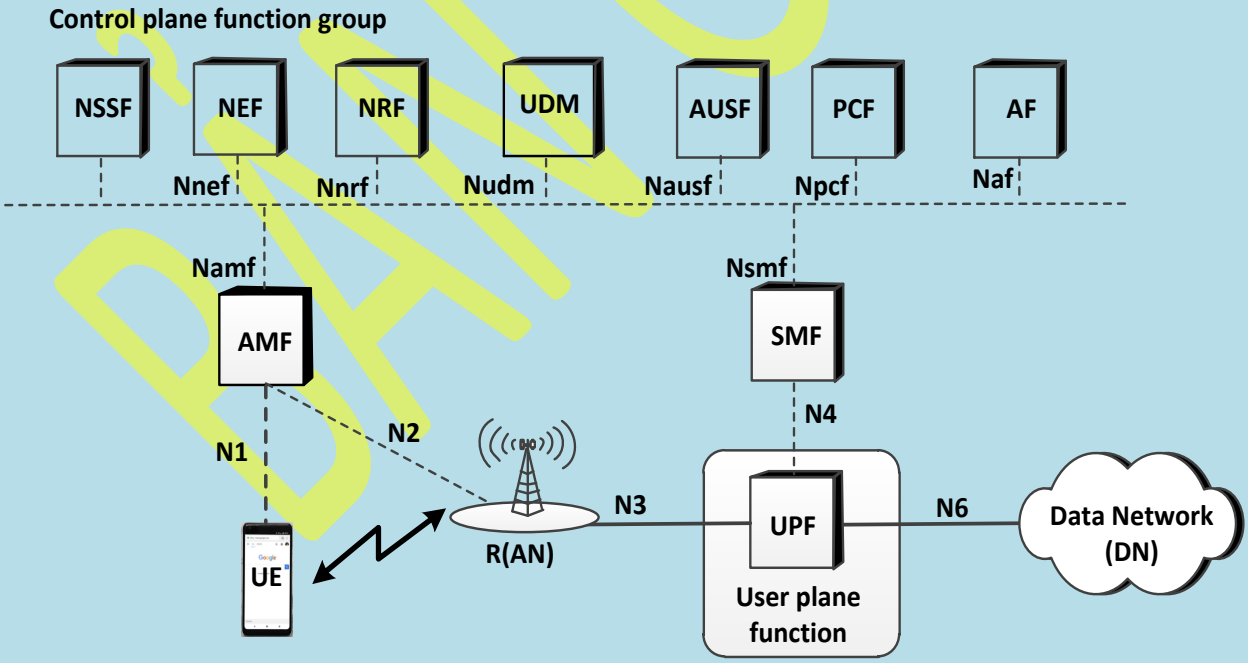
9.1.1. Kiến trúc hệ thống 5G dựa trên dịch vụ

Phần này trình bày tổng quan kiến trúc hệ thống 5G (5GS) không chuyển mạng dựa trên dịch vụ (SBA: Service Based Archtecture) của 3GPP TS 23.501 v15 trong đó các giao diện dựa trên dịch vụ được sử dụng trong mặt phẳng điều khiển.

Kiến trúc 5GS SBA (5G Service Bassed Architecture: kiến trúcs 5G dựa trên dịch vu) dựa trên sự tương tác giưã các phần tử mạng của mặt phẳng CP (Control Plane: mặt phẳng điều khiển). Các khái niệm then chốt của kiến trúc này như sau:

- Phân tách các chức năng mặt phẳng người dùng (UP: User Plane) và mặt phẳng điều khiển (CP: Control Plane) để có thể định cỡ, phát triển và triển khai độc lập, chẳng hạn có thể đặt các chức năng này tập trung hoặc phân bố
- Modul hóa thiết kế chức năng, chẳng hạn hỗ trợ cắt lát mạng linh hoạt và hiệu quả
- Tại nơi có thể áp dụng, có thể định nghĩa các thủ tục (tập các tương tác giữa các chức năng mạng) như là các dịch vụ, vì thế có thể tái sử dụng chúng
- Cho phép từng chức năng mạng tương tác trực tiếp với chức năng mạng khác khi cần thiết. Kiến trúc này không cản trở việc sử dụng một chức năng trung gian để hỗ trợ các bản tin mặt phẳng điều khiển định tuyến.

Hình 9.1 cho thấy kiến trúc 5G dựa trên dịch vụ



Control plane function group: nhóm chức năng mặt phẳng điều khiển

Hình 9.1. Kiến trúc 5GS dựa trên dịch vụ (SBA: Service Based Architecture.

Các chức năng trên hình 9.1 như sau:

(1) AMF (Access and Mobility Management Function: chức năng quản lý truy nhập và di động): bao gồm kết cuối giao diện RAN CP (N2) và giao diện NAS (N1), mật mã và bảo vệ tính toàn vẹn NAS, quản lý đăng ký/kết nối/khả năng đạt đến kết nối/di động, can thiệp theo luật, nhận thực truy nhập và trao quyền, neo an ninh, quản lý ngữ cảnh thông tin.

(2) SMF (Session Management Function: chức năng quản lý phiên): quản lý phiên, ấn định và quản lý địa chỉ IP của UE, chọn/điều khiển các chức năng UP, kết cuối các giao diện tới PCF, áp đặt chính sách và QoS, chức năng chuyển mạng, SMF kết nối đến UDF qua giao diện N4

(3) AUSF (Authentication Server Function: chức năng server nhận thực) cung cấp các chức năng xác thực và trao quyền

(4) NEF (Network Exposure Function: chức năng để lộ mạng): cung cấp các phương tiện để thu thập,lưu trữ và để lộ một cách an toàn các dịch vụ và các khả năng do các chức năng mạng của 3GPP cung cấp (chẳng hạn đến phía thứ ba hay giữa các NF). Đây là một cổng API cho phép các người dùng bên ngoài, các xí nghiệp hay các nhà khai thác khác có khả năng giám sát, cung cấp và áp đặt chính sách ứng dụng cho người dùng bên trong mạng

(5) NRF (Network Repositor Function: chức năng kho chứa của mạng) lưu giữ và cung cấp thông tin về các thể hiện NF khi triển khai/cập nhật/loại bỏ các thể hiện NF để hỗ trợ chức năng phát hiện. Đây là một chức năng mới cung cấp chức năng đăng ký và phát hiện để các phần tử mạng có thể phát hiện lẫn nhau và giao tiếp qua các API.

(6) PCF (Policy Contrrol Function: chức năng điều khiển chính sách) hỗ trợ chương trình khung thống nhất để điều chỉnh các hành vi , cung cấp các quy tắc chính sách đến các chức năng mặt phẳng điều khiển để thực thi

(7) UDM (Unified Data Management: quản lý số liệu thống nhất): hỗ trợ kho chứa chứng nhận xác thực (Authentification) và chức năng xử lý, lưu trữ các chứng nhận an ninh dài hạn và các thông tin đăng ký thuê bao. Giống như HSS (Home Subscriber Server: smays chủ thuê bao nhà) trong 4G, nhưng sử dụng được cho cả truy nhập cố định và di động

(8) AF (Application Function: chức năng ứng dụng) trình bày mọi chức năng CP bổ sung cần thiết cho các lát cắt mạng đặc thù, chủ yếu cung cấp bởi phía thứ ba

(9) UPF (U Plane Function: chức năng mặt phẳng người dùng) thực hiện các tác vụ sau: điểm neo cho di động nội tại hoặc giữa các RAT (công nghệ vô tuyến), điểm kết nối với mạng ngoài của phiên PDU, định tuyến và chuyển phát, xử lý QoS, giám sát gói và áp đặt thực thi quy tắc chính sách, can thiệp theo luật, tính và báo cước.

(10) NSSF (Network Slice Selection Function: chức năng chọn lát cắt mạng) hỗ trợ chọn tập các thể hiện lát cắt mạng để phục vụ UE, quyết định NSSAI (Network Slice Selection Asistance Information: thông tin hỗ trực chọn cắt lát mạng) được phép, quyết định AMF sẽ được sử dụng cho UE

(11) R(AN): (Radio (Access Network): mạng truy nhập (vô tuyến))

(12) UE (User Equipement: thiết bị người dùng)

Ngoài ra hệ thống 5G (5GS) có có các phần tử sau:

(1) NWDAF (Network Data Analytíc Function: chức năng phân tích dữ liệu mạng). NWDAF nết nối với các chức năng mạng khác nhau không được trình bày trên hình 10.1.

(2) SEPP (Security Egge Protection Proxy: đại diện bảo vệ an ninh biên)

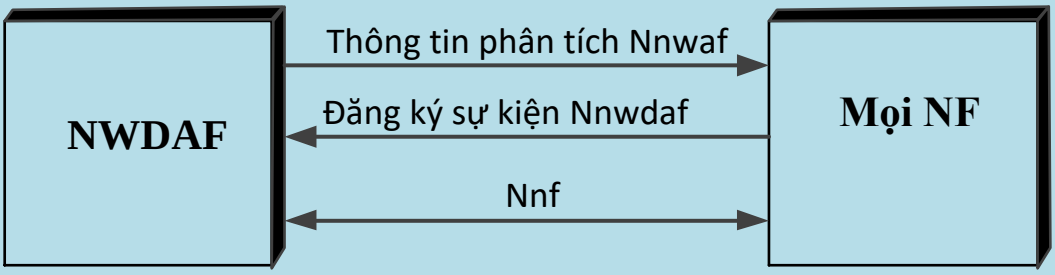
(3) CHF (Charging Function: chức năng tính cước)

(4) OAM (Operation, Administration and Maintenance: khai thác, quản lý và bảo dưỡng)

9.1.2. NWDAF

NWDAF là một chức năng phân tích dữ liệu trong các mạng truyền thông di động 5G, nó cung cấp phân tích mạng dựa trên yêu cầu nhận được từ các chức năng mạng khác. Vì là nguồn dữ liệu, NWDAF có thể sử dụng mọi chức năng mạng khác. Vì thế tồn tại quan hệ hai chiều giữa NWDAF và các chức năng mạng (NF) như mô tả trong hình 9.2.

Lưu ý rằng Nnwwdaf trình bày giao diện dựa trên dịch vụ của NWDAF và Nnf trình mày giao diện dựa trên dịch vụ của mọi NF (chẳng hạn, Npcf trình bày giao diện dựa trên dịch vụ của PCF). Như thấy trên hinhg vẽ, NWDAF có thể cung cấp hoặc số liệu phân tích cho các NF khavs (nghĩa là, thông tin phân tích) hay các NF có thể yêu cầu đăng ký từ NWDAF cho chuyển phát dữ liệu (ví du: đăng ký sự kiện) Ngoài ra, NWDAF lấy dữ liệu từ các NF khác bằng cách sử dung giao diện Nnf.

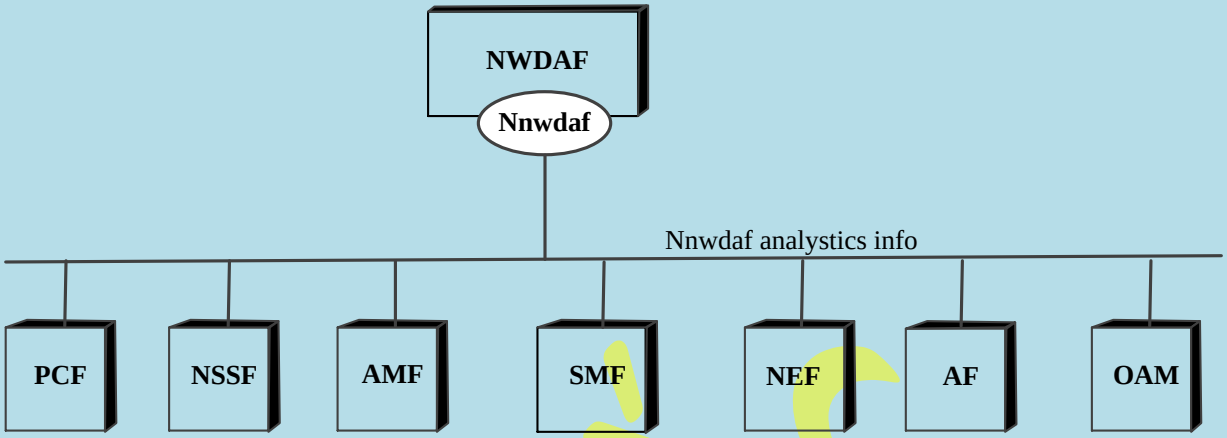


Hình 9.2. Kiến trúc để lộ thu tập dưa liệu và phân tích dữ liệu mạng

Tồn tại hai dịch vụ Nnwdaf (Nnwdaf-EventsSubsription service): đăng ký các sự kiện (nghĩa là, đăng ký phân tích) và thông tin phân tích. Đăng ký các sự kiện Nnwdaf cho phép các phần tử tiêu dùng NF (NF Comsumer) đăng ký hoặc hủy đăng ký các sự kiện phân tích khác nhau và thông báo các phần tử tiêu dùng NF có đăng ký tương ứng về các sự kiện quan sát được. Mặt khác, thông tin phân tích Nnwdaf cho phép các phần tử tiêu dùng NF yêu cầu và nhận một kiểu thông tin sự kiện phân tích khác từ NWDAF.

Dịch vụ đăng ký các sự kiện Nnwdaf là một bộ phận của giao diện dựa trên dịch vụ các sự kiện Nnwwdaf được thể hiện bởi chức năng phân tích dữ liệu mạng. Các phần tử tiêu thụ của dịch vụ này là:

(1) PCF (Policy Contrrol Function: chức năng điều khiển chính sách)
(2) NSSF (Network Slice Selection Function: chức năng chọn lát cắt mạng)
(3) AMF (Access and Mobility Management Function: chức năng quản lý truy nhập và di động)
(4) SMF (Session Management Function: chức năng quản lý phiên)
(5) NEF (Network Exposure Function: chức năng để lộ mạng)
(6) AF (Application Function: chức năng ứng dụng)
(7) OAM (Operation, Administration and Mentenance: khai thác, quả lý và bảo dưỡng)
(8) CEF (Charging Ennablement Function: chức năng hỗ trợ tính cước)
Kiến trúc tham chiếu cho dịch vụ đăng ký các sự kiện Nnwdaf được cho trên hình 9.3.



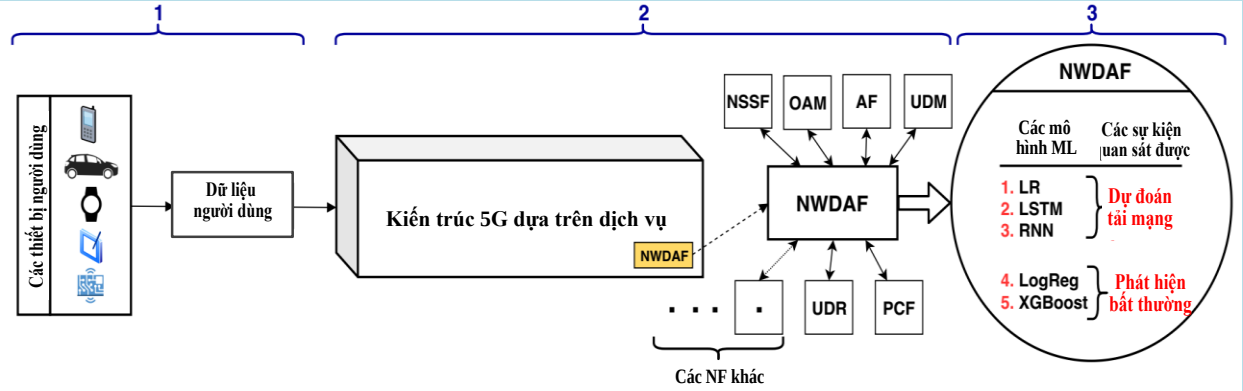
Nnwdaf analystics info: thông tin phân tích từ Nnwdaf

Hình 9.3. Kiến trúc tham chiếu cho dịch vụ đăng ký các sự kiên Nnwdaf

Vì thế có thể coi dịch vụ đăng ký các sự kiện là một dịch vụ nữa của dịch vụ tính cước, trong đó dịch vụ đăng ký các sự kiện là dịch vụ yêu cầu áp dụng các kỹ thuật AI/ML. Với thông tin phân tích Nnwdaf, có thể quan sát được các sự kiện sau:

- Thông tin hành vi bất thường của một nhóm UE hay một UE cụ thể,
- Thông tin hành vi được kỳ vọng của một nhóm UE hay một UE cụ thể,
- Hiệu suất tải mạng trong một vùng được quan tâm
- Mức tải mạng của thực thể lát cắt mạng,
- Thông tin phân tích tải NF cho một NF cụ thể,
- Mẫu truyền thông cho một UE cụ thể,
- Thông tin nghẽn dữ liệu người dùng đối với một vị trí cụ thể,
- Thông tin liên quan tính di động đối với một nhóm UE hay một UE cụ thể
- Thống kê thay đổi QoS và thay đổi QoS tiềm năng trong một vùng nào đó,
- Trải nghiệm dịch vụ đối với một ứng dụng hay đối với một lát cắt mạng.

Hình 9.4 mô tả quy trình mức cao của một hệ thống xử ký ba sự kiện đầu tiên được xét ở trên. Như mô tả trong hình 9.4, dữ liệu nhận được từ UE được chuyển đến 5GSBA, nơi có NWDAF và các phần tử mạng (NF). NWDAF được nối đến các phần tử mạng khác qua giao diện SBI (servive based interface), NWDAFvà các NF trao đổi qua lại dữ liệu giữa chúng. Khi này NWDAF thu thập thông tin từ các NF khác nhau và khớp nối một số mô hình ML phù hợp để dự đoán hiệu suất tải mạng và phát hiện các bất thường tả mạng. Với cấu hình topo cho trước, hệ thống có thể sử dụng các dữ liệu được gán nhãn để đào tạo, vì thế nó chọn mô hình ML đẹp nhất tùy thuộc vào đặc tính của cấu hình topo.



Hình 9.4. Mô tả quy trình mức cao của một hệ thống: (1) số liệu được UE tạo ra và gửi đến 5G SBA, (2) NWDAF nhận thông tin từ thông tin được gửi từ UE và trao đổi với qua lại các NF liên quan, (3) các hàm ML trong NWDAF tạo ra thông tin phân tích mạng.

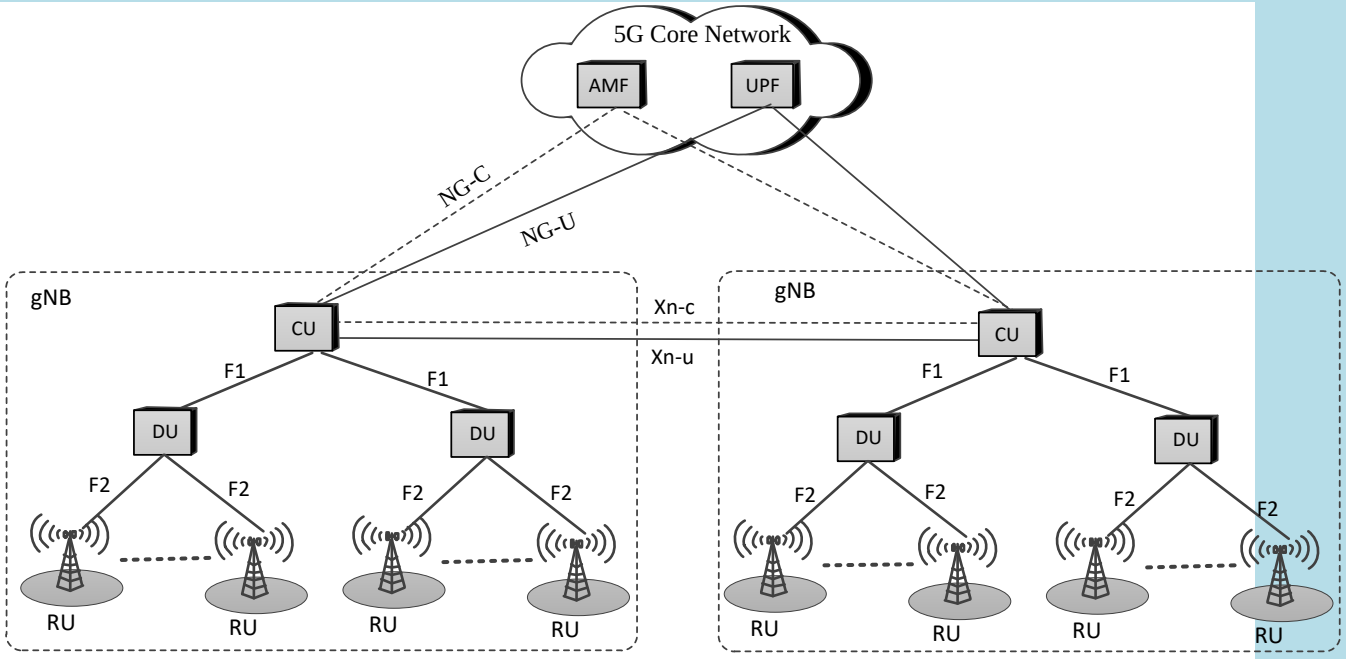
9.1.3. Kiến trúc mạng truy nhập vô tuyến 5G

Hình 9.5 cho thấy kiến trúc truy nhập vô tuyến 5G (5G RAN: 5G Radio Access Network) gồm các gNB cấu thành từ cáci phần tử cơ sở sau:

- CU (Central Unit: khối trung tâm) chịu trách nhiệm xử lý các lớp cao hơn lớp vật lý. Thực hiện CU có thể là phần cứng dành riêng hoặc đám mây vô tuyến. Một mặt CU được nối đến 5GC (5G Core Network: mạng lõi 5G), mặt khác nó được nối đến DU (Distributed Unit: khối phân bố). DU được nối đến các RU (Radio Unit: đơn vị vô tuyến).
- DU xử lý các quá trình quan trọng thời gian khi trễ không được phép quá lớn.
- Các RU được đặt tại trạm ô cùng với anten và khối RF.

Các giao diện sau đây được định nghĩa cho cấu trúc truy nhập vô tuyến 5G:

- Giao diện F1 kết nối CU với một hay nhiều DU. Một CU có thể quản lý nhiều DU
- Giao diện F2 kết nối DU với RU. Một DU có thể kết nối đến nhiều RU.
- Giao diện Xn kết nối các CU với nhau
- Giao diện NG (Next Generation: thế hệ sau) nối CU với 5GC. Giao diện này được chia thành NG-U cho UP (User Plane: mặt phẳng người dùng) và NG-C cho CP (Control Plane: mặt phẳng điều khiển), trong đó NG-U được gọi là N3 và NG-C được gọi là N2.



5G Core Network: mạng lõi 5G
AMF (Access Management Function): chức năng quản lý truy nhập
UPF (User Plane Function): chức năng mặt phẳng người dùng

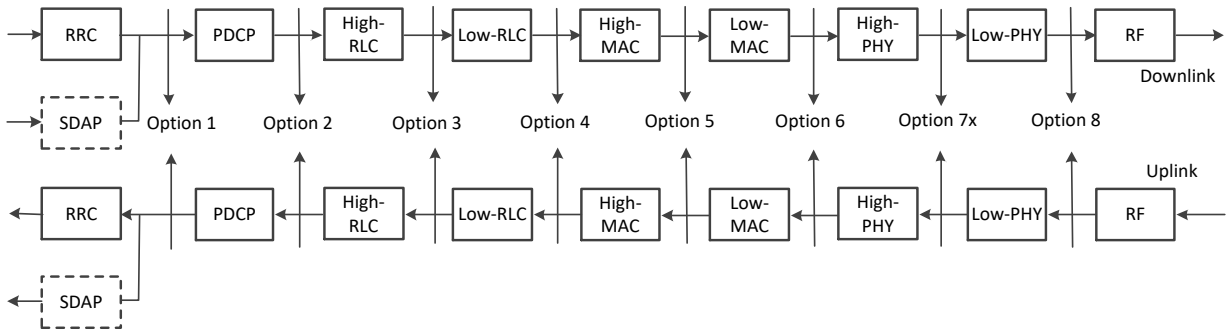
CU (Celtral Unit): đơn vị trung tâm
DU (Distributed Unit): đơn vị phân tán
RU (Radio Unit): đơn vị vô tuyến

NG-C: giao diện mặt phẳng điều khiển thế hệ sau
NG-U: giao diện mặt phẳng người dùng thế hệ sau

Hình 9.5. Phân chia gNB thành gNB CU, DU và RRH

9.1.4. Phân chia các chức năng giữa CU, DU và RU của gNB

Phân chia các chức năng của gNB cho phép đáp ứng tốt nhất các yêu cầu trong các kịch bản triển khai khác nhau. Hình 9.6 tổng kết các tùy chọn phân chia chức năng đối với 5G NR do 3GPP đề xuất.



Option: tùy chọn; High-RLC: RLC cao; Low-RLC: RLC thấp; High-MAC: MAC cao; Low-MAC: MAC thấp; High-Phy: lớp vật lý cao; Low-Phy: lớp vật lý thấp

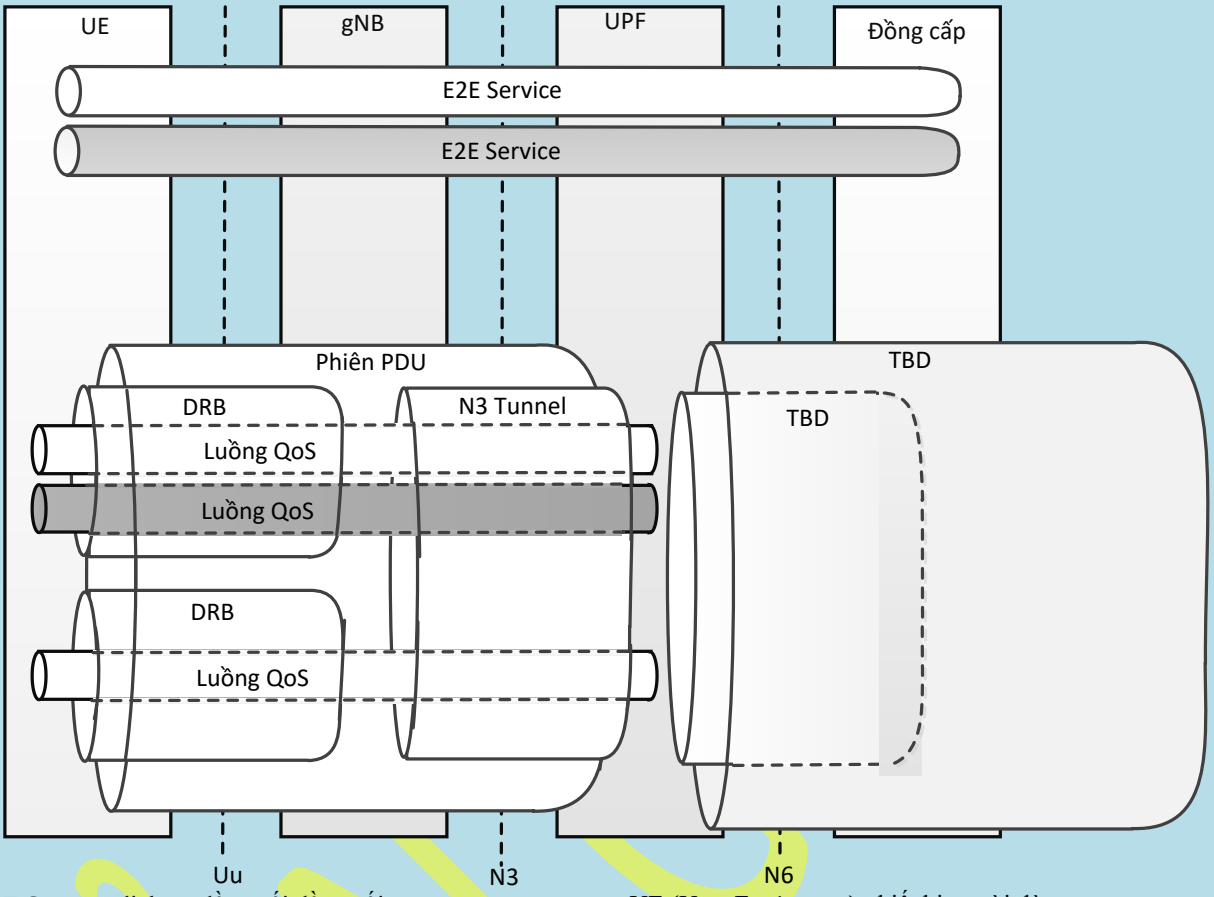
Hình 9.6. Tổng kết các tùy chọn phân chia chức năng đối với 5G NR do 3GPP đề suất.

9.1.5. Kiến trúc QoS trong hệ thống 5G

Hình 9.7 cho thấy kiến trúc QoS trong hệ thống 5G:

- Đối với mỗi UE, 5GC thiết lập một hay nhiều phiên PDU (Protocol Data Unit: đơn vị dữ liệu giao thức);
- Đối với mỗi UE, 5G-RAN thiết lập ít nhất một DRB (Data Radio Bearer: kênh mang vô tuyến dữ liệu) cho phiên PDU. Sau đó có thể các DRB cho một hay nhiều luồng QoS của phiên này được lập cấu hình;
- 5G-RAN ấn định tài nguyên các gói thuộc các phiên PDU khác nhau lên các DBR khác nhau;
- Các bộ lọc gói của NAS (NAS Filter) trong UE và trong 5G CN liên kết các gói đường lên (UL:Uplink) và các gói đường xuống (DL: Dowlink) với các luồng QoS;
- Các quy tắc ấn định tài nguyên mức AS ("AS" Filter) trong UE và 5G RAN liên kết QoS đường lên và đường xuống với các DBR.

BẢN GỐC



E2E Service: dịch vụ đầu cuối đầu cuối
TBD (Transport Block Data): số liệu khối truyền tải
DRB (Data Radio Bearer): kênh mang vô tuyến dữ liệu

UE (User Equipment): thiết bị người dùng
5G RAN (5G Radio Access Network): mạng truy nhập vô tuyến
5G CN (5G Core Network): mạng lõi 5G

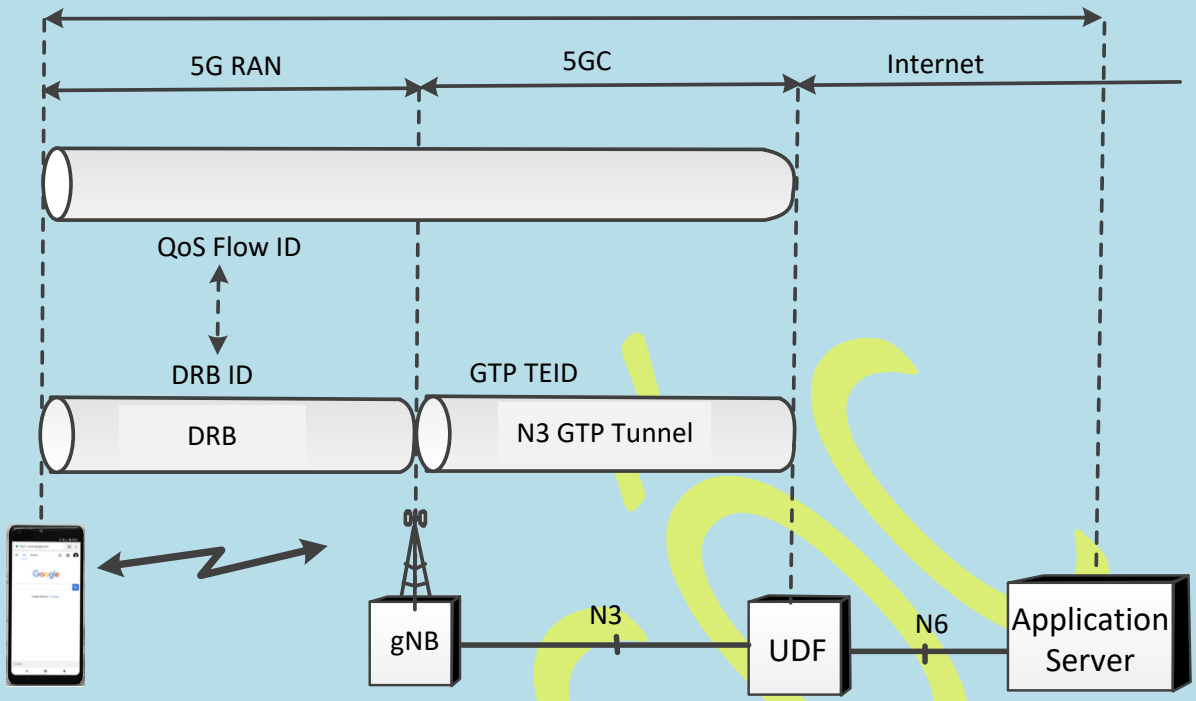
Hình 9.7. Kiến trúc QoS trong hệ thống 5G

5GC và 5G RAN đảm bảo chất lượng dịch vụ (độ tin cậy và trễ yêu cầu) bằng cách ấn định tài nguyên các gói vào các luồng QoS và các DRB tương ứng. Như vậy sẽ có hai bước ấn định tài nguyên các luồng IP lên các luồng QoS (NAS) và từ các luồng QoS lên các DRB.

Trong NR, DRB định nghĩa xử lý gói trên giao diện vô tuyến (Uu). Mỗi một DRB phục vụ các gói được xử lý cùng một chuyển phát. Các DRB khác nhau có thể được thiết lập cho các xử lý chuyển phát khác nhau.

Trên đường xuống, 5G RAN ấn định tài nguyên các luống QoS lên các DRB dựa trên QoS Flow ID (đánh dấu N3) và các hồ sơ QoS tương ứng.

Hình 9.8 cho thấy thí dụ truyền một luồng QoS từ UE đến mọt server ứng dụng đồng cấp.



QoS Flow TI (ID luồng QoS), DRB ID (ID kênh mang vô tuyến), GTP TEID (GPRS Tunnelling Protocol Tunnel Endpoint Identifier): ID điểm cuối đường hầm của giao thức truyền đường hầm GPRS

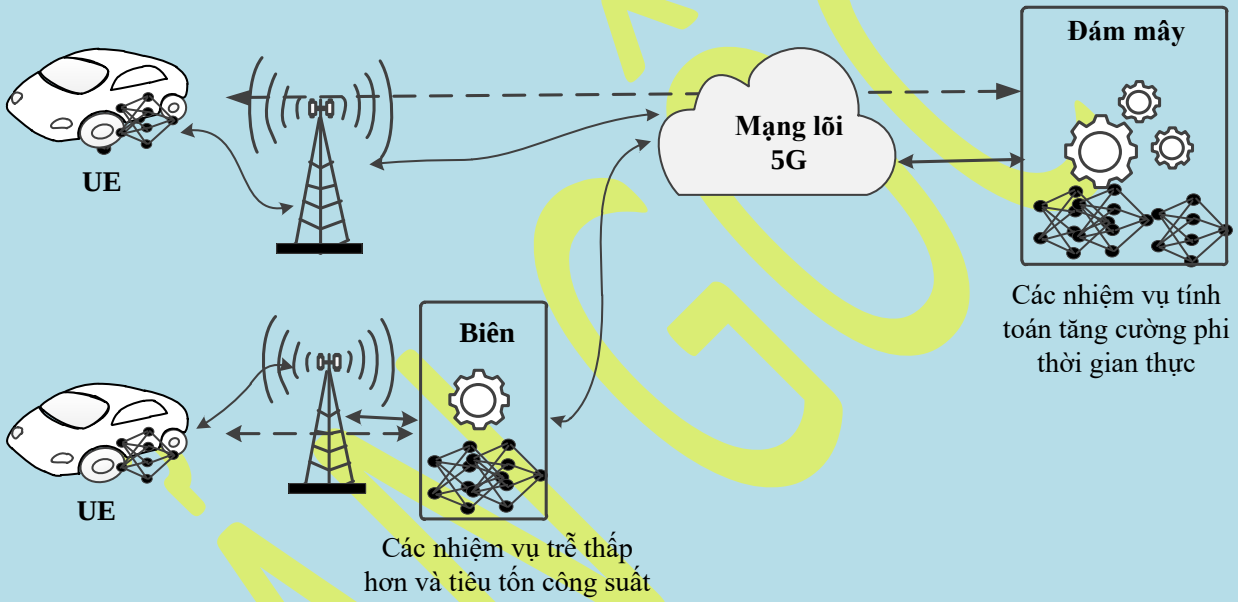
Hình 9.7. Thí dụ truyền một luồng QoS từ UE đến một server ứng dụng

9.2. AI/ML TRONG CÁC MẠNG TRUYỀN THÔNG DI ĐỘNG

AI/ML đóng một vai trò ngày càng tăng trong các mạng truyền thông di động cho phép các mạng này thông minh, hiệu quả và thích ứng hơn và cả trong các ứng dụng của người dùng khác nhau. Vì thế các mạng tương lai cần không phải chỉ hỗ trợ AI/ML cho hoạt động hiệu quả của chính mình và còn phải cung cấp truyền thông, hỗ trợ lưu trữ và tính toán cho các dịch vụ dựa trên AI/ML. Trong vấn đề này, tiêu chẩn hóa hương đến hai khía cạnh sau:

1) **AI là trung tâm (NET for AI: Mạng cho AI).** Khía cạnh này giải quyết vấn đến hỗ trợ hiêu quả các chức năng AI/ML như hỗ trợ học liên kết (FL), trao đổi dữ liệu và mô hình giữa các ứng dụng người dùng và các server đám mây và đảm bảo QoS mong muốn cho các yêu cầu truyền thông như thí dụ minh họa trong hình 9.8. Hình 9.8 cho thấy một xe ô tô tự hành được kết nối đến mạng 5G. Xa ô tô sử dụng các mô hình AI/ML phân cấp. Mô hình ML trong xe quản lý các nhiệm vụ thời gian thực như

phanh gấp và tránh chướng ngại. Đối với các nhiệm vụ phức tạp và tính toán mạnh hơn, các mô hình ML có sẵn tại biên và đám mây. Các mô hình biên (đám mây biên chẳng hạn) được sử dụng cho các nhiệm vụ trễ trung bình (phi thời gian thực, tương ứng) như tối ưu hóa tuyến (hệ thống quản lý pin, tương ứng). Triển khai phân cấp này đảm bảo ấn định nhiệm vụ hiệu quả dựa trên các yêu cầu trễ và tính toán. Các mô hình ML trong ô tô có thể sử dụng dữ liệu từ các bộ cảm biến của nó, mô hình tại biên có thể sử dụng dữ liệu từ cá ô tô ở gần còn các mô hình tại đám mây có thể sử dụng dữ liệu toàn cục.Tuyến dữ liệu trong biên và trong đám mây được thể hiện bằng các đường đứt nét.



Hình 9.8. Thí dụ minh họa ML phân cấp trong các ô tô tự hành được kết nối đến mạng 5G

2) **Mạng là trung tâm (AI for Network: AI cho mạng).** Khía cạnh này xử lý thu thập, lưu trữ và quản lý an ninh dữ liệu từ các phần tử mạng khác nhau và sau đó sử dụng dữ liệu này cho đào tạo các mô hình được tích hợp để cải thiện hiệu suất mạng.

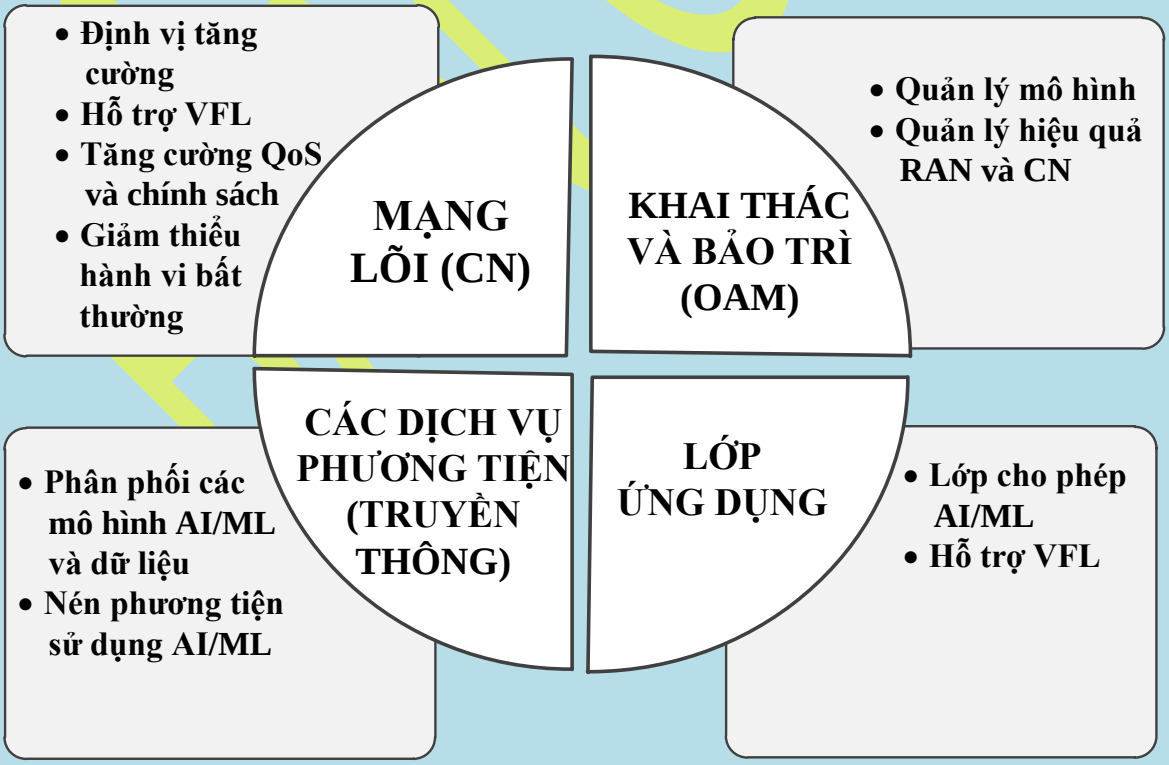
Dưới đây ta xét các thách thức trong cả hai kịch bản nói trên.

1) Tính toán phân cấp. Vì các thiết bị di động thường được treng bị pin với các khả năng tính toán và lưu trữ hạn chế, cxử lý ác nhiệu vụ phức tạp tại chõ có thể không khả thi. Để giải quyết vấn đề này, khối lượng công việc AI/ML có thể được phân chia giữa thiết bị di động và server biên và (hoặc) đám mây. Thách thức ở đây là việc quyết

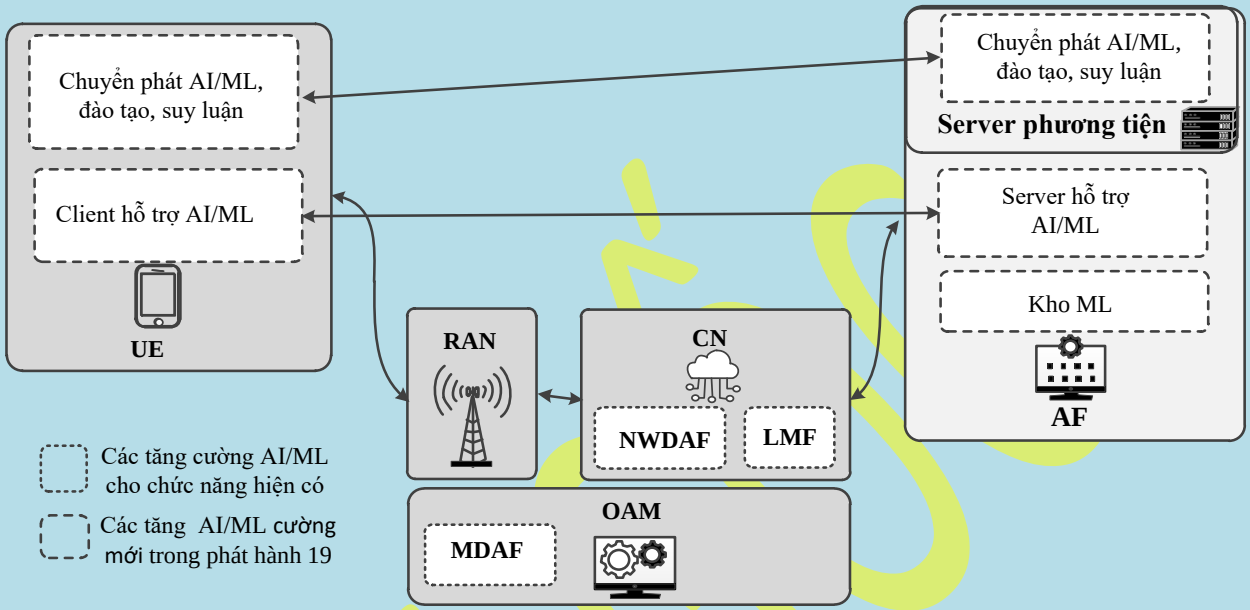
phân chia ML như thế nào, nút nào sử dụng nó, các chỉ số hiệu suất chính (KPI) gì cho mỗi núy, tiêu thụ bao nhiêu năng lượng và v.v…

2) **Phân phối và chia sẻ ML**. Mức độ phức tạp của mô hình MT tăng theo độ sâu của mạng neuron sâu (DNN), vì số lớp nhiều hơn yêu cầu tài nguyên toán nhiều hơn dẫn đến tăng độ phức tạp mô hình. Ngoài ra, hiệu suất mô hình chịu ảnh hưởng của phân phối dữ liệu đào tạo, điều này có thể thay đổi theo thời gian và khác nhau trên các vùng địa lý, nên thường cần có các mô hình tách riêng để duy trì độ chính xác trong từng ngữ cảnh. Việc chọn đúng mô hình và quyết định nút tính toán tối ưu là mộth thức quan trọng. Do lưu trữ hạn chế tại UE, có thể phải tải xuống các mô hình từ biên/đám mây theo yêu cầu. Để chuyển giao mô hình giữa UE và biên /đám mây, cần phải trao đổi một khối lượng lớn thông tin trên mạng. Trao đổi này không chỉ bao hàm thông tin thô mà cả dữ liệu tính toán và các thông số mô hình trên cả đường lên lẫn đường xuống. Để xử lý sự tăng cao trong trao đổi dữ liệu, mạng phải hỗ trợ các các yêu cầu các dịch vụ truyền thông và lưu trữ cao hơn.
3) **Học phân tán/ liên kết**. Học phân tán là phương thức học trong đó đào tạo các mô hình ML được trải rộng trên nhiều nút. FL là một kiểu của học phân tán trong đó nhiều nút (các UE hay các nút biên) hợp tác đào tạo một mô hình dùng chung mà không trao đổi dữ liệu thô. Thay vào đó, mỗi nút đào tạo mô hình tại chỗ bằng cách sử dụng dữ liệu của mình và chỉ chia sẻ các cập nhật mô hình (các gradient hay các trọng số)) với một thực thể trung tâm, thực thể này tổng hợp chúng để cải thiện mô hình toàn cục. Phương phát tổng hợp phổ biến nhất là lấy trung bình lặp. Mỗi nút đào tạo mô hình được tải xuống bằng cách sử dụng dữ liệu tại chỗ và gửi các kết quả tạm thời trên đường lên của 5G. Server tổng hợp các cập nhật này, và gửi các mô hình được cập nhật trở lại các nút này thông qua đường xuống của 5G cho vòng đào tạo tiếp theo. Trễ của quá trình FL được quyết định bởi tốc độ hội tụ (số lần lặp trước khi quá trình hội tụ đến một sự đồng thuận và trễ truyền thông cộng tính toán của mối lần lặp. Trễ và tốc độ dữ liệu trong các chuyển giao này đóng vai trò then chốt trong việc đảm bảo độ chính xác của mô hình.
4) **Các thách thức trong AI cho mạng**. Giống như các thác thức trong "mạng cho AI" được chỉ ra ở trên, cũng có rất nhiều thách thức khi sử dụng công nghệ AI/ML để cải thiện hiệu suất và khai thác mạng. Các kỹ thuật dựa trên AI/ML phụ thuộc và các khối lượng dữ liệu chất lượng lớn để hoạt động hiệu quả. Chẳng hạn, trong lĩnh vực an ninh mạng, các công cụ AI/ML cần nhiều terabyte để xây dựng các mô hình chính xác cho phát hiện các mối đe dọa. Tương tự, cần khối lượng lới dữ liệu để sử dụng các sơ đồ quản lý tài nguyên dựa trên ML để cải thiện hiệu suất mạng. Tuy nhiên dữ liệu này không có sẵn trong miền công cộng vì các ngành công nghiệp không sẵn lòng chia sẻ các nhật ký mạng và các dữ liệu nhạy cảm với các phía thứ ba. Điều này có

thể là một vấn đề quan trọng đối với các nhà mạng đáp ứng nhu cầu của một số lượng nho các thuê bao. Vì thế cần thiết phải thiết lập một tập dữ liệu mạng mở.
Để xử lý vấn đề không có sẵn dữ liệu này, mạng cũng phải cố gắng sử dụng dữ liệu tổng hợp và điều này gây thách thức do các vấn đề thiên lệch phân phối. Các công nghệ AI/ML và cảm biến cũng cần làm việc cùng nhau để cải thiện hiệu suất mạng, chẳng hạn: sử dụng dữ liệu do mạng cảm ứng có thể hỗ trợ các sơ đồ quản lý tài nguyên dựa trên ML. Tuy nhiên, ở đây lại gặp một thách thức là dữ liệu được cảm ứng từ môi trường vô tuyến có thể không phải bao giờ cũng có các nhãn yêu cầu để sử dụng. Đôi khi các mô hình AI/ML có thể tạo ra các cảnh báo giả hay bỏ qua các mối đe dọa nghiêm trọng trong giám sát mạng. Các sơ đồ dựa trên AI/ML cũng cần có công suất xử lý đáng kể dẫn đến tăng các yêu cầu về phần cứng và khả định cỡ.
Để xử lý các thách thức nói trên, 3GPP định nghĩa các hướng dẫn và các yêu cầu tính năng rõ ràng. Các tính năng như vùng phủ, tối ưu hóa dung lượng và quản lý búp sóng được hỗ trợ bởi AI/ML đã được đưa vào RAN cùng với QoS và quản lý các ứng dụng trong CN. Ngoài ra, quản lý mô hình AI/ML để hỗ trợ các hoạt động khai thác và bảo trì cũng đã được đưa vào. Hình 9.9tổng kết các tăng cường AI/ML của phát hành 19 trên các miền khác nhau của hệ thống 5G. Đối với các tăng cường liên quan đến AI/ML trong phát hành 19, hình 9.10 cung cấp tổng quan ~~và bảng 1 liệt kê tất cả~~ các mục công việc và nghiên cứu liên quan.



Hình 9.9. Các phần tử khác nhau trong hệ thống 5G và các tăng cường. Hình tròn trắng cho thấy các phần tử mạng được tăng cường, còn các hộp xám bên cạnh tổng kết các tăng cường.



Hình 9.10. Tổng quan các tăng cường AI/ML trong phát hành 19

Để tạo điều kiện chia sẻ hoặc phân phối mô hình ML và FL, các yêu cầu hệ thống được 3GPP đưa vào để chuyển giao mô hình sử dụng kết nối mạng trong phát hành 18. Các yêu cầu này bao gồm các tốc độ trải nghiệm đường lên và đường xuống cho các kịch bản khác nhau như chuyển giao mô hình phân chia,suy luận phân chia hay FL tương ứng với các dịch vụ như nhận biết ảnh, truyền luồng video, robot và nhận dạng giọng nói. Chẳng hạn, giới hạn trên của đường lên cho suy luận phân chia và tải xuống mô hình được thiết lập là 1,1 Gbit/s cho đường xuống, trong khi trong các vùng nóng giới hạn tại 4 Gbit/s. Với các yêu cầu điểm chuẩn này, mô hình AI/ML có thể tiềm năng ưu tiên và chọn các người sử dụng trong các vùng phủ tốt là ccas thành viên phù hợp cho FL.

Trong phát hành 19, 3GPP đưa vào kết nối trực tiếp thiết bị đến thiết bị (D2D: Device to Device) để hỗ trợ tích hợp AI/ML cho một loạt các ứng dụng bao gồm lái xe, điều khiển từ xa robot và nhận dạng video. Điều này bao gồm cả hỗ trợ chuyển giao mô hình AI/ML trên kết nối trực tiếp thiết bị 5G cho học/ suy luận phânn tán. Chẳng hạn, trong kịch bản suy luận phân tán, một thiết bị có dữ liệu đầu vào giá trị nhưng với công suất/ tài nguyên tính toán thấp, có thể tính toán một số lớp đầu tiên của một mô hình AI và giảm tải dữ liệu trung gian đến một thiết bị khác bằng cách sử dụng kết nối thiết bị trực tiếp cho kết

quả suy luận cuối cùng. Đối với kết nối D2D, 3GPP định nghĩa các chỉ số KPI, các tiêu chuẩn hiệu suất QoS và các yêu cầu hệ thống đặc thù. Chẳng hạn, trong các hoạt động AI/ML phân chia cho suy luận AI/ML giữa các điểm cuối sử dụng các kết nối thiết bị trực tiếp, tốc độ dữ liệu trải nghiệm cực đại giới hạn tại 1,5 Gbit/s và trễ từ đầu đến cuối giới hạn tại 10 mi-li giây cho các ứng dụng robot nhà máy. Một thí dụ của yêu cầu hệ thống là hệ thống 5G phải có khả năng bổ sung hay loại bỏ năng động các UE vào/khỏi nhiệm vụ AI/ML khi truyền thông thông qua kết nối thiết bị trực tiếp.

9.2. CÁC TĂNG CƯỜNG AI/ML CHO MẠNG LÕI 5G

3GPP đưa vào các tăng cường AI/ML cho 5G CN ngay từ phát hành 17 với một số hỗ trợ chức năng ML trong NWDAF (Network Data Analytics Function). Trong phát hành 18, hệ thống 5G hỗ trợ phân phối, chuyển giao và đào tạo mô hình ML cho các ứng dụng như nhận biết vdeo/ giojng nói , điều khiển robot, ô tô v.v… Tập trung chính trong phát hành 19 là hỗ trợ cá tính năng sau:

- Tăng cường các dịch vụ định vị sử dụng AI/ML
- Cho phép hệ thống 5G hỗ trợ các hoạt động AI/ML với sự hỗ trợ của NWDAF cho VFL
- Cho phép hệ thống 5G sử dụng AI/ML để điều khiên chính sách cùng với cải thiện QoS.

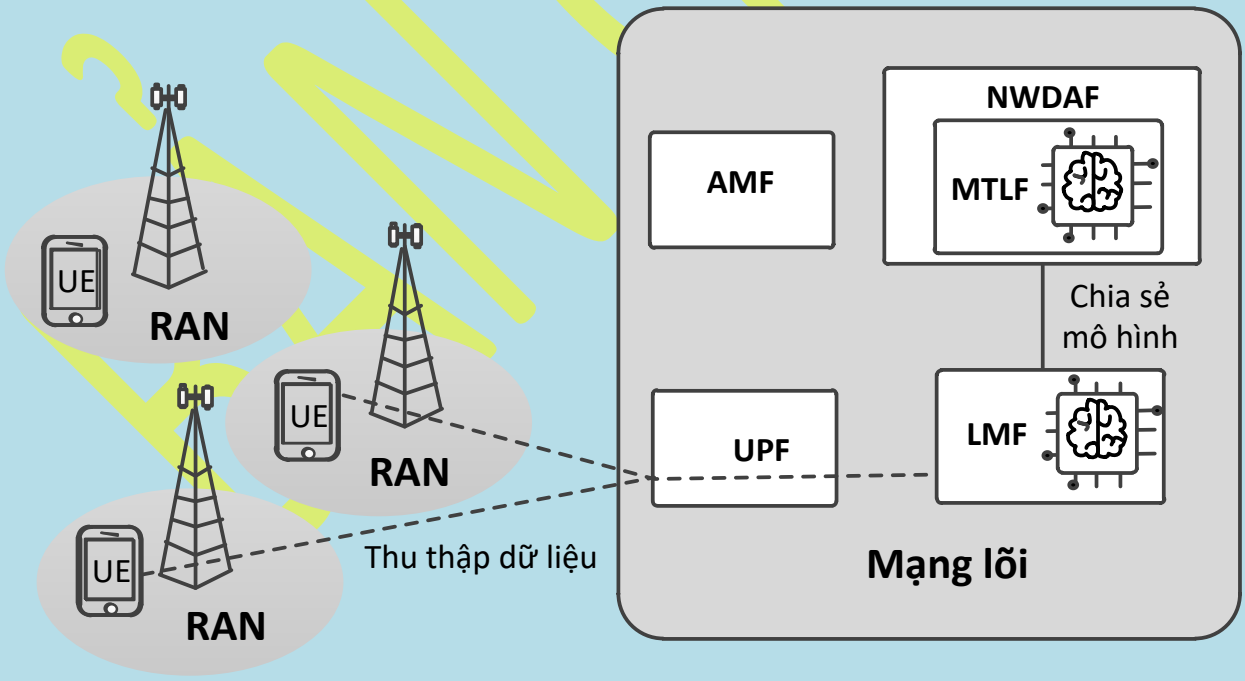
NWDAF là một NF trong mạng lõi chủ yếu tham gia hỗ trợ tất cả các ninh năng nói trên. Các chức năng của nó bao gồm thu thập dữ liệu từ các NF, AF và OAM, để lộ các phân tích dữ liệu, đào tạo mô hình ML, hỗ trợ FL chiều ngang (HFL: Horizonal FL) và FL chiều đứng (VFL: Vertical FL). NWDAF cấu trúc từ hai hai chức năng, chức năng logic đào tạo mô hình (MTLF: Model Traing Logical Function) và chức năng logic phân tích (AnLF: Analytics Logical Function). MTLF chủ yếu đào tạo các mô hình ML còn AnLF sử dụng các mô hình được đào tạo cho suy luận và chiết suất dữ liệu phân tích. AnLF cũng tạo điều kiện cho để lộ dữ liệu phân tích. MTLF và AnLF có thể là bộ phận của cùng một NWDAF hoặc được triển khai trong các thực thể NWDAF khác nhau.MTLF được đưa vào phát hành 17, nó được tăng cường để hỗ trợ tất cả các tính năng nói trên trong phát hành 19.

Dưới đây ta sẽ xét chi tiết các tính năng được liệt kê ở trên.

9.2.1. Hỗ trợ các dịch vụ định vị

Chức năng quản lý vị trí (LMF: Loacation Management Function) trong mạng lõi chịu trách nhiệm thu thập dữ liệu từ UE/RAN, tính toán thông tin vị trí và quản lý

tổng thể các dịch vụ vị trí bao gồm cả cung cấp dịch vụ cho người tiêu thụ. Các chức năng mạng (NF) trong mạng lõi (CN) hay các ứng dụng (AF) có thể yêu cầu thông tin vị trí của UE. Các phương pháp định vị dựa trên AI/ML được được đưa vào RAN để cải thiện độ chính xác của định vị trong phát hành 18. TRong phát hành 19, các phương pháp này được tích hợp vào kiến trúc CN tại LMF và NWDAF. Trong các phương pháp này, mô hình ML có thể được phân tán tại UE hay nút RAN hay nút LMF hay NWDAF. Hình 9.11 cung cấp tổng quan định vị được hỗ trợ bởi UE và RAN với mô hình ML tại LMF. Kiến trúc các dịch vụ định vị được định nghĩa trong 3GPP TS 23.273. LMF thu thập dữ liệu đầu vào từ UE hoặc RAN để thực hiện tính toán vị trí. Sử dụng dữ liệu này mô hình ML có thể được đào tạo tại chỗ tại LMF hay LMF có thể yêu cầu NWDAF đào tạo mô hình ML. Tương ứng, việc giám sát hiệu suất cho phương pháp này có thể được thực hiện hoặc tại LMF hoặc tại NWDAF. LMF sau đó tính toán thông tin vị trí dựa trên dữ liệu đo bằng cách sử dụng mô hình ML. Dựa trên kết quả này, LMF có thể thay đổi phương pháp định vị hoặc đào tạo lại mô hình. Bảo vệ an ninh và quyền riêng tư, như trao quyền (Authorization) LMF trước khi nó có thể thu nhận một mô hình ML từ NWDAF và sự chấp thuận của người dùng cho thu thập dữ liệu từ UE, phải được xem xét. Các khía cạnh an ninh liên quan đến vấn đề này được đề cập trong3GPP TS 33.501.



Hình 9.11. Định vị AI/ML dựa trên LMF. LMF rút ra vị trí của UE dựa trên các đầu vào từ UE và RAN. LMF có thể lấy các mô hình ML từ MTLF của MWDAF.

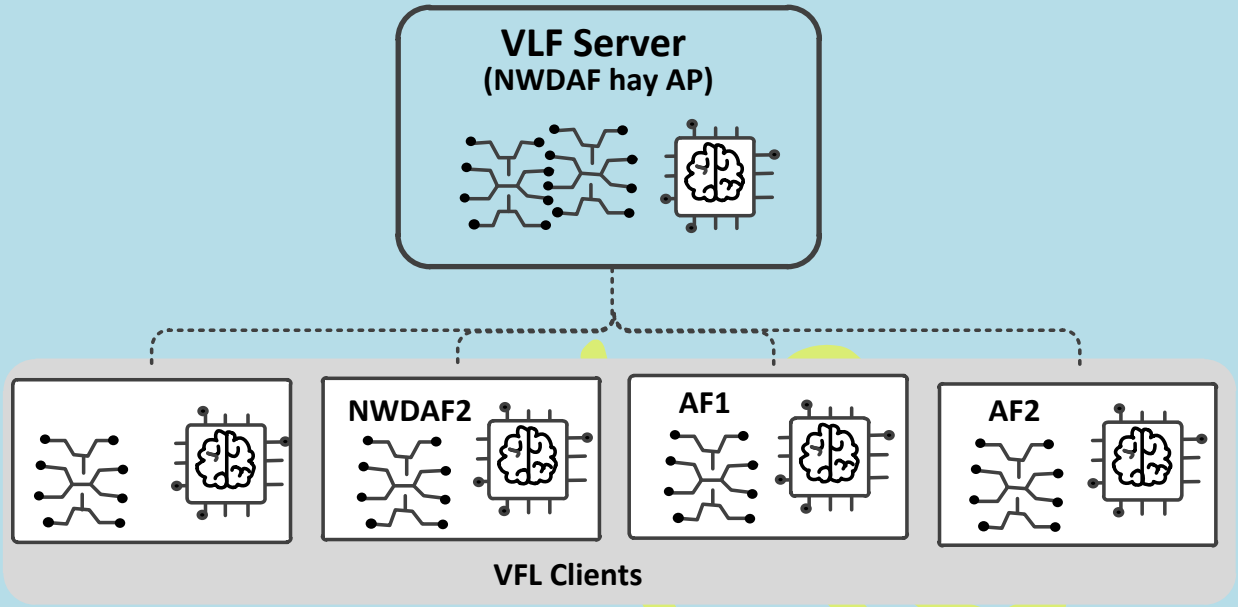
9.2.2. Hỗ trợ học liên kết chiều đứng (VFL)

Học máy liên kết sử dụng nhiều thực thể để kết hợp đào tạo một mô hình ML. Học máy liên kết chiều đứng (VFL: Vertical Federated Learning) là một loại học máy liên kết trong đó nhiều thực thể cộng tác đào tạo một mô hình ML mà không để lộ tập dữ liệu gốc được sử dụng bởi từng thực thể cho đào tạo mô hình. Do đó, nó ẩn tang duy trì quyền riêng tư của dữ liệu thô. Đây là một tính năng quan trọng đối với các mạng di động để bảo vệ quyền riêng tư của người dùng trong khi vẫn hỗ trợ các hoạt đông AI/ML với các thực thể bên ngoài như các AF và các nhà mạng khác. Hỗ trợ VFL trong 5G CN vho phép sử dụng đa dạng các phân tích để đào tạo các mô hình ML mà vẫn duy trì quyền riêng tư của người dùng.

NWDAF và AF có thể cùng nhau đào tạo để phát triển một mô hình (xem hình 9.12). Một thực thể (hoặc NWDAF hoặc AF) có thể hoạt động như là một VFL server và các thực thể khác (có thể bao gồm nhiều NWDAF hay nhiều AF) có thể hoạt động như là các client để đào tạo mô hình ML. VFL server yêu cầu các VFL client đào tạo cục bộ (đào tào tại chỗ) bằng các tập dữ liệu mà chúng có và chia sẻ các kết quả trung gian. Sau đó VFL server kết hợp các mô hình này để rút ra một mô hình ML cuối cùng. Nếu AF là một AF không tin tưởng (AF không do nhà mạng làm chủ), nó có thể thông qua chức năng để lộ mạng (NEF: Network Exposure Function) để hỗ trợ các tương tác VFL. Các luồng thông tin chi tiết cho đào tạo và suy luận VFL được định nghĩa trong 3GPP TS 23.288.

Trong 3GPP TS 33.501, các luồng thông tin chi tiết được mô tả đề cập đến trao quyền (Authorization) NWDAF hay AF nhận vai trò FFL server. Trao đổi thông tin trong quá trinh VFL được bảo vệ bởi nan ninh lớp vận tải (TLS: Transport Layer Security).

BAN HOC



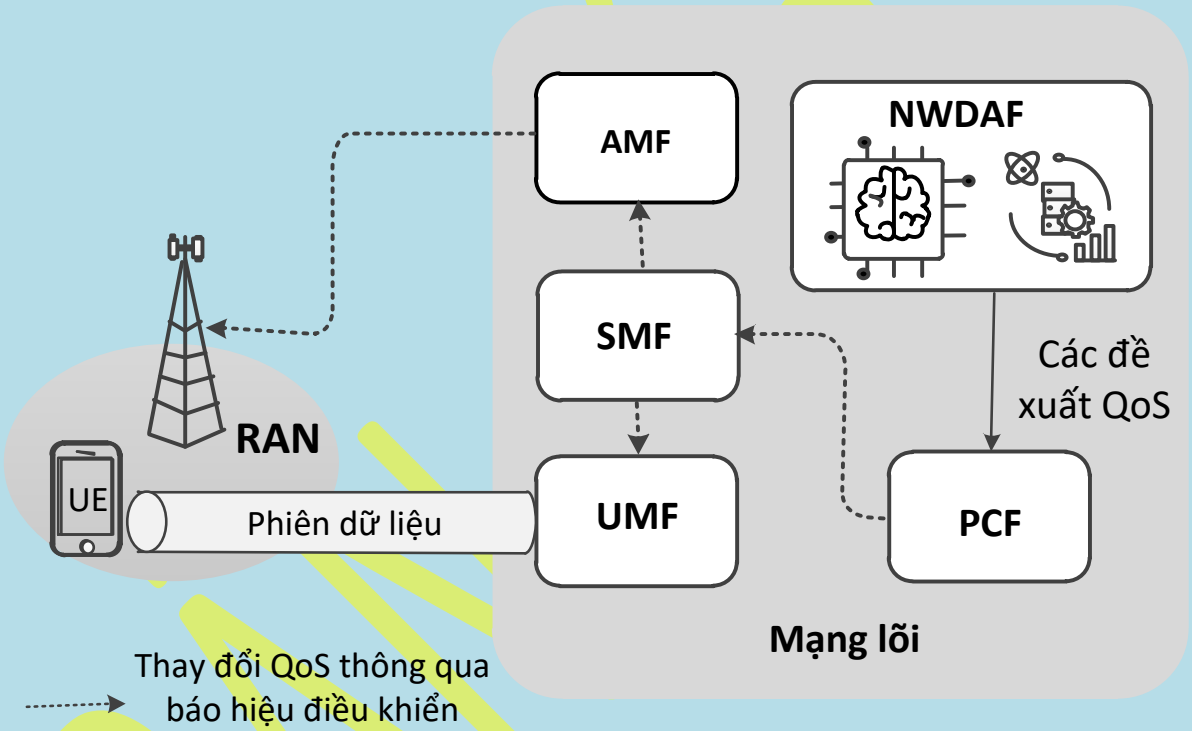
Hình 9.12. Hỗ trợ học liên kết chiều thẳng đứng (VLF). NWDAF hoặc AF có thể là VFL server hoặc là các VFL client. VLF server tạo ra mô hình cuối cùng dựa trên các đầu vào từ các VLF client.

9.2.3. Các tăng cường điều khiển chính sách và QoS sử dụng NWDAF

Chức năng điều khiển chính sách (PCF: Policy Control Function) trong CN quản lý khung điều khiển chính sách và tính cước. Các hoạt động của nó bao gồm quản lý truy nhập và di động của UE liên quan đến chính sách, thông tin chính sách UE và các chính sách phiên dữ liệu. Điều khiển QoS là một phần của quản lý phiên dữ liệu. Cho đến phát hành 18, PCF được xác định bởi các tham số QoS cho từng dữ liệu cá thể bằng cách xem xét các yêu cầu dịch vụ được cung cấp bởi các AF. Các giá trị này được áp dụng cho phiên dữ liệu (luồng dịch vụ) thông qua chức năng quản lý phiên (SMF: Session Management Function). Vì vậy, PCF sẽ kiểm tra xem các tham số được áp dụng có thỏa mãn các yêu cầu dịch vụ hay không. Nếu không đáp ứng, nó sẽ cập nhật các tham số QoS và gửi chúng đến SMF để áp dụng lại các giá trị được thay đổi. PMF sau đó sẽ kiểm tra lại các phân tích dịch vụ mới để quyết định xem các thay đổi QoS có giúp được không. Sự kiểm tra lại và áp dụng này sẽ xảy ra nhiều lần trước khi các yêu cầu dịch vụ được yêu cầu có thể được đáp ứng.

Trong phát hình 19, mục đích là đơn giản hóa quá trình này bằng cách sử dụng các phương pháp AI/ML, như thấy trong hình 9.13, hỗ trợ điều khiển chính sách và quản lý QoS dựa trên ML được đưa vào. NWDAF thu thập các phân tích liên quan đến chính

sách từ các thực thể AF, các NF và OAM để thực hiện các dự đoán. PCF có thể sử dụng tính năng hỗ trợ chính sách của NWDAF để cung cấp các tham số QoS và yêu cầu chất lượng dịch vụ trải nghiệm (QoE: Quality of Experience). Đáp lại, NWDAF cung cấp các tham số QoS có thể có và QoE dự đoán tương ứng của chúng. PCF sử dụng các chế độ này để thực hiện các điều chỉnh và đạt được một QoE tốt hơn. QoS và hỗ trợ chính sách này có thể được cung cấp cho một UE hoặc nhiều UE. QoE có thể được yêu cầu cho một ứng dụng cụ thể hay nhiều ứng dụng, nó có thể liên kết với các luồng của một hay nhiều phiên dữ liệu trong một UE cụ thể hay nhiều UE.

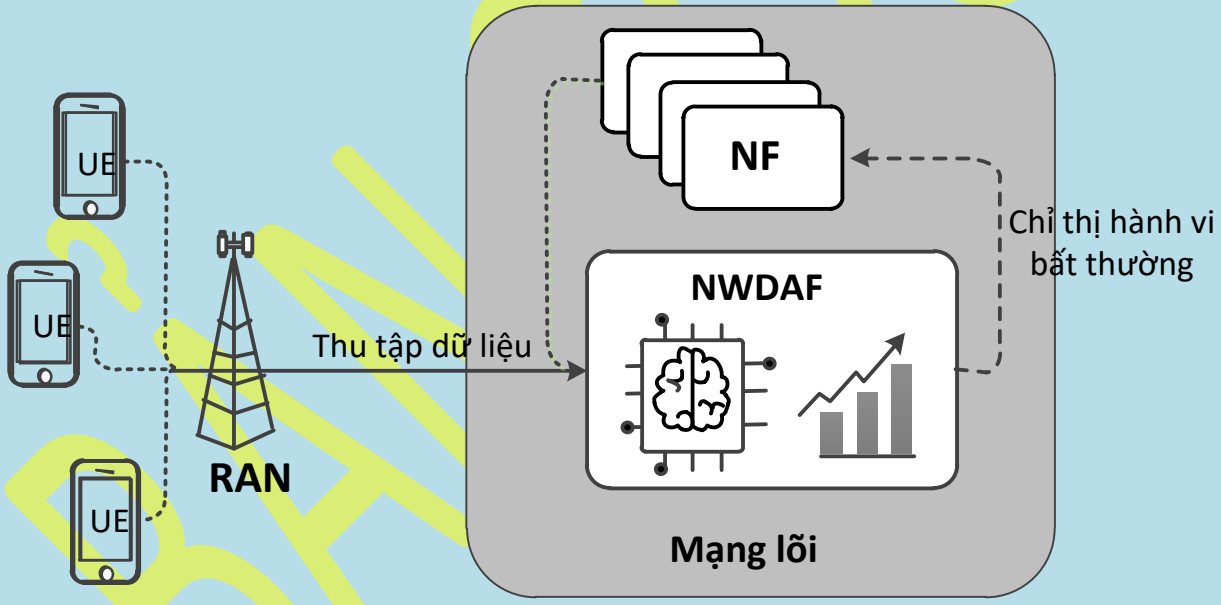


Hình 9.13. Các tăng cường chính sách QoS dựa trên AI/ML. NWDAF cung cấp các thông số QoS được dự đoán cho PCF, QoS được áp dụng lên UPF thông qua SMF và lên RAN thông qua AMF cho phiên dữ liệu.

9.2.4. Giảm thiểu và ngăn ngừa hành vi bất thường khi sử dụng NWDAF

Hanh vi bất thường trong mạng 5G ở đây đề cập đến các tăng bất ngờ trong khối lượng báo hiệu (cũng còn để cập đến là bão báo hiệu). NWDAF đã luôn sẵn sàng cung cấp các phân tích về thông tin tải NF, các nghi ngờ tấn công DDOS (Distributed Denial of Service: từ chối dịch vụ phân tán). Tuy nhiên báo hiệu thái quá có thể là do các thiết bị internet vạn vật băng hẹp (NB-IoT: Narrowband Internet of Things) chưa được nghiên cứu trước đây. Kết hợp tất cả các tính năng này, tập trung của phát hành 19 là phát triển các cơ chế để ngăn chặn bão báo hiệu với sự hỗ trợ của phân tích của NWDAF.

Như được thể hiện trong hình 9.14, NWDAF biết được hành vi bất thường trong mạng bằng cách sử dụng một mô hình ML được đào tạo. Nó giám sát mạng cho các nguồn gốc bất thường trong các mức báo hiệu để nhận dạng và biết được bão báo hiệu. Nó nhận dạng các nguồn (các NF và hay các UE) của báo hiệu thái quá. NWDAF giám sát các ý đồ nhắn tin thành công và thất bại của các UE, từ các NF, tải tại các NFvà csc dữ liệu lên quan khác để nhận dạng nguồn gốc của bão báo hiệu.Dựa trên các phân tích của mình, nó có thể báo cáo về các NF bị ảnh hưởng và nguyên nhân có thể đối với báo hiệu bất thường và nguồn của nó (các UE hay các NF). NWDAF cung cấp các phân tích bão báo hiệu này cho các NF khác, như AMF (Access and Mobility Management Function: chức năng quản lý truy nhập và di động), SMF (Session Managemet Function: chức năng quản lý phiên), UDM (Unified Data Management: quản lý dữ liệu thống nhất), NRF (Network Repository Function: chức năng kho mạng), PCF, OAM và AF. NWDAF cũng có thể báo cáo một bão báo hiệu được dự đoán với một thời gian hiệu lực. Việc giảm thểu vấn đề này sẽ dựa trên chính sách của nhà mạng hay cấu hình mạng.



Hình 9.14. Giảm thiểu hành vi bất thường sử ụng AI/ML. NWDAF dự ddoensd các đột biến báo hiệu dựa trên các đầu vào từ UE, RAN và các NF. Các dự đoán sẽ được chia sẻ với các NF để giảm thiểu.

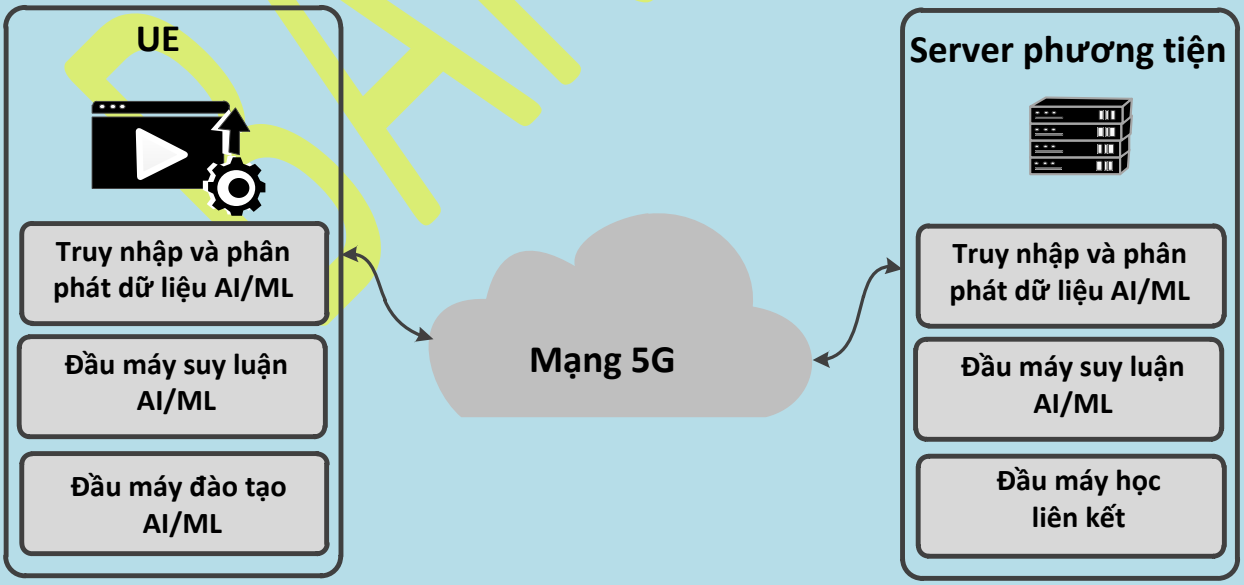
Hỗ trợ các dịch vụ định vị và hỗ trợ VFL, cả hai điều chỉ ra cáchthức có thể hỗ trợ hiệu quả các ứng dụng AI/ML thông qua mạng khi truyền thông qua mạng, đây là trường hợp sử dụng “mạng cho AI” (Network for AI). Trái lại, giảm thiểu bão báo hiệu và xử lý điều khiển chính sách có thể được xử lý hiệu quả sử dụng các dự đoán của mô hình ML. Các tính năng này chỉ ra cách thức có thể sử dụng AI trong mạng để

cải thiện quản lý nó và hiệu suất của nó, đây là trường hợp sử dụng "AI cho mạng" (AI for Network).

9.3. CÁC TĂNG CƯỜNG AI CHO CÁC DỊCH TIỆN PHƯƠNG TIỆN (MEDIA SERVICE), LỚP ỨNG DỤNG , CÁC HOẠT ĐỘNG VÀ QUẢN LÝQUẢN LÝ

9.3.1. Các tăng cường AI cho các dịch vụ phương tiện

Trong phát hành 19, 3GPP đưa vào các mô hình AI/ML để hỗ trợ hiệu quả một dải rộng các ứng dụng xử lý phương tiện, trải rộng từ các trường hợp sử dụng phi thời gian thực như nhận biết hình ảnh, giọng nói/khuôn mặt đến các ứng dụng thời gian thực như phiên dịch trực tiếp, giọng nói-văn bản, phát hiện đồ vật và tăng cường chất lượng truyền luồng video. Cách thức các mô hình và dữ liệu có thể được phân phối trên hệ thống 5G và được phân chia giữa các điểm cuối khác nhau đã được nghiên cứu. Để cho phép xử lý phương tiện dựa trên AI/ML, ba kịch bản phân chia đã được xem xét: suy luận AI tại UE, suy luận AI tại server phương tiện và phân chia suy luận giữa UE và server phương tiện. AI/ML cũng đang được sử dụng để tối ưu hoá các giải thuật nén phương tiện và tăng cường truyền dẫn, triển khai các mô hình AI/ML và dữ liệu. Điều này bao gồm cả nghiên cứu các kỹ thuật nén cho cả các mô hình và dữ liệu. Để hỗ trợ vấn đề này, các chức năng logic mới, như đầu máy suy luận AI/ML, phân phát/truy nhập dữ liệu AI/ML và đầu máy FL, đã được đưa vào cả ở UE và server phương tiện. Hình 9.15 mô tả các chức năng logic mới này được đưa vào để tăng cường các dịch vụ phương tiện. Các kết quả của nghiên cứu này được kỳ vọng dẫn đến công việc tạo ra tiêu chuẩn trong phát hành 20. Tính năng này phù hợp với mô hình " mạng cho AI" trong hệ thống 5G.



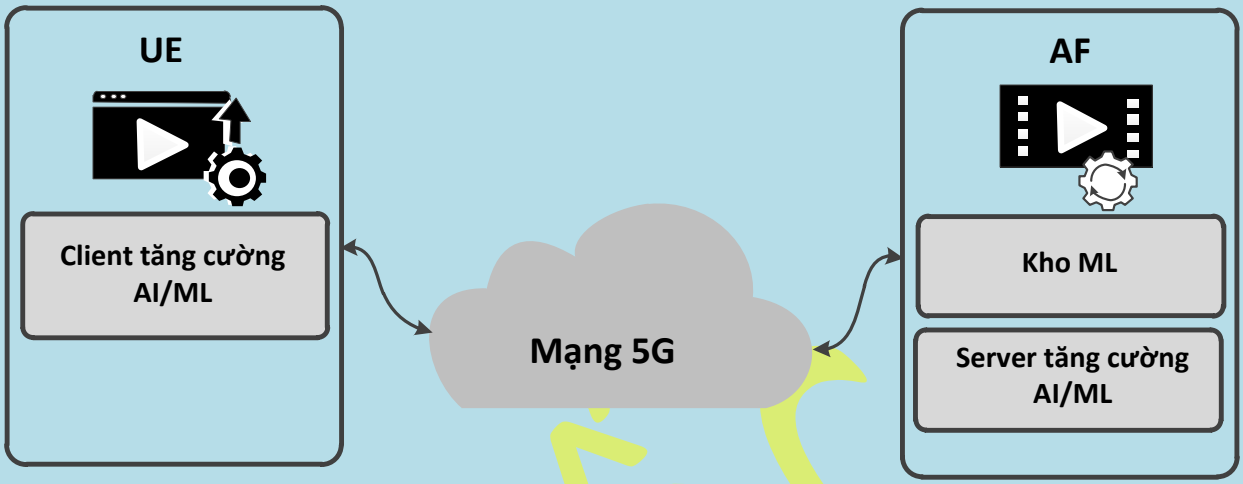
Hình 9.15. Các tăng cường AI/ML cho các dịch vụ phương tiện. Các hộp xám cho thấy các tăng cường AI/ML.

9.3.2. Các tăng cường AI/ML cho lớp ứng dụng

3GPP đưa vào khung phân tích dữ liệu lớp ứng dụng trong phát hành 18 để cung cấp để lộ và các dịch vụ giá trị gia tăng chung cho các nhà cung cấp dịch vụ cho ngành dọc (các ngành hàng , lĩnh vực chuyên biệt hoặc phân khúc thị trường cụ thể) và các nhà cung cấp dịch vụ ứng dụng (ASP: Application Service Provider) tại lớp ứng dụng. Khung hỗ trợ các phân tích dữ liệu liên quan đế hiệu suất ứng dụng từ đầu đến cuối, tải server biên, các API dịch vụ, độ chính xác định vị, hiệu suất cắt lát mạng và phân tích. Trong phát hành 19, các tăng cường bổ sung được đề suất để củng cố hỗ trợ phân tích dữ liệu thông qua các kỹ thuật AI/ML, cùng với các mở rộng để cho phép truyền thông UE đến tại lớp ứng dụng. Một lớp cho phép AI/ML được đưa vào tại lớp ứng dụng để tạo điều kiện cho các hoạt động AI/ML cho một hai nhiều ứng dụng dọc (Vertical Application), hoặc tại UE và (hoặc) tại AF. Logic hoạt động AI/ML lớp ứng dụng được điều khiển bởi một AF. Lớp cho phép AI/ML có thể được triển khai trong các cấu hình khác nhau: trung tâm tại một nền tảng đám mây đóng vai trò tổng hợp dữ liệu từ nhiều nền tảng biên, tại chỗ tại nền tảng biên hay theo cách phân cấp trên cả các nền tảng biên và đám mây. Triển khai phân cấp này đặc biệt có lợi cho các trường hợp có sự tham gia của FL, học phân tán hay suy luận phân tán.

Hình 9.16 mô tả các tăng cường cho lớp ứng dụng. Server cho phép AI/ML tại AF quản lý hành vi AI/ML của các ứng dụng dọc và hỗ trợ các nhiệm vụ liên quan đến chuyển phát và phân phối các mô hình AI/ML và dữ liệu tại lớp cho phép AI/ML. Ngoài ra một kho ML mới được đưa vào tại lớp] ứng dụng này để hỗ trợ lưu trữ và thu nhận các dịch vụ AI/ML, bao gồm mô hình ML và đăng ký/ các cập nhật thành viên cho trác trường hợp sử dụng đứng (Vertical Use Case). Việc đưa vào lớp này đem đến sự hỗ trợ cho các chức năng AI/ML khác nhau, bao gồm quản lý vòng đời, đào tạo và phân phối mô hình, cập nhật thành viên FL, quản lý hoạt động AI/ML, chọn thành viên VFL, triển khai học chuyển giao cho HFL (Horizontal Federated Learning: học liên kết theo chiều ngang) và phân tích tiên tiến để cải thiện hiệu suất năng lượng trong mạng dữ liệu.

Tính năng này phù hợp với “mạng cho AI” trong hệ thống 5G.



Hình 9.16. Các tăng cường AI/ML cho lớp ứng dụng trong phát hành 19. Các hộp xám cho thấy các tăng cường AI/ML.

9.3.3. Các tăng cường AI/ML cho các hoạt động và quản lý

Hệ thống OAM chịu trách nhiệm cho quản lý, lập cấu hình, tối ưu hóa và duy trì các tài nguyên mạng và các dịch vụ trong mạng di động. Để triển khai hiệu quả các khả năng AI/ML trong hệ thống 5G, điều tối quan trọng là quản lý các mô hình ML và các chức năng suy luận AI/ML theo các đặc trưng và các yêu cầu đặc thù của các trường hợp sử dụng mà chúng hỗ trợ. Các khả năng AI/ML đã được đưa vào khung MDA (Management Data Analyrics: phân tích dữ liệu quản lý) trong phát hành 18. Các khả năng quản lý AI/ML bao gồm quản lý và các hoạt động cho đào tạo ML, kiểm tra ML, phỏng tạo AI/ML, triển khai thực thể ML suy luận AI/ML. Trong phát hành 19, khung MDA đã được mở rộng để hỗ trợ các khả năng quản lý AI/ML cho các kỹ thuật học tiên tiến như học chuyển giao, FL, học tăng cường và học phân tán.

Chưc năng MDA (MDA Function) cung cấp dịch vụ MDA cho phép mọi người tiêu thụ có cấp quyền yêu cầu và nhận được các phân tích. Mục đích chính của các tăng cường này là cho phép quản lý hiệu quả các khả năng của AI/ML trong RAN và trong mạng lõi 5G. Thí dụ cho các khả năng này là:

- **Tối ưu hóa phủ sóng và dung lượng dựa trên AI/ML trong RAN**. Tính năng này phát hiện và giải quyết các vấn đề liên quan đến vùng phủ trong RAN. Để hỗ trợ tính năng này, MDAF cung cấp các mô hình ML được đào tạo cho gNB và gNB sử dụng nó cho suy luận.
- **Đào tạo ML và suy luận AI/ML cho mạng lõi 5G**. NWDAF hỗ trợ chức năng hoạt động ML, nó có thể nhận các mô hình ML đã đào tạo từ MDAF. Hỗ trợ tính năng suy luận AI/ML có thể được cung cấp bởi MDAF dựa trên lập cấu hình.

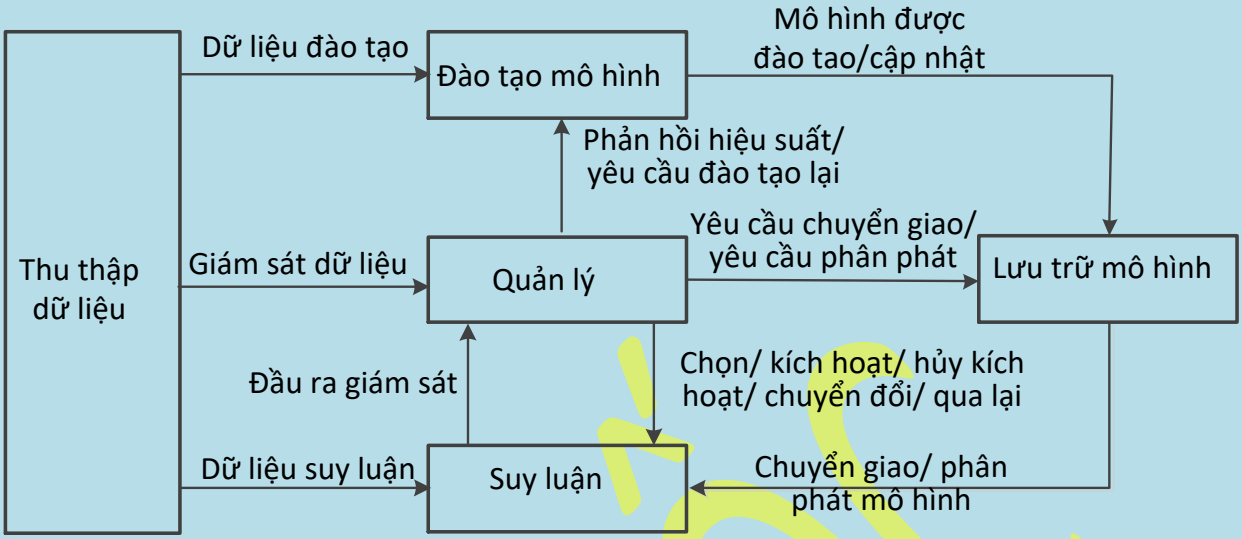
Đối với các tính năng này, các chức năng đào tạo và suy luận AI/ML có thể được triển khai bên trong RAN/CN hay MDAF hoặc phân chia giữa hai thực thể. Các tính năng này trươc hết được đồng chỉnh hướng đến tối ưu hóa mạng tốt hơn bằng cách sử dụng AI/ML (trường hợp sử dụng "AI cho mạng").

Dưới đây là một vài nét tổng quan công việc tiêu chuẩn hóa được tiến hành sau phát hành 19.

Phát hành 19 đang tiến gần đến hoàn thiện, trong khi một số danh mục nghiên cứu 5G nâng cấp (5G Advanced) và 6G của phát hành 20 đang ở các giai đoạn ban đầu. Một số nghiên cứu AI/ML hiện nay đang được tiến hành trong phát hành 20 tron nhóm công tác SA nhắm đến mở rộng phạm vi các dịch vụ AI/ML để tăng cường hiệu quả của kiến trúc 5G. Một lĩnh vực chính của tập trung này là khám phá cách thức có thể sử dụng phân tích dữ liệu để tối ưu hóa hiệu suất mặt phẳng người dùng. Một tập trung quan trọng khác là tính bền vững: đánh giá tiêu thu năng lượng AI/ML, tính hiệu quả, sự sử dụng tài nguyên trên các bước của vòng đời cũng như hỗ trợ sử dụng các nguồn năng lượng tái tạo. Nhiều nỗ lực được thực hiện để tối ưu hóa quá trình đăng ký và phát hiện mô hình ML trong khung quản lý AI/ML. Tại lớp ứng dụng, giám sát hiệu suất mô hình ML, đánh giá sự chính xác và đảm bảo hỗ trợ liên tục dịch vụ trên các miền của nhiều nhà khai thác đang được nghiên cứu.

9.4. KHUNG 3GPP AI/ML CHUNG

Để thiết lập một khung AI/ML 3GPP chung cho giao diện vô tuyến, những nỗ lực đáng kể từ các thành viên 3GPP đã được dành để xây dựng các thuật ngữ chung liên quan đến các chức năng, thủ tục và giao diện AI/ML. Hình 9.17 phác thảo bộ khung chức năng 3GPP AI/ML cho giao diện vô tuyến NR. Bộ khung mô tả một tập hợp các chức năng cốt lõi, bao gồm thu thập dữ liệu, đào tạo mô hình, quản lý, suy luận và lưu trữ mô hình.



Hình 9.17. Phác thảo bộ khung chức năng 3GPP AI/ML cho giao diện vô tuyến NR

Mô hình AI/ML cần được phát triển, triển khai và quản lý trong toàn bộ vòng đời- một quy trình được gọi là quản lý vòng đời mô hình AI/ML (AI/ML Model LCM: AI/ML Model Life Cycle Managment). 3GPP đã nghiên cứu hai phương pháp riêng biệt để quản lý vòng đời của mô hình AI/ML tại thiết bị người dùng (UE). Phương pháp đầu tiên được phân loại là LCM dựa trên chức năng (Function Based LCM) . Chức năng đề cập đến một tính năng được AI/ML hỗ trợ hoặc nhóm tính năng được hỗ trợ bởi một cấu hình. nhận dạng chức năng AI/ML (AI/ML Function Identification) thúc đẩy sự hiểu biết lẫn nhau giữa mạng và UE về chức năng AI/ML. Quá trình nhận dạng chức năng có thể được tích hợp trong bộ khung báo hiệu theo khả năng UE hiện có. Về cơ bản, cấu hình được điều chỉnh phù hợp với các điều kiện được chỉ ra bởi khả năng UE. Sau đó, sau khi nhận dạng các chức năng, UE có thể báo cáo các bản cập nhật liên quan đến các chức năng có thể áp dụng trong số những chức năng được định cấu hình hoặc được nhận dạng. Trong LCM dựa trên chức năng, mạng chỉ ra lựa chọn, kích hoạt, hủy kích hoạt, chuyển đổi và quay lại (Fallback) chức năng AI / ML thông qua báo hiệu 3GPP. Đáng chú ý, (các) mô hình AI/ML chính xác hỗ trợ cho một chức năng nhất định có thể không được xác định trên mạng.

Phương pháp thứ hai được phân loại là LCM dựa trên nhận dạng mô hình (Model Identity based LCM) . ID mô hình (Model ID) đóng vai trò là mã định danh đặc biệt cho mô hình AI/ML, trong đó mô hình có thể là logic và ánh xạ của nó đến mộ mô hình vật lý khi triển khai. Việc xác định mô hình AI/ML đảm bảo sự hiểu biết lẫn nhau giữa mạng và UE liên quan đến mô hình AI/ML được đề cập. Cụ thể, mô hình AI/ML được nhận dạng bằng ID

mô hình được chỉ định tại mạng và UE cho biết mô hình AI/ML được hỗ trợ của nó cho mạng. Bên cạnh ID mô hình, mô hình có thể có các điều kiện đi kèm như một phần của định nghĩa khả năng UE cũng như các điều kiện bổ sung (ví dụ: kịch bản, trang web và bộ dữ liệu) xác định khả năng ứng dụng của mô hình. Trong LCM dựa trên ID mô hình, cả mạng và UE đều có thể thực hiện lựa chọn, kích hoạt, hủy kích hoạt, chuyển đổi và quay lại mô hình AI/ML bằng cách sử dụng ID mô hình tương ứng.

Việc triển khai thương mại tính năng hỗ trợ AI/ML phụ thuộc vào khả năng mang lại hiệu suất đáng tin cậy trên nhiều kịch bản, cấu hình và điều kiện cụ thể của địa điểm trong hệ thống truyền thông di động. Để đạt được mục tiêu này, 3GPP đã nghiên cứu ba cách tiếp cận: tổng quát hóa mô hình (model generalization), chuyển đổi mô hình (model switching) và cập nhật mô hình (model update). Tổng quát hóa mô hình nhằm mục đích phát triển một mô hình có thể tổng quát hóa cho các tình huống, cấu hình hoặc địa điểm khác nhau. Ngoài ra, một tập hợp các mô hình cụ thể có thể được phát triển — từ kịch bản cụ thể đến cấu hình hoặc địa điểm cụ thể.

Trong phạm vi mô hình này, kỹ thuật chuyển đổi mô hình (Model Switching Technique) được khai thác để giải quyết hiệu quả các tình huống, cấu hình hoặc địa điểm khác nhau. Quá trình cập nhật mô hình, thường liên quan đến việc tinh chỉnh, đòi hỏi sự thích ứng linh hoạt của cấu trúc mô hình hoặc các tham số của nó để đáp ứng với những thay đổi trong kịch bản, cấu hình hoặc các site. Một nguyên tắc then chốt làm nền tảng cho các phương pháp tiếp cận này là đảm bảo rằng hiệu suất của các tính năng hỗ trợ AI/ML vẫn ở mức bằng hoặc tốt hơn so với các hoạt động không dựa trên AI/ML cũ. Do đó, giám sát hiệu suất là điều bắt buộc đối với các tính năng hỗ trợ AI/ML, yêu cầu các chức năng như tính toán các chỉ số hiệu suất được giám sát, báo cáo kết quả giám sát và cơ chế báo hiệu điều khiển để nhanh chóng phục hồi sau lỗi.

Các trường hợp sử dụng khác nhau yêu cầu mức độ cộng tác khác nhau giữa mạng và UE cho các hoạt động AI/ML tương ứng. 3GPP đã xác định ba cấp độ hợp tác mạng và UE đặc biệt:

- **Cấp độ x–không hợp tác**: Ở cấp độ này, các hoạt động AI/ML dựa trên việc triển khai độc quyền mà không có bất kỳ cải tiến tiêu chuẩn cụ thể nào được điều chỉnh cho các chức năng của AI/ML.
- **Cấp độ y – cộng tác dựa trên báo hiệu mà không cần chuyển giao mô hình**: Ở cấp độ này, các hoạt động AI/ML tích hợp các cải tiến tiêu chuẩn dành riêng để tạo điều kiện thuận lợi cho quá trình mà không liên quan đến việc chuyển gia mô hình. Ở đây, 'chuyển gioa mô hình' đề cập đến việc phân phối mô hình AI/ML qua giao diện vô

tuyến từ thực thể này sang thực thể khác, được thực hiện theo cách không trong suốt (not transparent) đối với các cơ chế tín hiệu 3GPP.

- **Cộng tác dựa trên tín hiệu cấp z với chuyển giao mô hình**: Trong cấp này, các hoạt động AI/ML không chỉ bao gồm tích hợp báo hiệu mới mà còn tận dụng các khả năng chuyển giao mô hình nâng cao.

Về bản chất, các cấp độ cộng tác này bao gồm một phạm vi từ sự tham gia tối thiểu đến tích hợp sâu, biểu thị tính linh hoạt và khả năng thích ứng của các hoạt động AI/ML trong các bối cảnh khác nhau.

9.5. AI/ML TRONG NGHIÊN CỨU CÁC KHÍA CẠNH HỆ THỐNG CỦA 3GPP

9.5.1. Công nghệ cho phép tự động hóa mạng (eNA: enabler for Network Automation)

Tự động hóa mạng được đề cập như là sử dụng phần mềm (thường được tăng cường bởi các chức năng tiên tiến của các công nghệ AI) để thực hiện nhiều nhiệm vụ. Các nhiệm vụ này bao gồm: quy hoạch, triển khai, lập cấu hình, sắp xếp và điều phối (Orchestration), phân tích và đảm bảo các mạng và các dịch vụ. Ngay từ phát hành 15, 3GPP đã đưa ra thực thể Chức năng phân tích số liệu mạng (NWDAF: Network Data Analytics Function). Chức năng của thực thể này như là một hộp đen AI để đảm bảo phân tích hiệu năng và dự báo liên quan mạng được rút ra từ chọn lựa dữ liệu đặc thù từ một hay nhiều nguồn dữ liệu, chẳng hạn các chức năng mạng (Network Functions), quản trị và bảo trì vận hành (OAM: Operation Administration and Maintenance) và UE.

Các phát hành tiếp theo đã mở rộng khả năng của nó để bao hàm cả thu thập dữ liệu và các tính năng để lộ các phân tích mạng (Network Anlalytics Exprosure). Các nâng cao này là then chốt trong việc tăng chức năng và sự hữu ích của tính năng này trong kiến trúc 5GC. Đặc biệt là hiện nay các kiểu phân tích khác nhau được đặc tả và phụ thuộc và nhận dạng (hay các nhận dạng) của phân tích (Analytics ID) được yêu cầu bởi một phần tử mạng người tiêu dùng (Consumer-Network Function: chức năng người tiêu dùng-mạng). Có lẽ là, thực hiện trên học liên kết (FL) sớm nhất trong 3GPP là tại eNA, tại đây các NWDAF khác nhau có thể liên kết hướng dẫn một mô hình để phân tích mạng. Tiêu chuẩn này cũng đặc tả các giao thức để các NWDAF hình thành quá trình FL hoặc mhư là server hoặc client.

9.5.2. AIML của Phát hành 18

Năm 2019, cuộc họp TSG SA#86 chấp thuận mục làm việc (Work Item) tập trung lên nghiên cứu các đặc trưng lưu lượng và các yêu cầu hiệu năng liên quan đến AMMT

(Artificial Intelligence and Machine Learning Model Trasfer: chuyển giao mô hình trí tuệ nhân tạo và máy học) trong hệ thống 5G (5GS: 5G System). Mục tiêu là nói rõ các tiêu chí hiệu năng đặc thù bao gồm trễ đầu cuối đầu cuối, tốc độ số liệu trải nghiệm và tính khả dụng của dịch vụ truyền thông. Ngoài ra các yêu cầu dịch vụ như quản lý QoS AI/ML, phân bố và chuyển giao mô hình/dữ liệu AI/ML, cũng như hiệu suất mạng và giám sát/dự báo sử dụng tài nguyên cũng được định nghĩa. Các nỗ lực này nhằm cho phép 5GS hỗ trợ một loạt các hoạt động AI/ML cho các ứng dụng đa dạng khác nhau bao gồm (nhưng không phải tất cả) nhận biết hình ảnh/tiếng nói, biên tập/các nâng cao truyền thông (Media Editing/Enhancements), điều khiển robot và các ứng dụng ô tô.

- Phân chia hoạt động AM/ML giữa các điểm cuối AM/ML;
- Phân bố mô hình/dữ liệu AM/ML và chia sẻ trên 5GS;
- Phân bố/FL trên 5GS.

Nhóm công tác về các khía cạnh hệ thống 3GPP SA1 (AS: System Aspect) đã hoàn thiện nghên cứu giai đoạn 1 về AMMT bằng việc giải quyết các trường hợp sử dụng và các yêu cầu hiệu suất tiềm năng để hỗ trợ AI lớp ứng dụng, phân bố và chuyển giao mô hình ML trong 5GS. Nghiên cứu này cũng bao gồm cả nhận dạng các đặc trưng lưu lượng đi kèm với chuyển giao, phân bố AI/ML và đào tạo các ứng dụng khác nhau. Bắt đầu của nghiên cứu các thủ tục này là khám phá các khả năng của nền tảng 5GS để hỗ trợ các hoạt động AI/ML của lớp ứng dụng được bắt đầu bởi 3GPP SA2 vào cuối năm 2021.

SA 2 đã giải quyết thành công một số vấn đề quan trọng không chỉ giới hạn đến giám sát sử dụng tài nguyên mạng mà còn bao gồm cả bảo vệ thông tin 5GC (5G Core: lõi 5G) không bị lộ đến UE và phía thứ ba, cải thiện QoS và chính sách, đặc biệt là miền các hoạt động FL phức tạp. Các giải quyết này đóng góp vào sự bền vững và hiệu quả tổng thể của hệ thống, đảm bảo tính riêng tư được nâng cao, hiệu quả và các khả năng phát triển các tiêu chuẩn viễn thông.

Trong phát hành 18, việc đưa FL vào 3GPP đánh dấu một cột mốc quan trọng. Các thực thể trước hết liên quan đến đóng góp vào FL là NWDAF và AF (Application Function: chức năng ứng dụng). NWDAF đóng vai trò trung tâm trong hướng dẫn đào tạo mô hình và các cập nhật thông qua một FL server NWDAF thực hiện điều phối nhiều FL Client NWDAF. Sự cộng tác này hỗ trợ việc kết hợp dữ liệu từ nhiều dạng nguồn khác nhau trong khi vẫn đảm bảo an ninh và tính riêng tư. Mặt khác, AF chịu trách nhiệu lập lịch cho các UE tham gia và để lộ (Explosure) các yêu cầu thông qua NEF (Network Explosure Function: chức năng để lộ hay phơi sáng mạng) của 5GC. Quá trình này có một tập các nguyên tắc để đảm bảo sự tham gia hiệu quả cả các UE trong quá trình FL trong khi vẫn duy trì tính toàn vẹn và hiệu năng mạng.

9.5.3. FL được hỗ trợ bởi truyền thông thiết bị đến thiết bị (D2D: Device to Device)

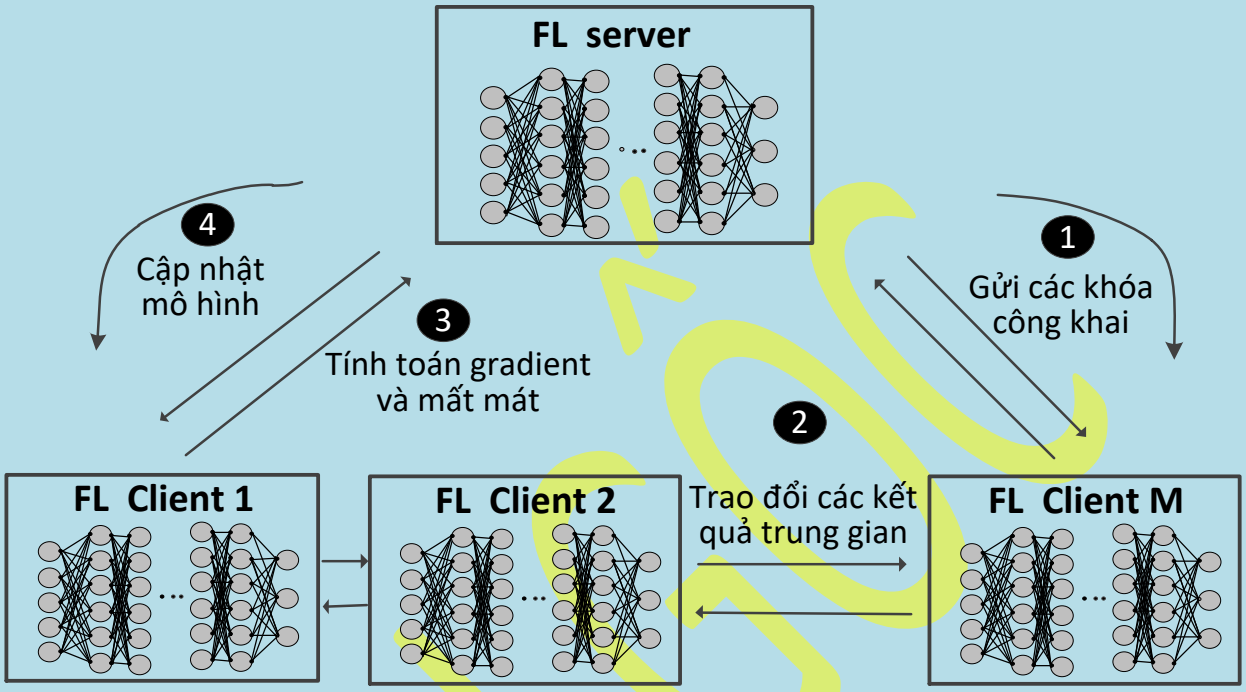
Để giảm tải (Offloading) các nhiệm vụ từ một UE này sang một UE khác trong vùng gần, nhiều nhà cung cấp dịch vụ đầu cuối đã đề xuất FL dựa trên (hay được hỗ trợ) bởi D2D.Giải pháp này nhằm giảm bớt các công sức tính toán và bảo tồn công suất bằng cách phân phát các nhiệm vụ giữa các tiết bị gần nhau. Trong trường hợp này, các UE chuyển tiếp (Relay UE) cần thủ tục khám phá khi chọn các FL client, đôi khi với sự hỗ trợ của AF. Vì thế cần nghiên cứu chính sách và điều khiển QoS bao gồm các ngưỡng QoS được tổng hợp, để hỗ trợ FL trên D2D hiệu quả.

Đáng tiếc là còn quá sớm để quyết định quy mô mà chức năng AI/ML dựa trên D2D có thể đóng góp cho hệ thống khi chưa có sự triển khai trước tiên một bộ khung AI/ML đầy đủ trong cả RAN và 5GC. Tuy nhiên có thể kỳ vọng rằng sử dụng FL được D2D hỗ trợ có thể cung cấp tiềm năng các lợi ích đang kể liên quan đến hiệu suất chuyển tải nhiệm vụ và tối ưu hóa tài nguyên một khi một bộ khung AL/ML bền vững được thiết lập trên toàn mạng.

9.5.4. AIML của Phát hành 19

Đối với các mục tiêu đặc thù để nâng cao hỗ trợ cho AI/ML trong 5GC của phát hành 19, ưu tiên được dành cho dịch vụ định vị (LCS: Location Service) liên quan đến các nâng cao như là một tính năng trụ cột trên toàn miền của AM/ML. Ưu tiên này nhấn mạnh tầm quan trọng đặc biệt của tích hợp xuôn xẻ các khả năng AI/ML vào các dịch vụ định vị trong bộ khung của 5GC. Trung tâm của nỗ lực này là nghiên cứu một hô hình AI/ML cho định vị dựa trên LMF (Location Management Function: chức năng quản lý vị trí). Nghiên cứu này bao hàm nhiều khía cạnh khác nhau như: huấn luyện mô hình như thế nào, thu thập số liệu như thế nào và suy luận được như thế nào. Đồng thời các khía cạnh quan trọng như: thu thập số liệu của UE, chuyển giao/chuyển phát mô hình đến UE, đồng bộ nhận dạng và quản lý cũng được lập lịch cho nghiên cứu và thảo luận tiếp theo. Các khía cạnh này được giải quyết cùng với các các quyết định và các kết luận được đưa ra trong thời gian hội nghị toàn thể vào tháng 9 năm 2024 để đảm bảo xem xét và đồng bộ toàn diện với việc phát triển các tiêu chuẩn và và các yêu cầu từ góc nhìn RAN. Hai mục tiêu được ưu tiên khác là hỗ trợ học liên kết dọc chiều dọc (VFL: Vertical Federated Learning), điều khiển chính sách được NWDAF hỗ trợ (NWDAF Assisted Policy) và giảm thiểu hành vi không bình thường của mạng (Network Abnormal Behavior Mitigation). Về tính năng VFL, nó đánh dấu một cột mốc quan trọng, vì nhóm nghiên

cứu xem xét kỹ lưỡng sự bắt nguồn của dữ liệu từ cùng một không gian nhưng lại được đặc trưng bởi các không gian tính năng khác nhau. Hình 9.18 cho thấy thí dụ về một kiến trúc điển hình liên quan đến VFL.



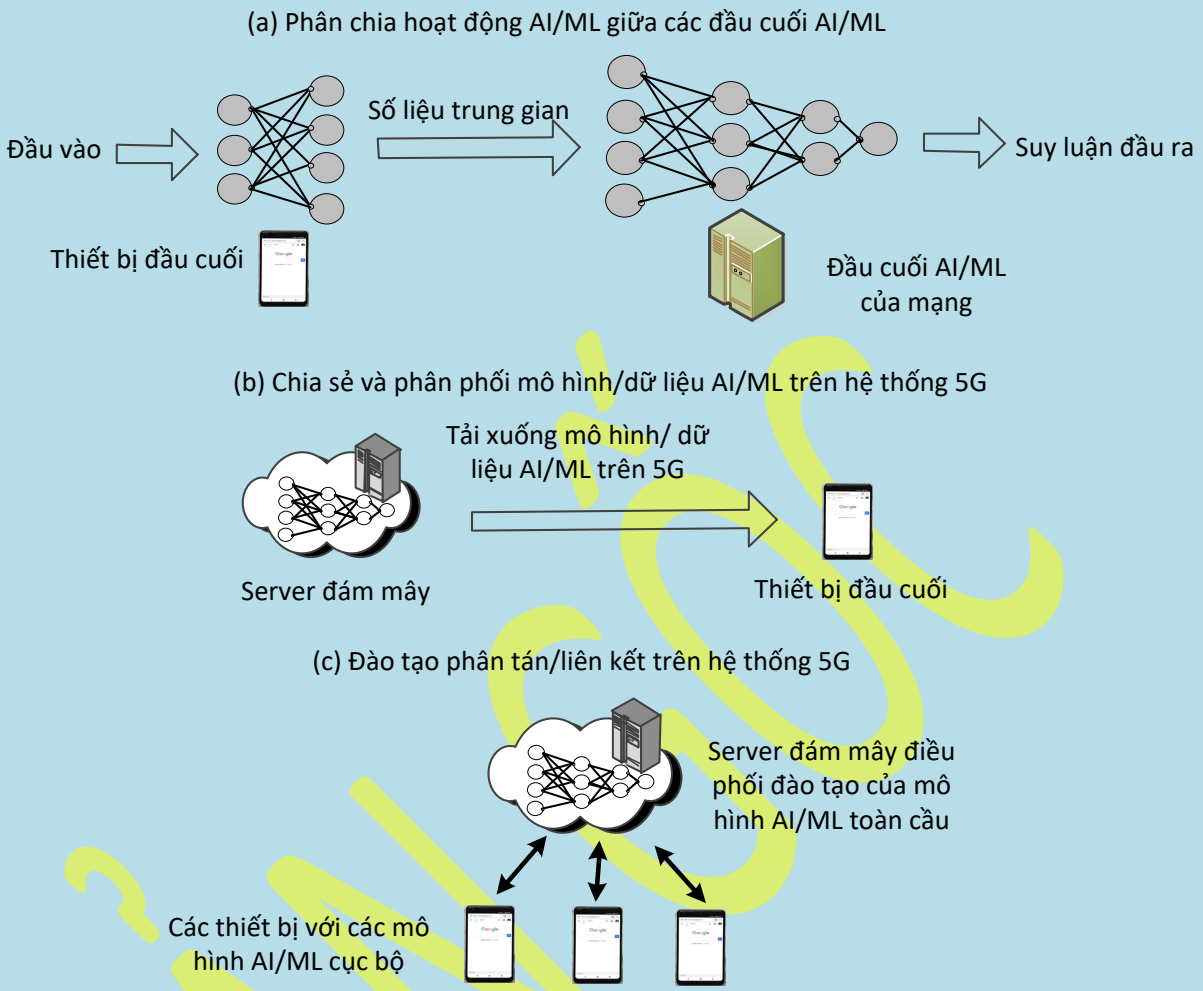
Hình 9.18. Kiến trúc đối với hệ thống VFL

9.6. HOẠT ĐỘNG CỦA AI/ML LỚP ỨNG DỤNG TRONG HỆ THỐNG 5G

9.6.1. Chuyển giao mô hình AI/ML (AI/ML Model Transfer)

Các mô hình AI/ML tìm thấy ứng dụng đầu tiên trong các thiết bị di động trong hệ thống 5G, ví dụ nhận dạng ảnh, nhận dạng giọng nói và xử lý video. Do đa dạng ứng dụng và lưu trữ có hạn, nên không thể tải trước tất cả các mô hình xuống UE. Vì thế, tải xuống các mô hình AI/ML là cần thiết. Đối với một số ứng dụng, UE có thể không có đủ tài nguyên tính toán để hực hiện suy luận, trong trường hợp này có thể phải giảm tải suy luận từ UE đến đám mây hoặc biên 5G. Ngoài ra cần chia sẻ dữ liệu đào tạo trên các thực thể khác nhau để liên kết đào tạo một mô hình AI/ML toàn cục trong hệ thống 5G.

Khuynh hướng chuyển giao các mô hình AI/ML và dữ liệu dẫn đến các kiểu lưu lượng mới cần phải phục vụ trong hệ thống 5G. 3GPP SAI chịu trách nhiệm nhận dạng dịch vụ và các yêu cầu hiệu năng mới trong các hệ thống 3GPP. Các nghiên cứu đề cập đến ba trường hợp sử dụng như mô tả trên hình 9.19.



Hình 9.19. Các trường hợp sử dụng của các hoạt động AI/ML trong hệ thống 5G được nhận dạng trong phát hành 18 của 3GPP

Trong trường hợp sử dụng thứ nhất (hình 9.19(a)), hoạt động AI/ML được phân chia giữa các điểm cuối. Mục đích của phân chia này là để giữ các phần dịch vụ nhạy cảm tính riêng tư – hay độ trễ của hoạt động này tại UE, nhưng giảm tải các phần nhạy cảm tính toán – năng lượng đến các điểm cuối mạng. Trong trường hợp sử dụng thứ hai (hình 9.19(b) chia sẻ và phân bố mô hình/ dữ liệu AI/ML (Model/ Data Distribution and Sharing) được thực hiện trên hệ thống 5G để có thể tải xuống từ mạng đến thiết bị mô hình thích ứng khi cần thiết. Trong trường hợp sử dụng thứ ba (hình 9.19(c)), học phân tán/liên kết (Distributed/ Federated Learning) được thực hiện trên hệ thống 5G, tại đây UE thực hiện đào tạo một phần dựa trên dữ liệu cục bộ và thực thể trung tâm đào tạo mô hình toàn cục bằng cách tổng hợp các kết quả cục bộ từ các UE.

Nghiên cứu đã nhận dạng các các yêu cầu dịch vụ mới và các chỉ số hiệu năng quan trọng (KPI: Key Performance Indicator) cho đào tạo, suy luận, tải xuống, giám sát, dự đoán và

quản lý mô hình AI/ML trên hệ thống 5G. Sau khi hoàn thành nghiên cứu, 3GPP SA1 tiếp tục với mục công tác tiếp theo cho hệ thống 5G để hỗ trợ ba trường hợp sử dụng nói trên. Các yêu cầu này được đưa ra trong đặc tả kỹ thuật (TS 22.261).

Từ góc độ các yêu cầu dịch vụ, TS 22.261 đặc tả rằng đối với chuyển giao mô hình AI/ML, hệ thống 5G sẽ có thể hỗ trợ chức năng dịch vụ mới liên quan đến, ví dụ: giám sát sử dụng tài nguyên, giám sát hiệu năng mạng, dự báo và thay đổi thông báo và quản lý chất lượng dịch vụ (QoS). Việc áp dụng các yêu cầu này tùy thuộc và chính sách nhà mạng, sự chấp thuận của người dùng và các yêu cầu pháp lý.

Từ góc độ yêu cầu hiệu năng, TS 22.261 đặc tả các KPI cho chuyển giao mô hình AI/ML trong hệ thống 5G gồm độ trễ đầu cuối đầu cuối, tốc độ dữ liệu trải nghiệm, mức độ khả dụng của dịch vụ truyền thông, trong số các KPI khác. Chẳng hạn, tốc độ dữ liệu đường xuống trải nghiệm yêu cầu là 1,1 Gbit/s cho nhận dạng ảnh liên quan đến phân phối mô hinhd AI/ML. Một thí dụ khác là nhân dạng ảnh AI/ML phân chia giữa UE và server mạng, trễ tối thiểu được phép đường lên đầu cuối đầu cuối là 2 ms cà tốc độ dữ liệu đường lên trải nghiệm yêu cầu là 1,08 Gbit/s.

9.6.2. Hỗ trợ hệ thống 5G

3GPP SA1 tiếp tục cải thiện chuyển giao AI/ML trong dịch vụ được nhận dạng bởi hệ thống 5G và các yêu cầu hiệu năng, nhiều trong số chúng liên quan đến hỗ trợ hoạt động AI/ML của lớp ứng dụng trong UE. Mặc dù hệ thống 5G có NWDAF trong 5GC để cho phép tự động hóa mạng, vẫn chưa có đặc tả các giải pháp truyền tải được tối ưu cho hoạt động AI/ML lớp ứng dụng trong UE. Vì thế vẫn cần phải phát triển hệ thống 5G đế hỗ trợ các dịch vụ dựa trên AI/ML. Đây chính là động lực cho mục nghiên cứu hỗ trợ của hệ thống 5G cho các dịch vụ dựa trên AI/ML đang được tiến hành bởi 3GPP SA2, một nhóm công tác chịu trách nhiệm cho phát triển kiến trúc hệ thống 3GPP tổng thể.

Từ góc nhìn các yêu cầu kiến trúc, một chức năng ứng dụng (AF Application Function) điều khiển logic hoạt động AI/ML lớp ứng dụng. AF là một chức năng của mặt phẳng điều khiển tương tác với 5GC để cung cấp các dịch vụ hỗ trợ, chẳng hạn ảnh hưởng của ứng dụng lên định tuyến lưu lượng. Các yêu cầu AF cho sự trợ giúp của 5GC để hỗ trợ hoạt động AI/ML lớp ứng dụng phải được 5GC cấp quyền.

3GPP SA2 khám phá một số mở rộng kiến trúc và chức năng để hõ trợ hoạt động AI/ML lớp ứng dụng. Một vấn đề quan trọng là cách thức giám sát sử dụng tài nguyên mạng liên quan đến hiệu suất của UE như tốc độ dữ liệu và độ trễ. Chức năng để lộ mạng (NEF: Network Exposure Function) trong 5GC có thể hỗ trợ giám sát khả năng cho phép cấu hình, phát hiện và báo cáo các sự kện giám sát đến phía ngoài được cấp quyền. Các sự

kiện giám sát mới có thể được đưa ra để đo và dự đoán sử dụng tài nguyên mạng để hỗ trợ hoạt động AI/ML lớp mạng. Ngoài ra vệc mở rộng để lộ thông tin 5GC cho UE hoặc cho phía ngoài được cấp quyền có thể có ích để hỗ trợ hoạt động AI/ML lớp ứng dụng. Thông tin trợ giúp cho để lộ có thể gồm các tình trạng của UE hoặc mạng, và các dự báo hiệu suất theo, ví dụ: vị trí UE, tải và QOS. Một vấn đề quan trọng nữa là tăng cường khả năng cung cấp, cho phép một bên ngoài cung cấp thông tin cho 5GC tạo điều kiện hỗ trợ hoạt động AI/ML lớp ứng dụng. Một thí dụ về cung cấp thông tin tham số từ bên ngoài là các hành vi của UE được kỳ vọng, như tốc độ UE kỳ vọng và các đặc tính UE kỳ vọng. Cũng cần nghiên cứu các tăng cường 5GC khác như cơ chế định tuyến lưu lượng để hỗ trợ lưu lượng AI/ML.

Hệ thống 5G có các cơ chế đầy đủ cho định nghĩa QoS, thực hiện, điều khiển và giám sát để hỗ trợ các ứng dụng khác nhau. Hoạt động AI/ML lớp ứng dụng dựa trên QoS được cung cấp bởi hệ thống 5GC phía dưới. Cần nghiên cứu xem có cần tăng cường bộ khung chính sách và mô hình QoS hiện thời cho lưu lượng ứng dụng AI/ML, để lộ giám sát và thông tin trạng thái về sử dụng tài nguyên mạng cho phía thứ ba được cấp quyền và thông báo hoạt động AI/ML lớp ứng dụng về các dự báo về sự các sự thay đổi tình trạng mạng. Một mục tiêu khác của nghiên cứu 3GPP SA2 là việc nghiên cứu sự trợ giúp của 5GC để tạo điều kiện thuận lợi cho hoạt động học liên kết của lớp ứng dụng trên hệ thống 5G, bao gồm chọn lựa và quản lý thành viên liên kết, giám sát và để lộ, và ấn định tài nguyên mạng.

9.6.3. Các khía cạnh về tính riêng tư và an ninh

3GPP SA2 tiếp tục nghiên cứu hỗ trợ hệ thống 5G cho hoạt động AI/ML được mô tả trong phần trước nhận thấy rằng các dữ liệu khác nhau cần được truyền giữa 5GC và AF. Một số dữ liệu (thí dụ: phân tích QoS, thông tin phân bố địa lý) có thể nhạy cảm tính riêng tư của người dùng và không được để lộ đến AF không được cấp quyền. Một số dữ liệu (ví dụ: phân tích tải mạng) phản ánh trạng thái mạng phải được xử lý một các an ninh để phòng ngừa dữ liệu này bị sử dụng bởi các kẻ tấn công tiềm năng. Vì thế nghiên cứu các khía cạnh tính riêng tư và an ninh của các hoạt động AI/ML lớp ứng dụng là rất quan trọng để tránh rủi do và đe dọa tiềm năng đối với hệ thống 5G và các người dùng.

3GPP SA3, một nhóm công tác chịu trách nhiệm cho an ninh và tính riêng tư trong các hệ thống 3GPP, đang tiến hành nghiên cứu an ninh và tính riêng tư của các dịch vụ dựa trên AI/ML và các ứng dụng trong 5G. Mục đích của nghiên cứu này là để nhận dạng các vấn đề then chốt, các đe dọa tiềm năng, các yêu cầu và các giải pháp. Một vấn đề then chốt

được nhận dạng là tính riêng tư và cấp quyên cho để lộ thông tin hỗ trợ của 5GC, bao gồm xác định thông tin hỗ trợ nhậy cảm tính riêng tư và các phương pháp để 5GC bảo vệ và cấp quyền AF truy nhập thông tin này. Các cơ chế bảo vệ tính riêng tư và cấp quyền thích hợp cần phải có để tôn trọng tính riêng tư và đảm bảo an ninh mạng. Một giải pháp là sử dụng lại các cơ chế hiện thời (ví dụ: cấp quyền mở (OAuth: Open Authorization)) cho để lộ thông tin hỗ trợ của 5GC đến AF.

Một giải pháp khác đang được nghiên cứu trong 3GPP SA3 là cấp quyền để lộ thông tin hỗ trợ 5GC dựa trên hồ sơ (Profile Based). Một hồ sơ của UE sẽ xác định liệu một AF đặc thù có thể yêu cầu hoặc thay đổi thông tin nhất định của một UE liên kết. Nó có thể gồm nhận dạng UE, nhận dạng AF, định danh dịch vụ kỳ vọng (Expected Service Identifier), thông tin hỗ trợ 5GC đích, thời gian hết hạn và các chính sách bảo vệ và cấp quyền. Hồ sơ có thể được lưu trữ trong quản lý dữ liệu thống nhất (UDM: Uniffied Data Management)/một kho số liệu thống nhất (UDR: Uniffied Data Repository). Một AF gửi yêu cầu truy nhập thông tin hỗ trợ 5GC đến NEF/NWDAF (Network Exposure Function/ Network Data Analytics Function: chức năng để lộ mạng/chức năng phân tích dữ liệu mạng). Nhận được yêu cầu này, NEF/ NWDAF trực tiếp xác định hồ sơ UE nếu có trong NEF/NWDAF; nếu không NEF/NWDAF nhận hồ sơ UE từ UDM/UDR. Dựa trên hồ sơ UE, NEF/NWDAF gửi đi thông tin hỗ trợ 5GC cùng với cơ chế bảo vệ an ninh tương ứng đến AF yêu cầu.

9.7. AI/ML CHO CÁC PHƯƠNG TIỆN TRUYỀN THÔNG (MEDIA) VÀ QUẢN LÝ AI/ML TRONG 5G

9.7.1. AI/ML cho các phương tiện truyền thông

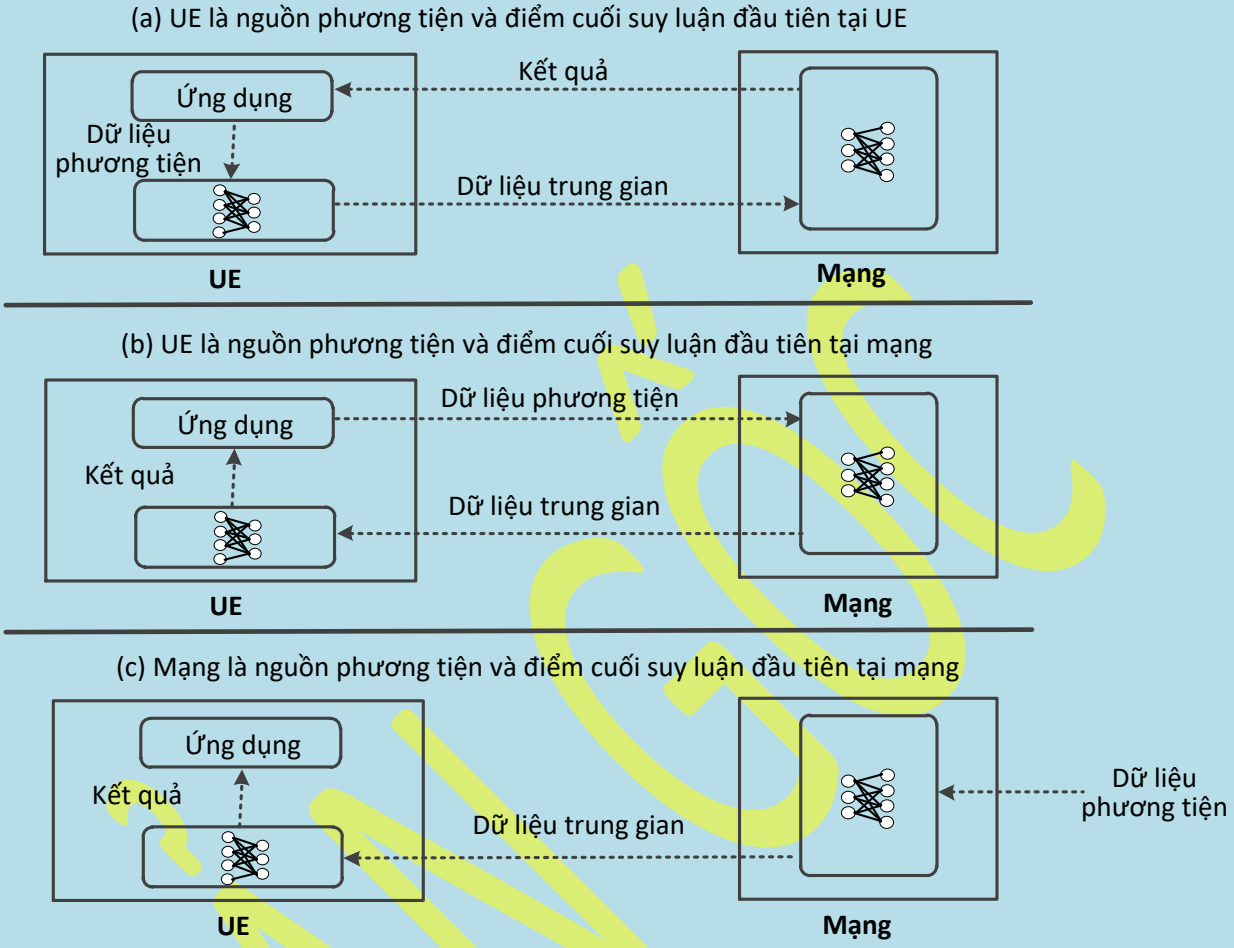
Rất nhiều các ứng dụng AI/ML là các phương tiện truyền thông (gọi tắt là phương tiện) như phân loại ảnh, nhận dạng giọng nói và tăng cường chất hượng video. Các ứng dụng này đòi hỏi xử lý tính toán ngay càng cao và chuyên sâu hơn để xử lý khối lượng dữ liệu AI/ML và mức độ mức tạp của mô hình ngày càng tăng. Trong khi đó, các dịch vụ phương tiện mới nổi như thực tế ảo (VR), thực tế tăng cường (AF) và trò chơi đám mây cũng đang rất thành công. Các ứng dụng di động như vậy phụ thuộc rất nhiều vào các chức năng AI/ML. Chẳng hạn, các dịch vụ phương tiện AR dựa trên kính đeo đòi hỏi ngữ cảnh của môi trường xung quanh, có thể được cung cấp bằng cách sử dụng phân đoạn, nhận dạng và phân loại đối tượng với trợ giúp của AI/ML. AI/ML cũng tìm được ứng dụng trong các giải thuật nén phương tiện để làm cho các bộ mã hóa thích ứng hơn.

Ngoài ra các mô hình AI/ML cho các ứng dụng phương tiện đòi hỏi nén phù hợp để tạo điều kiện dễ dàng hơn cho truyền dẫn hiệu suất. Tất nhiên nhóm các chuyên gia hình ảnh động (MPEG: Moving Picture Experts Group) cũng đang nghiên cứu mã hóa video mạng thần kinh sâu để tối ưu hóa nén phương tiện và nghiên cứu đại diện mạng thần kinh cho các mạng thần kinh nén cho các ứng dụng phương tiện.

Các xu hướng mới nhất nói trên trong tác động lẫn nhau giữa AI/ML và các ứng dụng phương tiện đòi hỏi nghiên cứu sâu để hiểu biết được ảnh hưởng của chúng lên hệ thống 5G. Vì thế, 3GPP SA4 chịu trách nhiệm cho phát triển các đặc tả codec phương tiện, các khía cạnh hệ thống và các khía cạnh chuyển phát nội dung phương tiện, đạng nghiên cứu AI/ML trong các dịch vụ phương tiện 5G. Mục tiêu chính của nghiên cứu này là nhận dạng các yêu cầu tương tác liên quan và các ràng buộc thực hiện của AI/ML trong các dịch vụ phương tiện 5G.

Lúc đầu, công việc của 3GPP SA4 có mục đích là mô tả các trường hợp sử dụng dựa trên AI/ML dựa trên kết quả nghiên cứu về chuyển giao mô hình AI/ML để phân loại ba hoạt động then chốt bao gồm phân chia hoạt động AI/ML, phân phối mô hình và học phân tán/liên kết. Tập các trường hợp sử dụng thứ nhất là nhận dạng đối tượng trong hình ảnh và video được thực hiện bởi chỉ suy luận tại UE, chỉ suy luận tại mạng và phân chia các suy luận. Tập thứ hai của các trường hợp sử dụng là tăng cường chất lượng video trong phát luồng trực tiếp. Một trong các trường hợp sử dụng này là mã hóa video đầu cuối đầu cuối dựa trên mạng thần kinh, trong đó phía gửi xử lý một video chất lượng cao bằng cách sử dụng một mạng thần kinh để tạo ra một luồng số liệu trung gian và phát nó cùng với mã hóa video với độ phân giải thấp hơn và phía thu xử lý luồng dữ liệu trung gian thu được và luồng video để tái tạo lại video chất lượng cao cho kết suất. Ta cũng có thể áp dụng mạng thần kinh để hậu xử lý một video đã được giải mã để nâng cao chất lượng video. Tập các trường hợp sử dụng thứ ba là thu tập phương tiện truyền thông từ một nhóm đông (Crowd-Sourcing Media Capture), trong đó mỗi UE sử dụng một mạng thần kinh chia sẻ để xử lý video/audio thu được của mình hay mạng thực hiện suy luận trên dự liệu phương tiện từ nhiều UE. Tập các trường hợp sử dụng cuối cùng là xử lý ngôn ngữ tự nhiên trên lời nói (Speech) chẳng nhận dạng lời nói và dịch tiếng nói (Voice).

Mục tiêu chính của nghiên cứu của 3GPP SA4 về AI/ML cho phương tiện truyền thông là mô tả kiến trúc dịch vụ của phương tiện cho AI/ML và các luồng dịch vụ liên quan. Cuối cùng, một nghiên cứu quan trọng là phân chia suy luận AI/ML giữa mạng và UE. Điểm phân chia phụ thuộc vào một số nhân tố bao gồm các khả năng của UE (ví dụ: bộ nhớ, tính toán, tiêu thụ năng lượng và độ trễ suy luận). Hình 9.20 minh họa thứ tự khác nhau của các hoạt động và các luồng dịch vụ tương ứng cho phân chia suy luận giữa mạng và UE.



Hình 9.20. Các cấu hình topo suy luận AI/ML được phân chia cho phương tiện

9.7.2. Quản lý AI/ML trong hệ thống 5G

AI/ML đang được sử dụng trong hệ thống 5G bao gồm quản lý và điều phối (Ochestration), mạng lõi và RAN. Để cho phép và tạo điều kiện thuận lợi cho hoạt động AI/ML trong hệ thống 5G, các mô hình AI/ML cần được tạo ra, được đào tạo, được kiểm tra, được phát triển và được quản lý trong suốt vòng đời. Trong phát hành 17, 3GPP tiên hành nghiên cứu chuẩn hóa quản lý đào tạo AI/ML, nhưng không để cập đến các khía cạnh khác như triển khai và suy luận AI/ML. Ngoài ra các khả năng quản lý AI/ML có thể không cần phối hợp với các khả năng của AI/ML trong mạng lõi và hỗ trợ các khả năng AI/ML trong RAN. 3GPP SA5, chịu trách nhiệm cho quản lý, điều phối và tính

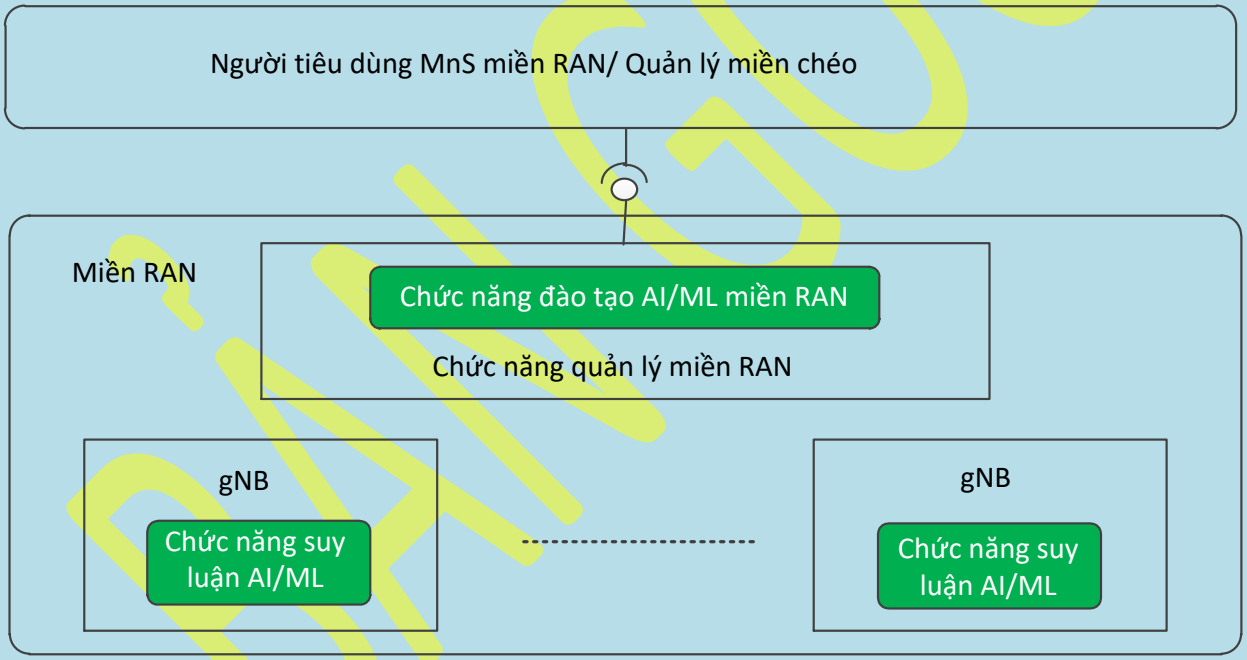
cước cho các hệ thống 3GPP, đang nghiên cứu quản lý để phối hợp các chức năng AI/ML trên các hệ thống 5G.

Quy trình làm việc của hoạt động AI/ML gồm ba giai đoạn: giai đoạn đào tạo (bao gồm đào tạo mô hình và kiểm tra), giai đoạn triển khai và giai đoạn suy luận. Quản lý AI/ML cho giai đoạn đào tạo cần hỗ trợ quản lý dữ liệu đào tạo, quản lý đào tạo, quản lý kiểm tra và xác nhận. AI/ML cho giai đoạn triển khai cần hỗ trợ điều khiển và giám sát triển khai, cho phép mô hình đã được đào tạo và đã được kiểm tra triển khai đến chức năng suy luận đích cũng như giám sát quá trình triển khai. Quản lý AI/ML cho giai đoạn suy luận cần hỗ trợ kích hoạt và hủy kích hoạt, điều kiển chức năng suy luận, quản lý hiệu năng suy luận, quản lý sự tin cậy (Trustworthiness) và điều phối suy luận.

3GPP SA5 đã nghiên cứu một mục đầy đủ các trường hợp sử dụng cho quản lý AI/ML và các yêu cầu tương ứng của chúng cũng như các giải pháp có thể có. Các khả năng của các trường hợp sử dụng cho quản lý cho giai đoạn đào tạo bao gồm dữ liệu sự kiện cho đào tạo, xác nhận thực thể, kiểm tra thực thể, đào tạo lại thực thể, đào tạo liên kết các thực thể, phân tích và báo cáo hiệu quả số liệu đào tạo, lập bản đồ và khai phá khả năng, quản lý cập nhật, đáng giá hiệu năng đào tạo, quản lý cấu hình đào tạo, học chuyển giao v.v. Các khả năng của các trường hợp sử dụng cho quản lý cho giai đoạn suy luận bao gồm theo dõi lịch sử suy luận, điều phối suy luận, điều khối khả năng, tải thực thể, mô phỏng suy luận, đánh giá hiệu năng suy luận, quản lý cấu hình suy luận, điều khiển cập nhật v.v.

Cần lưu rằng sự tin cậy được nhận dạng như là một khả năng quản lý cho cả giai đoạn đào tạo và giai đoạn suy luận. Mục tiêu của quản lý sự tin cậy là để đảm bảo mô hình AI/ML là bền vững, có thể giải thích được và tốt. Các yêu cầu của sự tin cậy có thể thay đổi tùy theo mức độ rủi do của trường hợp sử dụng yêu cầu cơ chế sự tin cậy. Từ bước đàu tiên, các chỉ số về sự tin cậy của AI/ML cần được định nghĩa. Phụ thuộc vào trường hợp sử dụng, người tiêu dùng dịch vụ dịch vụ quản lý AI/ML (AI/ML MnS: AI/ML Management Service) có thể lựa chọn một tập phù hợp các chỉ số sự tin cậy và yêu cầu một nhà sản xuất AI/ML MnS giám sát và và đánh giá các chỉ số được chọn. Tiền xử lý dữ liệu đào tạo/ kiểm tra/ suy luận có thể cần thiết tùy thuộc và phạm vi sự tin cây mong muốn của mô hình AI/ML tương ứng. AI/ML phải trang bị cho người tiêu dùng khả năng cung cấp đến nhà sản suất các yêu cầu sự tin cậy của xử lý dữ liệu cũng như cho phép nhà sản xuất để lộ các khả năng xử lý dữ liệu liên quan đến sự tin cậy được hỗ trợ cho người tiêu dùng. Tương tự, người tiêu dùng AI/ML MnS phải có thể truy vấn nhà sản xuất đào tạo AI/ML, nhà sản xuất suy luận, (hoặc) và nhà sản xuất đánh giá và các khả năng sự tin cậy được hỗ trợ và yêu cầu cấu hình, đo và báo cáo tập các đặc tính sự tin cậy được chọn.

3GPP SA5 cũng đã nghiên cứu các kịch bản triển khai cho chức năng đào tạo, chức năng kiểm tra, chức năng suy luận và các khả năng quản lý tương ứng. Chức năng đào tao/ suy luận có thể được đặt trong hệ thống quản lý miền chéo (Cross-Domain) hay trong hệ thống quản lý miền, ví dụ: phân tích dữ liệu trong MDAF (Management Data Analytics Function: chức năng phân tích dữ liệu quản lý) hay trong mạng lõi, ví dụ: trong NWDAF (Network Data Analytics Function: chức năng phân tích số liệu mạng) và RAN, ví dụ: trí tuệ RAN (RAN Intelligence) trong gNB. Các khả năng quản lý do nhà sản xuất MnS được đặt trong chức năng quản lý tương ứng MnF (Management Function: chức năng quản lý). Các chức năng khác nhau có thể được đặt cùng một chỗ hoặc được triển khai trong các thực thể khác nhau. Hình 9.21 minh họa một kịch bản triển khai cho trí tuệ RAN trong đó chức năng đào tạo và suy luận của AI/ML được đặt trong thực thể quản lý RAN và gNB và thực thể quản lý RAN cung cấp khả năng quản lý cho cả hai chức năng đào tạo và chức năng suy luận.



Hình 9.21. Thí dụ về kịch bản triển khai cho trí tuệ RAN

9.8. AI/ML CHO MẠNG TRUY NHẬP VÔ TUYẾN 5G

Có thể nói RAN là phần tử phức tạp nhất của các mạng tổ ong. Trong phát hành 18, 3GPP tham gia vào công việc chuẩn hóa cũng như nghiên cứu sử dụng AI/ML để tăng cường giao diện vô tuyến 5G NR, để cải thiện hiệu năng hệ thống và trải nghiệm người dùng. Phần này sẽ trình bày sự kết hợp AI vào mạng truy nhập vô tuyến của 5G NR-RAN

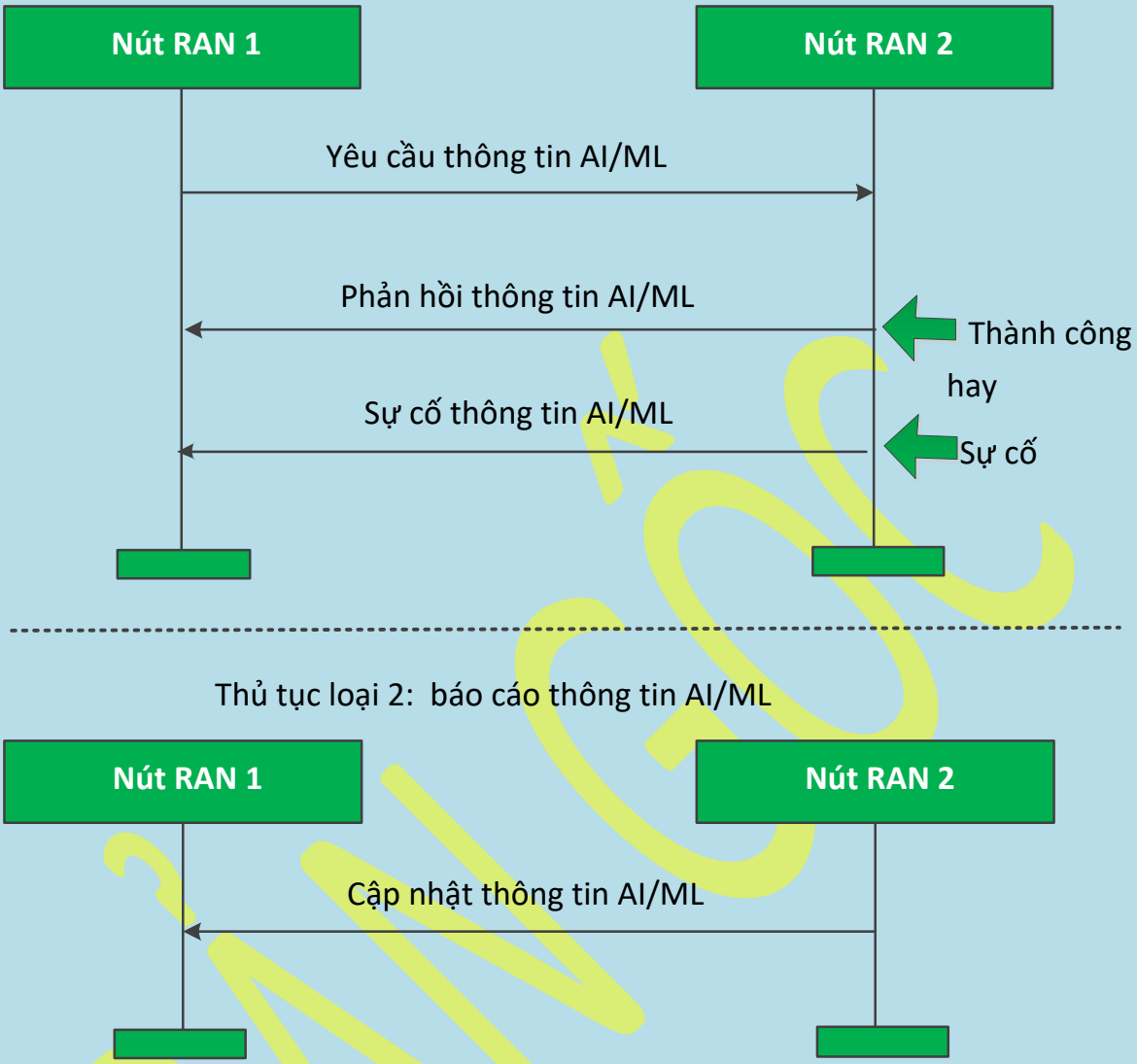
(5G NR-RAN: New Radio –Radio Access Network: mang truy nhập vô tuyến – vô tuyến mới của 5G)

9.8.1. NG-RAN được AI/ML hỗ trợ

Tiếp sau việc hoàn thiện của nghiên cứu trong phát hành 17, 3GPP RAN3, chịu trách nhiệm cho kiến trúc RAN tổng thể và đặc tả các giao thức cho các giao diện mạng liên quan, tiến hành công việc chuẩn hoá trong phát hành 18 cho trí tuệ RAN được hỗ trợ bởi AI/ML. Mục đích của công việc chuẩn hoá là đặc tả các tăng cường thu thập dữ liệu đặc thù và hỗ trợ báo hiệu để tiết kiệm năng lượng mạng dựa trên AI/ML, cân bằng tải và tối ưu hóa di động. Các tăng cường được đưa ra trong các giao diện và kiên trúc NR-RAN.Cả hai kiến trúc phân chia (thí dụ: gNB nguyên khối) và kiên trúc phân chia (ví dụ: gNB phân chia vào đơn vị trong tâm (CU) và đơn vị phân tán (DU) là phạm vi của công việc này. Đào tạo mô hình AT/ML và suy luận AI/ML có thể được đặt trong OAM (Operation and Maitenance: khai thác và bảo dưỡng) và gNB (hay gNB-CU cho kiến trúc phân chia) hay cả hai có thể được đặt trong gNB (hay gNB-CU cho kiến trúc phân chia).

So với các xử lý dựa trên đo truyền thống trong các đặc tả của 3GPP, trí tuệ RAN dựa trên AI/ML dựa rất nhiều vào sử dụng các dự đoán có thể được trao đổi giữa các gNB trên giao diện Xn. Các thí dụ về thông tin dự đoán bao gồm quỹ đạo UE chi tiết đến ô được dự báo và trạng thái tài nguyên được dự báo (ví dụ: số lượng UE hoạt động được dự báo, số lượng kết nối RRC (Radio Resource Control: điều khiển tài nguyên vô tuyến) được dự báo và tài nguyên vô tuyến được dự báo). Ngoài ra thông tin phản hồi cho biết các hoạt động dựa trên AI/ML ảnh hưởng lên hiệu năng RAN như thế nào có thể được trao đổi giữa các gNB. Các thí dụ về thông tin phản hồi bao gồm hiệu năng UE (ví dụ: thông lượng UE đường lên/ đường xuống, trễ gói, tỷ lệ lỗi gói) và số liệu đánh giá hiệu quả năng lượng.

Để hỗ trợ trao đổi thông tin giữa các gNB cho trí tuệ RAN được hỗ trợ bởi AI/ML, các thủ tục báo hiệu mới được đưa ra cho giao diện Xn. Có hai loại thủ tục cơ bản cho giao thức Xn là: loại 1: các thủ tục cơ bản với phản hồi (thành công hay thất bại) và loại 2: các thủ tục cơ bản không có phản hồi. Thủ tục loại 1 được sử dụng cho khởi đầu báo cáo thông tin AI/ML trong đó một nút của NG-RAN yêu cầu báo cáo thông tin liên quan đến AI/ML cho một nút NR-RAN khác. Thủ tục loại 2 được sử dụng để báo cáo thông tin AI/ML trong đó một nút NR-RAN báo cáo thông tin AI/ML sau khi thủ tục khởi đầu báo cáo thông tin AI/ML thành công. Hai loại thủ tục này được minh họa trên Hình 9.22.



Hình 9.22. Thủ tục trao đổi thông tin AI/ML trên Xn cho NR-RAN

9.8.2. AI/ML cho giao diện vô tuyến gốc 5G (5G Native Interface)

AI/ML cho giao diện vô tuyến gốc được dự đoán sẽ là giải pháp quan trọng trong 6G. Vì thế, điều quan trong là ngay từ bây giờ phải nghiên cứu sử dụng AI/ML trong giao diện vô tuyến 5G để chuẩn bị cho chuẩn hóa 6G, quá trình sẽ sẽ bắt đầu vào khoảng đầu năm 2025. Mục nghiên cứu của phát hành 18 của 3GPP về AI/ML cho giao diện vô tuyến NR đánh dấu một mốc quan trọng trong phát triển các mạng tổ ong vì đây là lần đầu tiên cách tiệp cận như vậy được tiếp nhận. Mục đích chính của mục nghiên cứu này thiết lập một bộ khung tổng quát để tăng cường giao diện vô tuyến bằng cách sử dụng AI/ML, cùng với nhiều đề tài khác nhau được nghiên cứu. Các đề tài này bao gồm định nghĩa các giai

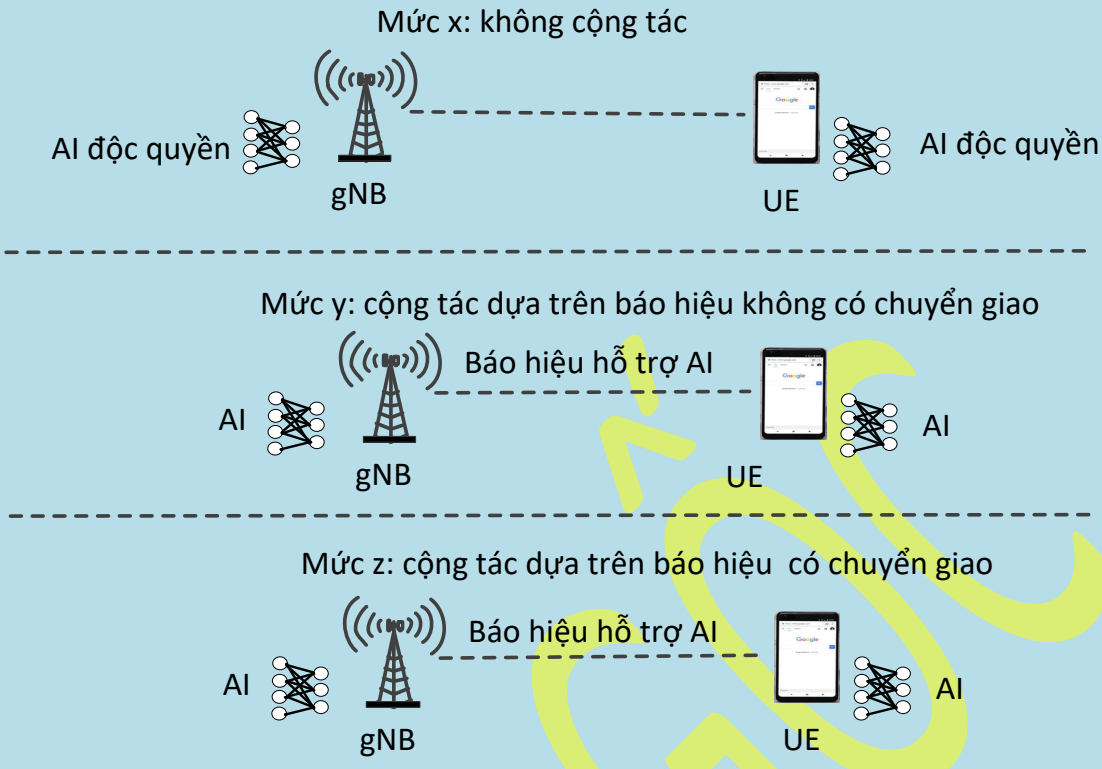
đoạn của các giải thuật AI/ML và các mức độ cộng tác giữa gNB và UE, các tập dữ liệu được yêu cầu cho đào tạo mô hình AI/ML, xác nhận và kiểm tra, quản lý vòng đời của các mô hình AI/ML. .

3GPP đã xác định ba mức độ hợp tác giữa gNB và UE:

- Cấp độ x–không cộng tác.
- Cấp độ y – hợp dựa trên báo hiệu mà có chuyển giao mô hình.
- Cộng tác dựa trên tín hiệu cấp z với chuyển giao mô hình.

Hình 9.23 minh họa các mức độ hợp tác gNB-UE x, y và z. Tại mức x, không có cộng tác giữa gNB và UE. Việc sử dụng các kỹ thuật AI/ML trong trường hợp này chỉ đơn thuần dựa trên các thực hiện và các giải pháp bản quyền. Cụ thể là, không có tăng cường đặc thù AI/ML dành riêng (ví dụ,: báo hiệu liên quan đến quản lý vòng đời) cho các hoạt động AI/ML tại mức x. Tại mức y, có cộng tác dựa trên báo hiệu không có chuyển giao. So với mức x, sự khác biệt ở chỗ đối với mức y ta có thể thay đổi giao diện vô tuyến và đưa ra báo hiệu mới để tạo điều kiện dễ dàng cho các tính năng dựa trên AI/ML hiệu quả khác, chẳng hạn đưa ra các phép đo và báo cáo mới. Mức z được định nghĩa là hợp tác dựa trên báo hiệu có chuyển giao mô hình, trong đó chuyển giao mô hình đề cập đến chuyển phát một mô hình AI/ML trên giao diện vô tuyến (hoặc các tham số của cấu trúc mô hình đã được biết tại đầu thu hoặc mô hình mới cùng với các tham số) và chuyển phát có thể chứa một mô hình đầy đủ hay một phần mô hình. Lưu ý rằng ranh giới giữa mức y và mức z được định nghĩa dựa trên định nghĩa là việc chuyển phát mô hình có trong suốt với báo hiệu 3GPP hay không. Cụ thể là, chuyển phát mô hình trong mức độ z không trong suốt với báo hiệu 3GPP, trong khi đó mức y bao gồm các trường hợp không chuyển phát mô hình và với chuyển phát mô hình trong suốt với báo hiệu 3GPP trên giao diện vô tuyến.

BẢN QUYỀN



Hình 9.23. Minh họa các mức độ hợp tác gNB-UE x, y và z

Mục nghiên cứu này được tập trung lên ba trường hợp sử dụng phục vụ như là một chương trình ban đầu để hiểu sâu sắc và toàn diện hơn về không gian giải pháp thông qua các so sánh đánh giá hiệu năng với các thực hiện dựa trên AI/ML liên quan.

9.8.3. Các trường hợp sử dụng

Trong 3GPP RAN1, đã có ba trương hợp sử dụng dựa trên AI đầu tiên: phản hồi CSI Channel Sate Information: thông tin trạng thái kênh), BM (Beam Management: quả lý búp soabfs) và các nâng cao độ chính xác định vị. Trong phát hành 19, BM và định vị đã được tiến triển đến công tác chuẩn mực. Tuy nhiên các kết quả đánh giá nhận được từ các công ty tham gia nghiên cứu liên qua đến tăng cường phản hồi CSI không đáp ứng kỳ vọng. Vì thế các công ty tham giá tiếp tục nghiên cứu làn thế nào để cải thiện tăng cao hiệu năng.

BM dựa trên AI bao gồm hai trường hợp sử dụng con (Sub Use Cases): dự báo búp sóng trong miền không gian và dự báo búp sóng trong miền thời gian. Trong miền không gian, một tập con của các búp sóng được chọn, được đặt tên là tập B (Set B), được đo để cung cấp đầu vào cho mô hình để nó dự báo các búp ứng cử cho tập A (Set A) (tập đầy đủ). Khi so sánh với các cơ chế trước đây, phương pháp này giảm kể chi phí và trễ. Trong

miền thời gian, lịch sử số liệu đo được dùng là dầu vào của mô hình để cho phép dự báo các búp ứng cử cho K khoảng thời gian tương lai, trong đó K bằng hoặc lớn hơn 1.Đáng lưu ý là trong các khoảng thời gian nhất định, UE không cần đo nhờ vậy giảm đáng kể chi phí đo.

Các búp sóng được đo nói trên có thể rộng hoặc hẹp vì các tiêu chuẩn hiện này không áp đặt bất kỳ ràng buộc lên chúng.

Các nghiên đối với định vị và phản hồi CSI dựa trên AI được tiến hành như sau. Trong định vị dựa trên AI, có hai phương pháp quan trọng: định vị AI/ML tực tiếp để tạo ra các tọa độ vị trí chính xác và định vị AI/MLđược hỗ trợ để cung cấp các kết quả đo trung gian giống như dữ liệu định thời (Timing Data). Phát hành 19 tập trung lên ba trường hợp định vị đặc thù để nhấn mạnh sử dụng các mô hình tại các phía khác nhau (UE, nút NG-RAN với lMF và gNB) mà không gây tổn hại tính riêng tư của người dùng.

Đối với phản hồi CSI dựa trên AI, hai trường hợp sử dụng con được định dạng. Trường hợp sử dụng con thứ nhất sử dụng mô hình hai phía để nén CSI trong miền không gian-thời gian nhằm giảm chi phí báo cáo trong khi yêu cầu đào tạo mô hình cộng tác. Tường hợp con thứ hai, sử dụng mô hình một phía UE để dự báo CSI trong miền thời gian cho phép giảm chi phí đo tín hiệu tham chẩn (RS: Reference Signal) và cho phép báo cáo lớp 1 (L1) giống như các phương pháp truyền thống.

Tóm lại mục tiêu của nghiên cứu ba trường hợp sử dụng nói trên như sau.

Phản hồi thông tin trạng thái kênh (CSI: Channel State Inforrmation): mục tiêu của trường hợp sử dụng này là sử dụng các kỹ thuật AI/ML để giảm chi phí CSI, cải thiện độ chính xác phản hồi và cho phép dự báo. Một lĩnh vực được tập trung là nén CSI miền không gian- thời gian bao gồm bộ mã hóa CSI dựa trên AI/ML tại UE và bộ giải mã CSI dựa trên AI/ML tại gNB. Một lĩnh vực được tập trung khác là dự báo CSI miền thời gian tại UE dựa trên các kết quả đo được trong quá khứ.

Quản lý búp sóng (Beam Management): Mục tiêu của trường hợp sử dụng này là sử dụng các kỹ thuật AI/ML để giảm chi phí quản lý búp sóng cũng như cải thiển độ chính xác lựa chọn búp sóng. Một lĩnh vực được tập trung là dự báo búp sóng đường xuống miền không gian với sử dụng một mô hình AI/ML để suy luận búp sóng đường xuống tốt nhất trong tập A (Set A) các búp đường xuống dựa trên các kết quả đo của tập B (Set B) các búp sóng đường xuống tại một thời điểm tương lai dựa trên các kết quả đo tập B các búp sóng đường xuống trong quá khứ.

Định vị (Positioning): Mục đích của trường hợp sử dụng này là sử dụng các kỹ thuật AI/ML để cải thiện độ chính xác định vị cho các kịch bản khác nhau bao gồm, ví dụ: các tình trạng NLoS nặng (Non Line of Sight: không trực xạ). Một lĩnh vực được tập trung là định vị được hỗ trợ bởi AI/ML với sử dụng một mô hình AI/ML để suy luận một thống kê đo trung gian cho định vị. Các thí dụ của thống kê đo trung gian là xác suất (LoS: Line of Sight: trực xa)/NLoS, thời gian tới, góc tới và góc khởi hành.

Tổng kết lại, hiểu biết toàn diện vai trò của 3GPP đối với hỗ trợ hợp nhất AI/ML vào giao diện vô tuyến nhận được từ các nghiên cứu nói trên sẽ đóng góp cho công việc chuẩn hóa không chỉ trong các phát hành tương lai của 5G mà còn cho các thế hệ của các hệ thống không dây di động tương lai được phát triển bởi 3GPP gồm cả 6G.

9.8.4. Quản lý mô hình

Quản lý mô hình bao gồm các thao tác khác nhau như: kích hoạt, thôi kích hoạt mô hình, chuyển mạch (Switching), cập nhật, chọn, giám sát, nhận dạng chức năng/mô hình và lùi lại (Fallback) v.v.

- Kích hoạt /thôi kích hoạt mô hình bao gồm cho phép/thôi cho phép một mô hình AI/ML cho một tính năng đạc thù được cho phép bởi AI/ML;
- Cập nhật mô hình dẫn đến quá trình cập nhật các tham số và (hoặc) cấu trúc của một mô hình;
- Nhận dạng chức năng/ mô hình đề cập đến quá trình /phương pháp nhận dạng một chức năng AI/ML cho hểu biết chung giữa mạng và UE;
- Fallback xẩy ra khi hiệu năng của mô hình hiện thời không đáp ứng được kỳ vọng, nhắc nhở UE/mạng quay trở lại cơ chế cũ ;
- Giám sát mô hình là một thủ tục thực hiện giám sát thực hiện suy luận của mô hình AI/ML;
- Chuyển mạch và chọn mô hình luôn thực hiện trong các kích bản có nhiều mô hình phục vụ một chức năng. Chọn mô hình là lựa chọn một mô hình để kích hoạt trong số số nhiều mô hình cho đối với cùng một tính năng được AI/ML cho phép. Chuyển mạch mô hình nghĩa là thôi kích hoạt một mô nhình AI/ML đang hoạt động và kích hoạt một mô hình AI/ML khác cho một tính năng đặc thù được AI/ML cho phép.

Trong phát hành 19, cần thiết phải định nghĩa báo hiệu và các chi tiết mới để tạo thuận tiện cho cho các thao tác nói trên. Tuy nhiên cũng gặp phải một số thách thức. Chẳng hạn, đối với nhận dạng, nếu nhận dạng chức năng được áp dụng cho quản lý mô hình, các thao tác nói trên sẽ dựa trên chức năng. Điều này là nẩy sinh một vấn đề nghiêm trọng: có

thể có nhiều mô hình cho một chức năng. Nếu độ chi tiết của nhận dạng không được chi tiết hóa đủ, một số mô hình có thể duy trì trong môt trạng thái không hiểu được đối với UE hoặc mạng. Ta xét kịch bản sau, có ba mô nhình được sử dụng cho dự báo tạo búp tại phía UE hoạt động tại tốc độ 3 km/h, 60 km/h và 120 km/h. Tuy nhiên mạng chỉ biết là UE có thể thực hiện dự báo sử dụng mô hình AI và không biết là các mô hình này có thể được phân biệt dựa trên tốc độ chuyển động của UE. Hậu quả là không rõ ràng đối với mạng để quyết định các thao tác quản lý mô hình. Tuy nhiên nếu nhận dạng mô hình được sử dụng, sẽ không cần giải quyết vấn đề nêu trên, vì quản lý mô hình có thể dựa trên ID của mô hình. Dù sao đi chăng nữa, sử dụng ID mô hình dẫn đến các phức tạp của chính nó. Chẳng hạn định nghĩa ID mô hình có thể là một quá trình phức tạp, nó có thể toàn cục hoặc địa phương, vật lý hoặc logic. Hơn nữa không thể loại trừ rằng một mô hình có thể phục vụ nhiều chức năng. Cho đến nay, 3GPP vẫn chưa áp đặt các giới hạn lên các phương pháp nhận dạng.

Từ góc nhìn RAN1, có vẻ như cả hai nhận dạng mô hình và nhận dạng chức năng đều có thể chấp nhận.

Đối với các trường hợp sử dụng dựa trên AI, rõ ràng rằng có các ảnh hưởng cần thiết của đặc tả để hỗ trợ quản lý mô hình. Khi mạng quản lý các mô hình phía UE, điều quan trong là cả mạng và UE phải hiểu giống nhau về thông tin mô hình chẳng hạn chức năng, các tham số và các điều kiện bổ sung. Điều này có thể giúp tận dụng báo cáo khả năng của UE. Liên quan đến các điều kiện bổ sung như các kịch bản, các site và các tập dữ liệu v.v., chúng có thể được phân loại vào hai nhóm: các điều kiện bổ sung phía mạng và các điều kiện bổ sung phía UE. Đồng bộ các điều kiện bổ sung giữa mạng và UE cũng quan trọng để đảm bảo tính nhất quán giữa đào tạo mô hình và suy luận. Ngoài ra các điều kiện bổ sung này có thể gồm chuyển mạch mô hình. Khi đã miêu tả rõ ràng các điều kiện bổ sung cho từng mô hình, việc tạo điều kện chuyển đổi sang mô hình đích dựa trên phát hiện các thay đổi môi trường sẽ trở nên khả thi hơn. Một thách thức quan trọng khác gắn liền với giám sát mô hình. Cho đến này, giám sát mô hình liên quan đến quan sát hiệu năng của mô hình đang hoạt động và đánh giá sự thích hợp của nó với kịch bản hiện thời. Điều này làm nẩy sinh các câu hỏi sau: số đo hiệu năng là gì? Cần thiết kế thủ tục giám sát như thế nào? Theo các nghiên cứu được tiến hành trong phát hành 18, có nhiều phương án khác nhau: chẳng hạn, các số đo hiệu năng có thể bao gồm: độ chính xác của suy luận, phân bố dữ liệu v.v. Trong phát hành 19, chúng ta có thể có thể giảm bớt lựa chọn phương án trong số các phương án nói trên. Những người ủng hộ cũng cho rằng có thể mở rộng cơ chế giám sát đến các mô hình không hoạt động để đánh giá chúng có thể được kích hoạt hay không. Nếu có thể áp dụng giám sát cho các mô hình không hoạt động, thì các mô hình không hoạt động này phải làm việc để tạo ra các kết quả dự báo.

Do đó những người không tán thành không hiểu được vì sao các mô hình không hoạt động lại được phép dựa báo. Tuy nhiên nếu các mô hình không hoạt động không làm việc thì vệc chọn mô hình sẽ trở nên không chắc chắn. Các vấn đề này vẫn chưa được giải quyết khiến mọi người quan tâm hơn đối với các kết quả nghiên cứu của phát hành 19. Ngoài ra áp dụng giám sát để đảm bảo sự nhất quán giữa đào tạo và suy luận mô hình cũng đã được đề xuất. Trong các trường hợp sử dụng dựa trên AI, duy trì độ chính xác dự báo bằng cách đồng bộ đào tạo mô hình và suy ulận với các kịch bản giống nhau đóng vai trò rất quan trọng. Giám sát phục vụ như là một phương tiện để cảm nhận các thay đổi kịch bản một cách gián tiếp. Dựa trên các kết quả giám sát, các thao tác quản lý mô hình khác nhau có thể được thực hiện, chẳng hạn: chọn, chuyển mạch, lùi lại (Fallback), kích hoạt hay thôi kích hoạt mô hình.

9.8.5. Đánh giá hiệu năng (Performance Evaluation)

Trong phát hành 18, ba trường hợp sử dụng mới được đưa ra trong lớp vật lý, vần đề đầu tiên và quan trọng là đánh giá các lợi ích của hiệu năng của ba trường hợp sử dụng này. Một số chỉ thị hiệu năng then chốt (KPI: Key Performance Indicators) đóng vai trò quan trọng trong đánh giá là hiệu năng suy luận, độ trễ, độ phức tạp tính toán, chi phí bổ sung và các yêu cầu phần cứng. Các phương pháp đánh giá và các số đo cho từng trường hợp sử dụng cần được thiết kế cho các tính năng và các yêu cầu đặc thù của nó.

Trong trường hợp sử dụng BM dựa trên AI, phương pháp mô phỏng hệ thống được áp dụng làm đường cơ sở. Các KPI quan trọng được xác định, chẳng hạn như, Top-K / 1 (%) là tỷ lệ phần trăm của búp sóng được hỗ trợ bởi thần đèn Top-1 (Top1 genie : một mô hình của AI) là một trong những búp sóng được dự đoán Top-K. Độ chính xác dự đoán búp sóng (%) với lề 1dB (1 dB margin) là tỷ lệ phần trăm của búp sóng được dự đoán Top-1 có L1-RSRP (Layer 1-Reference Signal Received Power: công suất tín hiệu tham chiếu thu – lớp 1) lý tưởng nằm trong phạm vi 1dB so với L1-RSRP lý tưởng của búp sóng có sự hỗ trợ của thần đèn Top-1.

Nhờ nỗ lực hợp tác của các công ty khác nhau, đối với trường hợp 1 (Case 1), người ta nhận thấy rằng bằng cách sử dụng tập B (Set B) các búp sóng cố định (trình bày gần đúng một phần tư tập A các búp sóng), đã có một sự cải thiện đáng kể độ chính xác búp sóng phát đường xuống top1 (Top1 DLTx Beam). Độ chính xác dự báo có thể đạt được 70% đến 90 %. . Đặc biệt là khi xét búp sóng phát đường xuống Top 1 với lề 1 dB, người ta đã chứng minh rằng độ chính xác vượt quá 90%.

Khi đánh giá định vị dựa trên AI, KPI chính trên tất cả các kịch bản và trường hợp sử dụng xoay quanh phần trăm Hàm phân phối tích lũy (CDF: Commulative Distribution Function) có độ chính xác theo chiều ngang, đặc biệt tập trung lên 90% (Baseline: đường cơ sở) và tùy chọn bao gồm 50%, 67%, 80%. Ngoài ra, báo cáo độ chính xác theo chiều dọc là tùy chọn. Trong bối cảnh định vị được hỗ trợ bởi AI / ML, đầu ra của các mô hình AI / ML có thể bao gồm nhiều loại thông tin khác nhau, chẳng hạn như Thời gian đến (ToA: Time of Arrival), Chênh lệch cường độ tín hiệu nhận được (RSTD: Received Signal Strength Difference), Góc khởi hành (AoD: Angle of Departure), Góc đến (AoA: Angle of Arrival), các chỉ báo LoS / NLoS, trong số những chỉ số khác.

Độ chính xác suy luận trong định vị dựa trên AI bị ảnh hưởng đáng kể bởi môi trường kênh, đặc biệt là trong các tình huống NLoS. Dựa trên đóng góp của các công ty rằng đố với hiệu suất nền tảng không tổng quátt hóa, mô hình AI/ML trải qua quá trình đào tạo và thử nghiệm bằng cách sử dụng các bộ dữ liệu từ cùng một kịch bản triển khai. Định vị dựa trên AI / ML cho thấy những cải tiến đáng kể về độ chính xác của định vị so với các phương pháp định vị phụ thuộc vào Công nghệ truy cập vô tuyến (RAT: Radio Access Technology) hiện có. Ví dụ: trong kịch bản InF-DH (Indoor Factory-Density High: mật độ cao nhà máy trong nhà) với cài đặt tham số cụm (Clustter Prarrameter) là 60%, 6m, 2m, định vị dựa trên AI / ML đạt được độ chính xác định vị ngang nhỏ hơn 1m ở CDF bằng 90%, trái ngược với >15m đối với các phương pháp định vị thông thường.

Về đánh giá hiệu suất của cải tiến phản hồi CSI dựa trên AI/ML, mô phỏng hệ thống đóng vai trò là cơ sở (Baseline). KPI và số liệu đo lường (Metrics) bao gồm:

- Việc đánh giá nén CSI bao gồm các chỉ số hiệu suất chính như tương đồng Cosin tổng quát bình phương (SGCS: Squared Generalized Cosine Similarity) và / hoặc Sai số bình phương trung bình chuẩn hóa (NMSE: Normalized Mean Square Error) để đánh giá độ chính xác của đầu ra CSI do AI / ML tạo ra. SGCS được tính toán riêng cho từng lớp, khi hạng> 1.
- Đánh giá dự đoán CSI liên quan đến việc tính toán KPI trung gian cho từng trường hợp dự đoán khi mô hình AI/ML tạo ra nhiều dự đoán.

Trong [29], kết quả mô phỏng nén CSI từ các công ty khác nhau được trình bày. Ví dụ: trong các kịch bản triển khai trong đó kịch bản A là UMi (Urban Micro: vùng vi mô trong thành phố) và kịch bản B là UMa (Urban Mcro: vùng vĩ mô trong thành phố), kịch bản A là UMa và kịch bản B là UMi, hoặc kịch bản A là UMa và kịch bản B là InH (Indoor Hospot: điểm nóng trong nhà), sự suy giảm hiệu suất từ -1,69% đến -31,6% được quan sát thấy. Đối với dự đoán CSI, Nếu tốc độ UE B là 30km/h, 60km/h hoặc 120km/h, hoặc

nếu tốc độ UE B là 10km/h và tốc độ UE A là 60km/h hoặc 120km/h, hiệu suất giảm từ trung bình đến đáng kể (giảm -2,01% đến -76,85%).

Dựa trên các quan sát nêu trên, kết quả đánh giá hiệu năng nén CSI và dự báo CSI rất không thỏa mãn. Vì thế trong phát hành chuẩn hóa không được tiến hành tiếp, thay vào đó nẫn tập trung duy trì trạng thái của các mục nghiên cứu.

9.9. CÁC TRƯỜNG HỢP SỬ DỤNG AI/ML CHO GIAO DIỆN VÔ TUYẾN CỦA 5G NR

9.9.1. Trường hợp sử dụng: phản hồi CSI

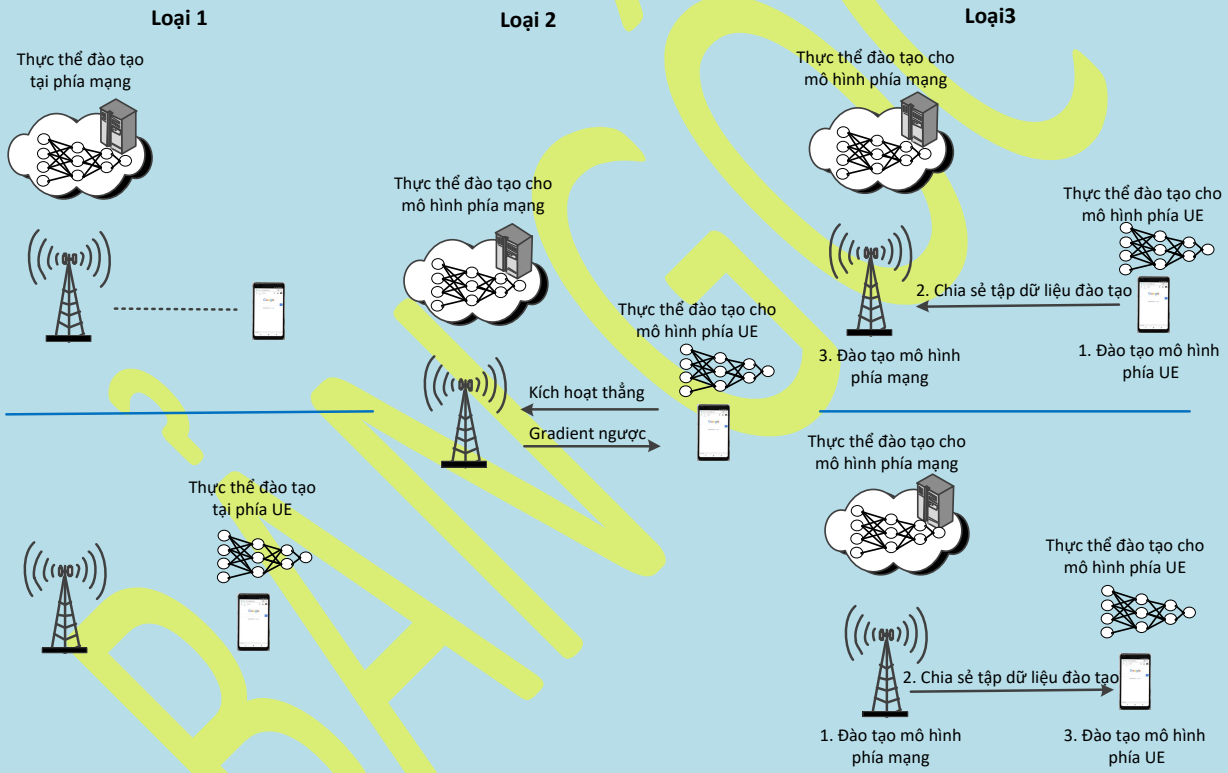
CSI (Channel State Information: thông tin trạng thái kênh) đề cập đến thông tin của kênh không dây đa đường giữa nút 5G gNB và UE. UE có thể đo tín hiệu tham chiếu đường xuống, tính toán CSI đường xuống và cung cấp báo cáo CSI cho gNB, do đó tạo điều kiện thuận lợi cho việc truyền đường xuống. Tận dụng các thuật toán dựa trên AI/ML có thể nâng cao hơn nữa phản hồi CSI, mang lại những lợi thế như giảm chi phí và cải thiện độ chính xác. 3GPP đã nghiên cứu hai trường hợp sử dụng con đại diện của việc sử dụng các thuật toán dựa trên AI/ML để tăng cường phản hồi CSI: nén CSI miền tần số không gian và dự đoán CSI miền thời gian bằng mô hình phía UE.

9.9.1.1. Nén CSI miền tần số không gian

Trong nén CSI dựa trên AI/ML, UE sử dụng bộ mã hóa CSI dựa trên AI / ML để tạo thông tin phản hồi CSI, trong khi bộ giải mã CSI dựa trên AI/ML tương ứng tại gNB được sử dụng để tái tạo CSI từ dữ liệu phản hồi nhận được. Đây là một ví dụ về mô hình AI/ML hai phía, trong đó hoạt động suy luận được phân chia giữa UE và gNB. Mô hình AI / ML hai phía này có thể được sử dụng để nén ma trận kênh thô do UE ước tính hoặc ma trận mã hóa trước có nguồn gốc từ ma trận kênh thô. Đáng chú ý, nén ma trận mã hóa trước phù hợp với bộ khung phản hồi CSI dựa trên bảng mã hiện có được đặc tả cho giao diện vô tuyến NR, do đó thu hút nhiều sự quan tâm hơn trong quá trình nghiên cứu 3GPP.

Sử dụng mô hình AI/ML hai phía cho giao diện vô tuyến mang lại vô số thách thức. Thách thức đầu tiên liên quan đến việc đào tạo mô hình AI/ML hai phía. Trong bối cảnh này, 3GPP đã điều tra ba loại đào tạo liên quan đến các mức độ hợp tác khác nhau giữa mạng và UE, như được minh họa trong Hình 9.24. Trong kịch bản đào tạo đầu tiên, được chỉ định là 'Loại 1', bộ mã hóa và bộ giải mã của mô hình hai phía được đào tạo chung ở một phía. Nếu đào tạo này được thực hiện ở phía mạng, mô hình bộ mã hóa nhận được có thể được chuyển sang UE và ngược lại. Chuyển sang 'Loại 2', bộ mã hóa và bộ giải mã của mô hình hai phía trải qua quá trình đào tạo lần lượt ở phía UE và phía mạng. Chính

xác, bộ mã hóa và bộ giải mã được đào tạo chung trong cùng một vòng lặp để lan truyền thẳng (Feedforward) và lan truyền ngược bằng cách trao đổi kích hoạt thẳng và gradient ngược giữa UE và mạng. Cuối cùng, 'Loại 3' bao gồm việc đào tạo riêng biệt của bộ mã hóa và bộ giải mã trong mô hình hai phía, được thực hiện trong các phiên đào tạo riêng biệt ở phía UE và mạng. Việc đào tạo riêng biệt có thể bắt đầu ở phía UE hoặc phía mạng. Lấy ví dụ về đào tạo riêng biệt bắt đầu từ phía UE. Trong giai đoạn đầu, một thực thể phía UE đào tạo mô hình bộ mã hóa CSI. Sau đó, thực thể phía UE chia sẻ một bộ dữ liệu đào tạo- bao gồm đầu ra bộ mã hóa và CSI mục tiêu — với một thực thể phía mạng. Tập dữ liệu này sau đó được thực thể phía mạng tận dụng để đào tạo mô hình giải mã CSI của nó.



Hình 9.24. Các kiểu đào tạo mô hình AI/ML cho nén CSI sử dụng mô hình hai phía

Các phương pháp đào tạo khác nhau mang lại những lợi thế và thách thức khác nhau. Trong số này, đào tạo loại 1 hứa hẹn mang lại hiệu suất vượt trội, nhưng việc triển khai và quản lý các mô hình có thể phức tạp. Lấy loại đào tạo 1 ở phía mạng làm ví dụ. Thực thể phía mạng sẽ liên quan đến nhà cung cấp loại UE được nhắm mục tiêu trong quá trình đào tạo để mô hình mã hóa CSI được đào tạo tương thích với việc triển khai UE. Tuy nhiên, nỗ lực này đòi hỏi phải tiết lộ thông tin độc quyền của nhà cung cấp UE cho phía

mạng, đây có thể là một thách thức đáng kể. Mặt khác, đào tạo loại 2 giữ tính bảo mật của thông tin độc quyền ở cả hai phía mạng và UE, vì thông tin mô hình vẫn nằm trong các miền tương ứng của chúng. Tuy nhiên, hai thực thể đào tạo tương ứng cần phối hợp các lần lặp lại đào tạo của họ để trao đổi các kết quả kích hoạt thẳng và lan truyền ngược, điều này có thể dẫn đến nỗ lực phối hợp đáng kể và chi phí. Đào tạo loại 3 nổi lên như một lựa chọn không chỉ bảo vệ thông tin độc quyền mà còn loại bỏ sự cần thiết phải cộng tác trong quá trình lặp lại đào tạo vì sự phối hợp có thể diễn ra bên ngoài quá trình đào tạo.

Sự phức tạp đào tạo vốn có trong mô hình AI/ML hai phía cho giao diện không vô tuyến càng được kết hợp bởi sự cần thiết phải có khả năng tương tác và tương thích của nhiều nhà cung cấp. Bộ giải mã CSI nằm ở gNB cần tương thích với các bộ mã hóa CSI khác nhau tại UE và ngược lại. Trong các trường hợp mô hình bộ giải mã CSI phổ biến được sử dụng cho nhiều mô hình bộ mã hóa CSI, thực thể đào tạo phía mạng- theo loại đào tạo 1 hoặc 2 -hải phối hợp với các nhà cung cấp UE để thực hiện các nỗ lực đào tạo chung. Đáng chú ý, việc phát hành một loại UE mới có khả năng kích hoạt đào tạo lại trên tất cả các nhà cung cấp. Những thách thức tương tự tồn tại trong đào tạo loại 1 hoặc 2 khi một mô hình bộ mã hóa CSI được chia sẻ được sử dụng cho nhiều mô hình bộ giải mã CSI. Những vấn đề này có thể được giảm thiểu trong đào tạo loại 3. Đặc biệt, nếu một mô hình bộ giải mã CSI chung được sử dụng cho nhiều mô hình bộ mã hóa CSI (hoặc ngược lại), việc đào tạo lại do sự phát hành một loại UE mới có thể chỉ liên quan đến nhà cung cấp UE liên kết và nhà cung cấp mạng do bản chất đào tạo riêng biệt trong loại đào tạo 3. Tóm lại, đối với phản hồi CSI dựa trên mô hình AI / ML hai phía, đào tạo loại 3 nổi lên như một cách tiếp cận thực tế và khả thi hơn so với đào tạo loại 1 và 2

Một cân nhắc quan trọng khác trong thiết kế nén CSI dựa trên AI/ML là khả năng tổng quát hóa mô hình AI/ML trên nhiều kịch bản và cấu hình. Cụ thể, thiết kế cần xem xét các kịch bản đa dạng, bao gồm các kịch bản triển khai khác nhau (ví dụ: vĩ mô đô thị, vi mô đô thị và trong nhà), phân phối UE, tần số sóng mang, v.v. Mô hình AI / ML cũng phải có thể mở rộng trên nhiều cấu hình khác nhau, chẳng hạn như băng thông kênh khác nhau, độ chi tiết tần số, tải trọng phản hồi CSI, cũng như các biến thể trong bố cục và số cổng ăng-ten. Các cấu hình đa dạng này có thể ảnh hưởng đến thiết kế mô hình AI/ML vì chúng có thể dẫn đến các khía cạnh khác nhau của đầu vào và đầu ra mô hình.

9.9.1.2. Dự đoán CSI miền thời gian với mô hình phía UE

Một thách thức trong bộ khung báo cáo CSI cũ của NR liên quan đến độ trễ thời gian giữa thời gian mà CSI được báo cáo tương ứng với thời điểm mà gNB thực sự sử dụng báo cáo CSI. Sự chậm trễ thời gian này dẫn đến tình huống CSI được báo cáo trở nên lỗi

thời, một hiện tượng thường được gọi là lão hóa kênh. Đáng chú ý, tốc độ mà CSI được báo cáo trở nên lỗi thời được khuếch đại bởi tốc độ UE cao hơn. Điều này trở nên đặc biệt rõ rệt trong bối cảnh các kịch bản đa đầu vào nhiều đầu ra nhiều người dùng (MU-MIMO), đặc biệt là trong các triển khai MIMO lớn. Hiệu suất của MU-MIMO đã được quan sát thấy giảm đi khi UE di chuyển ở tốc độ trung bình đến cao. Tận dụng các thuật toán AI/ML để dự đoán CSI nổi lên như một kỹ thuật đầy hứa hẹn để chống lại tác động của CSI lỗi thời.

Trái ngược với nén CSI dựa trên AI/ML, đòi hỏi mô hình hai phía, dự đoán CSI dựa trên AI/ ML trong miền thời gian có thể sử dụng mô hình một phía. Đào tạo mô hình một phía này có thể được thực hiện bởi một nhà cung cấp duy nhất và suy luận sau đó có thể được thực hiện bởi một phía (gNB hoặc UE). Xem xét khối lượng công việc trong phát hành 18, 3GPP đã tập trung chiến lược vào mô hình một phía UE để dự đoán CSI trong miền thời gian. Trong thiết lập này, đầu vào cho mô hình một phía bao gồm một chuỗi các kết quả đo CSI quá khứ do UE thực hiện. Kết quả đầu ra của mô hình này là CSI được dự báo cho một trường hợp thời gian trong tương lai, được dự đoán bởi UE.

Từ quan điểm của các tiêu chuẩn 3GPP, dự kiến rằng chúng ta có thể sử dụng phần lớn bộ khung CSI hiện có để hỗ trợ dự đoán CSI. Đặc biệt, mô hình AI/ML LCM để dự đoán CSI phía UE ở một mức độ lớn có thể sử dụng lại những gì được xác định cho các trường hợp sử dụng một phía UE khác vì tác động đặc tả kỹ thuật của các mô hình AI/ML một phía UE đã được nghiên cứu để quản lý búp sóng và định vị, như được mô tả trong các phần sau.

9.9.2. Trường hợp sử dụng: quản lý búp sóng (Beam Management)

Tạo búp được sử dụng phổ biến trong các hệ thống thông tin di động 5G mặt đất và không gian. Ngoài ra nó cũng được sử dụng cho RIS (Reconfiguration Intellegence Surface: mặt phẳng thông minh khả lập cấu hình) để hướng tới vô tuyến thông minh và khả lập trình trong 6G.

Chức năng quản lý búp sóng (Beam Management) trong NR được sử dụng để hỗ trợ tạo búp (Beamforming). Nó đặc biệt cần thiết cho các hệ thống sóng milimet 5G dựa trên tạo búp tia tương tự (Analog Beamforming). Trong quy trình cơ bản để quản lý tạo búp đường xuống, UE đo tín hiệu tham chiếu được liên kết với búp phát của gNB và kiểm tra các búp thu khác nhau của UE đối với mỗi búp phát của gNB để tìm một cặp búp đường xuống phù hợp. Quá trình này có thể tốn thời gian và kéo theo chi phí đáng kể về các tín hiệu tham chiếu. Các thuật toán dựa trên AI / ML mang lại tiềm năng nâng cao chức năng

quản lý búp sóng, mang lại những lợi thế bao gồm giảm chi phí, giảm thiểu độ trễ và cải thiện độ chính xác trong việc lựa chọn búp.

3GPP đã nghiên cứu hai trường hợp sử dụng con đại diện liên quan đến việc áp dụng các thuật toán dựa trên AI/ML để quản lý búp sóng. Chúng được gọi là 'dự đoán búp sóng đường xuống miền không gian' và 'dự đoán búp sóng đường xuống miền thời gian'.

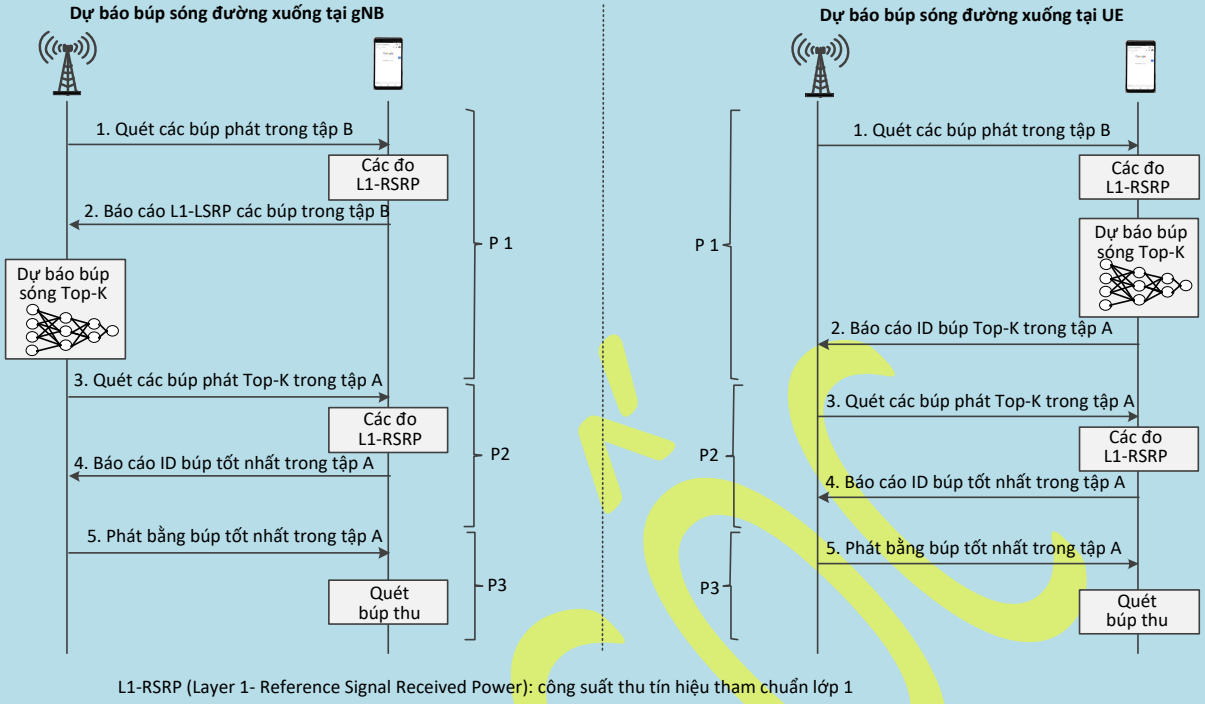
Dự đoán búp sóng đường xuống miền không gian tận dụng kết quả đo lường từ một tập hợp được chỉ định', được ký hiệu là 'tập B', để dự đoán búp sóng tốt nhất trong một tập hợp búp sóng đường xuống khác, được gọi là tập A' vào thời điểm hiện tại.

Dự đoán búp sóng đường xuống miền thời gian khai thác các kết quả đo lịch sử bắt nguồn từ 'tập B' để dự đoán búp sóng tốt nhất trong 'tập A' cho một hoặc nhiều trường hợp thời gian trong tương lai.

Đáng chú ý, 'Tập B' có thể tạo thành một tập hợp con của 'Tập A' hoặc hai tập hợp có thể khác nhau (ví dụ: 'Tập B' bao gồm các búp rộng trong khi 'Tập A' bao gồm các búp hẹp). Ngoài ra, trong bối cảnh dự đoán búp sóng đường xuống miền thời gian, 'Tập A' và 'Tập B' cũng có thể giống nhau.

Đầu vào điển hình cho mô hình AI/ML để dự đoán búp sóng đường xuống miền không gian hoặc miền thời gian là các kết quả đo công suất nhận được tín hiệu tham chiếu lớp 1 (L1-RSRP: Layer1- Received reference Power) của các búp sóng trong 'tập B.' Đầu ra điển hình từ mô hình AI / ML là các búp sóng top K được dự đoán trong ' tập A.' Đào tạo và suy luận mô hình AI / ML có thể nằm ở phía gNB hoặc ở phía UE. Trong các tình huống suy luận AI/ML xảy ra ở phía UE, UE cần báo cáo (các) búp sóng dự đoán của nó cho gNB. Ngoài ra, khi suy luận AI / ML diễn ra ở phía gNB, UE được yêu cầu báo cáo các phép đo L1-RSRP của nó cho các búp sóng trong ' tập B' cho gNB.

Để đảm bảo sự rõ ràng về dự đoán búp sóng đường xuống dựa trên AI / ML, bây giờ chúng ta trình bày chi tiết về cách dự đoán búp sóng có thể được tích hợp vào khuôn khổ quản lý búp sóng của NR hiện có. Nhớ lại rằng bộ khung hiện tại bao gồm ba quy trình được gọi là P1 (thiết lập cặp búp sóng ban đầu), P2 (tinh chỉnh búp sóng phát) và P3 (tinh chỉnh búp sóng nhận). Hình 9.25 minh họa về các quy trình quản lý búp sóng đường xuống với dự đoán búp sóng dựa trên AI / ML ở phía gNB và ở phía UE.



Hình 9.25. Quy trình quản lý búp sóng đường xuống với dự đoán búp sóng dựa trên AI/ML ở phía gNB (trái) và ở phía UE (phải)

Lấy dự đoán búp sóng ở phía gNB làm ví dụ. Đầu tiên, gNB quét qua các búp sóng phát khác nhau trong 'tập B', được đo bởi UE. Sau đó, UE truyền các kết quả đo L1-RSRP cho các búp sóng trong 'tập B' trở lại gNB. Sau khi nhận được báo cáo, gNB sử dụng các giá trị L1-RSRP nhận được làm đầu vào cho mô hình AI / ML của nó, sau đó đưa ra dự đoán về các búp sóng top-K trên cùng trong 'tập A.' Dựa trên kết quả suy luận, gNB sau đó quét các búp sóng top- K trên cùng được dự đoán trong 'tập A', UE đo các búp sóng này để xác định búp sóng phát tốt nhất mang lại giá trị L1-RSRP cao nhất. Sau đó, UE báo cáo ID của búp sóng tốt nhất trở lại gNB. Nếu có nhu cầu, gNB có tùy chọn kích hoạt quy trình P3. Trong giai đoạn này, gNB tiếp tục sử dụng búp sóng phát tốt nhất, trong khi UE thăm dò các búp sóng thu được khác nhau để xác định búp sóng phù hợp nhất.

9.9.3. Trường hợp sử dụng: định vị (positioning)

Các phương pháp định vị 5G NR hiện tại thường dựa trên hình học, bao gồm hai bước chính: 1) tiến hành đo tín hiệu vô tuyến và 2) tính toán ước tính vị trí bằng cách giải một hệ phương trình phi tuyến tính thiết lập mối quan hệ giữa vị trí của UE và các phép đo. Độ chính xác của các phương pháp định vị dựa trên hình học phụ thuộc rất nhiều vào tính khả dụng của các phép đo được liên kết đường trực xạ (LOS). Trong các tình huống liên

quan đến điều kiện LOS yếu hoặc môi trường đa đường dày đặc, chẳng hạn như lắp ráp thiết bị bên trong nhà máy trong nhà (Indoor Factory), độ chính xác của các phương pháp dựa trên hình học có xu hướng giảm sút. Các thuật toán dựa trên AI / ML có thể nâng cao độ chính xác định vị trong một loạt các tình huống, bao gồm cả những tình huống được đặc trưng bởi các điều kiện không LOS (NLOS) phổ biến.

3GPP đã nghiên cứu hai trường hợp sử dụng con đại diện liên quan đến việc áp dụng các thuật toán dựa trên AI / ML để định vị. Chúng được gọi là 'định vị AI / ML trực tiếp' và 'định vị hỗ trợ AI/ML'.

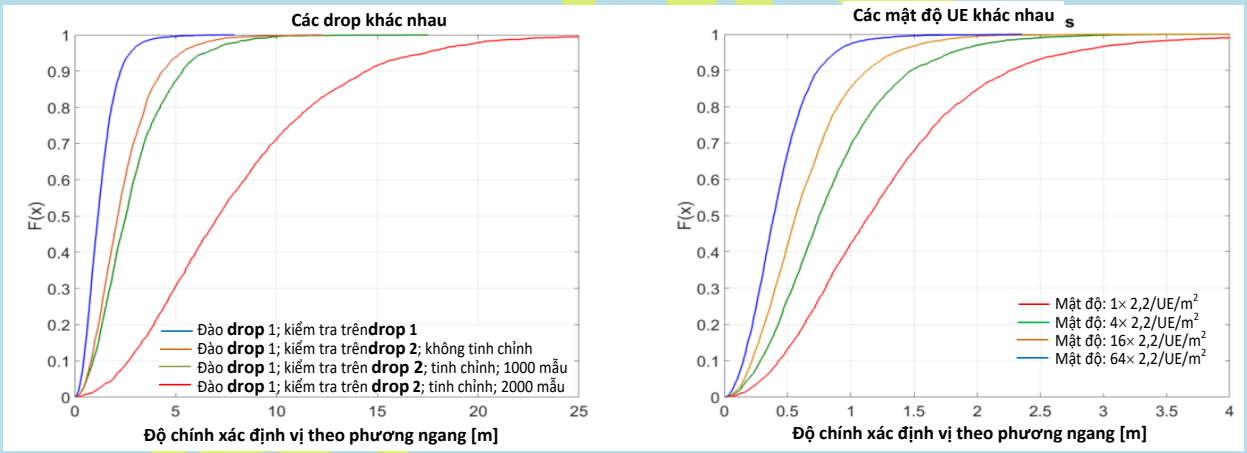
Định vị AI / ML trực tiếp sử dụng mô hình AI / ML để xác định trực tiếp vị trí của UE. Ví dụ: điều này có thể bao gồm định vị dựa trên dấu vân tay (Fingerprinting) bằng cách sử dụng các quan sát kênh, chẳng hạn như phản ứng xung kim kênh (CIR: Channel Impulse Response) hoặc hồ sơ độ trễ công suất (PDP: power delay profile), làm đầu vào cho mô hình AI / ML.

Định vị được hỗ trợ bởi AI / ML liên quan đến việc tận dụng mô hình AI / ML để tạo ra thống kê đo lường trung gian, là công cụ trong việc định vị. Điều này có thể bao gồm các phép đo như xác suất LOS / NLOS, góc đến/ góc khởi hành hoặc thời gian đến.

Đào tạo và suy luận mô hình AI/ML có thể nằm ở phía UE, phía chức năng quản lý vị trí (LMF: Location Management Function) hoặc phía gNB. Tùy thuộc vào vai trò của UE, LMF và gNB trong các quy trình định vị, 3GPP tập trung vào ba loại phương pháp định vị. Loại đầu tiên là định vị dựa trên UE, trong đó bản thân UE thực hiện định vị AI / ML trực tiếp hoặc định vị có sự hỗ trợ của AI / ML. Loại thứ hai là định vị dựa trên LMF có sự hỗ trợ của UE, trong đó UE cung cấp hỗ trợ cho LMF trong việc ước tính vị trí của UE. Trong kịch bản này, LMF có thể thực hiện định vị AI / ML trực tiếp hoặc UE có thể tham gia vào định vị có sự hỗ trợ của AI / ML. Loại thứ ba là định vị có sự hỗ trợ của gNB, trong đó gNB cung cấp hỗ trợ cho LMF trong việc ước tính vị trí của UE. Trong trường hợp này, LMF có thể thực hiện định vị AI / ML trực tiếp hoặc gNB có thể tham gia vào định vị được AI / ML hỗ trợ.

Khám phá khả năng tổng quát hóa của các mô hình AI/ML là một lĩnh vực nghiên cứu chính trong các trường hợp sử dụng trong phát hanhd 18 của 3GPP. Liên quan đến định vị dựa trên AI / ML, các khía cạnh khác nhau đã được xem xét để điều tra khả năng tổng quát hóa mô hình. Ví dụ: tập dữ liệu đào tạo và thử nghiệm được tạo theo các lần thả (Drop) khác nhau, các tham số lộn xộn khác nhau hoặc lỗi thời gian riêng biệt. Để giúp hiểu vấn đề này, phần bên trái của hình 10 trình bày một nghiên cứu điển hình minh họa khả năng khái quát hóa định vị AI/ML trực tiếp, sử dụng thiết lập mô phỏng 3GPP [30].

Kịch bản triển khai mô phỏng là nhà máy trong nhà (Indoor Factory) với điều kiện NLOS nặng. Mô hình AI/ML mô phỏng có dạng mạng neuron tích chập (CNN), trong đó CIR đóng vai trò là đầu vào mô hình và vị trí UE dự đoán tạo thành đầu ra mô hình. Hai tập dữ liệu riêng biệt, được ký hiệu là Drop 1 1' (thả 1) và 'Drop 1 2' (thả 2), được tạo bằng cách sử dụng các giá trị hạt giống ngẫu nhiên khác nhau và được sử dụng để đào tạo và thử nghiệm mô hình tương ứng. Về bản chất, hai Drop này có thể được hình dung là đại diện cho nhà máy trong nhà khác nhau có cùng sắp đặt lộn xộn. Kết quả cho thấy sự sụt giảm đáng kể về độ chính xác của định vị AI/ML trực tiếp khi đào tạo và thử nghiệm được thực hiện trên các Drop khác nhau. Một biện pháp khắc phục hợp lý cho vấn đề này là sử dụng tinh chỉnh mô hình AI / ML, như cũng được thể hiện trong phần bên trái của Hình 9.26. Điều đáng chú ý là mô hình ban đầu được đào tạo với tập dữ liệu gồm 16.000 mẫu. Do đó, việc sử dụng 1.000 (hoặc 2.000) mẫu tinh chỉnh tương đương với 6,25% (hoặc 12,5%) trong số 16.000 mẫu cần thiết để đào tạo mô hình AI/ML ban đầu.



Hình 9.26 Độ chính xác định vị của định vị AI / ML trực tiếp dưới các 'drop' khác nhau (trái) và mật độ UE khác nhau (phải)

Thu thập dữ liệu là một lĩnh vực khám phá quan trọng khác trong các trường hợp sử dụng được nghiên cứu trong phát hành 18 của 3GPP. Khi xem xét định vị dựa trên AI / ML, có một kỳ vọng trực quan rằng mật độ dữ liệu thu thập được tăng lên sẽ dẫn đến độ chính xác của định vị được nâng cao. Trực giác này được minh họa định lượng trong phần bên phải của Hình 9.26. Xu hướng rõ ràng cho thấy rằng việc cải thiện độ chính xác của định vị đi kèm với yêu cầu cao về thu thập dữ liệu. Điều này nhấn mạnh tầm quan trọng của các chiến lược thu thập dữ liệu để thực hiện hiệu quả định vị dựa trên AI/ML.

9.10. TỔNG KẾT

Chương này trình bày các hoạt động và nghiên cứu tích hợp AI/ML vào các hệ thống truyền thông di động 5G trong 3GPP, nhấn mạnh các tính năng quan trọng được đưa vào phát hành 19 trong các lĩnh vực khác nhau. Các tính năng này được phân loại thành hai lĩnh vực: "AI cho mạng" và "mạng cho AI". Kết hợp hai mô hình này cần tập trung nhiều tài nguyên và đòi đòi hỏi thay đổi căn bản trong kiến trúc mạng và các giao thức. Tuy nhiên xét đến sự hữu ích và sự chấp nhận rộng rãi các giải thuật AI/ML, các cơ quan chuẩn hoa cần cung cấp các khung hỗ trợ các hai mô hình này. Các nỗ lực đang được tiến hành trong phát hành 20 của 3GPP và các hệ thống 6G. Các nghiên cứu tương lai sẽ thúc đẩy mạnh quá trình tích hợp AI/ML và các mạng tương lai.

Chương này cũng trình bày các thành quả nghiên cứu mới nhất liên quan đến FL và quản lý búp sóng (BM). Chương này cũng nhấn mạnh các thức phải kế hệ thống để hỗ trợ tốt hơn tích hợp AI và truyền thông di động 5G.

Cuối chương này các trường hợp sử dụng AI/ML cho giao diện vô tuyến của 5G NR được trình bày.

BẢN GỐC

Chương 10

ỨNG DỤNG AM/ML CHO CÁC DỊCH VỤ PHƯƠNG TIỆN TRONG HỆ THỐNG TRUYỀN THÔNG DI ĐỘNG KHÔNG DÂY

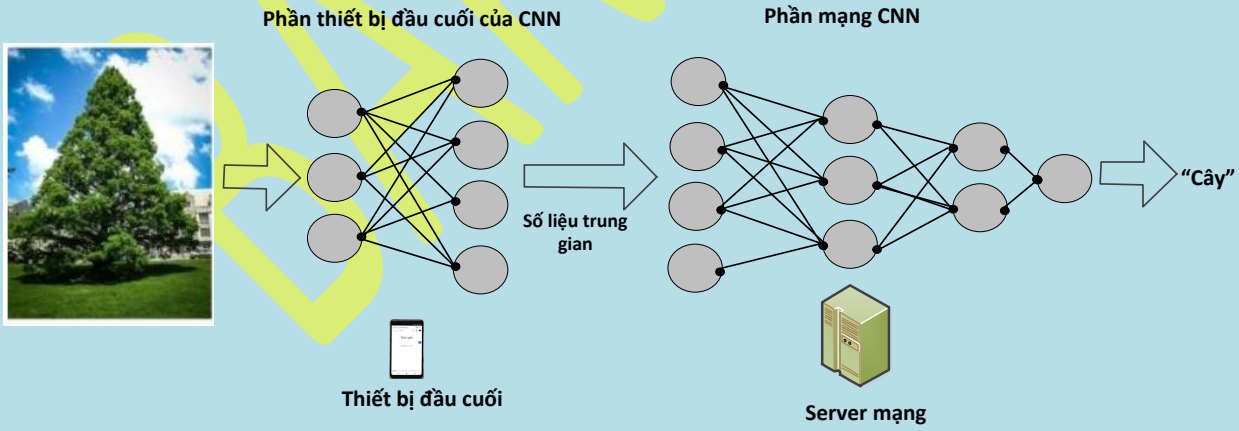
10.1. TỔNG QUAN

Các ứng dụng AI/ML ngày càng đòi hỏi tính toán chuyên sâu hơn, bộ nhớ và thiêu thụ công suất lớn hơn. Trong khi đó các thiết bị đầu cuối trong các hệ thống truyền thông di động không dây thường chỉ có khả năng tiêu thụ công suất giới hạn, khả năng tính toán và bộ nhớ giới hạn để thực hiện suy luận AI/ML tại chỗ hoàn toàn ngoại tuyến (suy luần thại thế bị đầu cuối di đông). Rất nhiều các ứng dụng AI/ML, chẳng hạn nhận dạng hình ảnh, có xu thể giảm tải (Offloading) xử lý suy luận từ thiết bị di động đến trung tâm dữ liệu internet (IDC: Internet Data Center). Chẳng hạn các ảnh được chụp bởi điện thoại thông minh thường được xử lý tại đám mây AI/ML thước khi được hiển thị cho người

dùng chụp chúng. Tuy nhiên các nhiệm vụ suy lận AI/ML dựa trên đám mây cũng cần phải xem xét đến áp lực tính toán tại các IDC, tốc độ số liệu/trễ yêu cầu và yêu cầu bảo vệ tính riêng tư.

Ảnh và video là các nguồn dữ liệu lớn nhất hiện nay trên internet. Các video chiếm trên 70% lưu lượng hàng ngày của internet. Các mô hình mạng tế bào thần kinh tích chập (CNN: Convolutional Neural Network) đang được sử dụng rộng rãi cho các nhiệm vụ nhận dạng hình ảnh/video trên các thiết bị di động, chẳng hạn phân loại hình ảnh, phân đoạn hình ảnh, phát hiện và định vị đối tượng, xác thực mặt, nhận dạng hành động, tăng cường hình ảnh (Enhanced Photography), VR/AR (Virtual Reality/ Augremented Reality: thực tế ảo/ thực tế tăng cường), các trò chơi video. Trong khi đó suy luận mô hình CNN đòi hỏi khả năng tính toán và nhớ lớn.

Rất nhiều nghiên cứu cho thấy rằng suy luận AI/ML cho xử lý hình ảnh với kết hợp thiết bị đầu cuối di động-mạng giảm áp lực tính toán, dấu chân bộ nhớ (Memory Foootprint), lưu giữ, công suất và tốc độ số liệu được yêu cầu trên thiết bị, giảm trễ đầu cuối đầu cuối và tiêu thụ năng lượng, cải thiện độ chính xác, hiệu năng và tính riêng tư đầu cuối đầu cuối đầu cuối so với các tiếp cận tại chỗ trên mỗi phía. Sơ đồ phân chia nhận dạng hình ảnh AI/ML được mô tả trên hình 10.1. CNN được phân chia thành hai phần dựa trên nhiệm vụ nhận dạng hình ảnh và môi trường hiện thời. Mục đích phân chia là chuyển tải các phần đòi hỏi tính toán và năng lượng cao đến server mạng. Serrver mạng thực hiện các lớp CNN còn lại. Trong khi mô hình được phát triển và được yêu cầu, hoạt động phân chia AI/ML vẫn dựa trên mô hình cũ.



Hình 10.1. Thí dụ phân chia nhận dạng ảnh AI/ML

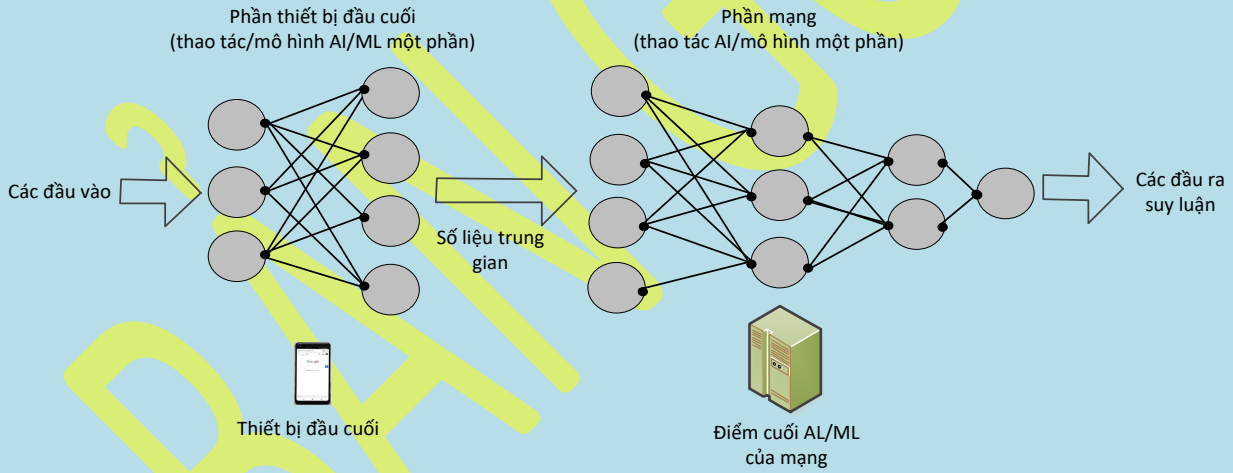
Do đặc điểm của một số thuật toán trong giai đoạn đào tạo mô hình, mỗi mô hình có một mức độ mạnh nhất định. Nên, nếu có lỗi trong quá trình truyền dữ liệu trung gian, mô hình có dung sai nhất định và vẫn có thể đảm bảo độ chính xác của kết quả suy luận. Vì

kết quả suy luận cần được chuyển tiếp đến UE (User Equipement: thiết bị người dùng) nên độ tin cậy của việc truyền kết quả suy luận cần được đảm bảo.

Các hệ thống truyền thông di động di động không dây có thể hỗ trợ ba kiểu hoạt động AI/ML:

- Phân chia hoạt động AI/ML giữa các điểm cuối;
- Phân bố mô hình/dữ liệu AI/ML và chia sẻ trên toàn bộ hệ thống
- Học phân tán/học liên kết (Distributed/Federated Learning) trên hệ thống.

Sơ đồ suy luận AI/ML được phân chia (Split AM/ML Inference) được mô tả trên hình 10.2. Hoạt động của AI/ML/mô hình được phân chia vào nhiều bộ phận tùy theo nhiệm vụ hiện thời của môi trường. Mục đích là giảm tải (Offloading) các phần đòi hỏi tính toán chuyên sâu và nhiều năng lượng vào các điểm cuối mạng trong khi để lại các phần nhậy cảm trễ và sự riêng tư tại các thiết bị đầu cuối. Thiết bị thực hiện thao tác/mô hình đến các bộ phận/lớp đặc thù sau đó gửi số liệu trung gian đến điểm cuối mạng. Điểm cuối mạng thực hiện các phần còn lại và cung cấp các kết quả suy luận ngược về thiết bị này.



Hình 10.2. Thí dụ về suy luận

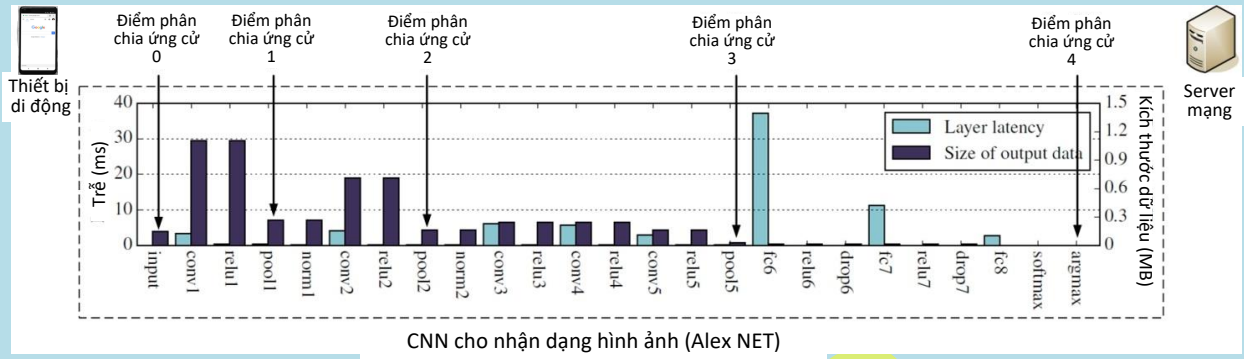
Các đầu cuối di động đa chức năng có thể cần chuyển mạch mô hình AI/ML để đáp ứng các thay đổi nhiệm vụ và môi trường. Điều kiện chọn mô hình đáp ứng là các mô hình này cần được chọn lựa phải khả dụng cho thiết bị di động. Tuy nhiên, do các mô hình AI/ML ngày càng đa dạng và tài nguyên được lưu giữ có hạn trong UE, nên không thể nạp trước tất cả các mô hình AI/ML ứng cử vào thiết bị. Phân bố mô hình trực tuyến (tải xuống mô hình mới) là cần thiết, trong đó một mô hình AI/ML có thể được phân bố từ một điểm cuối của mạng đến các thiết bị khi chúng cần thích ứng đến các nhiệm vụ và

các môi trường bị thay đổi. Cho mục đích này, hiệu năng mô hình tai UE cần được thường xuyên giám sát.

Sơ đồ học liên kết (FL: Federated Learning) được mô tả trên Hình 4.1. trong phần 4. Server đám mây đào tạo một mô hình toàn cục bằng cách kết hợp các mô hình địa phương được đào tạo một phần. Trong mỗi lần lặp đào tạo, UE thực hiện đào tạo dựa trên mô hình được tải xuống. Sau đó UE báo cáo các kết quả đào tạo tạm thời đến server đám mây thông qua các kênh đường lên của hệ thống truyền thông di động. Server tổng hợp các kết quả đào tạo tạm thời từ các UE và cập nhật mô hình toàn cục. Sau đó mô hình toàn cục đã cập nhật được phân bố trở lại các UE và các UE thực hiện đào tạo cho lần lặp tiếp theo.

10.2. LỰA CHỌN ĐIỂM PHÂN CHIA HOẠT ĐỘNG CỦA AI/ML GIỮA THIẾT BỊ ĐẦU CUỐI DI ĐỘNG VÀ MẠNG

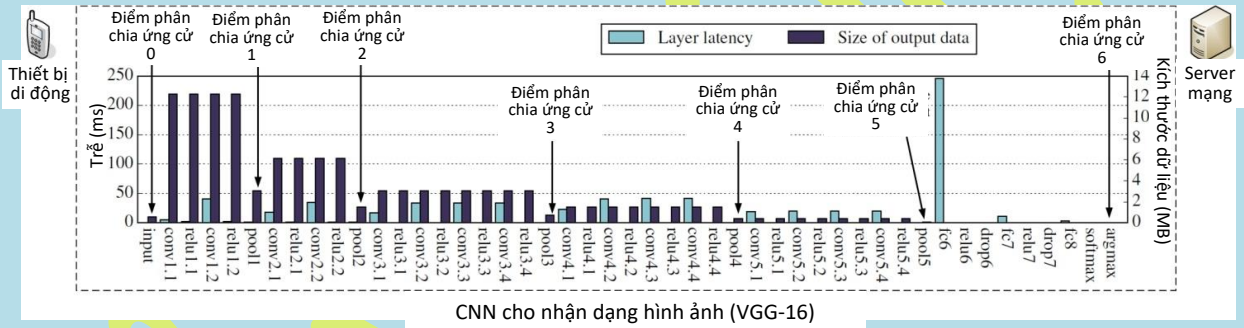
Các thuật toán nhận dạng hình ảnh AI/ML được phân chia có thể được phân tích dựa trên tính toán và đặc điểm dữ liệu của các lớp trong CNN. Hình 10.3 và 10.4 cho thấy đánh giá tài nguyên/truyền thông mức lớp cho các mô hình AlexNet và VGG-16 và dựa trên nghiên cứu trong công trình của Y. Kang và công sự. Các ví dụ sử dụng hình ảnh từ luồng video có tốc độ 30 khung hình/giây (FPS: Frame per Second) với hình ảnh đầu vào 227×227 cần được phân loại. Kích thước dữ liệu trung gian được truyền từ lớp CNN này sang lớp tiếp theo phụ thuộc vào vị trí của điểm phân chia. Do đó, tốc độ dữ liệu UL (Uplink: đường lên) cần thiết có liên quan đến điểm phân chia mô hình và tốc độ khung hình để nhận dạng hình ảnh cũng được quan sát bởi nghiên cứu trên. Các điểm phân chia ứng cử tương ứng được chỉ ra trên các hình 9.3 và 9.4. Tốc độ dữ liệu UL cần thiết cho các điểm phân chia khác nhau nằm trong khoảng từ 4,8 đến 65 Mbit/s đối với AlexNet được liệt kê trong bảng 10.1 và tốc độ dữ liệu UL cần thiết cho các điểm phân chia khác nhau nằm trong khoảng từ 24 đến 720 Mbit/s đối với VGG-16 được liệt kê trong bảng 10.2. Trong trường hợp hình ảnh có độ phân giải cao hơn, tốc độ dữ liệu cao hơn sẽ được yêu cầu.



CNN cho nhận dạng hình ảnh (Alex NET)

Layer latency: trễ lớp; Size of output data: kích thước dữ liệu đầu ra

Hình 10.3. Đánh giá tài nguyên tính toán / truyền thông mức lớp cho mô hình AlexNet



CNN cho nhận dạng hình ảnh (VGG-16)

Layer latency: trễ lớp; Size of output data: kích thước dữ liệu đầu ra

Hình 10.4. Đánh giá tài nguyên tính toán/truyền thông mức lớp cho mô hình VGG-16

Bảng 10.1. Tốc độ dữ liệu UL yêu cầu cho các điểm phân chia khác nhau của mô hình nhận dạng hình ảnh @30FPS

Điểm phân chia	**Kích thước dữ liệu đầu ra gần đúng (Mbyte)**	**Tốc độ dữ liệu UL yêu cầu (Mbit/s)**
Điểm phân chia ứng cử 0 (suy luận dựa trên đám mây)	0,15	36
Điểm phân chia ứng cử 1 (sau lớp pool)	0,27	65
Điểm phân chia ứng cử 2 (sau lớp pool2)	0,17	41
Điểm phân chia ứng cử 3 (sau lớp pool5)	0,02	4,8
Điểm phân chia ứng cử 4 (Suy luận dựa trên thiết bị)	NA	N/A

Bảng 10.2. Tốc độ dữ liệu UL yêu cầu theo trải nghiệm người dùng cho các điểm phân chia khác nhau của mô hình VGG-16 @30FPS

Điểm phân chia	**Kích thước dữ liệu đầu ra gần đúng (Mbyte)**	**Tốc độ dữ liệu UL yêu cầu theo trải nghiệm người dùng (Mbit/s)**
Điểm phân chia ứng cử 0 (suy luận dựa trên đám mây)	0,6	145
Điểm phân chia ứng cử 1 (sau lớp pool1)	3	720
Điểm phân chia ứng cử 2 (sau lớp pool2)	1,5	360
Điểm phân chia ứng cử 3 (sau lớp pool3)	0,8	192
Điểm phân chia ứng cử 4 (sau lớp pool4)	0,5	120
Điểm phân chia ứng cử 5 (sau lớp pool5)	0,1	24
Điểm phân chia ứng cử 6 (Suy luận dựa trên thiết bị)	N/A	N/A

10.3. CÁC CHẾ ĐỘ HOẠT ĐỘNG CHO CÁC AM/ML PHÂN CHIA GIỮA ĐẦU CUỐI DI ĐỘNG VÀ MẠNG

10.3.1. Kết nối giữa AI/ML và mạng truyền thông không dây di động qua đám mây biên

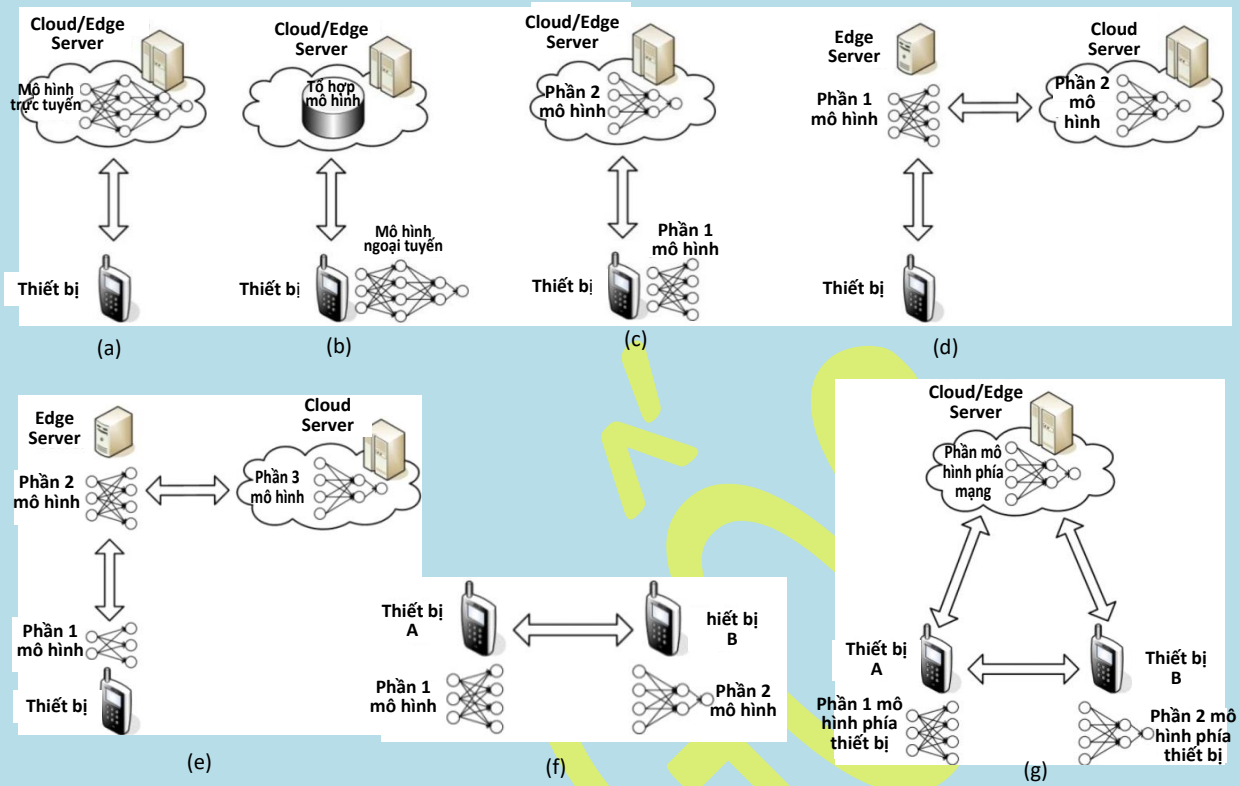
Các mạng truyền thông di động không dây tích hợp một dải rộng các nút được kết nối như: các thiết bị thông minh và các bộ cảm biến bằng cách cung cấp sô lượng lớn các ứng dụng. Để cho phép các ứng dụng trễ thấp hoạt động và đảm bảo an ninh cần chuyển dịch mô hình từ điện toán đám mây tập trung đến điện toán đám mây biên (Edge Cloud) gần thiết bị cuối hơn. Trong 5G điện toán đám mây biên được gọi là MEC (Mobile Edge Computing: điện toán di động biên). Nhiệm vụ chính của đám mây biên là: giảm tải quyết định, ấn định tài nguyên , triển khai server và chi phí truyền thông.

Một thách thức đối với mọi thiết bị người dùng là phải quyết định khi nào nó cần giam tải tính toán của mình lên server biên và khối lượng tính toán cần giảm tải. Quyết định vày rất quan trọng cả về tiết kiệm năng lượng lẫn hiệu quả tính toán. Quyết định này chịu ảnh hưởng của nhiều nhân tố như: an ninh và tính riêng tư của dữ liệu đối với người dùng, ưu tiên của người dùng thực hiện tính toán, chất lượng kết nối vô tuyến, mức độ khả dụng của đám mây biên và đám mây trung tâm và khả năng thực hiện xử lý, v.v. Khi xét đến các nhân tố này, có ba chiến lược cho giảm tải đến server biên: thực hiện cục bộ (xảy ra trên thiết bị người dùng), giảm tải toàn bộ hoặc một phần đến server biên. Mục đích của ba giải thuật này là cân bằng chỉ số: tiết kiệm năng lượng trên thiết bị người dùng và thỏa mãn ràng buộc trễ cần đảm bảo.

10.3.2. Các chế độ cho các hoạt động AI / ML phân chia giữa thiết bị đầu cuối di động và mạng lõi

Các chế độ cho các hoạt động AI / ML phân chia giữa thiết bị đầu cuối di động và mạng được minh họa trong Hình 10.5. Các chế độ nói chung có thể áp dụng cho đào tạo AI / ML cũng như suy luận. Trong phần này, chúng ta tập trung vào xử lý suy luận. Chế độ a) và b) là các sơ đồ truyền thống vận hành suy luận AI / ML hoàn toàn trên một điểm cuối. Chế độ c) - g) cố gắng chia suy luận AI/ML hoặc thậm chí mô hình thành nhiều phần theo nhiệm vụ và môi trường hiện tại, để giảm bớt áp lực tính toán, bộ nhớ/lưu trữ, năng lượng và tốc độ dữ liệu cần thiết trên cả thiết bị đầu cuối di động và điểm cuối mạng (NW), cũng như để có được hiệu suất suy luận mô hình tốt hơn về độ trễ, độ chính xác và bảo vệ quyền riêng tư.

BẢN NHÁP



Edge Server: máy chủ biên, Cloud Server: máy thủ đám mây, Cloud /Edge Server: máy chủ đám mây hoặc máy chủ biên

Hình 10.5. Các chế độ cho các hoạt động AI/ML phân chia giữa thiết bị và mạng

- Chế độ a): Suy luận dựa trên máy chủ đám máy hoặc máy chủ biên(Cloud/Edge-based Inferrence. Ở chế độ này (như thể hiện trong Hình 10.5 (a)), suy luận mô hình AI/ML chỉ được thực hiện hoặc trong máy chủ đám mây hoặc trong máy chủ biên. Thiết bị chỉ báo cáo dữ liệu cảm biến/nhận thức cho máy chủ và không cần hỗ trợ các hoạt động suy luận AI/ML. Máy chủ trả về kết quả suy luận cho thiết bị. Ưu điểm của chế độ này là hạn chế độ phức tạp của thiết bị. Một nhược điểm là hiệu suất suy luận phụ thuộc vào tốc độ dữ liệu truyền thông và độ trễ giữa thiết bị và máy chủ. Tải lên một số dữ liệu nhận thức theo thời gian thực (ví dụ: phát trực tuyến video có độ phân giải cao) yêu cầu tốc độ dữ liệu cao ổn định và một số dịch vụ AI/ML (ví dụ: robot điều khiển từ xa) yêu cầu độ trễ thấp ổn định, điều này rất khó đảm bảo trong hệ thống truyền thông di động do vùng phủ sóng mạng khác nhau. Và do việc tiết lộ dữ liệu nhạy cảm về quyền riêng tư cho mạng, các biện pháp bảo vệ quyền riêng tư tương ứng là cần thiết.
- Chế độ b): Suy luận dựa trên thiết bị (Device-based Inference)

Ở chế độ này (như thể hiện trong hình 10.5 (b)), suy luận mô hình AI / ML được thực hiện tại chỗ trên thiết bị di động UE (User Eqquipement: thiết bị người dung). Ưu điểm là, trong quá trình suy luận, thiết bị không cần giao tiếp với máy chủ đám mây hoặc máy chủ biên . Một động lực khác của chế độ này là bảo vệ quyền riêng tư tại nguồn dữ liệu, tức là thiết bị, mặc dù vấn đề bảo vệ quyền riêng tư cũng cần được xem xét ở phía thiết bị. Nhược điểm là có khả năng áp đặt tài nguyên tính toán/ bộ nhớ/ lưu trữ quá mức cho thiết bị.Và cũng không thể giả định rằng thiết bị luôn giữ tất cả các mô hình AI/ML tiềm năng cần thiết trên thiết bị. Trong một số trường hợp, thiết bị di động có thể cần lấy mô hình AI/ML từ đám mây/máy chủ biên, yêu cầu tốc độ tải xuống dữ liệu tương ứng từ hệ thống 5G và 6G.

- Chế độ c): Suy luận được phân chia giữa thiết bị-máy chủ đám mây hoặc máy chủ biên (Device-Cloud/edge Split Inference))

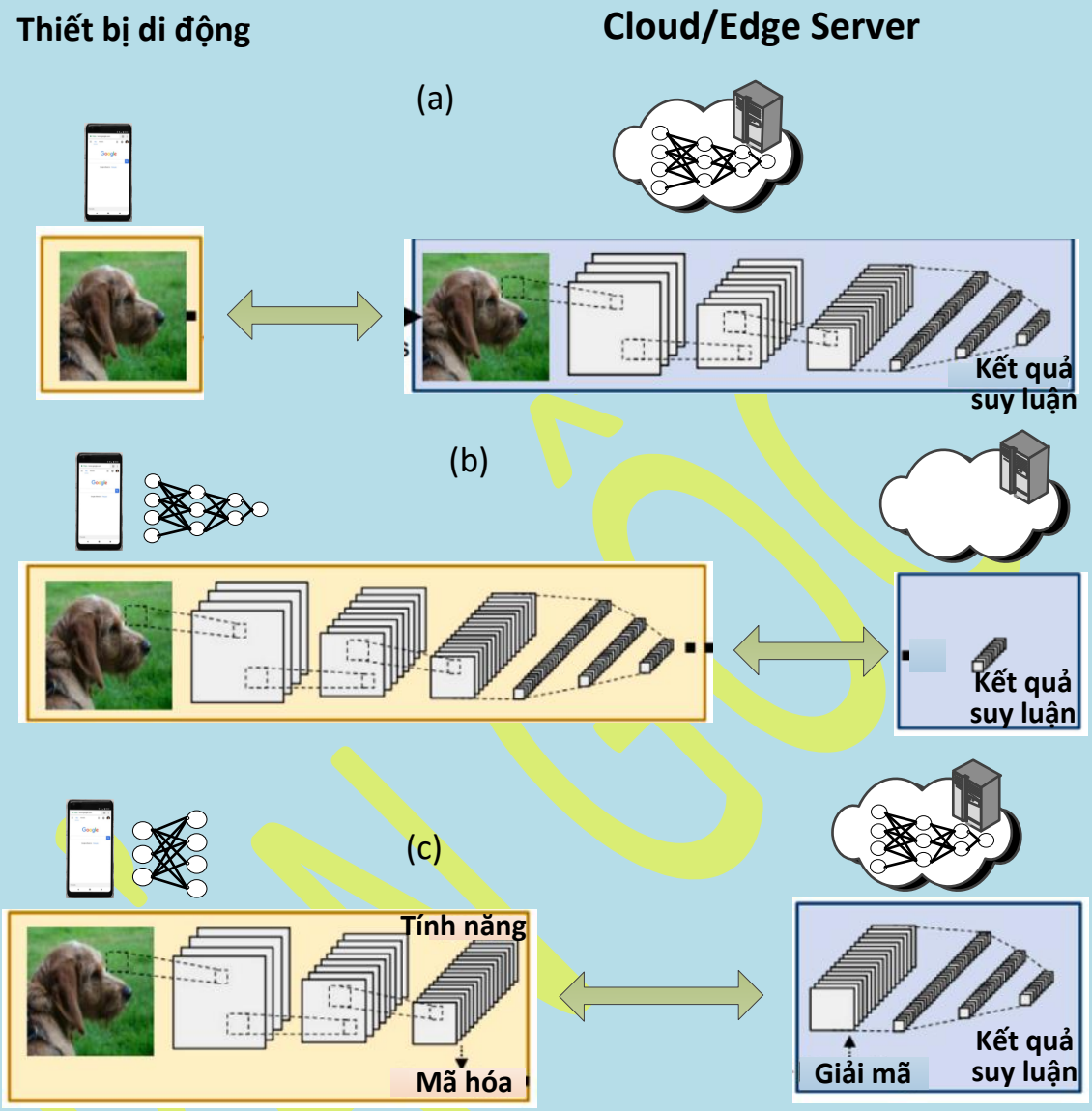
  Ở chế độ này (như thể hiện trong Hình 10.5 (c)), một hoạt động hoặc mô hình suy luận AI/ML trước tiên được chia thành hai phần giữa thiết bị và máy chủ đám mây/biên (Cloud Server/ Edge Server) theo các yếu tố môi trường hệ thống hiện tại như tốc độ dữ liệu truyền thông, tài nguyên thiết bị và khối lượng công việc máy chủ. Sau đó, thiết bị sẽ thực hiện suy luận AI/ML đến một phần cụ thể hoặc mô hình DNN lên đến một lớp cụ thể và gửi dữ liệu trung gian đến máy chủ đám mây/biên. Máy chủ sẽ thực hiện các phần/lớp còn lại và gửi kết quả suy luận đến thiết bị. So với Chế độ a) và b), chế độ này linh hoạt hơn và mạnh mẽ hơn đối với tài nguyên tính toán và điều kiện truyền thông khác nhau. Một liên kết quan trọng cho chế độ này là chọn đúng điểm phân chia tối ưu giữa phía thiết bị và phía mạng dựa trên các điều kiện.
- Chế độ d): Suy luận phân chia giữa máy chủ biên và máy chủ đám mây (Edge-Cloud Split Inference))

  Chế độ này (như thể hiện trong Hình 10.5 (d)) có thể được coi là phần mở rộng của Chế độ a). Sự khác biệt là mô hình DNN được thực thi thông qua sức mạnh tổng hợp biên-đám mây, thay vì chỉ được thực thi trên đám mây hoặc máy chủ biên. Phần nhạy cảm với độ trễ của hoạt động suy luận AI/ML hoặc các lớp của mô hình AI/ML có thể được thực hiện tại máy chủ biên. Các phần/lớp tính toán chuyên sâu mà máy chủ biên không thể thực hiện có thể được chuyển sang máy chủ đám mây. Thiết bị chỉ báo cáo dữ liệu cảm biến/nhận thức cho máy chủ và không cần hỗ trợ các hoạt động suy luận AI/ML. Dữ liệu trung gian được gửi từ máy chủ biên đến máy chủ đám mây. Cần chọn một điểm phân chia thích hợp để hợp tác hiệu quả giữa máy chủ biên và máy chủ đám mây.
- Chế độ e): Suy luận phân chia thiết bị- máy chủ biên- máy chủ đám mây (Device-Edge-Cloud Split Inference)

Chế độ này (như thể hiện trong Hình 10.5 (e)) là sự kết hợp của Chế độ c) và d). Hoạt động suy luận AI/ML hoặc mô hình AI/ML được phân chia trên thiết bị di động, máy chủ biên và máy chủ đám mây. Các phần/lớp tính toán chuyên sâu của hoạt động/mô hình AI/ML có thể được phân bố giữa đám mây và/hoặc máy chủ biên. Các phần/lớp nhạy cảm với độ trễ có thể được thực hiện trên thiết bị hoặc máy chủ biên. Dữ liệu nhạy cảm về quyền riêng tư có thể được để lại trên thiết bị. Thiết bị gửi kết quả dữ liệu trung gian từ tính toán của nó đến máy chủ biên. Và máy chủ biên gửi kết quả dữ liệu trung gian từ tính toán của nó đến máy chủ đám mây. Cần chọn hai điểm phân chia để hợp tác hiệu quả giữa thiết bị, máy chủ biên và máy chủ đám mây.

- Chế độ f): Suy luận phân chia thiết bị-thiết bị (Device-Device Split Inference)
  Chế độ này (như thể hiện trong Hình 10.5 (f)) cung cấp một suy luận phân chia phi tập trung. Một hoạt động hoặc mô hình suy luận AI/ML có thể được phân chia trên các thiết bị di động khác nhau. Một nhóm thiết bị di động có thể thực hiện các phần khác nhau của hoạt động AI/ML hoặc các lớp DNN khác nhau cho một nhiệm vụ suy luận và trao đổi dữ liệu trung gian với nhau. Tải tính toán có thể được phân bố trên các thiết bị trong khi mỗi thiết bị lưu giữ thông tin riêng tư cục bộ.
- Chế độ g): Suy luận phân chia thiết bị-thiết bị-đám mây/biên (Device-Edge/Cloud Inference)
  Chế độ g) có thể được kết hợp thêm với chế độ c) hoặc e). Như thể hiện trong Hình 2.4 (g), một hoạt động hoặc mô hình suy luận AI/ML trước tiên được chia thành phần thiết bị và phần mạng. Sau đó, phần thiết bị có thể được thực thi theo cách phi tập trung, tức là phân chia thêm trên các thiết bị di động khác nhau. Dữ liệu trung gian có thể được gửi từ một thiết bị đến máy chủ đám mây hoặc máy chủ biên. Hoặc nhiều thiết bị có thể gửi dữ liệu trung gian đến máy chủ đám mây hoặc máy chủ biên.

Hình 10.6 cho thấy ba kịch bản xử lý AI/ML dựa trên CNN khác nhau: (a) Suy luận chỉ được thực hiện tại server đám mây hoặc server biên, (b) suy luận chỉ được thực hiện tại UE và (c) suy luận được UE giảm tải đến máy chủ đám mây hoặc máy biên.



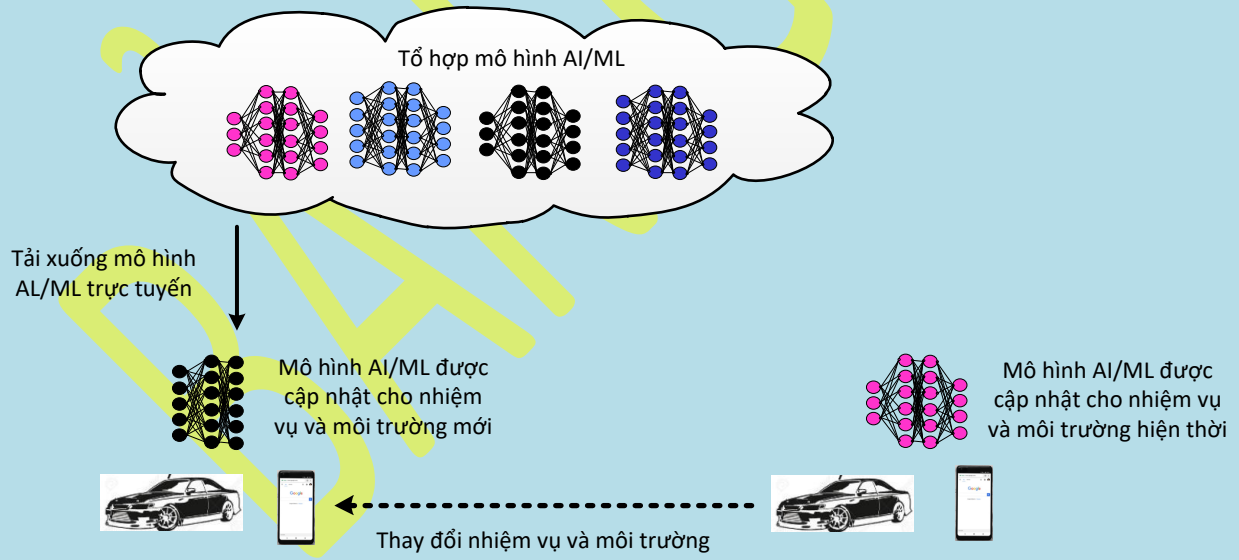
Hình 10.6.Minh họa các kịch bản phân chia xử lý hoạt động AI/ML dựa trên mạng CNN

10.4. PHÂN PHỐI VÀ CHIA SẺ MÔ HÌNH AI/ML TRONG CÁC HỆ THỐNG TRUYỀN THÔNG DI ĐỘNG KHÔNG DÂY

Đối với các nhiệm vụ suy luận yêu cầu độ trễ thấp và muốn lưu giữ dữ liệu nhạy cảm về quyền riêng tư ở phía UE, suy luận AI/ML ngoại tuyến là mong muốn, thay vì suy luận dựa trên đám mây. Tuy nhiên, mô hình AI/ML ngoại tuyến chạy trên thiết bị di động phải

có độ phức tạp tính toán tương đối thấp và dung lượng lưu trữ nhỏ. Một cách tiếp cận để kích hoạt các mô hình DNN ngoại tuyến trên thiết bị di động là nén mô hình để giảm các yêu cầu về tài nguyên và tính toán của nó. Tuy nhiên, nén DNN sẽ dẫn đến mất độ chính xác của suy luận và khả năng thích ứng với các nhiệm vụ và môi trường khác nhau. Một giải pháp cho thách thức này là chọn mô hình thích ứng để suy luận từ một tập hợp các mô hình được đào tạo. Việc lựa chọn mô hình được thúc đẩy bởi quan sát rằng mô hình tối ưu để suy luận phụ thuộc vào dữ liệu đầu vào và yêu cầu độ chính xác. Các thiết bị đầu cuối di động đa chức năng thường cần chuyển đổi mô hình AI/ML để đáp ứng các thay đổi nhiệm vụ và môi trường.

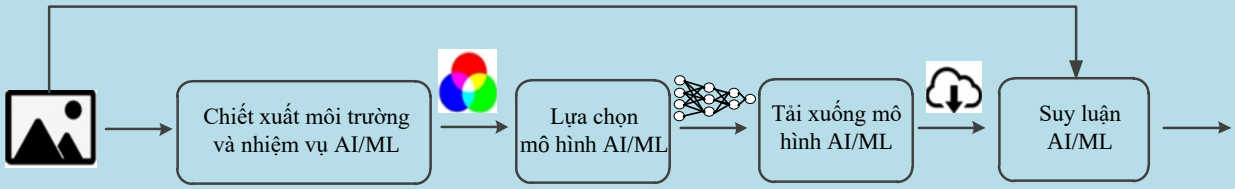
Điều kiện của việc lựa chọn mô hình thích ứng là các mô hình được chọn đã có sẵn cho thiết bị di động. Tuy nhiên, với thực tế là các mô hình DNN ngày càng trở nên đa dạng và với tài nguyên lưu trữ hạn chế trong UE, việc tải trước tất cả các mô hình AI/ML ứng cử viên lên UE là không khả thi. Phân bố mô hình trực tuyến (tức là tải xuống mô hình mới) hoặc học chuyển giao trực tuyến (tức là cập nhật mô hình một phần) là cần thiết. Như minh họa trong Hình 10.7, mô hình AI/ML có thể được phân phối từ điểm cuối mạng (NW endpoint) đến các thiết bị khi chúng cần để thích ứng với các nhiệm vụ và môi trường AI / ML đã thay đổi.



Hình 10.7. Tải xuống mô hình AI/ML qua hệ thống 5G/6G

Mô hình được phân phối có thể được xác định theo hai cách: được yêu cầu bởi một thiết bị hoặc được điều khiển bởi một máy chủ mạng. Điều kiện của cơ chế đầu tiên là thiết bị có thể đưa ra quyết định lựa chọn / lựa chọn lại mô hình dựa trên sự hiểu biết về nhiệm vụ AI/ML sắp tới, môi trường và danh sách các mô hình có sẵn tại máy chủ mạng. Như

thể hiện trong hình 10.8, bộ chọn mô hình trên thiết bị được đào tạo để chọn DNN tốt nhất cho các dữ liệu đầu vào khác nhau.



Hình 10.8. Lựa chọn mô hình AI/ML và tải xuống

Tốc độ dữ liệu để tải xuống các mô hình cần thiết phụ thuộc vào các yếu tố sau:

- Kích thước của mô hình:
  Điều này phụ thuộc vào các ứng dụng AI/ML khác nhau. Cùng với yêu cầu hiệu năng ngày càng tăng đối với các hoạt động AI/ML, kích thước của các mô hình cũng tiếp tục tăng lên, mặc dù các kỹ thuật nén mô hình đang được cải thiện.
- Độ trễ tải xuống yêu cầu
  Điều này phụ thuộc vào tốc độ mô hình cần sẵn sàng trên thiết bị. Nó bị ảnh hưởng bởi mức độ mà ứng dụng sắp tới có thể được dự đoán. Xem xét tính không thể đoán trước của hành vi người dùng và thời gian chờ đợi điển hình mà người dùng có thể chịu đựng, việc tải xuống mô hình AI/ML cần được hoàn thành trong vài giây hoặc thậm chí trong mili giây. Khác với video được phát trực tuyến có thể được phát khi một phần nhỏ được nhớ đệm, mô hình DNN chỉ có thể được sử dụng cho đến khi toàn bộ mô hình được tải xuống hoàn toàn.

Cần lưu ý rằng suy luận AI/ML dựa trên mạng và phân chia thường yêu cầu tốc độ dữ liệu đường lên cao và không đổi để liên tục giảm tải dữ liệu cảm biến/trung gian sang máy chủ đám mây hoặc máy chủ biên. Ngược lại, phân bố mô hình AI/ML chủ yếu yêu cầu tốc độ dữ liệu đường xuống cao trong một đợt bùng phát. Điều này làm cho việc phân bố mô hình phù hợp hơn với các hệ thống truyền thông di động chiếm ưu thế của đường xuống (ví dụ: sử dụng tỷ lệ DL-to-UL cao) hoặc các hệ thống có vùng phủ sóng không ổn định. Tất nhiên điều kiện là tài nguyên tính toán của thiết bị di động có thể đủ khả năng thực thi mô hình AI / ML trên bo mạch. Nếu tải tính toán vượt quá khả năng của thiết bị, suy luận dựa trên mạng hoặc phân chia phải được áp dụng.

10.5. NGUYÊN LÝ HỌC PHÂN TÁN/ LIÊN KẾT TRÊN CÁC HỆ THỐNG TRUYỀN THÔNG DI ĐỘNG KHÔNG DÂY

Với khả năng liên tục cải tiến của camera và cảm biến trên thiết bị di động, dữ liệu đào tạo có giá trị, cần thiết cho việc đào tạo mô hình AI/ML, ngày càng được tạo ra trên các thiết bị. Đối với nhiều nhiệm vụ AI/ML, dữ liệu bị phân mảnh được thu thập bởi thiết bị di động là điều cần thiết để đào tạo một mô hình toàn cục. Trong các phương pháp truyền thống, dữ liệu đào tạo được thu thập bởi thiết bị di động được tập trung vào trung tâm dữ liệu đám mây để đào tạo tập trung.

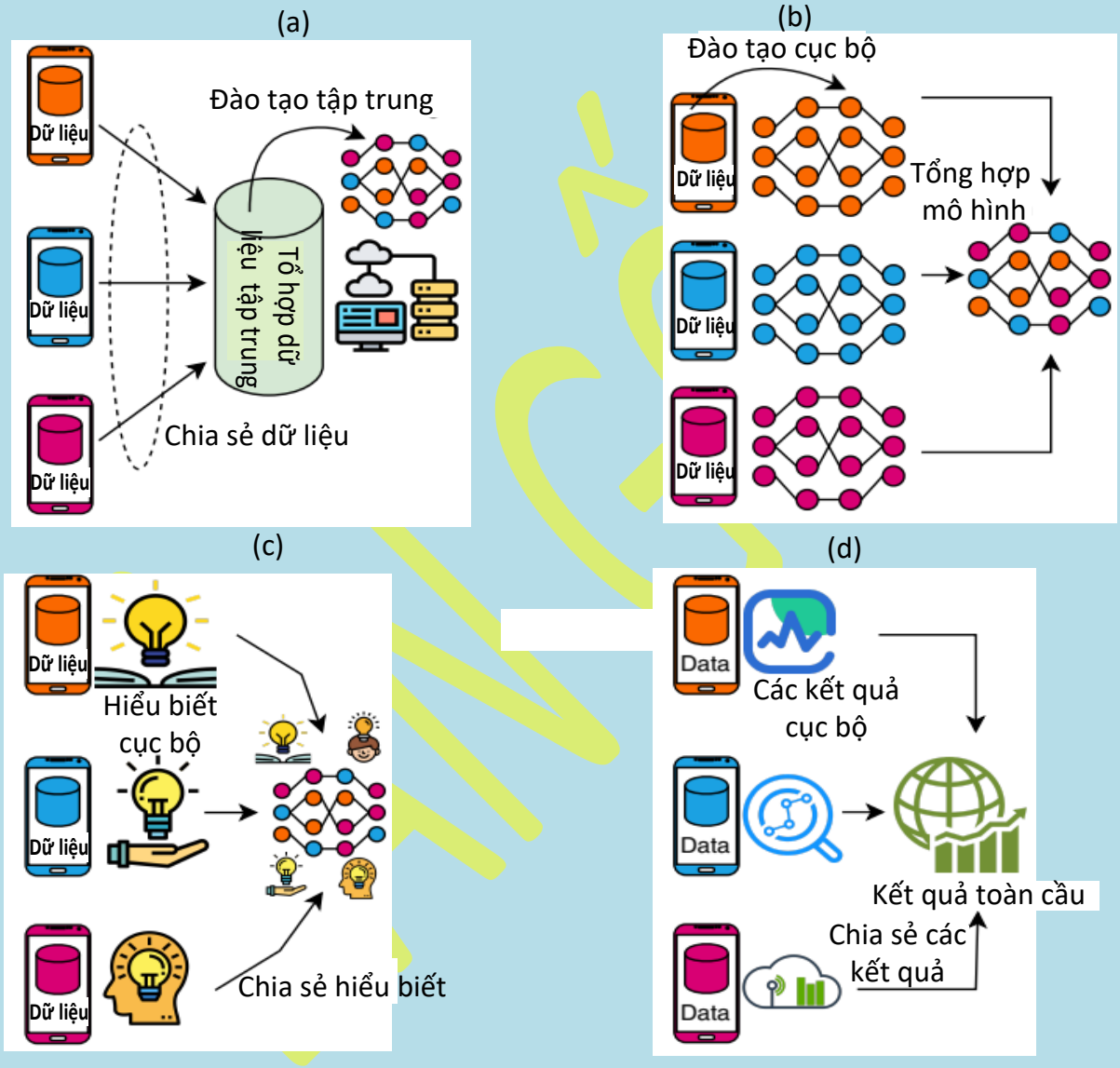
Tuy nhiên, việc đào tạo mô hình AI/ML thường yêu cầu một tập dữ liệu lớn và tài nguyên tính toán đáng kể cho nhiều lần lặp lại cập nhật trọng số. Ngày nay, hầu hết các nhiệm vụ đào tạo mô hình AI/ML được thực hiện trong các trung tâm dữ liệu đám mây lớn vì mức tiêu thụ tài nguyên của giai đoạn đào tạo vượt quá đáng kể giai đoạn suy luận. Trong nhiều trường hợp, việc đào tạo một mô hình DNN vẫn mất vài giờ đến nhiều ngày. Tuy nhiên, đào tạo dựa trên đám mây có nghĩa là lượng dữ liệu đào tạo khổng lồ phải được vận chuyển từ các thiết bị lên đám mây, gây ra chi phí giao tiếp quá lớn cũng như áp lực bảo mật dữ liệu ở phía mạng. Tương tự như suy luận AI/ML phân chia được giới thiệu trong phần 10.3.2, các nhiệm vụ đào tạo mô hình AI/ML cũng có thể hoạt động theo cách phối hợp thiết bị đám mây. Học tập phân tán và học tập liên kết là những ví dụ theo cách này.

10.5.1. Nguyên lý chung của học phân tán (Distributed Learning)

Khi các giải thuật ML cần được triển khai theo cách phân tán với nhiều server đám mây/biên và nhiều thiết bị (IoT hoặc di động), Giải pháp tiếp cận chung là chia sẻ dữ liệu, mô hình, sự hiểu biết hoặc kết quả giữa các client. Các cách tiếp cận này được đề cập đến như là cách tiếp cận chia sẻ dữ liệu, chia sẻ mô hình, chia sẻ sự hiểu biết và chia sẻ kết quả.

Hình 9.9 minh họa bốn cách tiếp cận học phân tán. Trong học phân tán, hệ thống thường bao gồm một server trung tâm và nhiều thiết bị phân tán. Trong cách tiếp cận chia sẻ số liệu (Data Sharing) như minh họa trên hình 10.9(a), server đào tạo mô hình học toàn cục (Global Learning Model) dựa trên các dữ liệu được chia sẻ từ các thiết bị phân tán. Trong cách tiếp cận chia sẻ mô hình (Model Sharing) như minh họa trên hình 10.9(b), các thiết bị phân tán phân tán đào tạo các mô hình tương ứng của mình sử dụng các dữ liệu cục bộ của chúng và chỉ chia sẻ mô hình được đào tạo với server để tổng hợp mô hình. Học liên kết (FL: Federated Learning) là thí dụ về cách tiếp cận này, trong đó mô hình được tổng hợp được phân phối từ server đến các thiết bị để khởi đầu một vòng mới. Khác với hai cách tiếp cận trên, trong đó dữ liệu và các mô hình được chia sẻ, cách tiếp cận thứ ba như minh họa trên hình 9.9(c) chia sẻ hiểu biết (Sharing Knowledge) cho phép các thiết bị chiết suất hiểu biết của mình dựa trên dữ liệu cục bộ sau đó chia sẻ nó với server học.

Trong cách tiếp cận cuối cùng như minh họa trên hình 9.9.(d), cách tiếp cận chia sẻ kết quả (Sharing Result) , các thiết bị phân tán chia sẻ đầu ra của các mô hình được đào tạo của chúng với server.



Hình 10.9. Minh họa các cách tiếp cận học phân tán chính

Học phân tán trên biên được nghiên cứu rộng rãi, đây là một cách tiếp cận hiệu quả cho các mô hình học sâu. Nhân tố động lực chính là hứa hẹn một mô hình được cải thiện kết hợp được dữ liệu từ các nguồn khác nhau, trong khi có thể kiểm soát được an ninh, chi phí truyền thông và tính toán.

Học phân tán trên biên đang được nghiên cứu rộng rải, đây là một cách tiếp cận hiệu quả cho các mô hình học sâu. Nhân tố động lực chính là nó hứa hẹn một mô hình được cải thiện kết hợp được dữ liệu từ các nguồn khác nhau, trong khi có thể kiểm soát được an ninh, chi phí truyền thông và tính toán. Có hai cách tiếp cận học phân tán chính (xem A survey: Distributed Machine Learning for 5G and beyond): học liên kết (FL) và học có giám sát (SL: Supervised Learning). Phần dưới đây ta sẽ xét học liên kết.

10.5.2. Kiến trúc tổng quát của học liên kết của học liên kết (Federated Learning)

Ở chế độ học liên kết (FL: Federated Learning), máy chủ đám mây hoặc máy chủ biên đào tạo một mô hình toàn cục bằng cách tổng hợp các mô hình cục bộ được đào tạo một phần bởi từng thiết bị cuối. Hình 10.10 minh họa kiến trúc tổng quát của học liên kết, trong đó một tập K người dùng (ví dụ các client hay cá thiết bị) đào tào các mô hình cục bộ bằng cách sử dụng các dữ liệu cục bộ của mình sau đó chia sẻ các cập nhật của mô hình cục bộ với server biên để tổng hợp mô hình (Model Aggregation). Mỗi người dùng có một tập dữ liệu gồm D_k mẫu rời rạc $D_k=\{1,\ldots, D_k\}$ và tổng số mẫu dữ liệu của tất cả các người dùng là: $D = \sum_{k=1}^{K} D_k$.

Như thấy trên hình 10.10, một vòng truyền thông của FL báo gồm ba bước sau:

- **Khởi đầu mô hình**. Server biên khởi đầu khởi đầu một trọng số $\boldsymbol{w}^0$ của mô hình toàn cục và phát quảng bá giá trị của nó đến các người dùng được chọn trong vòng truyền thông hiện thời. Lưu ý rằng do tính không đồng nhất của dữ liệu, các tài nguyên bị hạn chế và các điều kiện kênh, nên chỉ có một tập con các người dùng được chọn để đóng góp cho xử lý FL. Ngoài ra server biên phải đặc tả các tham số của các mô hình cục bộ (ví dụ: độ chính xác của mô hình cục bộ và tốc độ học) và các tham số của mô hình toàn cục (ví dụ: độ chính xác của mô hình toàn cục);
- **Đào tạo cục bộ**. Tại vòng truyền thông t, từng người dùng k đào tạo mô hình dựa trên tập dữ liệu thô của mình. Mục tiêu của đào tạo tại chỗ của người dùng k là cực tiểu hóa hàm mất mát (Loss Function) $f_k(\boldsymbol{w})$ như sau:

$$\boldsymbol{w}_k^t = \arg\min f_k \quad \boldsymbol{w}$$

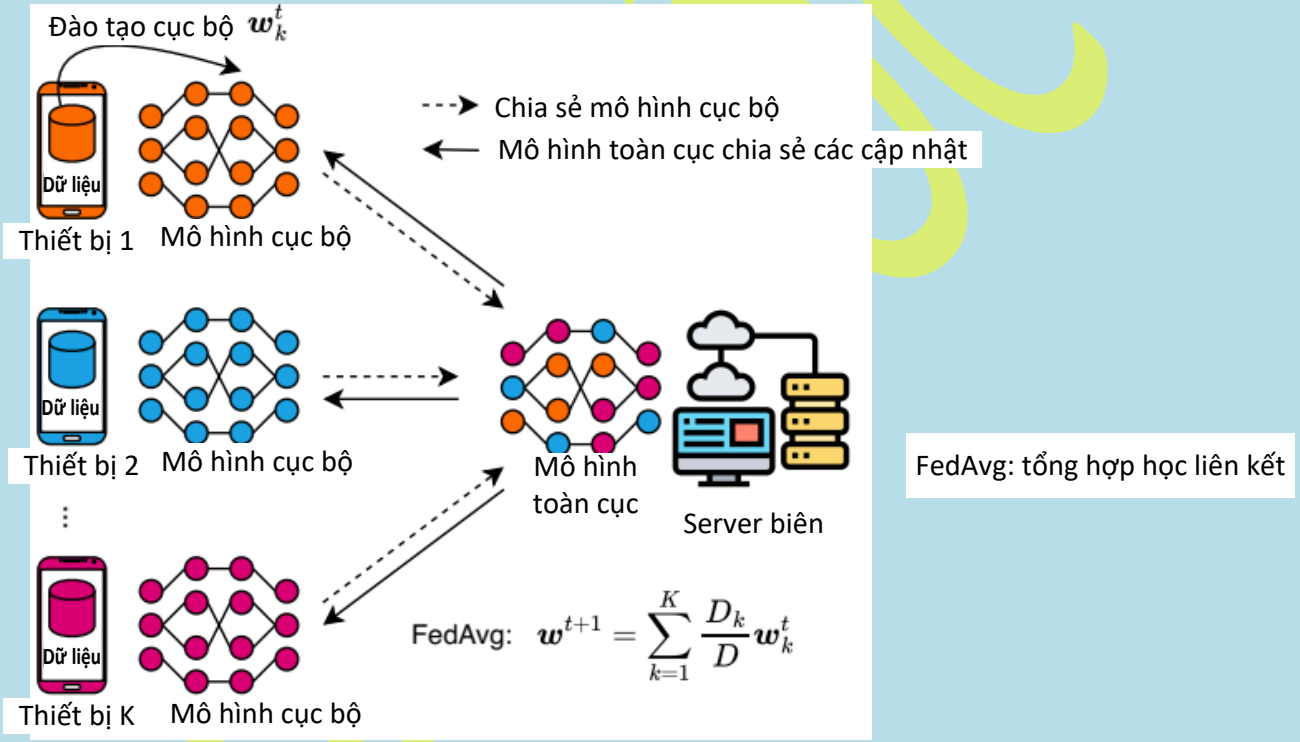
Khi này, các người dùng đã chia sẻ các tham số của mô hình cục bộ với server biên để tổng hợp mô hình (Model Aggregation). Các hàm mất mát khác nhau có thể được sử dụng cho các nhiệm vụ học khác nhau. Chẳng hạn mô hình hồi quy tuyến tính trên tập dữ liệu $(\boldsymbol{x},\boldsymbol{y})$ được nhận dạng bởi hàm mất mát:

$$f(\boldsymbol{w}) = \frac{1}{N}(\boldsymbol{y} - \hat{\boldsymbol{y}})^2,$$

trong đó $\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x}$ là vectơ giá trị thực tế với bỏ qua độ lệch $\boldsymbol{b}$ và $\hat{\boldsymbol{y}}$ là vectơ giá trị được dự báo.

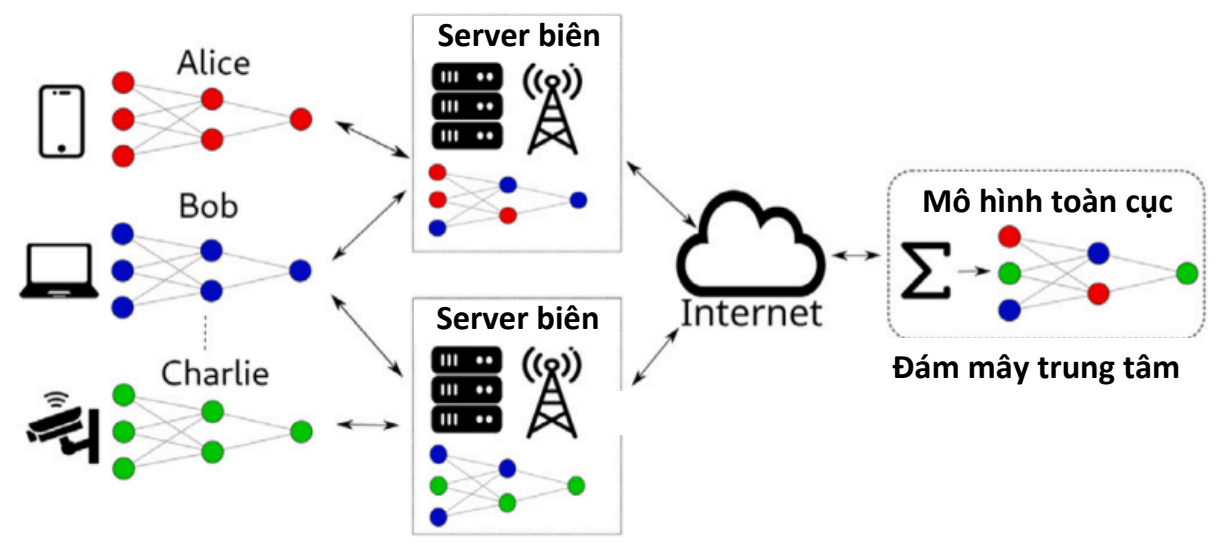
- **Tổng hợp mô hình**. Sau khi server biên nhận được các cập nhật từ các người dùng được lựa chọn, tham số mô hình toàn cục có thể nhận được bằng cách cực tiểu hóa hàm mất mát như sau:

$$\text{FedAvg} = \boldsymbol{w}^{t+1} = \sum_{k=1}^{K} \frac{D_k}{D} \boldsymbol{w}^t$$



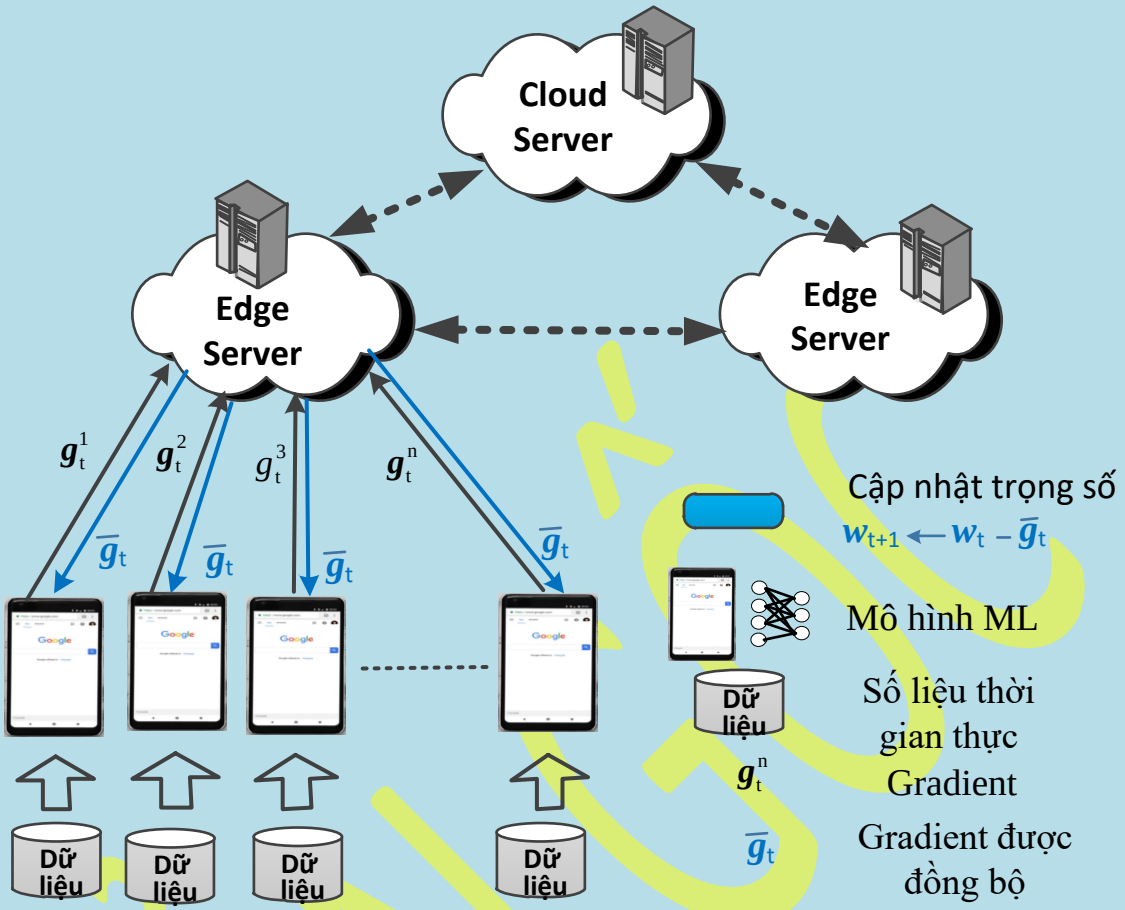
Hình 10.10. Minh họa kiến trúc tổng quát của học liên kết

Hình 10.11 cho thấy kiến trúc FL với sự tham gia đồng thời của server biên và đám mây trung tâm. Các thiết bị cuối (di động hoặc IoT) chuyển các trọng số đến đám mây biên. Sau suy luận, đám mây biên chuyển các trọng số đến đám mây trung gian để tổng hợp mô hình toàn cục. Cuối cùng mô hình mới này được phân phối ngược trở lại các thiết bị biên và từ Server biên nó được phân phối đến các thiết bị cuối.



Hình 9.11. Xử lý FL với sự tham gia đồng thời của server biên và đám mây trung tâm

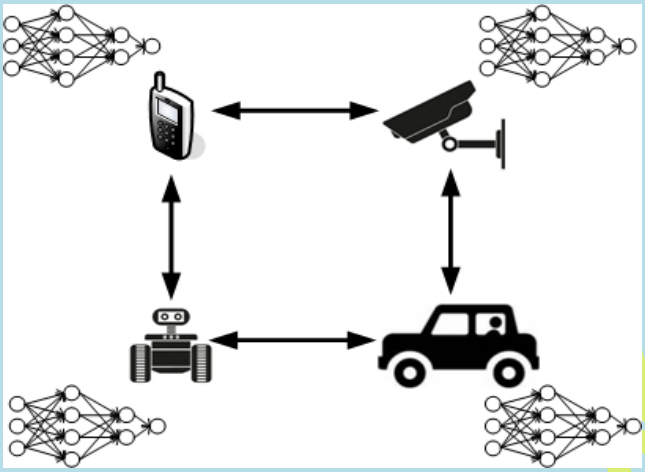
Thay vì gửi các trọng số đên server biên, các máy di động có thể gửi grandient cục bộ đến server và theo chiều ngược lại server biên gửi gradtent tàon cục đến các máy di động để chúng cập nhật mô hinh. Trong kịch bản máy di động và máy chủ biên, để đạt được dự báo thời gian thực tại phía người dùng, máy chủ biên thường đóng vai trò một máy chủ tham số cho cả hai: thực hiện và đào tạo ML. Máy chủ biên thực hiện đồng bộ gradient bằng cách thu thập tất cả các gradient và lấy trung bình chúng cho các tham số cập nhật (hình 10.12). Các tham số được cập nhật được gửi trở lại các cho các máy di động biên (được gọi là đồng bộ tham số)



Cloud Server: máy chủ đám mây, Edge Server: máy chủ biên

Hình 10.12. Thí dụ về đào tạo tại biên

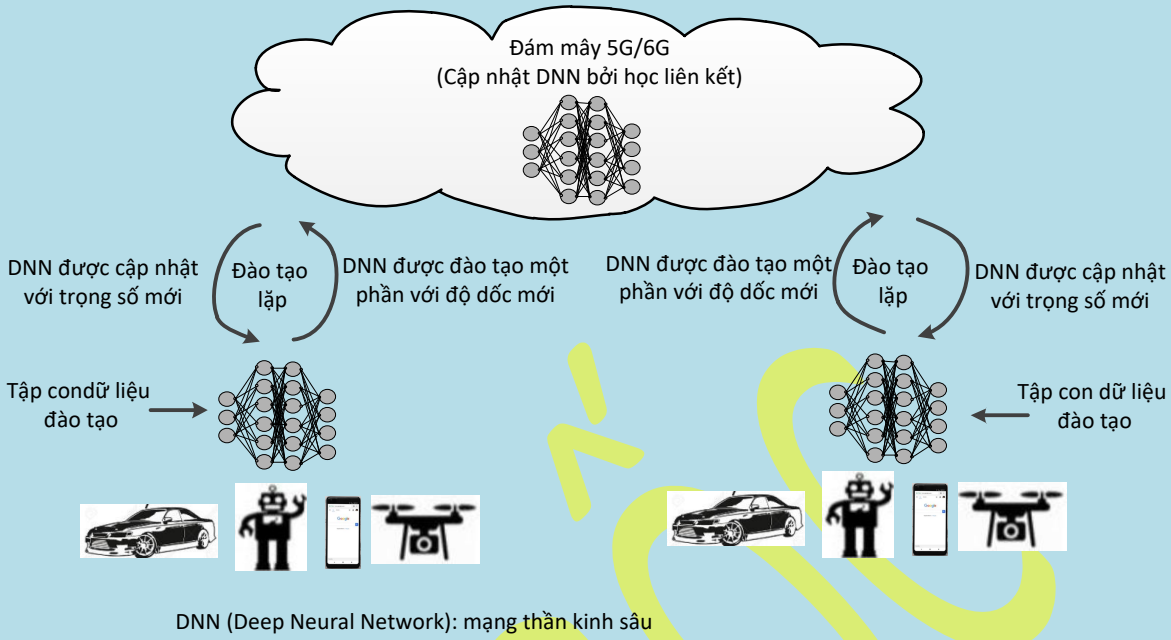
Trong chế độ học phân tán, như thể hiện trong Hình 10.13, mỗi nút tính toán đào tạo mô hình DNN của riêng mình cục bộ với dữ liệu cục bộ, bảo toàn thông tin cá nhân cục bộ. Để có được mô hình DNN toàn cục bằng cách chia sẻ cải tiến đào tạo cục bộ, các nút trong mạng sẽ truyền thông với nhau để trao đổi các bản cập nhật mô hình cục bộ. Ở chế độ này, mô hình DNN toàn cục có thể được đào tạo mà không cần sự can thiệp của trung tâm dữ liệu đám mây.



Hình 10.13. Học phân tán

9.5.3. Học liên kết dựa trên mô hình trung bình lặp trong 5G/6G

Thuật toán Federated Learning dễ chịu nhất cho đến nay dựa trên mô hình trung bình lặp đi lặp lại. Như được mô tả trong Hình 10.14, trong mỗi lần lặp lại đào tạo, UE thực hiện đào tạo dựa trên mô hình được tải xuống từ máy chủ AI bằng cách sử dụng dữ liệu đào tạo cục bộ. Sau đó, UE báo cáo kết quả đào tạo tạm thời (ví dụ: gradient hoặc trọng số cho DNN) cho máy chủ đám mây thông qua các kênh truyền thông không dây đường lên của 5G (5G UL: 5G Uplink)). Máy chủ tổng hợp các gradient/trọng số từ UE và cập nhật mô hình toàn cục. Tiếp theo, mô hình toàn cục được cập nhật được phân bố cho các UE thông qua các kênh truyền thông không dây đường xuống của 5G (5G DL: 5G Downlink). Sau đó, các UE có thể thực hiện đào tạo cho lần lặp tiếp theo.



Hình 10.14. Học liên kết trên dựa trên mô hình trung bình lặp trong 5G/6G

10.5.4. Các yêu cầu về hiệu năng đối với học phân tán/ liên kết

Các yêu cầu về hiệu năng đối với học tập phân tán/liên kết được liệt kê dưới đây. Các yêu cầu đối với các liên kết truyền thông không dây di động (ví dụ: tốc độ dữ liệu, độ trễ, độ tin cậy) có thể được rút ra từ các yêu cầu sau.

- Mất mát đào tạo
  Mất mát đào tạo là khoảng cách giữa đầu ra chính xác và đầu ra được tính toán bởi mô hình DNN, cho biết mô hình DNN được đào tạo phù hợp với dữ liệu đào tạo như thế nào. Mục đích của nhiệm vụ đào tạo là giảm thiểu mất mát đào tạo. Mất mát đào tạo chủ yếu bị ảnh hưởng bởi chất lượng của dữ liệu đào tạo và hiệu quả của các phương pháp đào tạo, tức là liệu ý nghĩa của dữ liệu đào tạo có thể được khám phá đầy đủ và đúng đắn hay không. Đối với học liên kết, chỉ khi dữ liệu đào tạo cục bộ có giá trị có thể được học đầy đủ trong thời gian lặp lại và các bản cập nhật đào tạo cục bộ có thể được báo cáo chính xác cho máy chủ đám mây trong khoảng thời gian mục tiêu, mất mát đào tạo mới có thể được giảm thiểu.
  Điều này ngụ ý rằng các yêu cầu đối với các thiết bị tham gia vào quá trình đào tạo về tốc độ dữ liệu UL có thể đạt được, độ trễ và độ tin cậy để báo cáo các bản cập nhật được đào tạo và tốc độ dữ liệu UL có thể đạt được, độ trễ và độ tin cậy để phân bố mô hình để đào tạo trong lần lặp tiếp theo. Và để giảm thiểu mất mát đào

tạo với tính không đồng nhất của thiết bị (trong hiệu năng tính toán và truyền thông), cần lựa chọn thiết bị đào tạo và cấu hình đào tạo trước khi đào tạo được thực hiện trong một lần lặp lại. QoS của các bản tin điều khiển liên quan, ví dụ như cho yêu cầu đào tạo, báo cáo tài nguyên đào tạo, lựa chọn thiết bị đào tạo, cấu hình đào tạo và phân bổ tài nguyên cho báo cáo cập nhật đào tạo, cũng cần được đảm bảo.

- Trễ đào tạo

  Độ trễ đào tạo là một trong những chỉ số hiệu năng cơ bản nhất của nhiệm vụ đào tạo mô hình AI/ML vì nó ảnh hưởng trực tiếp đến thời điểm mô hình được đào tạo có sẵn để sử dụng. Ngày nay, đào tạo dựa trên đám mây thường mất vài giờ đến nhiều ngày. Độ trễ của quá trình học phân tán/liên kết sẽ mất nhiều thời gian hơn nếu độ trễ tính toán hoặc độ trễ truyền thông không được giảm thiểu.

  Độ trễ của quá trình học tập phân tán/liên kết được xác định bởi tốc độ hội tụ (ví dụ: số lần lặp lại trước khi quá trình đào tạo hội tụ đến sự đồng thuận) và độ trễ của mỗi lần lặp bao gồm độ trễ tính toán và độ trễ truyền thông. Độ trễ tính toán phụ thuộc vào tài nguyên tính toán/bộ nhớ có sẵn trên thiết bị đào tạo. Độ trễ truyền thông phụ thuộc vào tốc độ dữ liệu DL khả dụng để phân bố mô hình và tốc độ dữ liệu UL khả dụng để cập nhật mô hình được đào tạo. Độ trễ của toàn bộ quá trình đào tạo được xác định bởi độ trễ lớn hơn giữa độ trễ tính toán và độ trễ các liên kết truyền thông. Do đó, độ trễ của các liên kết tính toán và truyền thông cần được giảm thiểu một cách hợp tác. Nếu độ trễ giao tiếp không phù hợp với độ trễ tính toán, liên kết giao tiếp sẽ trở thành nút thắt cổ chai và kéo dài toàn bộ quá trình đào tạo.

  Đối với học liên kết đồng bộ, trong mỗi lần lặp, độ trễ đào tạo được xác định bởi thiết bị cuối cùng báo cáo bản cập nhật đào tạo của nó vì tổng hợp liên kết có thể hoàn tất khi tất cả các bản cập nhật đào tạo cần thiết được thu thập chính xác. Điều đó có nghĩa là tính không đồng nhất của thiết bị (về hiệu suất tính toán và truyền thông) cũng sẽ ảnh hưởng lớn đến độ trễ đào tạo tổng thể. Thay vì yêu cầu độ trễ truyền UL của một thiết bị cụ thể, độ trễ tổng thể cần thiết cho tất cả các thiết bị đào tạo để tải lên các bản cập nhật đào tạo (độ trễ nhóm thiết bị) cần được xác định. Và QoS của các bản tin điều khiển để giảm thiểu độ trễ của nhóm thiết bị, ví dụ như cho yêu cầu đào tạo, báo cáo tài nguyên đào tạo, lựa chọn thiết bị đào tạo, cấu hình đào tạo và phân bổ tài nguyên cho báo cáo cập nhật đào tạo, cũng cần được đảm bảo.

- Hiệu quả năng lượng

  Đối với học phân tán/liên kết, cả quá trình tính toán và truyền thông đều tiêu tốn năng lượng đáng kể. Kiến trúc và giao thức học phân tán cũng nên xem xét các

hạn chế về năng lượng trên các thiết bị đào tạo và hiệu quả năng lượng trên thiết bị cũng như phía mạng.

- Quyền riêng tư
  Dữ liệu cảm biến/thu thập hỗ trợ đào tạo và suy luận trong server đám mây/biên thường mang thông tin riêng tư của các người dùng. Cần xem xét các kiểu bảo vệ quyền riêng tư khác nhau trong cả trường hợp xử lý dữ liệu tại thiết bị lẫn trường hợp báo cáo nó đến server đám mây/biên. So với báo cáo nó đến server, giữ dữ liệu thô tại thiết bị sẽ giảm đáng kể áp lực bảo vệ quyền riêng tư tại mạng.
  Ví dụ: học liên kết là một cách tiếp cận dễ chịu để tránh tải dữ liệu thô từ thiết bị lên mạng, như yêu cầu của đào tạo dựa trên đám mây.

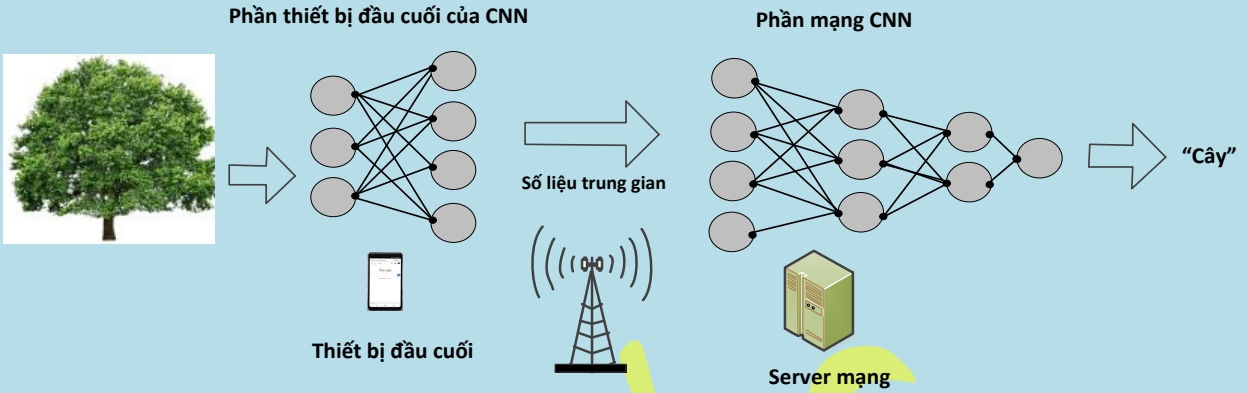
Các phần dưới đây sẽ xét các kịch bản sử dụng và các yêu cầu tiềm năng để các hệ thống 5Ghỗ trợ AI/ML, phân bố và chuyển giao mô hình (tải xuống, tải lên các cập nhật v.v.). Do hiện này chưa có 6G, nên các kịch bản sử dụng được 3GPP nghiên cứu trong các hệ thống 5G nhưng cũng hữu dụng cho khảo sát trong 6G. Các nghiên cứu trên các kịch bản này sử dụng này bao gồm các khía cạnh sau: phân chia hoạt động AI/ML giữa các điểm cuối, phân phối và chia sẻ mô hình/ dữ liệu AI/ML trên hệ thống 5G, học phân tán/liên kết trên hệ thống 5G.

10.6. CÁC KỊCH BẢN HOẠT ĐỘNG CỦA AI/ML ĐƯỢC PHÂN CHIA GIỮA CÁC ĐIỂM CUỐI TRONG CÁC HỆ THỐNG 5G VÀ CÁC YÊU CẦU HIỆU NĂNG.

10.6.1. Nhận dạng hình ảnh dựa trên AM/ML được phân chia

10.6.1.1. Kịch bản

Sơ đồ phân chia nhận dạng hình ảnh AI/ML được mô tả trên hình 10.15. CNN được phân chia thành hai phần dựa trên nhiệm vụ nhận dạng hình ảnh và môi trường hiện thời. Mục đích phân chia là giảm tải các phần đòi hỏi tính toán và năng lượng cao đến server mạng. Serrver mạng thực hiện các lớp CNN còn lại.



Hình 10.15. Thí dụ nhận dạng hình ảnh "cây" dựa trên AM/ML được phân chía

10.6.1.2. Quá trình diễn ra

Điều kiện ban đầu

Các điểm cuối AI/ML liên quan (ví dụ: UE, máy chủ đám mây/biên AI/ML) chạy các ứng dụng cung cấp khả năng suy luận mô hình AI/ML để nhận dạng hình ảnh và hỗ trợ hoạt động nhận dạng hình ảnh AI/ML phân chia.

Các luồng dịch vụ diễn ra như sau:

1) Ứng dụng nhận dạng hình ảnh dựa trên AI/ML được người dùng yêu cầu bắt đầu nhận dạng hình ảnh/ video do UE chụp.
2) Trong điều kiện chế độ phân chia và điểm phân tách đã được xác định, ứng dụng nhận dạng hình ảnh dựa trên AI/ML trong một điểm cuối AI /ML liên quan thực hiện phần được ấn định của mô hình AI / ML và gửi dữ liệu trung gian đến điểm cuối tiếp theo trong đường ống AI / ML (AI/MLPipeline.)
3) Sau khi tất cả các điểm cuối AI/ML liên quan hoàn thành việc đồng suy luận, kết quả nhận dạng hình ảnh sẽ được cung cấp cho người dùng sử dụng kết quả.
4) Các ứng dụng nhận dạng hình ảnh dựa trên AI / ML trong các điểm cuối thực hiện nhận dạng hình ảnh phân chia cho đến khi nhiệm vụ nhận dạng hình ảnh kết thúc.

Làm lại Bước 3) và 4) để chuyển mạch/ chọn lại điểm/ chế độ phân chia nếu cần để thích ứng với các điều kiện thay đổi.

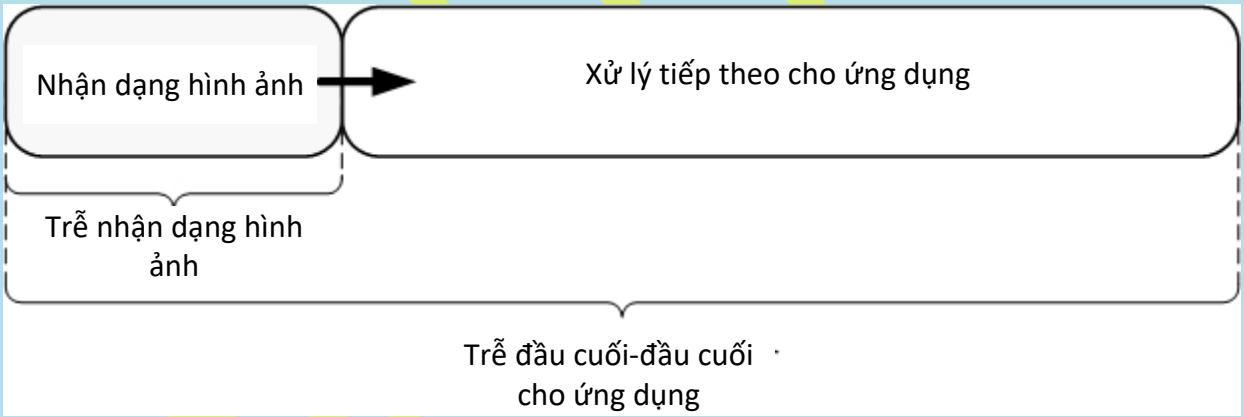
Điều kện kết thúc.Các đối tượng trong hình ảnh hoặc video đầu vào được nhận dạng, độ chính xác và độ trễ nhận dạng cần được đảm bảo.

Nhiệm vụ nhận dạng hình ảnh có thể được hoàn thành trong điều kiện tài nguyên năng lượng và tính toán sẵn có của UE. Và các các tài nguyên điện toán, truyền thông và năng lượng tiêu thụ trên các điểm cuối AI/ML được tối ưu hóa.

10.6.1. 3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

"Độ trễ nhận dạng hình ảnh" có thể được định nghĩa là độ trễ từ hình ảnh được chụp đến kết quả nhận dạng hình ảnh được xuất ra ứng dụng người dùng. Theo nguyên tắc phân tích các yêu cầu về độ trễ và tốc độ dữ liệu của nhận dạng hình ảnh phân chia được giới thiệu trong phần 10.2, độ trễ nhận dạng hình ảnh có liên quan đến ứng dụng người dùng mà nhận dạng được sử dụng.

Thị giác máy tính và nhận dạng hình ảnh đã được sử dụng rộng rãi cho nhiều ứng dụng di động quan trọng như nhận dạng đối tượng, nâng cao ảnh, giám sát video thông minh, AR di động, ô tô điều khiển từ xa, điều khiển công nghiệp và robot. Nhận dạng hình ảnh thường là một bước của quy trình xử lý của ứng dụng. Và độ trễ nhận dạng là một phần của độ trễ đầu cuối, như được mô tả trong Hình 10.16.



Hình 10.16.Trễ nhận dạng hình ảnh là một phần của trễ đầu cuối đầu cuối

Ví dụ, nếu kết quả nhận dạng hình ảnh chỉ được sử dụng để nhận dạng đối tượng, ví dụ như nhận dạng đối tượng chưa được biết cho người dùng điện thoại thông minh hoặc tìm kiếm tội phạm trong cơ sở dữ liệu đối với an ninh thông minh (Intelligence Security), thì có thể chấp nhận được rằng việc nhận dạng hình ảnh kết thúc trong vài giây. Nếu kết quả nhận dạng hình ảnh được sử dụng làm đầu vào cho một ứng dụng nhạy cảm với thời gian khác, ví dụ như màn hình AR /chơi game, ô tô điều khiển từ xa, điều khiển công nghiệp và robot, độ trễ nghiêm ngặt hơn nhiều sẽ được yêu cầu. Dựa trên các yêu cầu về độ trễ

đầu cuối của các ứng dụng, yêu cầu về độ trễ nhận dạng hình ảnh có thể được rút ra, như được liệt kê trong Bảng 10.3.

Bảng 10.3. Trễ nhận dạng hình ảnh và tốc độ dữ liệu đường lên đối với tải lên dữ liệu trung gian

Ứng dụng người dùng	Độ trễ: maximum			Tốc độ dữ liệu UL trải kinh nghiệm củ người dùng	
	Trễ đầu cuối-đầu cuối	Trễ nhận dạng hình ảnh	Trễ tải lên số liệu trung gian	AlexNet (H. 9.3, chú thích 4)	VGG-16 (H. 9.4, chú thích 4)
Nhận dạng đối tượng một lần trên điện thoại thông minh	Vài giây	~1s	~100ms	1,6~21,6Mbit/s	8~240Mbit/s
Nhận dạng người trong hệ thống giám sát an ninh	Vài giây	~1s	~100ms	1,6~21,6Mbit/s	8~240Mbit/s
Nâng cấp ảnh trên điện thoại thông minh	Vài giây	~1s	~100ms	1,6~21,6Mbit/s	8~240Mbit/s
Nhận dạng video	Vài giây	33ms@30FPS	~10ms	16~216Mbit/s	80Mbit/s~2,4Gbit/
Màn hình AR/chơi game	7~15ms (chú thích 1)	<5ms	2ms	80Mbit/s~1.08Gbit/s	0.4~12Gbit/s
Lái xe từ xa	10ms (chú thích 2)	<5ms	2ms	80Mbit/s~1.08Gbit/s	0,4~12Gbit/s

BAN GOC

Robot điều khiển từ xa	10~100ms (chú thích 3)	<5ms	2ms	80Mbit/s~1.08Gbit/s	0.4~12Gbit/s

Chú thích 1: độ trễ chuyển động từ VR đến photon nằm trong khoảng 5-15ms.
Chú thích 2: độ trễ một chiều cần thiết để lái xe từ xa là 5ms. Độ trễ khứ hồi được giả định là 10ms.
Chú thích 3: độ trễ đầu cuối-đầu cuối cần thiết cho robot điều khiển từ xa hoạt động bằng video là 10 ~ 100ms.
Chú thích 4: Như được liệt kê trong bảng 10.1 và bảng 10.2, kích thước dữ liệu trung gian cho các điểm phân tách của AlexNet v
VGG-16 lần lượt là 0,02 ~ 0,27MByte và 0,1 ~ 3MByte.

10.6.2. Nhận dạng phương tiện nâng cao: học sâu dựa trên các ứng dụng thị giác (Deep Learning Based On Vision Application)

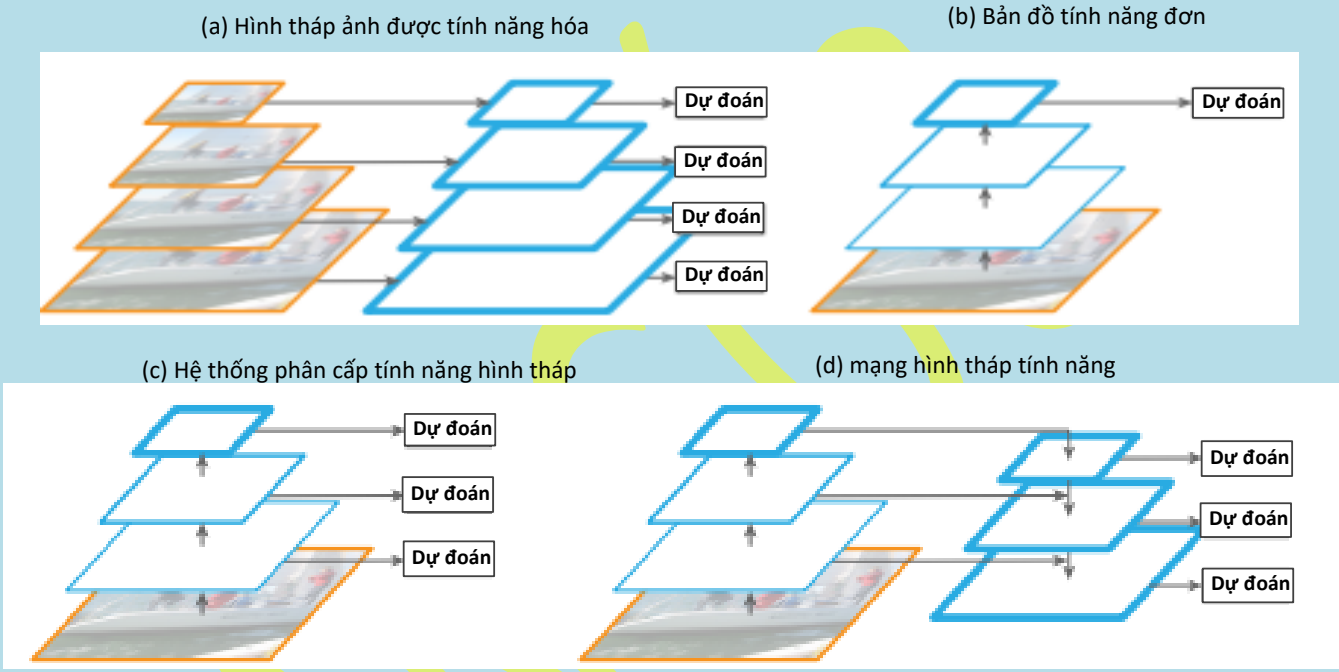
10.6.2.1. Tổng quan

Nhận dạng đối tượng tại các tỷ lệ (kích thước) (tiếng Anh: Scale) rất khác nhau là một thách thức cơ bản trong thị giác máy tính (Computer Vision). Các kim tự tháp tính năng **(**Feature Pyramid**)** còn được gọi là kim tự tháp ảnh được tính năng hóa (Featerized Image Pyramid) tạo thành cơ sở của một giải pháp tiêu chuẩn như trên hình 10.17 (a). Các kim tự tháp là bất biến tỷ lệ/ kích thước (Scale Invariant) theo nghĩa là sự thay đổi tỷ lệ/ kích thước của đối tượng được bù đắp bằng cách dịch mức của nó trong kim tự tháp. Thuộc tính này cho phép một mô hình phát hiện đối tượng trên một dải rộng các tỷ lệ bằng cách quét mô hình trên cả hai: các vị trí và các mức của hình tháp.

Kim tự tháp ảnh được tính năng hóa được sử dụng nhiều trong thời đại các tính năng được thiết kế thủ công. Đối với các nhiệm vụ nhận dạng, các tính năng được thiết kế thủ công phần lớn đã được thay đổi bằng các tính năng được tính toán bởi các mạng tích chập sâu (CNN). CNN cũng bền vững hơn đối với sự thay đổi tỷ lệ (kích thước) khi các tính năng được tính toán trên một đầu vào đơn (hình 10.17(b)).Tuy nhiên cho dù có tính bền vững, các kim tự tháp vẫn cần nhận được các kết quả chính xác nhất. Tất cả các mục hàng đầu (Top Entries) trong các thách thức phát hiện (Dection Challenge) của ImageNet và COCO sử dụng kiểm tra đa tỷ lệ (đa kích thước) trên các kim tự tháp hình ảnh được tính năng hóa. Ưu điểm nguyên tắc của tính năng hóa từng mức của một kim tự tháp hình ảnh là nó tạo ra một mô hình tính năng đa tỷ lệ (đa kích thước) trong đó tất cả các mức đều có ngữ nghĩa mạnh mẽ bao gồm cả các mức phân giải cao.

Tuy nhiên tính năng hóa từng mức của kim tự tháp hình ảnh có các hạn chế rõ ràng. Thời gian suy luận tăng đáng kể (bốn lần theo một nghiên cứu) làm cho cách tiếp cận này không thực tế đối với các ứng dụng thực tế. Hơn nữa mạng đào tạo sâu đầu cuối-đầu cuối trên kim tự tháp hình ảnh không khả thi liên quan đến bộ nhớ, vì thế các kim tự tháp hình ảnh chỉ được sử dụng trong thời gian thử nghiệm.

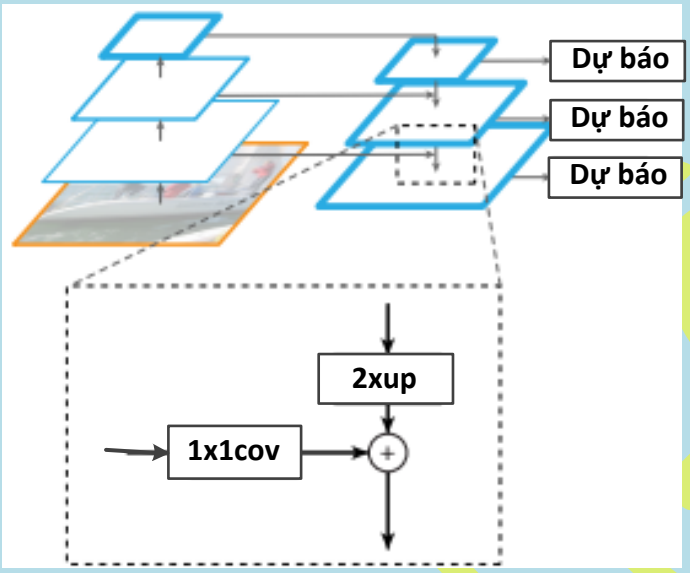
Tuy nhiên các kim tự tháp hình ảnh không phải là cách duy nhất để tính toán một mô hình tính năng đa kích thước (đa tỷ lệ). CNN sâu tính toán phân cấp tính năng theo lớp và với các lớp lấy mẫu phụ (Sub-Sampling Layers), phân cấp tính năng có hình dáng kim tự tháp đa kích thước cố hữu (hình 10.17(c)). Phân cấp tạo ra các khoảng trống ngữ nghĩa lớn gây ra nởi các độ sâu khác nhau. Các bản đồ độ phân giải cao có các tính năng mức thấp gây bất lợi cho khả năng tượng trưng đối với nhận biết đối tượng.



Hình 10.17.(a) Sử dụng một kim tự tháp hình ảnh (Image Pyramid) để xây dựng một kim tự tháp tính năng (Feature Pyramid). Các tính năng được tính toán dựa trên từng tỷ lệ (kích thước) của hình ảnh một cách độc lập, sơ đồ này phát hiện chậm. (b) Các hệ thống phát hiện gần đây sử dụng phân cấp các tính năng tỷ lệ (kích thước) đơn để phát hiện nhanh hơn. (c) Tái sử dụng phân cấp tính năng hình tháp được tính toán bởi CNN như thể là kim tự tháp được tính năng hóa. (d) Mạng kim tự tháp tính năng (FPN: Feature Pyramid Network) nhanh giống như (b) và (c) nhưng chính xác hơn. Trong hình vẽ này, các bản đồ tính năng được biểu thị bằng các đường nét ngoài mầu xanh và các đường dầy hơn biểu thị các tính năng có ngữ nghĩa mạnh hơn.

Mạng kim tự tháp tính năng (FPN: Feature Pyramid Network) trên hình 9.17(d) cho phép lợi dụng hình dạng kim tự tháp của phân cấp tính năng của CNN trong khi tạo ra một kim tự tháp có ngữ nghĩa mạnh tại tất cả các tỷ lệ (kích thước). FPN dựa trên một kiến trúc cho phép kết hợp các tính năng ngữ nghĩa mạnh độ phân giải thấp với các tính năng ngữ

nghĩa thấp độ phân giải cao thông qua đường đi từ đỉnh-xuống (Top-Down Pathway) và các kết nối bên cạnh (Laterral Conections). Kết quả cho ta một kim tự tháp tính năng giầu ngữ nghĩa và được xây dung nhanh chóng từ một tỷ lệ (kích thước) hình ảnh đầu vào đơn (hình 10.18).



Hình 10.18.Minh hoạ khối cơ sở của kết nối bên cạnh và đường từ đỉnh xuống (Top-Down Pathway) được hòa nhập bởi phép cộng

10.6.2.2. Kịch bản

Ta xét một trường ứng dụng sau đây. Một khách du lịch đang lang thang quanh một thành phố và khám phá các điểm tham quan và danh lam thắng cảnh của thành phố. Người dùng nhìn thấy một đối tượng đẹp và cô ấy quyết định quay video về đối tượng đó. Ứng dụng sử dụng các thuật toán học sâu để xử lý video và xác định đối tượng quan tâm và cung cấp thông tin lịch sử về video đó cho người dùng. Hơn nữa, ứng dụng sử dụng deep learning để tái tạo mô hình 3D của đối tượng quan tâm bằng cách sử dụng video 2D đã chụp.

Để làm ví dụ, ta ngghiên cứu các phương pháp phát hiện đối tượng dựa trên Mạng kim tự tháp tính năng (FPN: Feature Paramid Network)). Các mạng này thường bao gồm một xương sống (Blackbone) FPN và một đầu thực hiện suy luận theo nhiệm vụ cụ thể. FPN xử lý các hình ảnh đầu vào ở các tỷ lệ (kích thước) khác nhau để cho phép phát hiện các

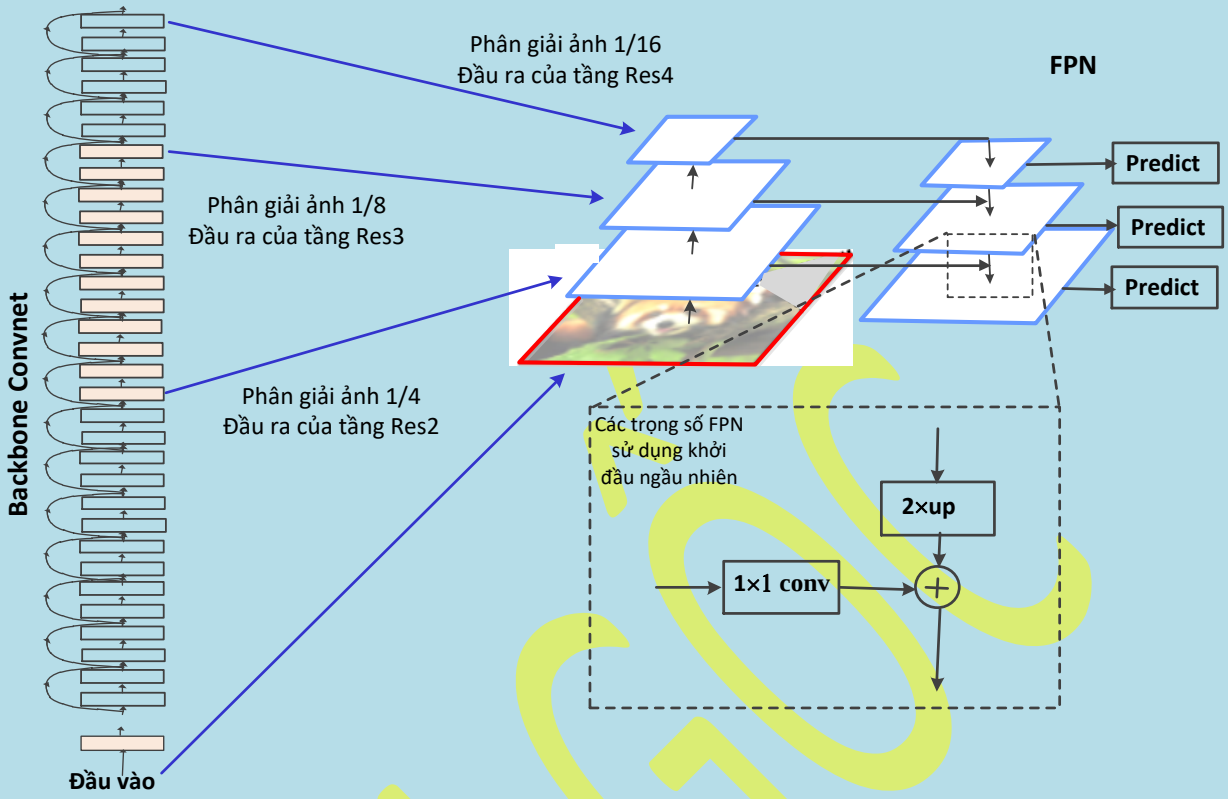
tính năng tỷ lệ nhỏ và tỷ lệ lớn. Ví dụ, đầu có thể phân đoạn các đối tượng, suy ra một hộp giới hạn cho các đối tượng hoặc phân loại các đối tượng.

Xương sống FPN tạo thành phần phức tạp nhất của mạng và rất thích hợp cho giảm tải đến biên/đám mây. Xương sống là một phần chung của một loạt các mạng có thể thực hiện các nhiệm vụ khác nhau. Các bản đồ tính năng được tạo ra sau đó có thể được gửi trở lại UE để suy luận theo nhiệm vụ cụ thể.

Phân tích kiến trúc mạng được thể hiện trong hình 9.19 dưới đây.

Như thể hiện trong hình 10.19, kiến trúc CNN cổ điển được sử dụng làm cốt lõi. FPN được sử dụng để trích xuất các đặc điểm (Feature) ở các tỷ lệ khác nhau của hình ảnh, làm cho nó bất biến theo tỷ lệ. Các nhiệm vụ dự đoán (Predict) tạo thành phần đầu mạng. Bằng cách cắm các đầu mạng khác nhau, các nhiệm vụ AI khác nhau có thể được thực hiện. Điều này làm cho mạng trở thành Mạng đa nhiệm (MTN: Multi-Task Network;). Ví dụ: Mạng đề xuất khu vực (RPN: Regional Proposal Network ; Mạng hoàn toàn tích chập cho phép dự đoán ranh giới đối tượng và điểm số đối tượng (Objectness Score) tại mọi vị trí) có thể được thêm vào để phát hiện và các đối tượng khung trong chuỗi đầu vào bằng cách xuất ra các hộp giới hạn. Các Đầu Nhiệm vụ cụ thể (Task Specific Head) khác có thể được thêm vào để phát hiện con người và tư thế, phân loại đối tượng, theo dõi đối tượng, v.v.

BẢN NHÁP



Hình 10.19. Thí dụ mạng đa nhiệm (Multi Task Network)

Từ hình 10.19 ta thấy, sau khi các tính năng được lấy ra bởi sương sống (Backbone) (hình kim tự tháp theo đường từ dưới lên đỉnh phía trái), mạng kim tự tháp tính năng (hình kim tự tháp theo đường từ đỉnh xuống dưới) cải thiện độ phân dải của các tính năng mức cao theo đường từ đỉnh xuống và hòa nhập chúng với các tính năng mức thấp bằng cách thực hiện lấy mẫu tăng (hay lấy mẫu lại) dựa trên phương pháp nội suy chẳng hạn và hòa nhập bằng phép cộng.

10.6.2.3. Quá trình diễn ra

Điều kiện ban đầu. Người dùng muốn nhận thông tin tức thời và cấu trúc lại để nâng cao trải nghiệm của họ. Thiết bị của người dùng hoạt động bằng pin.

Các luồng dịch vụ diễn ra như sau:

1. Người dùng mở ứng dụng máy ảnh của họ và bắt đầu quay video
2. Ứng dụng xử lý trước video để chuẩn bị cho việc suy luận
3. Ứng dụng truyền các tính năng (Feature) và / hoặc video được trích xuất đến biên/đám mây để xử lý.

4. Mạng thực hiện suy luận được phân chia (Split Infnerence) (ví dụ: chỉ chạy Brackbone) và truyền kết quả trở lại client
5. Ứng dụng chạy suy luận theo nhiệm vụ cụ thể để giải quyết nhiệm vụ cụ thể quan tâm (ví dụ: phát hiện đối tượng, theo dõi,...)
6. Ứng dụng sử dụng các nhãn (Label) và loại (Class) được suy ra để nâng cao chế độ xem của người dùng

Điều kện kết thúc. Người dùng nhận được thông tin nâng cao được trích xuất từ video về đối tượng quan tâm mà người dùng đang quay video.

10.6.2.4. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Các yêu cầu KPI tiềm năng để hỗ trợ trường hợp sử dụng như được đưa ra trong Bảng 10.4, dựa trên hai ví dụ về mô hình/thuật toán phát hiện đối tượng:

- Độ trễ phát trực tuyến đường lên không cao hơn [100-200ms] và người dùng trải nghiệm tốc độ dữ liệu UL là [100-1500] kbit / s;
- Độ trễ phát trực tuyến đường xuống không cao hơn [100-500ms] và người dùng trải nghiệm tốc độ dữ liệu DL là [32-150] Mbit / s.

Bảng 10.4. Nhận dạng: phân tích độ trễ và tốc độ dữ liệu UL / DL mà người dùng đã trải nghiệm

Nhiệm vụ công nhận	**Trễ : maximum (chú thích 4)**	**Tốc độ dữ liệu DL trải nghiệm của người dùng**		**Tốc độ dữ liệu DL trải nghiệm của người dùng**	
		Faster R-CNN (chú thích 1)	**YOLOv3 (chú thích 2)**	**Faster R-CNN**	**YOLOv3**
Phát trực tuyến (Streaming) đường lên	100-200ms			100-1000 kbit/s	200-1500 kbit/s
Suy luận FPN chung	100-500ms	FPN: 4-10fps Sum(Pi)~1MB/khung 32-100Mbit/s không nén (chú thích 3) Hệ số nén 10~100	Nhiều thang đo (tương tự FPN): 1.5 MB bản đồ tính năng/khung 40-150 Mbit/s không nén Hệ số nén 10~100		
Phân loại đối tượng	20-50ms	Thực hiện trên UE	Thực hiện trên UE		
Phát hiện hộp giới hạn	20-50ms	Thực hiện trên UE	Thực hiện trên UE		
Theo dõi đối tượng	50-150ms	Thực hiện trên UE	Thực hiện trên UE		
Truy xuất thông tin nâng cao		vài kByte trên một yêu cầu		Vài kByte trên một yêu cầu	

Kết xuất (hiển thị) lớp phủ	10ms	Thực hiện trên UE	Thực hiện trên UE
Chú thích 1: Faster R-CNN sử dụng kích thước hình ảnh đầuvào 3x224x224. Video được thu nhỏ trên UE xuống độ phân giải mục tiêu đó và sau đó được nén (ví dụ: sử dụng HEVC) và truyền trực tuyến đến biên để xử lý thêm. Chú thích 2: YOLOv3 sử dụng kích thước hình ảnh đầu vào là 3x416x416. Video đã quay được giảm tỷ lệ trên UE xuống độ phân giải mục tiêu và nén trước khi phát trực tuyến đến biên (edge). Chú thích 3: Faster R-CNN sử dụng FPN với ResNet 101 làm xương sống.; do đó dẫn đến các bản đồ tính năng {P2=(256x56x56), P3=(256x28x2), P4=(256x14x14), P5=(256x7x7)}. Chú thích 4: các ước tính độ trễ giả định độ trễ tổng thể khoảng 1 giây từ khi người dùng hướng đến một đối tượng cho đến khi thông tin lớp phủ (Overlay Information) được hiển thị cho người dùng.			

10.6.3. Nâng cao chất lượng phương tiện truyền thông: Nâng cấp chất lượng luồng video trực tuyến (Video Streaming Upgrade)

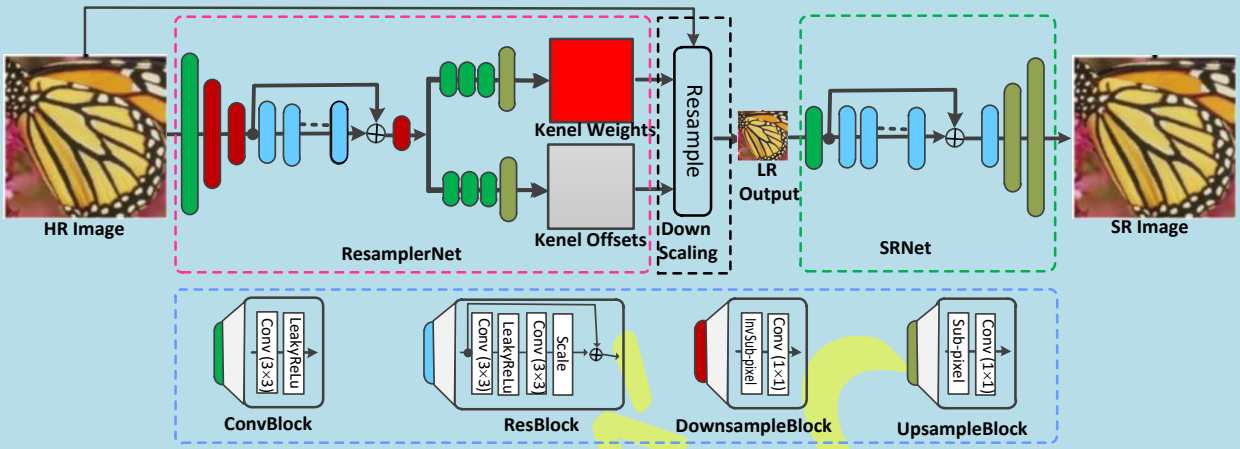
10.6.3.1 . Kịch bản

Ta xét một trường hợp cụ thể sau đây . Một người dùng chơi VR game với kính thực tế ảo (VR Headset). Game được chiết xuất (Render) trên đám mây và phát luồng ngược về UE. Người dùng muốn nhận được trải nghiệm chơi game mạnh đòi hỏi viedo chất lượng cao, chẳng 8K per eye at 120 frames per second (8K mỗi mắt tại tốc độ 120 khung hình trên giây) (8K có nghĩa là vào khoảng 8000 pixel theo chiều ngang của khung hình, chẳng hạn phân giải 7680×4320). Game server chỉ có thể tạo ra dữ liệu video 4K (chẳng hạn phân giải 4096×2160 đối với 4K UHD) do hạn chế của phần cứng, tải và nối mạng.

AI được sử dụng để nâng cấp nội dung 4K thành nội dung 16K (chẳng hạn phân giải 15360× 8640 đối với UHD) để có trải nghiệm người dùng tốt hơn.

Hình 10.20 dưới đây cho thấy một ví dụ về một mạng như vậy.

BẢN NỘC



HR Image (High Resolution Image): ảnh phân giải cao; LR Image (Low Resolution Image): ảnh phân giải thấp; SR Image (Super Resolution Image): ảnh siêu phân giải; ResamplerNet: mạng lấy mẫu lại; SRNet: mạng siêu phân giải; Kenel Weight: trọng số hạt nhân, Kenel Offset: bù trừ hạt nhân; Down Scaling: giảm tỷ lệ; LR Output: đầu ra phân giải thấp; ConvBlock (Convolution Block): khối tích chập; ResBlock (Residual Block): khối dư; DownsmpleBlock: khối giảm mẫu; UpsampleBlock: khối tăng mẫu.

Các hàm trong các khối như sau:

ConvBlock bao gồm: Conv(3×3), LeakyReLu; ResBlock bao gồm: Conv(3×3), LeakyReLu, Conv(3×3), Scale; DownBlock bao gồm: InvSub-pixel, Conv(1×1); UpsampleBlock bao gồm: Sub-pixel, Conv(1×1);

Hình 10.20. Ví dụ về một mạng nâng cấp phát luồng video

Video độ phân giải thấp được truyền đến UE, UE sẽ xử lý video để suy ra phiên bản độ phân giải cao. Các phần lấy mẫu xuống (Down Sampling) và lấy mẫu lên (Up Sampling) của mạng được khớp để tạo ra kết quả tốt nhất. Bất kỳ bản cập nhật nào đối với phần lấy mẫu xuống của mạng sẽ yêu cầu cập nhật ở phía UE đến phần lấy mẫu lên của mạng. Ngoài phiên bản LR của video, trọng số mô hình và cập nhật cấu hình topo có thể cần được gửi đến UE.

Các mô hình được đào tạo trước được tối ưu hóa cho loại nội dung (ví dụ: thể thao, phim hoạt hình, phim, ...) và hệ số tỷ lệ (Scale Factor). Khi người dùng chuyển sang một phần nội dung khác, thiết bị sẽ kiểm tra xem trọng số mô hình được đào tạo trước tương ứng đã được tải xuống trong bộ nhớ đệm hay chưa. Nếu không có sẵn, UE sẽ tải xuống trọng số mô hình được đào tạo trước cho phần nội dung đã chọn và dựa trên hệ số nâng cấp mong muốn. Nội dung mới được hiển thị cho người dùng trong vòng chưa đầy 3 giây kể từ khi người dùng chọn nội dung đó.

10.6.3.2. Quá trình diễn ra

Toàn bộ quá trình diễn ra cho kịch bản này như sau.

Điều kiện ban đầu. Máy chủ chơi game từ xa tạo siêu dữ liệu luồng cùng với luồng được trích xuất bằng cách chạy bộ mã hóa tự động AI (AI Autocoder) trên nội dung được chụp ban đầu.

Các luồng dịch vụ diễn ra như sau:

1. Người dùng bắt đầu phiên chơi game VR trên đám mây trên HMD (Head Mounted Display: màn hình đeo trên đầu) của họ;
2. Trò chơi được khởi chạy trên máy chủ đám mây và trò chơi có thể bắt đầu;
3. Máy chủ đám mây hiển thị và ghi lại nội dung và giảm tỷ lệ (Downscale) xuống video 4K;
4. Máy chủ đám mây cũng chạy DNN để tạo ra luồng siêu dữ liệu (Metadata Stream) sẽ được sử dụng để nâng cấp quy mô/ tỷ lệ (Upscaling)
5. UE sử dụng mạng DNN ngược để tăng tỷ lệ (Upscale) luồng 4K đã nhận lên 16K. Đầu vào cho DNN là video 4K và luồng siêu dữ liệu.
6. UE hiển thị chế độ xem 16K chất lượng cao trên HMD.

Điều kện kết thúc. Người dùng tận hưởng trải nghiệm VR chất lượng cao.

10.6.3.3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Các yêu cầu KPI tiềm năng để hỗ trợ trường hợp sử dụng là:
Hệ thống 5G/6G sẽ hỗ trợ cung cấp trọng số mô hình được đào tạo trước và cập nhật trọng lượng với độ trễ không quá 3 giây.
Bảng 10.5 dưới đây cung cấp ước tính về kích thước trọng số của mô hình.

Bảng 10.5. Kích thước trọng số của mô hình và tốc độ dữ liệu

Tên mô hình (chú thích 2)	Kích thước trọng số (chú thích 4)	Tóc độ số liệu (chú thích 3)
CAR 4x	Lấy mẫu giảm: ~40MB Lấy mẫu tăng: ~170MB	Tải xuống mô hình lấy mẫu tăng: 450Mbit/s (Tải xuống trong khoảng thời gian 3 giây)
SR-GAN 4x	Lấy mẫu giảm: N/A (bicubic) Lấy mẫu tăng: ~6MB	Tải xuống mô hình lấy mẫu tăng: 12Mbit/s (Tải xuống trong khoảng thời gian 3 giây)
SRResNet 4x	Lấy mẫu giảm: N/A (bicubic) Lấy mẫu tăng: ~6MB	Tải xuống mô hình lấy mẫu tăng: 12Mbit/s (Tải xuống trong khoảng thời gian 3 giây)
Chú thích 2: Kích thước của trọng số của các mô hình được đào tạo trước phụ thuộc vào bản chất của nội dung và hệ số tỷ lệ của video. Chú thích 3: Tải xuống trọng lượng mô hình được kích hoạt bởi sự thay đổi nội dung của người dùng và do đó dẫn đến lưu lượng truy cập bùng nổ về bản chất với thời gian nghỉ tương đối dài. Chú thích 4: tất cả các kích thước trọng sô mô hình là từ các mô hình PyTorch được đào tạo trước.		

10.7. ĐIỀU KHIỂN ĐƯỢC PHÂN CHIA CHO ROBOT

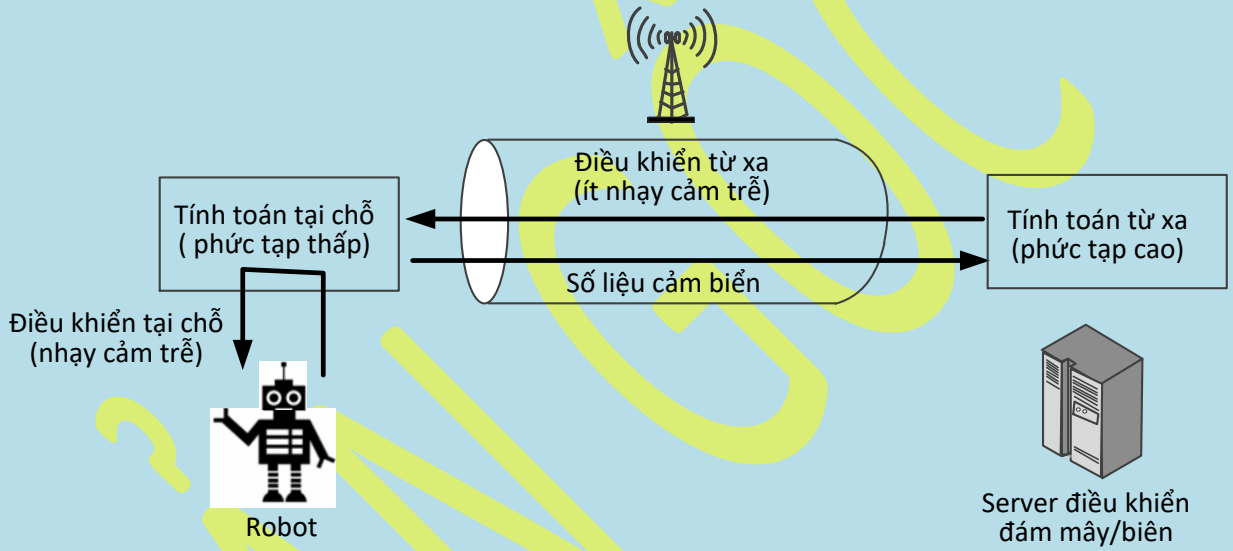
10.7.1. Mô tả

Robot di động đã và đang đóng một vai trò ngày càng quan trọng trong một số tình huống, ví dụ như nhà kho, cứu hộ thảm họa và nhà máy thông minh nhờ tính cơ động cao của chúng. Robot di động cần hoạt động trong một môi trường luôn thay đổi, do đó cần thực hiện cảm biến, lập kế hoạch và điều khiển nhanh chóng và đáng tin cậy. Nếu tính toán tương ứng được thực hiện trên bo mạch trong robot, nó sẽ yêu cầu các tính toán chuyên sâu dẫn đến tăng yêu cầu về khả năng tính toán và mức tiêu thụ điện năng. Tuy nhiên, yếu tố hình thức nhẹ luôn là yêu cầu đối với robot di động làm việc trong môi trường thực tế, điều này khiến robot không được trang bị một số lượng lớn CPU / GPU (Central Processing Unit/Graphic Processing Unit: đơn vị xử lý trung tâm/đơn vị xử lý đồ họa) và pin dung lượng lớn. Như ví dụ được dưa ra trong một nghiên cứu, một robot bốn chân tiên tiến có sẵn trên thị trường, mang 3 kg pin có năng lượng khoảng 650Wh, trong khi GPU cao cấp tiêu thụ hơn 250W năng lượng, ảnh hưởng đáng kể đến tuổi thọ pin nếu sức mạnh tính toán như vậy được nhúng ...

Việc giảm tải các tính toán từ robot lên đám mây đã được nghiên cứu trong nhiều tài liệu. Trong khi đó, dựa vào dữ liệu hoặc code (đoạn chương trình) từ mạng để hỗ trợ hoạt động của robot, các nhà thiết kế robot di động tự động phải xem xét các tình huống mà robot phải bao gồm khả năng xử lý cục bộ cho các phản hồi có độ trễ thấp trong thời gian chất lượng truy cập mạng thay đổi kém hơn.

Hệ thống kết quả (Resulting System: một hệ thống giám sát hướng đến kết quả được sử dụng trong các hoạt động hành động bên ngoài của EU) khác với hệ thống robot điều khiển hoàn toàn từ xa được mô tả trong [5], trong đó việc lập kế hoạch và điều khiển được thực hiện bằng điện toán đám mây và robot chỉ báo cáo dữ liệu cảm biến (bao gồm video) và nhận lệnh điều khiển. Vì điện toán đám mây hoàn chỉnh khó có thể đáp ứng yêu cầu độ trễ của vòng điều khiển phản hồi cấp ms của một số loại robot di động, ví dụ như robot có chân, nên việc điều khiển phân chia robot di động là một giải pháp dễ chịu trong trường hợp này.

[17] giới thiệu một robot điều khiển thăng bằng toàn bộ cơ thể được phân chia qua mạng 5G. Suy luận AI để điều khiển có thể được phân chia giữa robot và máy chủ đám mây: Như thể hiện trong Hình 10.21, phần phức tạp nhưng ít nhạy cảm trễ sẽ được chuyển tải đến tính toán từ xa trong máy chủ điều khiển đám mây hoặc biên. Phần độ phức tạp thấp chứa các điều khoản phản hồi lỗi và nhạy cảm trễ có thể được thực hiện hiệu quả bằng tính toán tại chỗ trong robot. Nếu rô bốt không nhận được điều khiển tối ưu từ "phần điều khiển từ xa" từ máy chủ điều khiển đám mây / biên (cloud/edge control server) do trễ truyền thông hoặc mất gói, nó có thể xấp xỉ hóa "phần điều khiển từ xa" bằng cách sử dụng ma trận phản hồi được tính toán trước đã nhận được trước đó. Và trong một khoảng thời gian nhất định, xấp xỉ hóa vẫn sẽ cho phép robot thực hiện kiểm soát phản hồi cho các nhiệm vụ bằng các xấp xỉ hóa và đảm bảo rằng robot vẫn có thể hoạt động.

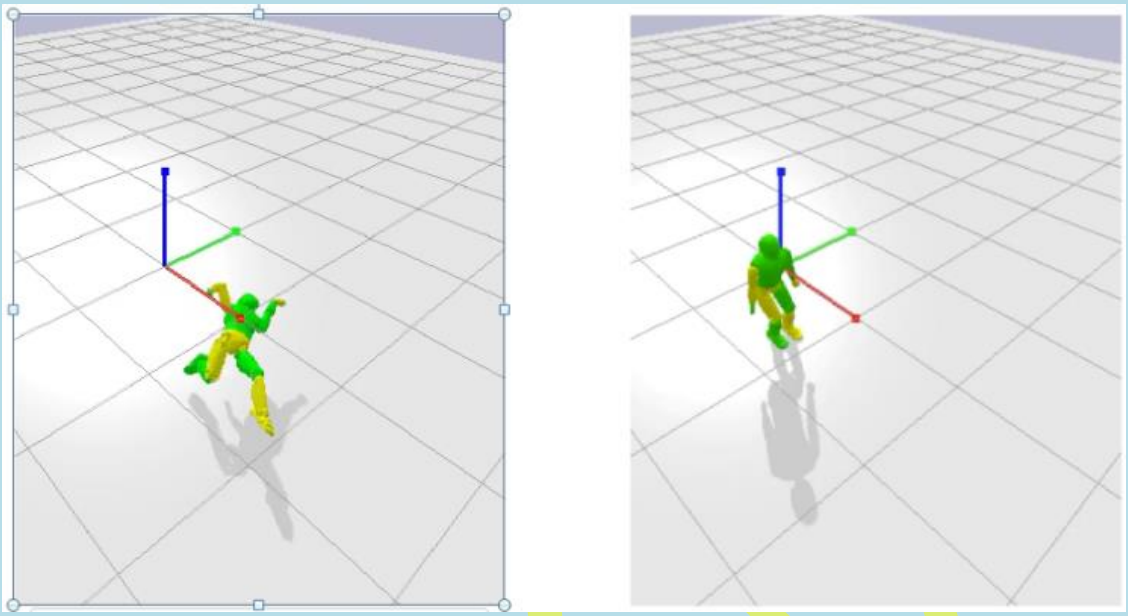


Hình 10.21. Điều khiển được phân chia cho robot có chân trên mạng 5G

Kết quả trong [27] cho thấy, trong trường hợp robot được điều khiển hoàn toàn bởi máy chủ đám mây, robot không thể hoàn thành nhiệm vụ đi bộ nếu độ trễ khứ hồi (Round Trip Latency) lớn hơn 3ms (từ gửi dữ liệu cảm biến đến nhận lệnh điều khiển, bao gồm cả xử lý ở đám mây / biên). Do các lệnh điều khiển bị trễ, robot sẽ bị đổ (như trong Hình 109.22(a). Tuy nhiên, nếu điều khiển phân chia được sử dụng, độ trễ khứ hồi 25ms trong trường hợp xấu hơn có thể được duy trì và robot vẫn có thể thực hiện nhiệm vụ đi bộ (như thể hiện trong Hình 10.22(b).

(a) Điều khiển từ xa 5G không có điều khiển tại chỗ

(b) Điều khiển từ xa 5G có điều khiển tại chỗ tại root



Hình 10.22. Hiệu suất mô phỏng của điều khiển thăng bằng toàn bộ cơ thể rô-bốt trên mạng 5G với độ trễ khứ hồi 25ms

10.7.2. Quá trình diễn ra

Quá trình trên hình 10.21 diễn ra như sau.

Điều kiện ban đầu. Các điểm cuối AI/ML liên quan (ví dụ: UE (robot), AI/ML đám mây/máy chủ biên) chạy các ứng dụng cung cấp khả năng suy luận mô hình AI/ML cho nhiệm vụ điều khiển robot và hỗ trợ hoạt động điều khiển robot phân chia.

Các luồng dịch vụ diễn ra như sau:

2. UE (robot) được kết nối với server điều khiển đám mây/biên thông qua mạng 5G;
3. Việc phân chia các hoạt động điều khiển ở phía robot và phía server điều khiển đám mây/biên cho nhiệm vụ điều khiển robot được xác định bởi robot hoặc mạng;
4. Trong điều kiện chế độ phân chia và điểm phân chia được xác định, robot thực hiện tính toán tại chỗ dựa trên dữ liệu cảm biến thu thập được và gửi các dữ liệu cảm biến này đến server điều khiển đám mây/biên. Server điều khiển đám mây/biên thực hiện tính toán từ xa và cung các đầu ra trở lại robot;
5. Robot điều khiển chuyển động của nó dựa trên kết hợp đầu ra của các tính toán điều khiển tại chỗ và từ xa;
6. Bắt đầu với bước 3) với nhiều thao tác điều khiển hơn, cho đến khi nhiệm vụ điều khiển robot kết thúc.

1. **Kết quả**. Robot nhận điều khiển tại chỗ và từ xa với độ chính xác và độ trễ cần thiết, để hoàn thành các nhiệm vụ di chuyển, có nghĩa là nhiệm vụ cân bằng và nhiệm vụ đi bộ.

10.7.4. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Nếu mọi thứ được thực hiện trên phần biên (Edge), robot cần gửi dữ liệu cảm biến 592B trên mỗi chu kỳ điều khiển (mỗi mili giây) và nhận 200B trên mỗi chu kỳ điều khiển từ "bộ điều khiển từ xa", dẫn đến tốc độ dữ liệu UL là 4,7Mbit / s và tốc độ dữ liệu DL là 1.6Mbit / s. Tuy nhiên, độ trễ khứ hồi tối đa bị giới hạn ở 3ms và độ trễ một chiều để tải xuống "phần điều khiển từ xa" cần được giới hạn ở 1ms. Nếu tuân theo chiến lược phân chia, việc tải xuống "phần điều khiển từ xa" từ máy chủ điều khiển đám mây/biên yêu cầu tải xuống dữ liệu bùng nổ 40kB trên mỗi chu kỳ điều khiển (mỗi mili giây) vì cần thêm thông tin để đảm bảo bộ điều khiển cục bộ có thể tiếp quản trong trường hợp có độ trễ không mong muốn, dẫn đến tốc độ dữ liệu DL của người dùng là 320Mbit/s. Trong trường hợp đó, độ trễ khứ hồi tối đa là 25ms có thể được chấp nhận giữa mỗi chu kỳ điều khiển và độ trễ một chiều để tải xuống "phần điều khiển từ xa" có thể được nới lỏng xuống còn 12ms.

Điều này ngụ ý sự đánh đổi giữa tốc độ dữ liệu DL và độ trễ: So với toàn quyền kiểm soát ở biên, chế độ điều khiển phân chia yêu cầu tốc độ dữ liệu DL có trải nghiệm cao hơn của người dùng, nhưng nới lỏng yêu cầu độ trễ nghiêm ngặt.

Bảng 10.6 cung cấp các yêu cầu về tốc độ dữ liệu và độ trễ đối với điều khiển robot.

Bảng 10.6 Các yêu cầu về tốc độ dữ liệu và độ trễ đối với điều khiển rô-bốt.

Chế độ điều khiển	Tốc độ dữ liệu UL trải nghiệm của người dùng cho tải lên dữ liệu cảm biến	Kích thước của "Phần điều khiển từ xa": tối đa	Tốc độ dữ liệu DL trải nghiệm của người dùng để tải xuống "Phần điều khiển từ xa"	Độ trễ khi tải xuống "Phần điều khiển từ xa": tối đa
Điều khiển hoàn toàn tại biên	4,7Mbit/s	200Bytes	1,6Mbit/s	1ms
Điều khiển phân chia	4,7Mbit/s	40kBytes	320Mbit/s	12ms

10.8. CÁC HOẠT ĐỘNG TÍNH TOÁN PHÂN CHIA CHUYỂN GIAO MÔ HÌNH ĐẶC THÙ PHIÊN

10.8.1. Ngữ cảnh

Một UE, để đạt được kết quả cho người dùng, sử dụng tính toán phân chia (Tính toán phân chia là để giảm tải nhiệm vụ tính toán chuyên sâu giữa UE và mạng). Các nhiệm vụ tính toán chuyên sâu (học máy, tính toán phức tạp sử dụng dữ liệu đầu vào và mô hình, v.v.) có thể được giảm tải hoàn toàn hoặc một phần. Trường hợp sử dụng này xem xét một mục đích sử dụng cụ thể - hiển thị thực tế tăng cường trong kính thực tế ảo với tài nguyên tính toán khiêm tốn. Quyết định làm thế nào để phân chia nhiệm vụ tính toán giữa UE và các tài nguyên tính toán khác có thể phụ thuộc vào các điều kiện của mạng truyền thông và các tài nguyên tính toán có sẵn trong UE.

10.8.2. Quá trình diễn ra

Điều kiện ban đầu.

Alice có kính thực tế tăng cường, một UE với sức mạnh tính toán hạn chế. Cô rời khỏi một chiếc xe buýt và đứng ở trạm xe buýt, nơi đằng sau một màn hình quảng cáo lớn, một gNB được lắp đặt. Kính của Alice có thể truy cập thông qua điểm truy cập. Cô ấy tìm cách tăng cường tầm nhìn của mình về thành phố bằng các chỉ đường và chú thích (giờ mở cửa, lịch sử địa phương, mô tả doanh nghiệp, v.v.). Tăng cường cảnh trực quan của thành phố trong thời gian thực là một nhiệm vụ chuyên sâu về tính toán, được thực hiện bởi một mô hình được phát triển thông qua ML. Mô hình có hai điểm phân chia ứng cử, mỗi điểm phân chia ứng cử có khối lượng công việc và yêu cầu truyền thông khác nhau được (được thể hiện hiển thị trong phần dưới). Chiến lược phân chia tính toán này đã được cài đặt trong UE và server ứng dụng để điểm phân chia có thể được điều chỉnh

động dựa trên sự thay đổi của hiệu suất truyền thông và / hoặc khả năng của UE. Server ứng dụng có thể nằm trong miền MNO (MNO: Mobile Network Operator: nhà mạng di động,tức là một ứng dụng đáng tin cậy) hoặc bên ngoài miền MNO (tức là một ứng dụng phía thứ ba được xác thực). Trường hợp sử dụng này không xem xét cách xác định chiến lược phân chia: liệu điều này có thể được thực hiện một cách tự động hay chiến lược có hình thức nào.

Bảng 10.7 liệt kê khối lượng công việc và yêu cầu truyền thông cho các điểm phân chia.

Bảng 10.7. Khối lượng công việc và yêu cầu truyền thông cho các điểm phân chia

	Tốc độ dữ liệu đầu ra UL (Mbit/s)	**Tải tính toán trong UE**
Điểm phân chia ứng cử 1	120	Thấp
Điểm phân chia ứng cử 2	24	Cao

Kính có khả năng tính toán hạn chế và khả năng này thay đổi theo thời gian. Phương tiện để xác định trạng thái hiện tại (thông tin AI-ML) của UE có sẵn cho mạng (thông qua lớp ứng dụng). Điều kiện được xác định ban đầu là UE có khả năng hỗ trợ một trong hai điểm phân chia ứng cử 1 hoặc 2.

Tài nguyên truyền thông mạng rất lớn, nó được quyết định bởi dịch vụ thực tế tăng cường để áp dụng điểm phân chia ứng cử viên 1, nên tính toán được thực hiện chủ yếu trong mạng, nhận được lượng lớn dữ liệu do kính của cô ấy cung cấp và điều này giúp giảm tính toán trong UE. Lượng lớn dữ liệu được truyền qua luồng QoS với tốc độ dữ liệu đảm bảo (GBR: Garanteed Bit Rate) 200 Mbit/s.

Các luồng dữ liệu diễn ra như sau:

Trường hợp A (điểm phân chia được điều chỉnh dựa trên hiệu năng truyền thông):

Alice đi bộ ra khỏi trạm xe buýt và vùng lân cận của điểm nóng.

Vì Alice đứng cách điểm phát sóng điểm nóng gNB vài mét, Tài nguyên truyền thông không đủ dẫn đến gNB phục vụ không thể giữ luồng QoS với GBR 200 Mbit/s nữa (GBR: Guaranteed Bit Rate: tốc độ bit được đảm bảo). Do đó, điểm quyết định chính sách (có thể ở bất cứ đâu – ta bỏ qua những gì đưa ra quyết định và cách thức) xác định hạ cấp sẽ được thực hiện tại thời điểm A GBR từ 200 Mbit/s xuống 30 Mbit/s và ngay lập

tức thông báo cho UE và server ứng dụng của việc hạ cấp này. Chiến lược phân chia tính toán cho ứng dụng AR bây giờ phải được điều chỉnh, tức là thay đổi sang điểm phân chia ứng cử viên 2, trong đó cần phải tính toán nhiều hơn tại chỗ nhưng tốc độ bit cần thiết cho truyền UL giảm xuống còn 24 Mbit/s.

Chiến lược và ràng buộc cho việc phân chia công việc nằm ngoài phạm vi của trường hợp sử dụng này. (Chúng có thể bao gồm, ví dụ: kết quả một phần có thể được gửi đến UE, có thể hoạt động dưới mức tối ưu với tài nguyên giảm, thông tin mô hình có thể được gửi ở dạng mất dữ liệu/nén vẫn hữu ích, v.v.) Trong mọi trường hợp, một trong những đầu vào quan trọng cho quyết định phân chia công việc là tập hợp các tài nguyên truyền thông hiện có sẵn.

Mạng cung cấp thông tin tài nguyên mạng hiện tại liên quan đến hiệu năng truyền thông từ UE đến mạng bao gồm các tham số QoS mới (GBR = 20Mbit/s), thông tin điều kiện (thời điểm-A) để cập nhật QoS mới, cũng như hiệu năng đầu cuối giữa UE và tài nguyên tính toán (ví dụ: trong Môi trường lưu trữ dịch vụ). Thông tin này được khả dụng (được để lộ) cho 'điểm quyết định chính sách' tính toán phân chia (có thể ở bất kỳ đâu - trong UE, tại biên, tại đám mây, v.v., điều này không liên quan đến trường hợp sử dụng.)

Trường hợp B (điểm phân chia được điều chỉnh dựa trên thông tin về AM/ML của UE):

Ban đầu, điểm phân chia ứng cử 2 được chọn khi khả năng của UE có thể hỗ trợ khối lượng công việc cao.

Thông tin tính toán của UE của Alice được giám sát tại lớp ứng dụng. Khi tài nguyên giao tiếp đủ để hỗ trợ điểm phân chia ứng cử 1, nếu các điều kiện của UE bị suy giảm đủ lớn (ví dụ: do pin cạn kiệt, thiếu bộ nhớ, giảm khả năng tính toán) thì đây sẽ là lý do để chọn điểm phân chia ứng cử 1.

Khi này, điểm quyết định tính toán phân chia điều chỉnh chiến lược tính toán phân chia:

Đối với trường hợp-A: để tránh gián đoạn dịch vụ, điểm quyết định tính toán phân chia chọn điểm phân chia-2 mới trước khi thời gian điểm-A đến. Cách điều này được truyền thông hoặc 'thực thi' nằm ngoài phạm vi của trường hợp sử dụng này và không đề xuất rằng điều này sẽ được tiêu chuẩn hóa.

Đối với trường hợp-B: để đảm bảo trải nghiệm người dùng, điểm quyết định tính toán phân chia chọn điểm phân chia-1 vì trạng thái của UE không đủ để hỗ trợ khối lượng công việc cao nữa.

Thông tin trạng thái UE được giả định là được thu thập thông qua lớp ứng dụng.

Điều kện kết thúc. Alice không nhận thức được sự thay đổi của điểm phân chia mô hình và tiếp tục tận hưởng hiệu năng có thể chấp nhận được khi cô ấy mạo hiểm vào thành phố, ngay cả khi có lẽ nó không tốt bằng khi cô ấy đứng ở trạm xe buýt. Lưu ý rằng trường hợp sử dụng này không kết thúc miễn là Alice tiếp tục sử dụng dịch vụ - vì hiệu suất giao tiếp UE với mạng có thể thay đổi bất cứ lúc nào.

10.9. PHÂN PHỐI VÀ CHIA SẺ MÔ HÌNH /DỮ LIỆU AI/ML TRÊN HỆ THỐNG 5G

10.9.1. Phân phối mô hình/dữ liệu cho nhận dạng ảnh (AI/ML model distribution for image recognition)

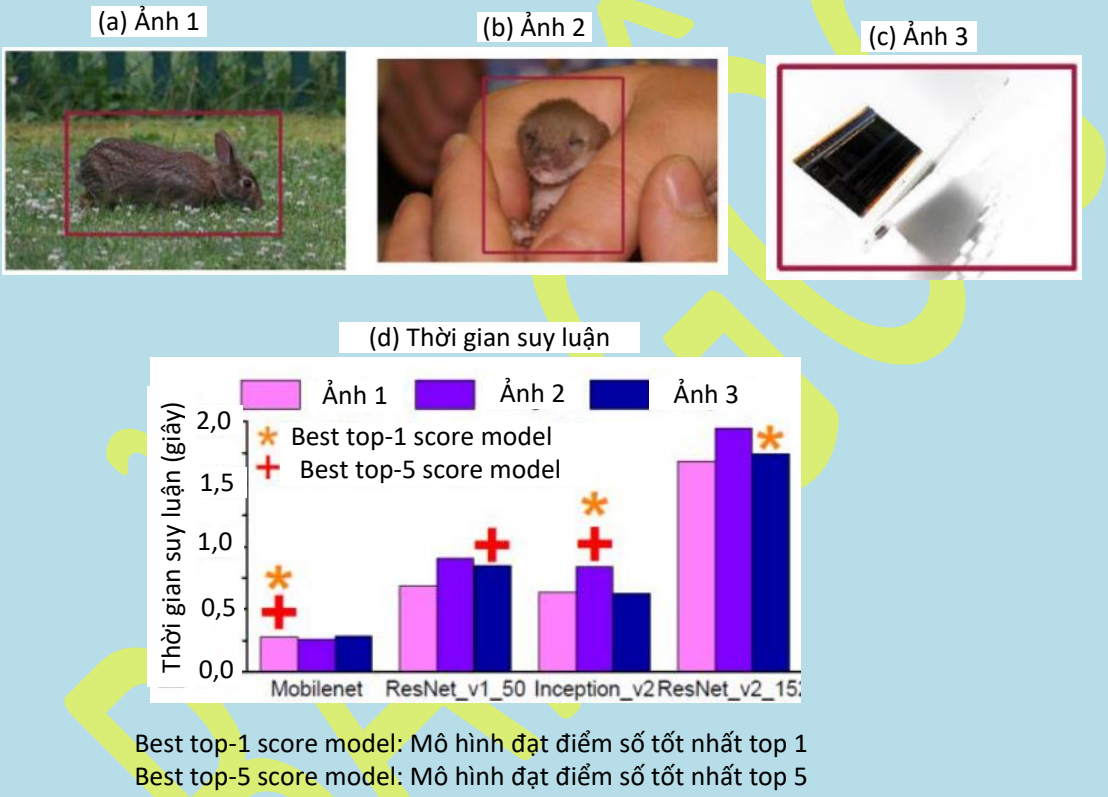
10.9.1.1. Mô tả

Nhận dạng hình ảnh là một lĩnh vực trong đó có sẵn một bộ mô hình AI/MLphong phú được đào tạo trước. Mô hình tối ưu phụ thuộc vào tính năng của hình ảnh/ video đầu vào, môi trường và yêu cầu độ chính xác. Mô hình được sử dụng để xử lý thị giác tại thiết bị cần được cập nhật thích ứng cho các đối tượng thị giác, phông nền, điều kiện ánh sáng, mục đích khác nhau (ví dụ: khôi phục hình ảnh so với phân loại) và thậm chí cả tỷ lệ nén mục tiêu. Mặc dù mô hình tĩnh cũng có thể hoạt động như mặc định trong một số trường hợp, nhưng việc điều chỉnh mô hình cho phù hợp với các điều kiện làm việc khác nhau sẽ giúp cải thiện độ chính xác nhận dạng và trải nghiệm người dùng tốt hơn.

Một ví dụ đã được đưa ra trong [20] cho động lực lựa chọn mô hình tối ưu cho các nhiệm vụ nhận dạng hình ảnh và các môi trường khác nhau. Như thể hiện trong Hình 9.23, 4 mô hình CNN điển hình đã được đánh giá và so sánh cho các nhiệm vụ nhận dạng hình ảnh khác nhau, ví dụ MobileNet_v1_025, ResNet_v1_50 (ResNet với 50 lớp), Inception_v2 và ResNet_v2_152 (ResNet với 152 lớp). Ví dụ này cho thấy rằng mô hình tốt nhất phụ thuộc vào loại hình ảnh đầu vào và yêu cầu nhiệm vụ. Đối với một thiết bị di động cần nhận dạng các nhiều loại hình ảnh và đáp ứng các yêu cầu khác nhau cho các ứng dụng khác nhau, mô hình cần được chuyển đổi thích ứng.

Trong trường hợp mô hình cần được chọn chưa được tải trước trong thiết bị, thiết bị cần tải xuống từ mạng trước khi nhiệmvụ nhận dạng hình ảnh có thể bắt đầu. Một mô hình có thể được tái sử dụng nếu nó được giữ trong bộ nhớ sau lần sử dụng trước đó. Nhưng do tài nguyên lưu trữ hạn chế, thiết bị không thể giữ lại tất cả các mô hình để sử dụng tiềm năng trong lưu trữ. Tốc độ dữ liệu để tải xuống các mô hình cần thiết phụ thuộc vào kích thước của mô hình và độ trễ tải xuống cần thiết.

Cùng với các yêu cầu về hiệu năng ngày càng tăng đối với các hoạt động AI/ML, kích thước của các mô hình cũng tiếp tục tăng lên, mặc dù các kỹ thuật nén mô hình đang được cải thiện. Các kích thước điển hình của các mô hình DNN điển hình để nhận dạng hình ảnh được liệt kê trong Bảng 10.8. Tham số DNN có thể được biểu thị bằng 32 bit để có độ chính xác suy luận cao hơn. Kích thước mô hình và chi phí tải xuống có thể bị nén nếu kích thước của một tham số giảm xuống còn 8 bit, bằng cách có khả năng hy sinh độ chính xác nhận dạng hình ảnh.



Hình 10.23. Ví dụ về việc chọn mô hình tối ưu cho các nhiệm vụ/môi trường nhận dạng hình ảnh khác nhau (hình được phỏng từ [20])

Chú thích: Điểm số top-k (Top-k score) là một chỉ số đánh giá được sử dụng trong học máy để đo lường hiệu suất của một mô hình trong nhiệm vụ phân loại đa lớp (Multi Class Classification Task). Nó đánh giá tần suất mô hình dự đoán chính xác lớp thực (True Class) trong các lớp dự đoán top k.

Bảng 10.8. Kích thước của các mô hình nhận dạng hình ảnh điển hình và tốc độ dữ liệu DL cần thiết để tải xuống trong 1 giây

Mô hình DNN để nhận dạng ảnh	Số lượng tham số (Triệu)	32 bit trên mỗi tham số		8 bit trên mỗi tham số	
		Kích thước của	Tốc độ dữ liệu	Kích thước của	Tốc độ dữ liệu DL

		mô hình (MByte)	DL yêu cầu (Mbit / s)	mô hình (MByte)	yêu cầu (Mbit / s)
AlexNet	60	240	1920	60	480
VGG16	138	552	4416	138	1104
ResNet-152	60	240	1920	60	480
ResNet-50	25	100	800	25	200
GoogleNet	6.8	27,2	217,6	6,8	54,4
Inception-V3	23	92	736	23	184
1.0 MobileNet-224	4,2	16,8	134,4	4,2	33,6

Độ trễ tải xuống mô hình yêu cầu phụ thuộc vào tốc độ mô hình cần sẵn sàng trên thiết bị. Nó bị ảnh hưởng bởi mức độ mà ứng dụng sắp tới có thể được dự đoán. Trong trường hợp sử dụng, chúng ta giả định thiết bị không thể dự đoán và tải xuống trước mô hình cần thiết. Độ trễ tải xuống mô hình yêu cầu phụ thuộc vào tốc độ mô hình cần sẵn sàng trên thiết bị. Nó bị ảnh hưởng bởi mức độ mà ứng dụng sắp tới có thể được dự đoán. Trong trường hợp sử dụng, chúng ta giả định thiết bị không thể dự đoán và tải xuống trước mô hình cần thiết. Khác với video được phát trực tuyến có thể được phát khi một phần nhỏ được nhớ đệm, mô hình DNN chỉ có thể được sử dụng cho đến khi toàn bộ mô hình được tải xuống hoàn toàn.

Ví dụ: nếu độ trễ tải xuống là 1 giây, tốc độ dữ liệu DL cần thiết nằm trong khoảng từ 134,4 Mbit/s đến 1,92Gbit/s trong trường hợp tham số 32 bit, như thể hiện trong Bảng 10.8. Trong trường hợp tham số 8 bit, tốc độ dữ liệu DL yêu cầu có thể được giới hạn ở mức 33,6Mbit / s ~ 1,1Gbit / s.

Một mô hình bao gồm cấu hình topo của mô hình và các thừa số trọng số của mô hình. Cấu hình topo phản ánh cấu trúc của mạng thần kinh (ví dụ: tế bào thần kinh của mỗi lớp, kết nối của các tế bào thần kinh giữa hai lớp lân cận). Kích thước của các tham số mô hình được thể hiện trong Bảng 10.8. Kích thước của tệp cấu hình của một mô hình (tức là cấu hình topo mô hình), thường không vượt quá 1Mbits. Khi một ứng dụng của UE yêu cầu một mô hình, server của bên thứ ba gửi một cấu hình mô hình có thể bao gồm hai phần, tức là topo mô hình và các thừa số trọng số mô hình. Bởi vì topo mô hình và các thừa số số trọng số mô hình đến từ cùng một server, có thể chúng có cùng bộ IP (IP tuple). Sai số truyền của topo mô hình rất quan trọng so với các yếu tố trọng số của mô hình (UE khó có thể chạy mô hình nếu topo mô hình có sai số trong khi các hệ số trọng số có thể có khả năng chịu sai số truyền cao do độ bền của mô hình). Do đó, việc truyền dữ liệu của topo mô hình quan trọng hơn và đòi hỏi độ tin cậy cao hơn.

10.9.1.2. Quá trình diễn ra

Điều kiện ban đầu.

Server AI/ML quản lý nhóm mô hình AI/ML và có khả năng tải mô hình được yêu cầu xuống ứng dụng cung cấp nhận dạng hình ảnh dựa trên AI/ML.

Hệ thống 5G có khả năng cung cấp thông tin liên quan đến mạng 5G cho máy chủ AI/ML.

Các luồng dữ liệu diễn ra như sau:

1. Ứng dụng nhận dạng hình ảnh dựa trên AI/ML được người dùng yêu cầu bắt đầu nhận dạng hình ảnh/video do UE chụp.

2. Mô hình AI/ML được tải xuống từ máy chủ mô hình vào ứng dụng nhận dạng hình ảnh dựa trên AI/ML thông qua mạng 5G.

3. Ứng dụng nhận dạng hình ảnh dựa trên AI/ML sử dụng mô hình AI/ML để suy luận cho đến khi hoàn thành nhiệm vụ nhận dạng hình ảnh.

4. Làm lại Bước 2) đến 3) để lựa chọn lại mô hình AI/ML và tải xuống nếu cần để thích ứng với các điều kiện thay đổi.

Điều kện kết thúc.

Các đối tượng trong hình ảnh hoặc video đầu vào được ứng dụng nhận dạng hình ảnh dựa trên AI/ML nhận dạng và độ chính xác và độ trễ suy luận cần được đảm bảo.

Nhiệm vụ nhận dạng hình ảnh có thể được hoàn thành theo nguồn năng lượng và tính toán sẵn có của UE.

10.9.1.3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Các yêu cầu KPI tiềm năng cần thiết để hỗ trợ trường hợp sử dụng bao gồm:

- Hệ thống 5G sẽ hỗ trợ tải xuống mô hình AI/ML để nhận dạng hình ảnh với độ trễ tối đa như trong Bảng 10.9.
- Hệ thống 5G sẽ hỗ trợ tải xuống mô hình AI / ML để nhận dạng hình ảnh với tốc độ dữ liệu DL trải nghiệm của người dùng như trong Bảng 10.9.
- Hệ thống 5G sẽ hỗ trợ tải xuống mô hình AI/ML để nhận dạng hình ảnh với tính khả dụng của dịch vụ truyền thông không thấp hơn 99.999%.

Bảng 10.9. Phân tích độ trễ tải xuống mô hình nhận dạng hình ảnh tải xuống đối với các ứng dụng ví dụ (tham số 8 bit cho DNN)

Ứng dụng của người dùng	**Độ trễ: tối đa**		**Tốc độ dữ liệu DL trải nghiệm người dùng đối với tải xuống mô hình**
	Độ trễ nhận dạng hình ảnh	**Độ trễ tải xuống mô hình**	
Nhận dạng đối tượng một lần trên điện thoại thông minh	~1s	1s (chú thích 1)	33,6Mbit/s~1,1Gbit/s
Nhận dạng người trong hệ thống giám sát an ninh	~1s	1s (chú thích 2)	33,6Mbit/s~1,1Gbit/s
Nâng cấp ảnh trên điện thoại thông minh	~1s	1s (chú thích 1)	33,6Mbit/s~1,1Gbit/s

NHận dạng video	33ms@30FPS	1s (chú thích 2)	33,6Mbit/s~1,1Gbit/s
Hiển thị AR/chơi game	<5ms (chú thích 3)	1s (chú thích 2)	33,6Mbit/s~1,1Gbit/s
Lái xe từ xa	<5ms (chú thich 4)	1s (chú thích 2)	33,6Mbit/s~1,1Gbit/s
Robot điều khiển từ xa	<5ms (chú thích 5)	1s (chú thích 2)	33,6Mbit/s~1,1Gbit/s

chú thích 1: một mô hình dưới tối ưu được cài đặt sẵn có thể được sử dụng tạm thời trong khi tải xuống mô hình tối ưu. Mô hình tối ưu nên được tải xuống khoảng 1 giây.

chú thích 2: đối với các ứng dụng có camera luôn bật, thiết bị có thể dự đoán mô hình cần thiết và bắt đầu tải xuống trước. Tải xuống mô hình trong vòng 1 giây là chấp nhận được.

chú thích 3: độ trễ chuyển động thành photon của VR (VR Motion-t0 Photon latence: trễ giữa chuyển động vật lý của người dùng, chẳng hạn người này quay đầu, và khi chuyển động này được phản ảnh trên màn hình) nằm trong khoảng 7-15ms. Có thể giả định rằng màn hình AR hoặc chơi game yêu cầu độ trễ đầu cuối tương tự. Xem xét thời gian cần thiết để kết xuất AR (ví dụ: 3D kết xuất các đối tượng ảo, kết xuất tăng cường chồng lấn), việc nhận dạng video nền sẽ được hoàn thành trong vòng 5ms.

chú thích 4: độ trễ một chiều cần thiết để lái xe từ xa là 5ms. Độ trễ khứ hồi được giả định là 10ms. Xem xét thời gian cần thiết của suy luận lái xe dựa trên thị giác tại máy chủ, quỹ độ trễ để nhận dạng đối tượng giao thông có thể được ước tính là 5ms.

chú thích 5: độ trễ đầu cuối cần thiết cho robot điều khiển từ xa hoạt động bằng video là 10 ~ 100ms. Xem xét thời gian cần để suy luận điều khiển robot tại máy chủ, nhận dạng thị giác robot cần được hoàn thành trong vòng 5ms.

10.9.2. Biên tập phương tiện theo thời gian thực với suy luận AI tại chỗ (Real time media editing with on-board AI inference)

10.9.2.1. Mô tả kịch bản

Điện thoại thông minh là thiết bị mà mọi người mang theo và sử dụng ở khắp mọi nơi để ghi âm và quay video. Nó cũng trở thành thiết bị đầu tiên trao đổi nội dung truyền thông với bạn bè và gia đình, để xuất bản trên phương tiện truyền thông xã hội. Điện thoại thông minh cao cấp ngày càng tích hợp CPU và GPU mạnh mẽ hơn và thậm chí cả bộ tăng tốc phần cứng AI chuyên dụng. Khi chất lượng máy ảnh và hình ảnh/video trở thành một yếu tố khác biệt giữa các điện thoại thông minh cao cấp, các mô hình AI/ML để nâng cao khả năng chụp ảnh xuất hiện tại chỗ trên các thiết bị cao cấp này. Các bộ tăng tốc AI dự kiến sẽ cho phép thực hiện các mô hình AI/ML phức tạp trực tiếp trên các thiết bị được kết nối của người dùng cuối; không chỉ nâng cấp ảnh mà cả nâng cao cao âm thanh chất lượng cao và phân tích nội dung video dự kiến sẽ được thực hiện tại chỗ trên điện thoại thông minh. Do đó, điện thoại thông minh sẽ trở thành một thiết bị để chỉnh sửa nội dung phương tiện trước khi chia sẻ qua mạng. Với sự ra đời của 5G/6G, các dịch vụ mới dựa trên tải xuống theo yêu cầu của các mô hình AI/ML lớn được thực thi thời gian thực (gần thời gian thực) trên thiết bị người dùng cuối sẽ xuất hiện; tùy thuộc vào dịch vụ, môi trường, sở thích của người dùng, đặc điểm của thiết bị, v.v., các mô hình DNN này sẽ cần được điều chỉnh hoặc cập nhật theo các ràng buộc về độ trễ nghiêm ngặt để không phải lưu trữ trước tất cả chúng tại chỗ.

Phân tích nội dung phương tiện kết hợp các nhiệm vụ như phát hiện đối tượng, phân đoạn, nhận dạng khuôn mặt, đếm người, theo dõi hoạt động của con người. Trong bảng 10.10, các ví dụ về các mô hình DNN thường được sử dụng để phát hiện đối tượng (với kích thước tương ứng của chúng) được liệt kê.

Bảng 10.10. Kích thước của các mô hình phát hiện đối tượng điển hình

Mô hình để phát hiện đối tượng	Số lượng tham số (Triệu)	Kích thước mô hình (MByte) các tham số 32 bit	Kích thước mô hình (MByte) các tham số 8 bit
MobileNet	3,2	12,8	3,2
DarkNet	20	80	20
SE ResNet	26	104	26
Inception v4	41	164	41
YOLONet	64	256	64
VGGNet	134	536	134

Biên tập nội dung phương tiện kết hợp các nhiệm vụ như cải thiện chất lượng âm thanh và video, dịch ngôn ngữ, ẩn danh khuôn mặt. Trong bảng 10.11 và 10.12, các ví dụ về các mô hình DNN thường được sử dụng cho siêu phân giải hình ảnh và siêu phân giải video được liệt kê.

Bảng 10.11. Kích thước của các mô hình siêu phân giải hình ảnh điển hình

Mô hình cho siêu phân giải hình ảnh	Số lượng tham số (Triệu)	Kích thước mô hình (MByte) các tham số 32 bit	Kích thước mô hình (MByte) các tham số 8 bit
RCAN	15,44	61,78	15,44
SAN	15,71	62,82	15,71
RDN	22,12	88,48	22,12
EDSR	40,73	16,92	40,73
OISR-RK3	41,91	167,64	41,91

Bảng 10.12. Kích thước của các mô hình siêu phân giải videođiển hình

Mô hình cho siêu phân giải video	Số lượng tham số (Triệu)	KÍch thước mô hình (MByte) các tham số 32 bit	KÍch thước mô hình (MByte) các tham số 8 bit
RBPN/4-PF	12,7	50,8	12,7
RBPN/6-PF	12,7	50,8	12,7
VSR-DUF	6,8	27,2	6,8
DRDVSR	0,7	2,8	0,7

Hai cài đặt được xem xét cho trường hợp sử dụng:

a) Người dùng độc lập: một người quay video hoặc bắt đầu cuộc gọi video trên UE của mình trong môi trường ồn ào, có điều kiện ánh sáng khó khăn và gắn thẻ tự động cho mô tả cảnh quay và các đối tượng được nhúng vào video.

a) Đám đông: Trong một sự kiện lớn, như một buổi hòa nhạc trực tiếp, vài nghìn người sử dụng UE của họ để quay phim hoặc chụp ảnh ban nhạc cùng một lúc và yêu cầu thêm thông tin về buổi hòa nhạc như danh sách đĩa nhạc, lời bài hát, nhận dạng khuôn mặt nghệ sĩ, thương hiệu nhạc cụ/thiết bị. Trong bối cảnh này, các UE yêu cầu tải xuống các mô hình DNN để cải thiện việc ghi lại và ghi lại buổi hòa nhạc, đồng thời cung cấp thông tin theo yêu cầu của những người tham dự buổi hòa nhạc. Mỗi UE có thể yêu cầu một số mô hình DNN để thực hiện các nhiệm vụ sau: chụp ảnh và tối ưu hóa video, nhận dạng khuôn mặt nghệ sĩ, nhận dạng nhạc cụ, cải thiện âm thanh và tạo lời bài hát. Với nhóm UE không đồng nhất, hàng nghìn mô hình DNN theo yêu cầu của ứng dụng/dịch vụ - có thể được yêu cầu tải xuống. Các mô hình DNN này được điều chỉnh hoặc cập nhật theo loại và phiên bản hệ điều hành UE, đặc điểm phần cứng và môi trường.

10.9.2.2. Quá trình diễn ra

Điều kiện ban đầu

Cài đặt cho trường hợp sử dụng này như sau. Alice đang tham dự một buổi hòa nhạc trực tiếp đông đúc. Cô ấy rất háo hức để có được các clip phim và hình ảnh như một món quà lưu niệm tuyệt vời của buổi hòa nhạc, nhưng cô ấy lo lắng về điều kiện khó khăn để có được món quà lưu niệm tuyệt vời này vì điều kiện về ánh sáng và âm thanh rất thay đổi, với ánh sáng hạn chế cho khán giả và quá nhiều ánh sáng trên hiện trường. Âm thanh nổi có thể thay đổi và không cân bằng tốt tùy thuộc vào vị trí của cô ấy trong số khán giả và có nền rất ồn ào.

Alice muốn lưu trữ các clip phim và hình ảnh chất lượng tốt trên tài khoản cá nhân của mình trên internet cho tương lai, đồng thời đăng ảnh và video được gắn thẻ tên nghệ sĩ và các thông tin liên quan khác trong buổi hòa nhạc. Vì Alice cũng là một nhạc sĩ nghiệp dư, cô ấy cũng muốn có được thông tin chi tiết theo thời gian thực về cấu trúc của bài hát, lời bài hát và nhạc cụ.

Các điều kiện ban đầu là:

a) Alice đang tham dự một buổi hòa nhạc trực tiếp đông đúc.

b) UE của cô đã được đăng ký vào mạng 5G.

c) Các ứng dụng của điện thoại thông minh của Alice có thể dựa vào các mô hình máy học được tinh chỉnh có sẵn thông qua mạng bao phủ phòng hòa nhạc. Các mô hình này là:

    1) Một mô hình ML cải thiện khả năng chụp ảnh cho phòng hòa nhạc này (một mô hình được tinh chỉnh đặc biệt cho địa điểm hòa nhạc này).

2) Mô hình ML cải thiện khả năng thu âm cho phòng hòa nhạc này (một mô hình được tinh chỉnh đặc biệt cho địa điểm hòa nhạc này).
3) Một mô hình ML chuyên về danh sách đĩa nhạc và lời bài hát của ban nhạc.
4) Một mô hình ML chuyên về nhận dạng khuôn mặt của nghệ sĩ.
5) Một mô hình ML chuyên về nhận dạng nhạc cụ.

Các mô hình được liệt kê ở trên là ví dụ cho trường hợp sử dụng này và không phải là danh sách đầy đủ.

Các luồng dịch vụ.

1) Ngay sau khi bắt đầu buổi hòa nhạc, Alice, cũng như hầu hết người hâm mộ, khởi chạy ứng dụng máy ảnh trên điện thoại di động của mình để quay cảnh và nhận thêm thông tin về ban nhạc, từng nghệ sĩ hoặc các bài hát hoặc nhạc cụ.
2) Cô ấy hướng máy ảnh của thiết bị của mình về phía hiện trường.
3) Môi trường rất tối với các đốm sáng mạnh. Để có đầy đủ chức năng và mang lại trải nghiệm người dùng tốt nhất, ứng dụng máy ảnh sẽ tải xuống các mô hình ML đặc biệt.
4) Các mô hình ML được đề xuất rất hiệu quả trong môi trường này nhưng cũng rất nặng về kích thước.
5) Các mô hình ML có thể được sử dụng cho toàn bộ quá trình chụp, đặc biệt nếu môi trường vẫn ổn định. Nếu môi trường thay đổi đáng kể hoặc có sẵn các mô hình ML tốt hơn, các mô hình ML có thể được cập nhật dần dần cho phù hợp với một số quy tắc vận hành (như số lượng cập nhật ML tối đa mỗi giây). Trong mọi trường hợp, ứng dụng máy ảnh tiếp tục hoạt động liền mạch.
6) Các luồng âm thanh và video được ghi lại, cải thiện chất lượng, xử lý để trích xuất và hiển thị thông tin bổ sung, đồng thời được lưu trữ trong thời gian thực trên chính điện thoại di động hoặc được cung cấp dưới dạng luồng trực tiếp.

Điều kiện kết thúc

Alice có thể thấy rằng ngay cả trong điều kiện ánh sáng khắc nghiệt và với nền ồn ào, ảnh và video rất tuyệt, thông tin bổ sung được cung cấp và tất cả đều được gắn thẻ chính xác theo yêu cầu.

Các điều kiện kết thúc là:

1) Ảnh và video được lưu trữ trên điện thoại di động với chất lượng cao được cải thiện, sẵn sàng để tải lên và chia sẻ trên phương tiện truyền thông xã hội hoặc chia sẻ trực tiếp trên phương tiện truyền thông xã hội.

2) Ghi âm có chất lượng cao với khả năng giảm tiếng ồn xung quanh, cải thiện cân bằng âm thanh nổi.

3) Thông tin bổ sung về ban nhạc, bài hát/lời bài hát, nhạc cụ, v.v. được hiển thị trên điện thoại di động và được lưu trữ trong siêu dữ liệu của bản ghi phương tiện.

4) Alice có thể trực quan hóa thông tin bổ sung và tải ảnh và video lên (các) mạng xã hội của mình với các thẻ và thông tin liên quan do mô hình cung cấp, đồng thời lưu trữ những thông tin trên máy chủ truyền thông cá nhân của cô ấy.

10.9.2.3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Các yêu cầu mới tiềm năng cần thiết để hỗ trợ kết quả trường hợp sử dụng của các tham số sau:

a) Kích thước mô hình AI/ML.

b) Độ chính xác của mô hình.

c) Hạn chế độ trễ của ứng dụng hoặc dịch vụ.

d) Số lượt tải xuống đồng thời, tức là số lượng UE yêu cầu tải xuống mô hình AI/ML trong cùng một khoảng thời gian.

Số lượng tải xuống đồng thời phụ thuộc vào mật độ UE trong khu vực được bao phủ và kích thước khu vực được bao phủ.

Các bảng 10.13, 10.14 và 10.15 chứa KPI cho các khía cạnh khác nhau của trường hợp sử dụng biên tập phương tiện theo thời gian thực.

Bảng 9.13. Kích thước điển hình của mô hình AI/ML cho trường hợp sử dụng

Mô hình AI/ML	Số lượng các tham số (trệu)	Kích thước mô hình AI (MByte)	Chú thích
MobileNet	3,2	3,2	Các tham số 8 bit
MobileNet	3,2	12,8	Các tham số 32 bit
RCAN	15,44	15,44	Các tham số 8 bit
DarkNet	20	20	Các tham số 8 bit
Inception v4	41	41	Các tham số 8 bit
RCAN	15,44	61,78	Các tham số 32 bit
YOLONet	64	64	Các tham số 8 bit
DarkNet	20	80	Các tham số 32 bit
VGGNet	134	134	Các tham số 8 bit
Inception v4	41	164	Các tham số 32 bit
YOLONet	64	256	Các tham số 32 bit

VGGNet	134	536	Các tham số 32 bit

Từ bảng 10.13, các mô hình AI/ML hiện có sẵn để xây dựng trường hợp sử dụng có kích thước thay đổi từ 3,2 MB đến 536 MB.

Như đã chỉ ra ở trên, kích thước của các mô hình AI/ML có thể được giảm bớt trước khi truyền bằng các kỹ thuật nén mô hình chuyên dụng. Ngược lại, các mô hình AI/ML với nhiều lớp mạng neuron hơn và kiến trúc phức tạp hơn phát sinh để giải quyết các tác vụ phức tạp hơn và cải thiện độ chính xác. Xu hướng này dự kiến sẽ tiếp tục trong những năm tới. Kích thước mô hình điển hình trong khoảng từ 3 MB đến 500 MB dường như là một sự thỏa hiệp hợp lý để xem xét cho trường hợp sử dụng này.

Trong phần sau, hai danh mục được xem xét cho kích thước mô hình AI/ML:

a) Kích thước mô hình AI/ML dưới 64 MB, có thể được liên kết với các mô hình được tối ưu hóa để truyền nhanh,

b) Kích thước mô hình AI/ML dưới 500 MB, có thể được liên kết với các mô hình được tối ưu hóa để có độ chính xác cao hơn.

Độ trễ tối đa trong chức năng của ứng dụng hoặc dịch vụ:

a) Dịch vụ videocall: độ trễ đầu cuối-đầu cuối dưới 200 ms,

b) Ứng dụng quay video, phát trực tuyến video và nhận dạng đối tượng: độ trễ dưới 1 giây.

Các kết quả tốc độ dữ liệu DL trải nghiệm của người dùng từ kích thước và giá trị độ trễ tối đa của mô hình AI/ML ở trên được tóm tắt trong bảng 10.14.

Bảng 10.14. Tải xuống mô hình AI/ML cho trường hợp sử dụng- UE đơn – các KPI

Tải xuống mô hình trường hợp sử dụng	Kích thước mô hình AI/ML	Độ trễ: tối đa	Tốc độ dữ liệu DL trải nghiệm bởi người dùng
Mô hình đơn / UE đơn	[3 MB – 64 MB]	<1 s	[24 Mb/s ~ 512 Mb/s]
Mô hình đơn / UE đơn	[3 MB – 64 MB]	<200 ms	[120 Mb/s ~ 2,56 Gb/s]
Mô hình đơn / UE đơn	[64 MB – 500 MB]	<1 s	[512 Mb/s ~ 4 Gb/s]
Mô hình đơn / UE đơn	[64 MB – 500 MB]	<200 ms	[2,56 Gb/s ~ 20 Gb/s]

Như đã chỉ ra ở trên, số lượt tải xuống đồng thời là tham số thứ ba để xác định các yêu cầu mới tiềm năng. Điều này tương ứng với số lượng UE tối đa yêu cầu tải xuống mô hình AI/ML trong cùng một cửa sổ thời gian và cùng một khu vực/ô được bao phủ.

Trường hợp của một phòng hòa nhạc là một minh họa cho kịch bản, "Truy nhập băng thông rộng trong đám đông (từ TS 22.261). Kịch bản này giả định mật độ người dùng tổng thể là 500 000 UE / km^2 (tức là 0,5 UE / m^2) và hệ số hoạt động là 30%.

Số lượng UE điển hình trong một phòng hòa nhạc thay đổi từ ~ 1000 chỗ ngồi đến ~ 5000 chỗ ngồi.

Dựa trên các số liệu này và giả định hoạt động UE, số lượng tải xuống đồng thời được ước tính như trong bảng 10.15.

Bảng 10.15. Số lượt tải xuống đồng thời ước tính

Số lượng UE	Diện tích ước tính (Mật độ 0.5 UE / m^2)	Hệ số hoạt động	Số lượt tải xuống đồng thời trong cùng một ô
1000	2000 m^2	1 %	10
5000	10000 m^2	1 %	50

Từ bảng 10.14 và bảng 10.15, các yêu cầu về khu vực được bao phủ được ước tính như sau:

Bảng 10.16. Yêu cầu về tốc độ dữ liệu DL khu vực được bao phủ ước tính

Số lượng UE	Hệ số hoạt động	Số lượt tải xuống đồng thời	Kích thước mô hình AI/ML model	Độ trễ: cực đại	Tốc độ dữ liệu DL trải nghiệm người dùng	Tốc lệ dữ liệu DL trải nghiệm người dùng tổng hợp cho khu vực được bảo hiểm
1000	1 %	10	[3 MB – 64 MB]	<1 s	[24 Mb/s ~ 512 Mb/s]	[240 Mb/s ~ 5,12 Gb/s]
			[3 MB – 64 MB]	<200 ms	[120 Mb/s ~ 2,56 Gb/s]	[1,2 Gb/s ~ 25,6 Gb/s]
			[64 MB – 500 MB]	<1 s	[512 Mb/s ~ 4 Gb/s]	[5,12 Gb/s ~ 40 Gb/s]
5000	1 %	50	[3 MB – 64 MB]	<1 s	[24 Mb/s ~ 512 Mb/s]	[1,2 Gb/s ~ 25,6 Gb/s]
			[3 MB – 64 MB]	<200 ms	[120 Mb/s ~ 2,56 Gb/s]	[6 Gb/s ~ 128 Gb/s]
			[64 MB – 500 MB]	<1 s	[512 Mb/s ~ 4 Gb/s]	[25,6 Gb/s ~ 200 Gb/s]

10.9.3. Phân phối mô hình AI/ML để nhận dạng giọng nói

10.9.3.1. Mô tả

Xử lý giọng nói dựa trên AI/ML đã được sử dụng rộng rãi trong các ứng dụng trên thiết bị di động (ví dụ: điện thoại thông minh, trợ lý cá nhân, dịch ngôn ngữ), bao gồm nhận dạng giọng nói tự động (ASR: Automatic Speech recognition), dịch giọng nói, tổng hợp giọng nói. Nhận dạng giọng nói để đọc chính tả, tìm kiếm và ra lệnh bằng giọng nói đã trở thành một tính năng tiêu chuẩn trên điện thoại thông minh và thiết bị đeo.

Các yêu cầu dịch vụ đối với ASR đã được giải quyết trong [29]. Các hệ thống ASR truyền thống dựa trên mô hình Markov ẩn (HMM: hidden Markov model) và mô hình hỗn hợp Gaussian (GMM: Gaussian mixture model). Tuy nhiên, các hệ thống HMM-GMM chịu WER (Word Error Rate :Tỷ lệ lỗi từ) tương đối cao với sự hiện diện của tiếng ồn môi trường. Mặc dù một số cải tiến đã được phát triển bao gồm "tăng cường tính năng" (cố gắng loại bỏ tiếng ồn làm hỏng khỏi các quan sát trước khi nhận dạng) và "thích ứng mô hình" (giữ nguyên các quan sát và thay vào đó cập nhật các tham số mô hình của bộ nhận dạng để đại diện hơn cho giọng nói được quan sát), các mô hình truyền thống khó có thể đáp ứng các yêu cầu của các ứng dụng thương mại. Các mô hình âm thanh dựa trên mạng neuron sâu (DNN) có độ bền tiếng ồn đáng chú ý và đã được sử dụng rộng rãi trong các ứng dụng ASR trong các thiết bị di động.

Ngày nay, hầu hết các ứng dụng ASR trên điện thoại thông minh đều hoạt động trong các máy chủ đám mây. Thiết bị đầu cuối tải giọng nói lên máy chủ đám mây, sau đó tải kết quả được giải mã trở lại thiết bị. Tuy nhiên, nhận dạng giọng nói dựa trên đám mây có khả năng gây ra độ trễ cao hơn (không chỉ do độ trễ mạng 4G / 5G mà còn do độ trễ internet) và vấn đề bảo mật và kết nối mạng độ tin cậy cần được xem xét.

Hệ thống nhận dạng giọng nói nhúng chạy trên thiết bị di động đáng tin cậy hơn và có thể có độ trễ thấp hơn. Hiện tại, một số ứng dụng ASR sẽ chuyển từ suy luận mô hình dựa trên đám mây sang suy luận mô hình ngoại tuyến khi vùng phủ sóng đường lên của người dùng di động trở nên yếu, ví dụ như khi vào tầng hầm hoặc thang máy. Tuy nhiên, các mô hình ASR cho máy chủ đám mây quá phức tạp đối với tài nguyên tính toán và lưu trữ trên thiết bị di động. Kích thước của mô hình ASR dựa trên ML chạy trên máy chủ đám mây đã tăng nhanh chóng trong năm gần đây, từ ~ 1GByte lên ~ 10GByte, không thể chạy trên thiết bị di động. Do hạn chế, chỉ các ứng dụng ASR đơn giản, ví dụ như phát hiện từ đánh thức (wakeword), mới có thể được triển khai trên điện thoại thông minh. Thực hiện các ứng dụng ASR phức tạp hơn, ví dụ như nhận dạng giọng nói liên tục từ vựng lớn (LVCSR: large vocabulary continuous speech recognition) vẫn là một lĩnh vực thách thức đối với một trình nhận dạng giọng nói ngoại tuyến.

Vào năm 2019, một trình nhận dạng LVCSR ngoại tuyến hiện đại dành cho thiết bị di động Android đã được công bố. Bộ nhận dạng đầu cuối phát trực tuyến dựa trên mô hình bộ chuyển đổi mạng neuron hồi quy (RNN-T: recurrent neural network transducer). Người ta nói rằng, bằng cách sử dụng tất cả các loại cải tiến và tối ưu hóa, dung lượng bộ nhớ có thể được giảm đáng kể và tính toán có thể được tăng tốc. Mô hình có thể được nén thành 80MB. Trong khi đó, mô hình ASR được nén để phù hợp với việc sử dụng trên thiết bị di động, độ bền đối với các loại tiếng ồn xung quanh khác nhau đã bị hy sinh. Khi môi trường tiếng ồn thay đổi, mô hình cần được chọn lại và trong trường hợp mô hình không được giữ trong thiết bị, mô hình cần được tải xuống từ máy chủ đám mây/biên của chủ sở hữu mô hình AI / ML thông qua mạng 5G.

10.9.3.2. Quá trình diễn ra

Điều kiện ban đầu

UE chạy một ứng dụng cung cấp khả năng suy luận mô hình AI/ML để nhận dạng giọng nói.

Máy chủ AI/ML quản lý nhóm mô hình AI/ML (AI/ML Model Pool) và có khả năng tải xuống mô hình được yêu cầu đến ứng dụng cung cấp nhận dạng giọng nói dựa trên AI/ML.

Hệ thống 5G có khả năng cung cấp thông tin liên quan đến mạng 5G cho máy chủ AI/ML.

Các luồng dịch vụ

1. Ứng dụng nhận dạng giọng nói dựa trên AI/ML được người dùng yêu cầu bắt đầu nhận dạng giọng nói được ghi lại.

2. Mô hình AI/ML được tải xuống từ máy chủ mô hình về ứng dụng nhận dạng giọng nói dựa trên AI/ML thông qua mạng 5G.

3. Ứng dụng nhận dạng giọng nói dựa trên AI/ML sử dụng mô hình AI / ML để suy luận cho đến khi nhiệm vụ nhận dạng giọng nói kết thúc.

4. Làm lại Bước 2) đến 3) để lựa chọn lại mô hình AI/ML và tải xuống lại nếu cần để thích ứng với các điều kiện thay đổi.

Điều kiện kết thúc

Nội dung trong giọng nói đầu vào được ứng dụng nhận dạng giọng nói dựa trên AI/ML nhận dạng và độ chính xác và độ trễ suy luận cần được đảm bảo.

Nhiệm vụ nhận dạng giọng nói có thể được hoàn thành theo nguồn năng lượng và tính toán sẵn có của UE.

10.9.3.4. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Mô hình ASR chỉ có thể được sử dụng cho đến khi toàn bộ mô hình được tải xuống hoàn toàn. Và thiết bị cần áp dụng mô hình ASR mạnh mẽ về tiếng ồn thích ứng với môi trường tiếng ồn đang thay đổi. Nói chung, micrô của thiết bị chỉ được bật khi ứng dụng nhận dạng giọng nói được kích hoạt. Thiết bị cần xác định môi trường tiếng ồn và tải xuống mô hình ASR tương ứng với độ trễ thấp (mức 1 giây). Kích thước của một số mô hình ASR điển hình được liệt kê trong Bảng 10.17, được sử dụng để lấy tốc độ dữ liệu cần thiết.

Bảng 10.17. Kích thước của các mô hình nhận dạng giọng nói điển hình và tốc độ dữ liệu DL trải nghiệm của người dùng để tải xuống trong 1 giây (8 bit cho mỗi tham số)

Mô hình DNN để nhận dạng giọng nói	Số lượng tham số (Triệu)	Kích thước của mô hình (MByte)	Tốc độ dữ liệu DL trải nghiệm của người dùng (Mbit

			/ s)
RNN-CTC	26,5	26,5	212
ResCNN-LAS	6,6	6,6	52,8
QuartzNet-15×5	19	19	152
Bộ nhận dạng giọng nói Gboard (Gboard speech recognizer)	N.A.	80	640

10.9.4. Quản lý mô hình AI như một dịch vụ (AI model management as a Service)

10.9.4.1. Mô tả

Các mô hình AI/ML có thể được phân loại và tạo ra từ các góc độ khác nhau, chẳng hạn như dịch vụ, người dùng, thời gian, vị trí, v.v. Vì vậy, có thể có một số lượng lớn các mô hình AI/ML (bao gồm dữ liệu cho các yếu tố cấu trúc và trọng số, độ dốc (gradient)) được sử dụng để suy luận và đào tạo. Do hạn chế về dung lượng lưu trữ, UE không phải lúc nào cũng tải trước tất cả các mô hình AI/ML cho các tác phẩm khác nhau. Đây có thể là một cơ hội thương mại mới khi các nhà khai thác cung cấp dịch vụ giúp quản lý và phân phối các mô hình AI/ML để UE có thể có được một mô hình phù hợp ngay lập tức.

Trong tương lai, dự kiến các công ty bên thứ 3 sẽ sử dụng các mô hình AI để hỗ trợ các loại dịch vụ khác nhau như hướng dẫn du lịch toàn cảnh sử dụng thực tế tăng cường (AR) trong khu nghỉ dưỡng. Tuy nhiên, hầu hết bên thứ 3 không có tài nguyên (máy chủ) ở những nơi phân tán. Xem xét kích thước lớn của các mô hình AI, thời gian tải xuống nghiêm ngặt (được minh họa trong trường hợp sử dụng 10.9.1, 10.9.2) và dung lượng lưu trữ UE hạn chế, các công ty bên thứ 3 ủy quyền cho các công ty chuyên nghiệp quản lý các mô hình AI của họ thay vì tự làm điều tương tự cục bộ.

Vì tài nguyên đám mây của nhà nhà mạng (Network Operator) có lợi thế để quản lý các mô hình AI/ML khác nhau một cách tập trung (ví dụ: máy chủ đám mây) và cục bộ (ví dụ: máy chủ MEC), các dịch vụ đó cũng có thể được để lộ cho người dùng bên thứ 3. Đặc biệt,

• Nhà điều mạng có nhu cầu riêng để duy trì một nhóm mô hình AI/ML cho hoạt động kinh doanh của riêng mình như tối ưu hóa mạng;

• Bên thứ 3 muốn sử dụng một số mô hình AI/ML phổ biến đã được lưu trữ trong đám mây của nhà mạng;

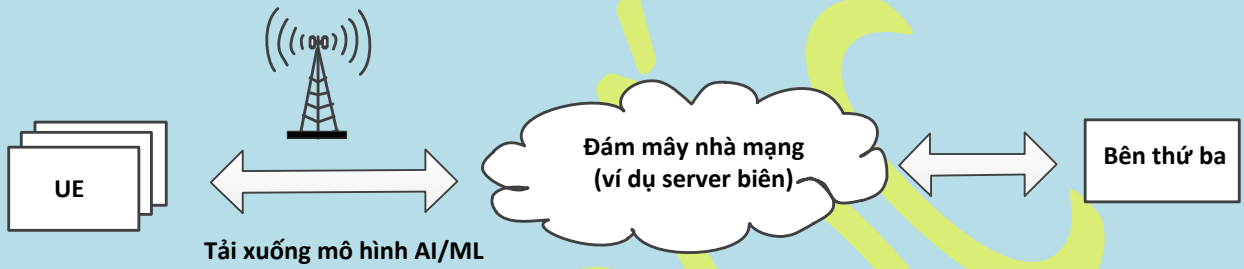
• Do hạn chế tài nguyên của bên thứ 3, họ muốn thuê tài nguyên đám mây của nhà mạng để quản lý các mô hình AI của họ.

Do đó, như thể hiện trong Hình 10.24, Quản lý mô hình AI bao gồm:

• Cho phép bên thứ 3 gọi các khả năng do hệ thống 5G/6G tiết lộ để tải lên / tải xuống / cập nhật / xóa / lưu trữ / giám sát các mô hình AI.

• Truyền mô hình AI cho người dùng một cách hiệu quả theo tình huống (ví dụ: đi vào một khu vực nhất định).

• Vì hệ thống 5G có thể thu thập dữ liệu truyền thông, trải nghiệm người dùng, v.v., nó có thể thực hiện phân tích dữ liệu cho trải nghiệm người dùng, có thể tạo ra kết quả phân tích để giúp đào tạo đám mây của nhà khai thác và cải thiện các mô hình AI cho bên thứ 3.



Hình 10.24. Đám mây nhà điều hành để quản lý mô hình

10.9.4.2. Quá trình diễn ra

Điều kiện ban đầu

Đám mây của nhà mạng (ví dụ: máy chủ biên) lưu trữ nhiều mô hình AI/ML theo yêu cầu của nhà khai thác hoặc bên thứ 3.

Đám mây của nhà mạng (ví dụ: máy chủ biên) có khả năng phân phối mô hình được lưu trữ trong đám mây của nhà mạng cho các thiết bị.

Các luồng dịch vụ

1. Công ty A cung cấp hướng dẫn viên du lịch toàn cảnh (panorama tourist) sử dụng công nghệ thực tế tăng cường (AR). Họ cung cấp dịch vụ hướng dẫn trong khu thương mại và khu du lịch. UE cần tải xuống mô hình A và B (cả hai đều là VGGnet 32 bit, 536MByte) tương ứng;

2. Để đạt được SLA (Service Level Agreement: thỏa thuận cấp độ dịch vụ) , công ty A cho nhà mạng biết rằng UE yêu cầu mô hình A ở khu vực A và mô hình B ở khu vực B;

3. Khi UE di chuyển đến một nơi, máy chủ biên địa phương sẽ kích hoạt để thiết lập khả năng tăng tốc QoS và UE tải xuống mô hình tương ứng (A hoặc B) được lưu trữ trong máy chủ biên địa phương kịp thời để người dùng có thể tận hưởng thế giới AR liên tục mà không bị gián đoạn rõ ràng khi mô hình thay đổi.

Điều kiện kết thúc

UE sử dụng Mô hình hướng dẫn du lịch toàn cảnh.

10.9.4.3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Tùy thuộc vào sự đồng ý của người dùng, chính sách của nhà khai thác và các yêu cầu quy định của khu vực hoặc quốc gia, hệ thống 5G/6G sẽ có thể cung cấp khả năng để lộ thông tin (ví dụ: tốc độ dữ liệu đo được, độ trễ, kết quả phân tích mạng) cho ứng dụng của bên thứ ba được ủy quyền để hỗ trợ đào tạo và giám sát các mô hình AI/ML.

Hệ thống 5G/6G sẽ có thể hỗ trợ ứng dụng của bên thứ ba được ủy quyền để phân phối mô hình AI/ML từ 3,2~536MB đến ứng dụng của bên thứ ba chạy trên thiết bị trong vòng chưa đầy 1 giây với mật độ người dùng lên đến 5000~10000/km2 trong khu vực đô thị.

10.9.5. Hệ thống nối mạng ô tô dựa trên AI/ML (AI/ML based Automotive Networked Systems)

10.9.5.1. Mô tả kịch bản

Trường hợp sử dụng này là minh họa cho một ví dụ cụ thể nhưng có thể dễ dàng mở rộng cho các tình huống nâng cao hơn liên quan đến các kịch bản thảm họa, hoạt động cứu hộ hoặc thậm chí là hoạt động của xe tự lái. Trọng tâm của kịch bản này là các hệ thống suy luận và học chuyển giao AI/ML giữa máy với máy được kết nối với nhau bằng mạng 5G/6G để cung cấp các ứng dụng và dịch vụ ô tô thông minh.

Có hai loại mô hình ML chính và xử lý mô hình được xem xét trong trường hợp sử dụng này:

1) Cập nhật các mô hình ML lớn bằng cách sử dụng đào tạo phi gian thực – Các mô hình ML này được đào tạo và tối ưu hóa với hàng triệu hoặc hàng tỷ tham số bằng cách sử dụng tài nguyên điện toán mở rộng để đạt được độ chính xác cao nhất có thể dựa trên các bộ dữ liệu đào tạo đầu vào cụ thể. Đào tạo này được thực hiện trong một khoảng thời gian dài và mô hình ML được đào tạo đầy đủ là mô hình cơ sở được cài đặt khi sản xuất ban đầu cho các thiết bị trong các trường hợp sử dụng được sẽ mô tả dưới đây. Các mô hình được đào tạo đầy đủ này có thể được cập nhật với dữ liệu mô hình AI-ML được cung cấp bên ngoài và cũng có thể tự cải thiện dựa trên dữ liệu cảm biến bên ngoài.

2) Chuyển giao và trao đổi một phần dữ liệu mô hình AI-ML – Trường hợp sử dụng này mô tả các ứng dụng khác nhau trong đó các loại hệ thống ML khác nhau được nối mạng để trao đổi các phần của dữ liệu mô hình AI-ML nhằm cải thiện độ chính xác dự đoán. Trong một số hệ thống, các mô hình ML cục bộ có thể liên tục cải thiện mô hình cơ sở của chúng bằng cách sử dụng dữ liệu thu thập được từ môi trường hiện có và đầu vào cảm biến khác. Các hệ thống này tải dữ liệu mô hình được cải thiện với tốc độ tương đối chậm lên mạng dựa trên đám mây lớn hơn, nơi xử lý thêm diễn ra để tinh chỉnh hơn nữa mô hình ML đầy đủ. Phương pháp cụ thể được sử dụng để cải tiến liên tục nằm ngoài phạm vi của trường hợp sử dụng này.

Các kịch bản ví dụ khác nhau được mô tả dưới đây.

Hệ thống AI/ML ứng phó khẩn cấp đối với phương tiện bị khuyết tật

Hình 9.25 minh họa một chiếc xe bị hỏng bị che khuất bởi một khúc cua mù trên đường. Phân tích này được coi là mức độ nghiêm trọng cao cho mục đích minh họa trường hợp sử dụng này.

Hình 10.25. Sự cố xe bị che khuất bởi một khúc cua mù nguy hiểm

Các cảm biến trên xe phát hiện những điều sau: hỏng hóc cơ học, giảm tốc độ, tọa độ GPS và nhiều tham số kỹ thuật khác nhau được sử dụng bởi các hệ thống suy luận ML trên xe để chẩn đoán ngay sự cố và thông báo cho các mạng xung quanh về mức độ nghiêm trọng của sự cố.

Các mô hình AI phi gian thực (Non-Real Time) khác nhau có thể sử dụng dữ liệu đào tạo lỗi thu thập được để tinh chỉnh các bản cập nhật mô hình dự đoán và phòng ngừa.

Camera giám sát giao thông và hệ thống an toàn giao thông MEC thành phố.

Camera giám sát sử dụng mô hình ML để phân tích dữ liệu cảm biến hình ảnh nhằm phát hiện và phân loại kiểu tai nạn bằng cách sử dụng suy luận ML với cải tiến mô hình AI-ML liên tục. Thông tin tai nạn đã xử lý này được tải lên cùng với các bản cập nhật mô hình lên MEC gần biên địa phương để dự đoán mô hình hệ thống AI /ML và xử lý thuật toán để đề xuất quá trình hành động tiếp theo ngay lập tức. Các hệ thống AI/ML này đề xuất một chương trình phát quảng bá cảnh báo tự động. Các hệ thống AI/ML tự động dựa trên các mô hình ML được đào tạo để dự đoán vị trí của phương tiện đặc biệt nguy hiểm do đường cong mù ẩn và thông báo cho Đơn vị Ứng phó Khẩn cấp địa phương để gửi trợ giúp thiết lập đường tránh giao thông sớm hơn trên đường. Gửi trở lại dữ liệu mô hình này trở lại camera giám sát để cải thiện thêm độ chính xác suy luận.

Việc tải lên dữ liệu, có thể được xử lý trước bởi mô hình AI/ML cục bộ, từ camera giám sát đến MEC có thể xảy ra với độ trễ thấp tùy thuộc vào mức độ nghiêm trọng của tai nạn trong khi dữ liệu được trả lại để cập nhật mô hình cục bộ có thể xảy ra trong khung thời gian chậm hơn nhiều.

Phản ứng đến gần của xe thông minh

Hình 10.26 minh họa một xe thông minh nhanh chóng đến gần một xe car được minh họa trong Hình 9.25. Các cảm biến của xe này không thể phát hiện mối nguy hiểm và phải dựa vào các hệ thống AI / ML khác để cảnh báo.

Hình 10.26. Xe tự hành đang tiến gần

Xe tự hành tiến gần nhận được nhiều thông báo cảnh báo và hệ thống AI/ML trong phương tiện này sử dụng các mô hình được đào tạo dự đoán và đề xuất các thao tác tự động thích hợp để di chuyển phương tiện đến làn đường xa để tránh bất kỳ nguy hiểm nào. Chia sẻ dữ liệu với các hệ thống khác cho phép liên tục tinh chỉnh các mô hình của họ; Việc chia sẻ dữ liệu này không có yêu cầu phi thời gian thực.

10.9.5.2. Quá trình diễn ra

Điều kiện ban đầu

Các kịch bản được mô tả ở trên giả định các điều kiện tiên quyết sau:

- Tất cả các hệ thống AI/ML hiện có đều có các mô hình AI/ML mặc định được đào tạo sẵn sẽ thực hiện dự đoán ở độ chính xác cơ bản và có khả năng cập nhật/tinh chỉnh lại các mô hình AI/ML để liên tục cải thiện độ chính xác cho mục đích cụ thể.

- Mộ số điều kiện bổ sung được cho tyrong bảng 10.18.

Bảng 10.18. Điều kiện tiên quyết trong trường hợp sử dụng

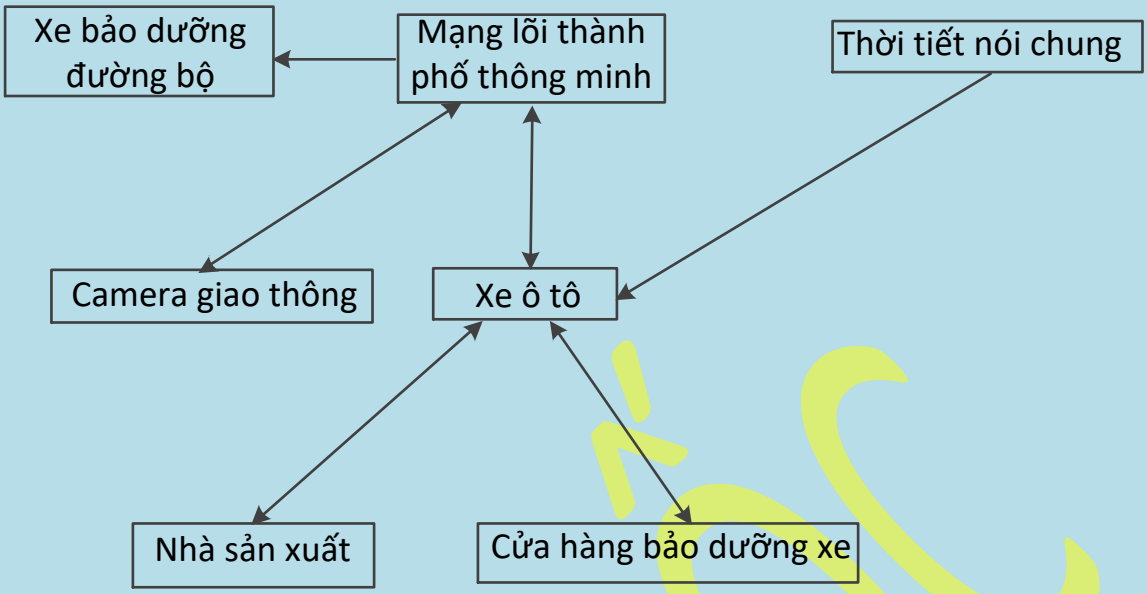
Xe tự hành bị sự cố	**Xe tự hành đang tiên gần**	**Camera giám sát đường bộ**	**Điều khiển giao thông thành phố thông minh**	**Mạng lưới dịch vụ khẩn cấp thành phố thông minh**
Mô hình xe AI/ML đã phát hiện ra lỗi. Tất cả các hệ thống truyền thông đều có sẵn và đang hoạt động.	Các mô hình Xe AI / ML đều hoạt động bình thường. Tất cả các hệ thống truyền thông đã sẵn sàng và hoạt động.	Camera được cấp nguồn, hoạt động và kết nối mạng.	Phần mềm và các hệ thống truyền thông đang chạy bình thường.	Phần mềm, Truyền thông và tài sản khẩn cấp đang chạy bình thường.

Các luồng dịch vụ

Các hệ thống AI/ML thích hợp sẽ cung cấp bảo mật và bảo vệ dữ liệu theo quy định của pháp luật và các chính sách thích hợp khác.

Các luồng dịch vụ được mô tả trong Hình 10.27 là tất cả các tương tác hệ thống thông minh AI/ML giữa máy với máy được thiết kế để chia sẻ thông tin với một loạt các nhu cầu từ độ trễ thấp nhất đối với các tình huống quan trọng (ví dụ: thông báo cảnh báo) và độ trễ nới lỏng cho các vấn đề ít nghiêm trọng hơn. Thông tin được chia sẻ (có thể bao gồm một bộ dữ liệu mở rộng) cũng có thể được trao đổi với độ trễ nới lỏng để hỗ trợ đào tạo và cập nhật mô hình AI/ML. Mỗi cấp độ dịch vụ có thể được sở hữu và điều hành bởi một tổ chức khác nhau.

Trường hợp sử dụng này giả định rằng không cần chia sẻ hoặc cập nhật quá [10%] tổng kích thước mô hình ML trong một lần truyền độ trễ thấp.



Hình 10.27. Ví dụ về tương tác AI-ML

Phương tiện đến các hệ thống AI khác

Ví dụ về trao đổi dữ liệu mô hình AI/ML là:

• Giám sát tình trạng xe và kiểm tra các mô hình ML - Các mô hình ML cục bộ cảm nhận và phản ứng với thông tin cảm biến va chạm, hỏa hoạn, nhiệt độ và điện. Các mô hình này liên tục cải tiến và tối ưu hóa nhưng không nhất thiết phải chia sẻ dữ liệu nếu hệ thống là danh nghĩa. Tuy nhiên, một khi tình huống khẩn cấp đã xảy ra, thông tin khẩn cấp có thể được gửi đến các dịch vụ ứng phó khẩn cấp thích hợp trong mạng lõi thành phố thông minh, camera cục bộ và cảm biến đường bộ. Thông tin mở rộng cũng có thể được chia sẻ với nhu cầu độ trễ thoải mái để hỗ trợ cập nhật/đào tạo mô hình.

• Cảm biến phương tiện cho các thay đổi của phương tiện dưới cảm biến và điều khiển mô hình ML. Các mô hình ML cục bộ cảm nhận và phản ứng với trượt bánh xe, trượt, tăng tốc và gãy. Các mô hình này liên tục cải tiến và tối ưu hóa nhưng không nhất thiết phải chia sẻ dữ liệu nếu hệ thống là danh nghĩa. Tuy nhiên, một khi tình huống khẩn cấp đã xảy ra, thông tin khẩn cấp ngay lập tức được phát đến các dịch vụ ứng phó khẩn cấp thích hợp và hệ thống khác áp dụng các mô hình ML liên quan để thông báo cho chúng. Thông tin mở rộng cũng có thể được chia sẻ với nhu cầu độ trễ nới lỏng để hỗ trợ cải thiện độ chính xác dự đoán của hệ thống.

Camera giám sát giao thông nối đến điều khiển giao thông thành phố

Camera cảm nhận tai nạn bằng cách sử dụng các mô hình AI/ML cục bộ và gửi thông báo đến kiểm soát giao thông. Kiểm soát giao thông có thể sử dụng nhiều camera để đưa tin về cùng một sự cố và do đó, cải thiện độ chính xác suy luận của nó. Thông tin mở rộng được trao đổi với nhu cầu độ trễ thấp có thể được sử dụng để đào tạo lại và phân phối lại các lớp mô hình AI tham chiếu cho phù hợp.

Cá ví dụ về trao đổi dữ liệu Mô hình AI/ML là:

• Camera và cảm biến đường xá phát hiện các tình trạng khẩn cấp- các mô hình ML của camera cục bộ và các bộ cảm biến đường xá liên tục cải thiện dự đoán của chúng về các tình huống khẩn cấp. Khi tình huống khẩn cấp được phát hiện, thông tin khẩn cấp từ các hệ thống này có thể được chia sẻ với hệ thống tương ứng để cải thiện phản ứng của các nhân viện ứng cứu khẩn cấp. Thông tin mở rộng có thể được chia xẻ với yêu cầu trễ lới lỏng để hỗ trợ cải thiện độ chính xác suy luận của thống.

Điều khiển giao thông thành phố nối đến mạng các dịch vụ khẩn cấp thành phố

Các mô hình AI/ML sử dụng tự động hóa và dự đoán để gửi các xe EMS (Emerrgency Medical Service: dịch vụ cấp cứu y tế) thích hợp đến hiện trường vụ tai nạn.

Ví dụ về trao đổi dữ liệu Mô hình AI/ML là:

• Camera, các bộ cảm biến đường xá và thời tiết sử dụng các mô hình ML. Các cảm biến mô hình ML liên tục cải thiện và tối ưu hóa kết quả dự đoán. Khi tình huống khẩn cấp được phát hiện, thông tin khẩn cấp từ các hệ thống này có thể được chia sẻ với các xe cấp cứu để, ví dụ: cải thiện thời gian định tuyến của họ qua giao thông dựa trên tình huống.

Xe bị chết máy nối đến nhà sản xuất

Xe bị chết máy gửi dữ liệu (có thể đã được xử lý trước, ví dụ: dữ liệu mô hình AI/ML) cho nhà sản xuất ô tô để giúp chẩn đoán sự cố. Nhà sản xuất sử dụng thông tin để cải thiện chất lượng, độ tin cậy và hiệu suất của sản phẩm. Các kỹ thuật học liên kết và/hoặc học tập phân tán có thể được sử dụng để cải thiện mô hình AI/ML trên xe.

Xe bị chết máy nối đến điều khiển giao thông thành phố (MTC: Metro Traffic Control)

Xe chết máy gửi thông tin khẩn cấp đến MTC để giúp cảnh báo những người khác. Ví dụ: vị trí đường cong mù cụ thể cùng với dữ liệu khu vực khác được sử dụng để cải thiện tất cả các phản ứng của MTC. Thông tin mở rộng cũng có thể được chia sẻ với nhu cầu độ trễ nới lỏng để cải thiện các mô hình phản hồi MTC.

Xe bị chết máy nối đến dịch vụ sửa chữa

Xe bị chết máy gửi dữ liệu (có thể đã được xử lý trước, ví dụ: dữ liệu mô hình AI/ML) đến dịch vụ sửa chữa để đảm bảo các xe phản ứng thích hợp cung cấp các công cụ và thiết bị phù hợp. Phản hồi dịch vụ sửa chữa sử dụng các kỹ thuật học liên kết và / hoặc học tập phân tán để cải thiện mô hình liên tục.

Xe đang đến gần truyền thông với điều khiển giao thông thành phố (MTC)

MTC cảnh báo phương tiện đang đến gần về mối nguy hiểm sắp xảy ra và tùy thuộc vào mức độ SAE (Society Automotive Engineers: hiệp hội kỹ sử ô tô) của tự động hóa lái xe, xe sẽ phản ứng thích hợp. Thông tin mở rộng cũng có thể được chia sẻ với nhu cầu độ trễ nới lỏng để hỗ trợ cải thiện độ chính xác dự đoán của hệ thống.

Điều kiện kết thúc

Tất cả các hệ thống được xác định trong kịch bản này sử dụng các thuật toán học liên kết hoặc học phân tán AI/ML độc lập, được nối mạng để tổng hợp thông tin từ nhiều nguồn nhằm cải thiện lớp của mô hình AI/ML hệ thống. Các hệ thống đào tạo AI/ML độc lập đảm bảo rằng các mô hình cải tiến được phân phối trở lại từng hệ thống cuối để cải thiện độ an toàn tổng thể và độ mạnh mẽ của phản ứng tiếp theo.

9.9.4.3. Các yêu cầu tiềm năng mới cần thiết để hỗ trợ trường hợp sử dụng

Bảng 10.19 cung cấp các mô hình ML ví dụ (kích thước và tốc độ dữ liệu DL) được xem xét cho các trường hợp sử dụng ở trên và bảng 10.20 phác thảo các yêu cầu về độ trễ. Giả định rằng tối đa 10% kích thước mô hình (đầy đủ) được trao đổi giữa các hệ thống tham gia.

Bảng 10.19. Các mạng / các mô hình ML và tốc độ dữ liệu DL tiềm năng

Mô hình DNN	32 bits per parameter		
	Mô hình đầy đủ (MBytes)	Dữ liệu được trao đổi (Mbytes) [10% mô hình đầy đủ]	Tốc độ dữ liệu người dùng cực đại (Mbit/s)
1.0 MobileNet-224	16,8	1,68	13,4
SSD-ResNet34	81	8,1	64,8
SSD-MobileNet-v1	27,3	2,37	21,8
MASK R-CNN	245	24,5	~100
DLRM	400	40,0	10

Bảng 10.20. Yêu cầu về độ trễ tiềm năng

Ứng dụng người dùng	Yêu cầu về độ trễ tiềm năng
	Dữ liệu trao đổi - Độ trễ tải xuống
Xe phát hiện lỗi và tấp vào lề để tránh	~500ms – 1sec

Ứng dụng người dùng	Yêu cầu về độ trễ tiềm năng
tai nạn	
Camera bên đường đi phát hiện nguy hiểm đường và cảnh báo mạng thành phố thông minh	~500ms – 1sec
Giao thông sắp tới phát hiện nguy hiểm và tránh tai nạn	~500ms – 1sec
Nhận dạng video	~500ms – 1sec
Thành phố thông minh phát hiện tai nạn và đưa ra cảnh báo cục bộ	~ lên đến vài giây
Nhà sản xuất ô tô/Bảo hiểm được cảnh báo về lỗi	Lên đến vài phút

Yêu cầu:

Hệ thống 5G/6G có thể hỗ trợ tải dữ liệu với dung lượng tối đa ~2-40 MB để cập nhật mô hình AI/ML cục bộ với độ trễ lên đến 500ms – 1s.

Hệ thống 5G/6G có thể hỗ trợ tải dữ liệu để cập nhật mô hình AI/ML cục bộ với tính khả dụng dịch vụ truyền thông lên đến 99.999%.

10.9.6. Giám sát mô hình AI/ML được chia sẻ (Shared AI/ML model monitoring)

10.9.6.1. Mô tả

Các mô hình AI/ML được đào tạo trên một tập dữ liệu đào tạo để hoàn thành một nhiệm vụ đã thiết lập. Các nhiệm vụ có thể thay đổi từ nhận dạng hình ảnh hoặc giọng nói đến dự báo để tối ưu hóa, ví dụ: hiệu suất chuyển giao (xem 3GPP TR 28.809) hoặc điều chỉnh các tham số được hỗ trợ của Mạng lõi (xem phần 5.4.6.2, TS 23.501).

Trong mỗi nhiệm vụ này, nhà cung cấp mô hình AI/ML được chia sẻ có thể được hưởng lợi từ việc chia sẻ mô hình AI/ML được đào tạo với (các) người tiêu dùng của mô hình AI/ML được chia sẻ hoặc có thể được hưởng lợi từ việc đào tạo mô hình AI/ML phân tán/liên kết hoặc đào tạo mô hình AI/ML phân chia qua hệ thống 5G/6G. Hơn nữa, giám sát mô hình AI/ML là một yêu cầu để cho phép học trực tuyến trong mạng (ví dụ: thông qua học tăng cường), một tập hợp các kỹ thuật phù hợp hơn để phản ứng kịp thời với sự suy giảm dịch vụ.

Các thuật ngữ sau được sử dụng trong trường hợp sử dụng này:

Mô hình AI/ML được chia sẻ (Shared AI/ML model): Mô hình AI/ML được chia sẻ giữa các ứng dụng khác nhau, ví dụ: mô hình AI/ML được đào tạo trước và cung cấp cho các người tiêu dùng khác nhau hoặc mô hình AI/ML được đào tạo bằng cách sử dụng phương pháp học tập phân tán/liên kết hoặc bằng cách chia giai đoạn đào tạo mô hình thành các phần khác nhau được thực hiện ở các vị trí mạng khác nhau.

Nhà cung cấp mô hình AI/ML được chia sẻ (Shared AI/ML model provider): máy chủ ứng dụng cung cấp hoặc quản lý "mô hình AI/ML được chia sẻ".

Người tiêu dùng mô hình AI/ML được chia sẻ (Shared AI/ML model consumer): ứng dụng, ví dụ: chạy trên UE, đang sử dụng/tiêu thụ "mô hình AI/ML được chia sẻ".

Do những thay đổi trong kịch bản (tức là trong bối cảnh thu thập dữ liệu đào tạo), mô hình AI/ML có thể mang lại hiệu suất kém so với hiệu suất của mô hình AI/ML được đo trong giai đoạn thử nghiệm mô hình. Điều này có thể xảy ra khi theo thời gian, sự phân phối của dữ liệu đầu vào để suy luận khác với sự phân phối của dữ liệu đào tạo hoặc nếu mô hình AI/ML được sử dụng trong một bối cảnh khác. Trong trường hợp này, nhà cung cấp mô hình AI/ML được chia sẻ sẽ có thể kịp thời phát hiện sự suy giảm hiệu suất và phản ứng để tránh sự suy giảm hoặc gián đoạn dịch vụ. Thông thường, việc cập nhật mô hình AI/ML được chia sẻ không chỉ phụ thuộc vào kết quả suy luận của một người tiêu dùng mô hình AI/ML được chia sẻ vì sự suy giảm hiệu suất có thể là do một số nguồn lỗi khác, ví dụ: lỗi đo lường đầu vào của mô hình. Để phát hiện mô hình AI/ML được chia sẻ lỗi thời, nhà cung cấp mô hình AI/ML được chia sẻ có thể sử dụng kết quả suy luận từ nhiều người tiêu dùng mô hình AI/ML được chia sẻ và một phân tích thời gian, không gian được thực hiện trước khi khởi động cập nhật mô hình AI/ML mới.

Do đó, nhà cung cấp mô hình AI/ML được chia sẻ, sau khi chia sẻ mô hình AI/ML qua hệ thống 5/6G với người tiêu dùng mô hình AI/ML được chia sẻ, cần theo dõi hiệu suất của mô hình để phát hiện sự suy giảm hiệu suất có thể xảy ra của mô hình AI/ML được chia sẻ (ví dụ: dựa trên phản hồi suy luận từ người tiêu dùng mô hình AI/ML chẳng hạn như mức độ tin cậy thấp hơn).

BẢN QUYỀN

Ngoài ra, nhà cung cấp mô hình AI/ML được chia sẻ có thể phân chia đào tạo mô hình AI/ML với người tiêu dùng mô hình AI/ML được chia sẻ để liên tục cải thiện hiệu suất của mô hình AI/ML được chia sẻ dựa trên phản hồi đào tạo và/hoặc suy luận cục bộ từ người tiêu dùng mô hình AI/ML được chia sẻ. Phần cục bộ của mô hình AI/ML được chia sẻ sẽ được đào tạo/tinh chỉnh theo hướng dẫn của nhà cung cấp mô hình AI/ML được chia sẻ. Đầu vào cho khóa đào tạo này sẽ là dữ liệu có sẵn tại người tiêu dùng mô hình AI/ML được chia sẻ. Đầu ra của đào tạo mô hình cục bộ hoặc suy luận tại người tiêu dùng mô hình AI/ML được chia sẻ có thể được cung cấp cho nhà cung cấp mô hình AI/ML được chia sẻ và được nhà cung cấp mô hình AI/ML dùng chung để cung cấp thêm thông tin cho hệ thống 5G/6G nhằm cải thiện hoạt động của nó.

10.9.6.2. Quá trình diễn ra

Điều kiện ban đầu

Nhà cung cấp mô hình AI/ML được chia sẻ lưu trữ nhiều mô hình AI/ML cùng với hiệu suất đã được đo của chúng trong giai đoạn thử nghiệm.

Nhà cung cấp mô hình AI/ML được chia sẻ có khả năng chia sẻ mô hình AI/ML với người tiêu dùng mô hình AI/ML được chia sẻ để tận dụng trên hệ thống 5G/6G.

Người tiêu dùng mô hình AI/ML được chia sẻ có thể chạy các ứng dụng yêu cầu sử dụng mô hình AI/ML và tải xuống từ nhà cung cấp mô hình AI/ML thông qua hệ thống 5G/6G.

10.8.6.3. Các luồng dịch vụ

1. Nhà cung cấp mô hình AI/ML được chia sẻ muốn tối ưu hóa hiệu suất của một số quy trình bằng hiệu suất của mô hình AI/ML được chia sẻ.

2. Nhà cung cấp mô hình AI/ML được chia sẻ gửi mô hình AI/ML được chia sẻ đã được đào tạo đến người tiêu dùng mô hình AI/ML được chia sẻ tại UE để tận dụng trên hệ thống 5G/6G.

3. UE nhận mô hình và sử dụng mô hình để thực hiện đào tạo và suy luận cục bộ bằng cách sử dụng dữ liệu có sẵn trên UE.

4. Nhà cung cấp mô hình AI/ML được chia sẻ giám sát kịch bản ngữ cảnh (ví dụ: dữ liệu UE có sẵn cho ứng dụng) trong đó UE đang chạy mô hình AI/ML được chia sẻ và hiệu suất của mô hình.

5. Nếu phát hiện thấy sự thay đổi trong kịch bản ngữ cảnh hoặc hiệu năng mô hình, ví dụ: dựa trên phản hồi suy luận từ người tiêu dùng mô hình AI/ML được chia sẻ, nhà cung cấp mô hình AI/ML được chia sẻ, để tránh suy giảm hiệu suất của mô hình, chia sẻ với người tiêu dùng mô hình AI/ML phiên bản cập nhật của mô hình AI/ML được đào tạo lại để nắm bắt được ngữ cảnh mới với hiệu suất tốt hơn dự kiến.

6. UE tiếp tục chạy mô hình cập nhật mà không bị giảm hiệu suất

Điều kiện kết thúc

Theo ví dụ trong trường hợp sử dụng, nhà cung cấp mô hình AI/ML được chia sẻ nhận được dự báo chính xác về hiệu suất mô hình AI/ML và người tiêu dùng mô hình AI/ML đang sử dụng mô hình AI/ML có hiệu suất cao.

Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Hệ thống 5G/6G sẽ có thể chuyển giao mô hình AI/ML cập nhật từ nhà cung cấp mô hình AI/ML được chia sẻ sang người tiêu dùng mô hình AI/ML được chia sẻ trong khoảng thời gian [1 giây-1 phút] đối với các model AI/ML có kích thước tối đa là [100-500] MB.

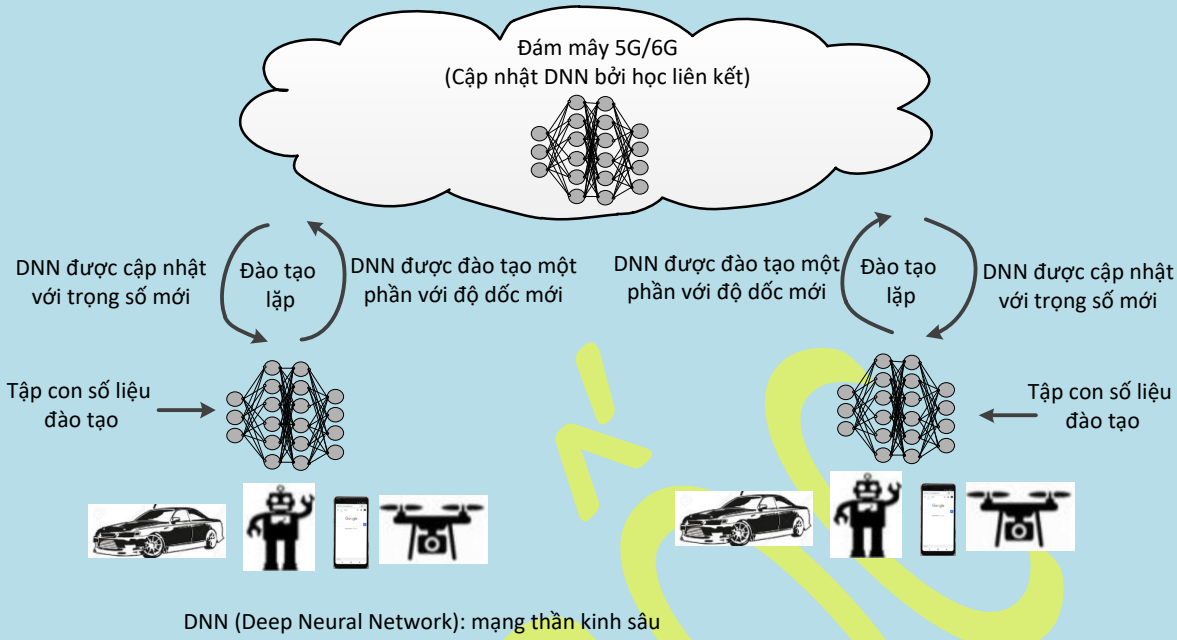
10.10. HỌC PHÂN TÁN/ LIÊN KẾT TRÊN HỆ THỐNG 5G (DISTRIBUTED/FEDERATED LEARNING OVER 5G SYSTEM

10.10.1. Học liên kết không nén để nhận dạng hình ảnh (Uncompressed Federated Learning for image recognition)

10.10.1.1. Mô tả

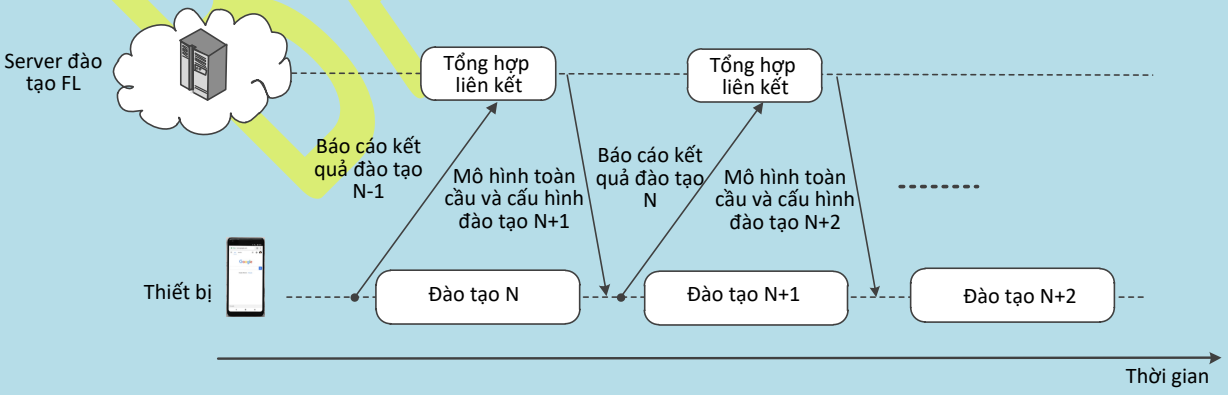
Ngày nay, máy ảnh trên điện thoại thông minh đã trở thành công cụ phổ biến nhất để chụp ảnh và quay video, chứa nhiều dữ liệu thị giác có giá trị để đào tạo mô hình nhận dạng hình ảnh. Đối với nhiều nhiệm vụ nhận dạng hình ảnh, hình ảnh/video được thu thập bởi thiết bị di động là điều cần thiết để đào tạo một mô hình toàn cục. Federated Learning (FL) là một cách tiếp cận ngày càng được sử dụng rộng rãi để đào tạo thị giác máy tính và các mô hình nhận dạng hình ảnh.

Như đã trình bầy trong 10.5, trong chế độ Federated Learning, máy chủ đám mây đào tạo một mô hình toàn cục bằng cách tổng hợp các mô hình cục bộ được đào tạo một phần bởi mỗi thiết bị đầu cuối dựa trên mô hình trung bình lặp đi lặp lại. Như được mô tả trong Hình 9.28, trong mỗi lần lặp lại đào tạo, một thiết bị thực hiện đào tạo dựa trên mô hình được tải xuống từ máy chủ AI bằng cách sử dụng dữ liệu đào tạo cục bộ. Sau đó, thiết bị báo cáo kết quả đào tạo tạm thời (ví dụ: độ dốc cho DNN) cho máy chủ đám mây thông qua các kênh 5G/6G UL. Máy chủ tổng hợp các gradient từ các thiết bị và cập nhật mô hình toàn cục. Tiếp theo, mô hình toàn cục được cập nhật được phân phối đến các thiết bị thông qua các kênh DL 5G/6G, các thiết bị có thể thực hiện đào tạo cho lần lặp tiếp theo.



Hình 10.28. Học liên kết trên hệ thống 5G

Một quy trình Federated Learning lặp đi lặp lại được minh họa trong Hình 10.29. Trong lần lặp lại đào tạo thứ N, thiết bị thực hiện đào tạo dựa trên mô hình được tải xuống từ máy chủ đào tạo FL bằng cách sử dụng hình ảnh/video được thu thập cục bộ. Sau đó, thiết bị báo cáo kết quả đào tạo tạm thời lặp lại thứ N (ví dụ: độ dốc (gradient) cho DNN) cho máy chủ thông qua các kênh UL 5G/6G. Trong khi đó, mô hình toàn cục và cấu hình đào tạo cho lần lặp thứ (N + 1) được gửi đến thiết bị. Khi máy chủ tổng hợp các gradient từ các thiết bị cho lần lặp thứ N, thiết bị sẽ thực hiện đào tạo cho lần lặp thứ (N + 1). Đầu ra tổng hợp liên kết được sử dụng để cập nhật mô hình toàn cục, mô hình này sẽ được phân phối cho các thiết bị, cùng với cấu hình đào tạo được cập nhật.



Hình `10.29. Dòng thời gian học liên kết để nhận dạng hình ảnh

Để sử dụng đầy đủ tài nguyên đào tạo tại thiết bị và giảm thiểu độ trễ đào tạo, đường đào tạo được hiển thị trong hình 10.29 yêu cầu báo cáo kết quả đào tạo cho lần lặp thứ (N-1) và phân phối mô hình/cấu hình đào tạo toàn cục cho lần lặp thứ (N + 1) được hoàn thành trong quá trình đào tạo của thiết bị cho lần lặp thứ N. ~~Phân tích trong phần 10.1.2 sẽ được phát triển dựa trên thời gian xử lý~~. Trong thực tế, dòng thời gian FL thư giãn hơn cũng có thể được xem xét với việc hy sinh tốc độ hội tụ đào tạo.

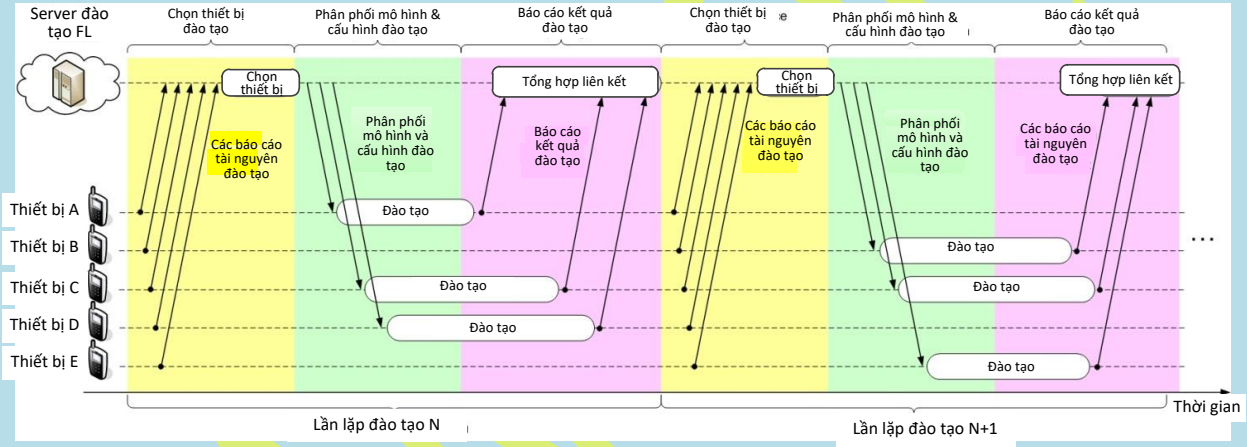
Thời gian đào tạo nên được giảm thiểu vì các thiết bị di động chỉ có thể ở trong môi trường trong một khoảng thời gian ngắn. Hơn nữa, xem xét dung lượng lưu trữ hạn chế tại thiết bị, có thể không thực tế khi yêu cầu thiết bị đào tạo lưu trữ một lượng lớn dữ liệu đào tạo trong bộ nhớ để đào tạo sau khi nó di chuyển ra ngoài môi trường.

Khác với đào tạo phi tập trung được vận hành trong các trung tâm dữ liệu đám mây, Federated Learning qua các hệ thống truyền thông không dây cần được sửa đổi để thích ứng với các điều kiện kênh không dây thay đổi, tài nguyên đào tạo không ổn định trên thiết bị di động và tính không đồng nhất của thiết bị. Giao thức học liên kết cho truyền thông không dây có thể được mô tả trong hình 10.30.

Đối với mỗi lần lặp, trước tiên các thiết bị đào tạo có thể được chọn. Các thiết bị đào tạo của ứng viên báo cáo tài nguyên tính toán của họ có sẵn cho nhiệm vụ đào tạo cho máy chủ FL. Máy chủ FL thực hiện lựa chọn thiết bị đào tạo dựa trên các báo cáo từ thiết bị và các điều kiện khác, ví dụ: điều kiện kênh không dây của thiết bị, vị trí địa lý, v.v.

Vì vậy, bên cạnh việc thực hiện nhiệm vụ học liên kết, các thiết bị đào tạo trong hệ thống truyền thông có dữ liệu khác để truyền ở đường lên (ví dụ: đối với các giao dịch dịch vụ đang diễn ra), có thể có mức độ ưu tiên cao và không chịu được độ trễ và việc truyền nó có thể ảnh hưởng đến khả năng tải lên mô hình được đào tạo cục bộ của thiết bị. Do đó, việc lựa chọn thiết bị phải tính đến sự đánh đổi để tải lên kết quả đào tạo so với việc tải lên dữ liệu đường lên khác. Hơn nữa, việc bỏ qua một thiết bị khỏi tổng hợp mô hình học tập liên kết cho một hoặc nhiều lần lặp lại ảnh hưởng đến sự hội tụ của mô hình học liên kết. Do đó, việc lựa chọn thiết bị đào tạo ứng viên qua các liên kết không dây phức tạp hơn so với học liên kết trong các trung tâm dữ liệu.

Sau khi các thiết bị đào tạo được chọn, máy chủ FL sẽ gửi các cấu hình đào tạo đến các thiết bị đào tạo đã chọn, cùng với mô hình toàn cục để đào tạo. Một thiết bị đào tạo bắt đầu đào tạo dựa trên mô hình toàn cục và cấu hình đào tạo nhận được. Khi kết thúc đào tạo cục bộ, một thiết bị báo cáo kết quả đào tạo tạm thời của nó (ví dụ: độ dốc cho DNN) cho máy chủ FL. Trong Hình 10.30, việc lựa chọn thiết bị đào tạo được thực hiện và các cấu hình đào tạo được gửi đến các thiết bị đào tạo khi bắt đầu mỗi lần lặp. Nếu các điều kiện (ví dụ: tài nguyên tính toán của thiết bị, điều kiện kênh không dây, các giao dịch dịch vụ khác của thiết bị đào tạo) không được thay đổi, việc lựa chọn lại thiết bị đào tạo và cấu hình lại đào tạo có thể không cần thiết cho mỗi lần lặp, tức là cùng một nhóm thiết bị đào tạo có thể tham gia đào tạo với cùng một cấu hình cho nhiều lần lặp. Tuy nhiên, việc lựa chọn các thiết bị đào tạo nên được xen kẽ theo thời gian để đạt được một lấy mẫu độc lập và phân phối giống hệt nhau từ tất cả các thiết bị, tức là tạo cơ hội công bằng cho tất cả các thiết bị đóng góp vào mô hình tổng hợp.



Hình 10.30. Giao thức Federated Learning điển hình trên hệ thống truyền thông không dây

10.10.1.2. Các yêu cầu tiềm năng mới cần thiết để hỗ trợ trường hợp sử dụng

Như đã xét ở trên, để giảm thiểu độ trễ đào tạo đối với học liên kết cho nhận dạng hình ảnh, tài nguyên tính toán cho nhiệm cụ đào tạo phải được sự dụng hết, nghĩa là cần duy trì đường ống đào tạo trên hình 10.29.

Nếu xét đào tạo mô hình CNN VGG16-BN 7 bit sử dụng ảnh 224×224×3 làm dữ liệu đào tạo, thì bảng 10.21 phải chỉ ra rằng tổng trễ độ trễ tải lên gradient, độ trễ tổng hợp liên kết và trễ tải xuống mô hình toàn cục không lớn hơn thời gian tính toán của GPU tại thiết bị cho một lần lặp. Đối với các kích thước lô (Batch) khác nhau, tải lên gradient và tải xuống mô hình toàn cục cho mỗi lần lặp cần phải kết thúc trong thời gian [52~162 ms], tương ứng.

Ngay cả khi nhiều thiết bị đào tạo có mặt trong một ô, tất cả các thiết bị đào tạo phải kết thúc tải lên trong thời gian trễ như trong bảng 10.21.

Kích thước của mô hình VGG_BN 8 bit là 32 MByte cho cả các gradient được đào tạo và mô hình toàn cục. Vì thế, để kết thúc tải lên gradient và tải xuống mô hình toàn cục, tốc độ dữ liệu đường lên và đường xuống được chỉ ra trong bảng 10.21là [6,5 Gbit/s đến 20,3 Gbit/s], tương ứng. Cũng cần lưu ý rằng 132 Mbyte là kích thước không nén. Kích thước này có thể giảm nếu các kỹ thuật nén mô hình/ dữ liệu được tiếp nhận.

Trong các yêu cầu cũ đối với hệ thống 5G, vùng phủ đầy đủ luôn được mong muốn đối với các UE. Tuy nhiên, nhiệm vụ đào tạo mô hình AI/ML có thể giảm nhẹ đến một mức độ nhất định các yêu cầu về vùng phủ mạng liên tục. Khi một server FL chọn các thiết bị đào tạo cho một nhiệm vụ học liên kết, nó có thể tìm cách chọn các UE trong một vùng phủ thỏa mãn, nếu chúng có thể thu thập dữ liệu đào tạo cần thiết. Điều này có nghĩa là, thậm chí trong một vùng phủ sóng không liên tục của sóng mm 5G, nhiệm vụ học liên kết vẫn có thể được thực hiện tốt. Điều này cung cấp cho các nhà mạng 5G một dịch vụ tốt hơn khi khảo sát sử dụng tài nguyên phổ tần FR2 (dải tần số 2 của 5G).

Các yêu cầu KPI cần thiết để hỗ trợ trường hợp sử dụng:

- Hệ thống 5G sẽ phải hỗ trợ tốc độ dữ liệu trải nghiệm DL và UL được cho trong bảng 10.21 để cho phép đào tạo liên kết không nén.

Bảng 10.21. Độ trễ và tốc độ dữ liệu trải nghiệm DL/UL cho học liên kết để nhận dạng ảnh

Kích thước lô mini (các ảnh)	Thời gian tính toán của GPU (ms)	Độ trễ cực đại cho tải lên gradient được đào tạo và phân bố mô hình toàn cục (xem chú thích 1)	Tốc độ dữ liệu DL/UL trải nghiệm người dùng cho tải lên gradient được đào tạo và phân bố mô hình toàn cục (xem chú thích 2)
64	325	3,24s	325Mbit/s
32	191	1,9s	55Mbit/s
16	131	1,3s	810Mbit/s
8	111	1,1s	960Mbit/s
4	105	1,04s	1,0Gbit/s

Chú thích 1: Độ trễ trong bảng được cho là 10 lần thời gian tính toán của GPU của thết bị cho kích thước lô mini được cho trước

Chú thích 2: Giá trị được cung cấp trong bảng là các yêu cầu có tính toán đối với mô hình VGG 16 BN 8bit với kích thước 132 Mbyte, khi cho trước kích thước lô min và khoảng thời gian của một lần lặp tính theo giây. Có thể giảm tốc độ dữ liệu DL/UL trải nghiệm người dùng, ví dụ: thiết lập tời gian một lần lặp lớn hơn, áp dụng FL nén hoặc sử dụng một mô nhình AI/ML khác.

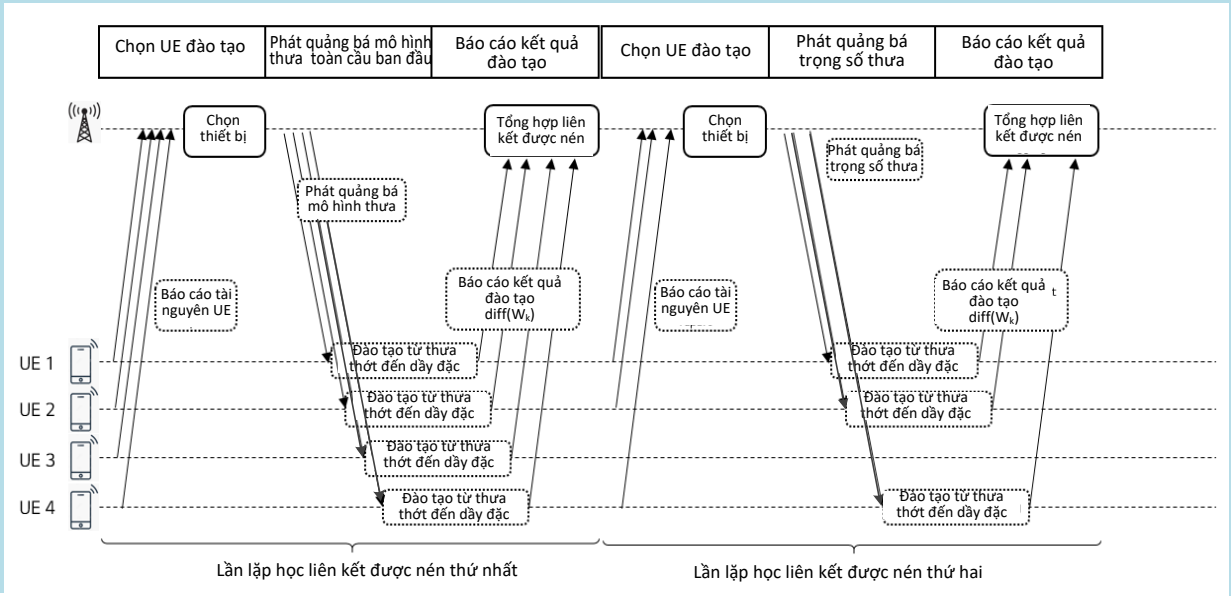
10.10.2. Học liên kết nén để xử lý hình ảnh/video

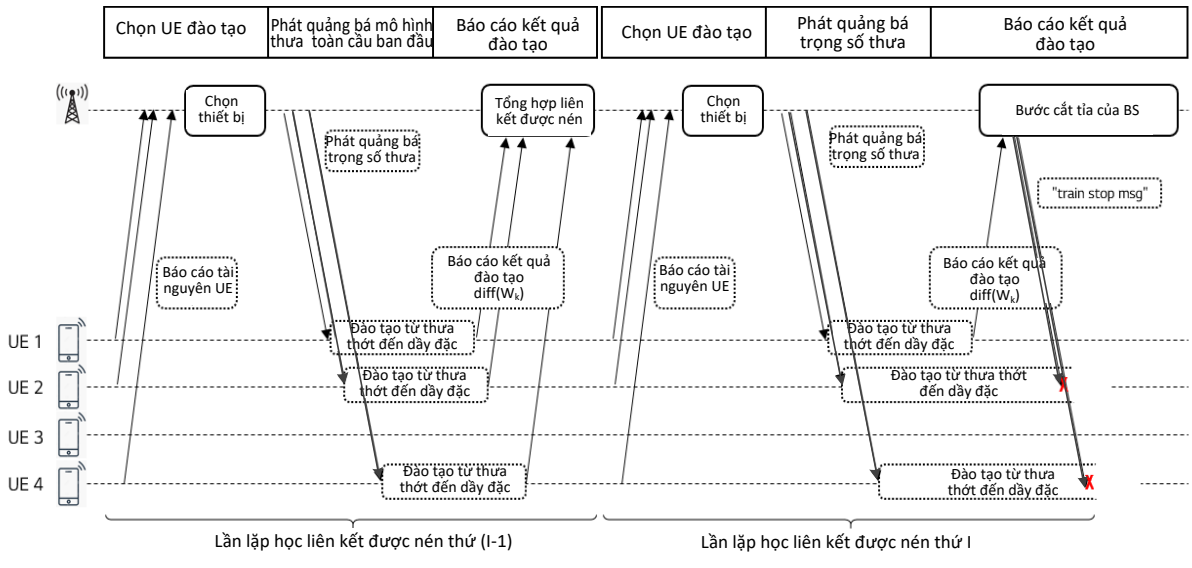
10.10.2.1. Mô tả

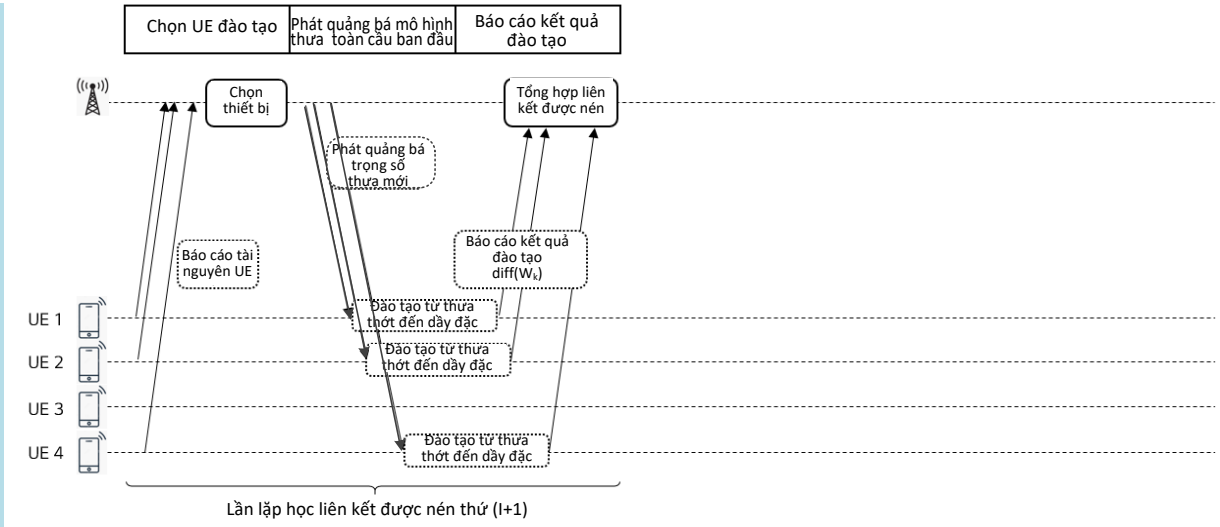
Học liên kết có thể được sử dụng để đào tạo các mô hình AI/ML dựa trên số lượng hình ảnh và video được tạo bởi camera trong thiết bị di động bằng cách trao đổi lặp đi lặp lại gradient của các mô hình cập nhật thay vì hình ảnh và video trực tiếp của người dùng. Bởi vì phương pháp này có thể sử dụng hình ảnh và video từ nhiều người dùng, hiệu năng của mô hình AI/ML được đào tạo có thể cao hơn đáng kể so với một trường hợp độc lập. Tuy nhiên, các phương pháp học liên kết cơ bản có thể có nhược điểm bởi lưu lượng truy cập đường lên lớn và chi phí tính toán cao trên thiết bị di động. Do đó, sẽ có lợi khi xem xét phương pháp học tập liên kết nén (CFL: Compressed Federated Learning), cho phép chuyển các mô hình nén (không đầy đủ) trong thời gian học tập.

Hình 10.31 cho thấy quy trình thiết yếu của CFL. CFL lặp đi lặp lại thực hiện một tập hợp ba giai đoạn hoạt động. Để mô tả các lần lặp lại trong CFL, ta xét một chu kỳ I: mỗi chu kỳ bắt đầu với lần lặp thứ 1 và kết thúc bằng lần lặp thứ I, ngay sau đó là lần lặp thứ 1 của chu kỳ tiếp theo (ví dụ: (I + 1) lần lặp thứ nhất). Đối với mỗi lần lặp, ba giai đoạn hoạt động bao gồm lựa chọn UE đào tạo, phân phối trọng số thưa thớt (trong trường hợp này ma trân trọng số sẽ đặt một số trọng số vào không để giảm kích thước mô hình) và các giai đoạn báo cáo kết quả đào tạo. Các hoạt động của ba giai đoạn này trong lần lặp đầu tiên và hoạt động trong lần lặp cuối cùng khác với các hoạt động trong lần lặp lại khác.

Mỗi lần lặp lại trong CFL bắt đầu với giai đoạn lựa chọn UE đào tạo, tại đó máy chủ CFL chọn một tập hợp người dùng có sẵn từ những người dùng ứng cử viên để liên kết với mô hình AI/ML cùng mục đích. Đối với những người dùng được chọn, sẵn sàng tham gia vào quá trình học tập vì đang ở trạng thái sẵn sàng, máy chủ CFL truyền thông tin cấu hình đào tạo. Ở giai đoạn tiếp theo, máy chủ CFL gửi mô hình toàn cục thưa thớt, có thể là phiên bản ban đầu của mô hình AI/ML trong lần lặp đầu tiên. Nếu không, mô hình toàn cục thưa thớt là một phiên bản tổng hợp dựa trên thông tin báo cáo của người dùng. Sau đó, mỗi UE đào tạo một mô hình nhận được sau khi mở rộng mô hình không gian và báo cáo kết quả đào tạo trung gian cho máy chủ CFL, trong đó kết quả đào tạo được nén để chỉ có các gradient trọng số giá trị đáng kể để áp dụng nén mô hình. Bằng cách đó, đường lên trong suốt yêu cầu có thể được giảm đáng kể so với phương pháp học liên kết cơ bản mà không cần nén. Trong lần lặp lại cuối cùng, lần lặp thứ I, Máy chủ CFL gửi 'thông báo dừng đào tạo' đến UE để UE có thể ngừng gửi bản cập nhật của nó lâu hơn và Máy chủ CFL thực hiện tinh chỉnh bằng cách cắt tỉa các nút không cần thiết. Trong suốt nhiều lần lặp lại này của một chu kỳ (tức là từ lần lặp lần 1 đến lần lặp lần thứ I) như trong hình, mô hình AI/ML sẽ được nâng cao dần dần dựa trên dữ liệu người dùng trong mạng di động với các yêu cầu giảm về thông lượng đường lên và đường xuống.


Chọn UE đào tạo
Phát quảng bá mô hình thưa toàn cầu ban đầu
Báo cáo kết quả đào tạo
Chọn UE đào tạo
Phát quảng bá trọng số thưa
Báo cáo kết quả đào tạo
Chọn thiết bị
Tổng hợp liên kết được nén
Phát quảng bá mô hình thưa
Báo cáo tài nguyên UE
Báo cáo kết quả đào tạo diff(Wk)
Đào tạo từ thưa thớt đến dầy đặc
UE 1
UE 2
UE 3
UE 4
Lần lặp học liên kết được nén thứ nhất
Lần lặp học liên kết được nén thứ hai


Chọn UE đào tạo
Phát quảng bá mô hình thưa toàn cầu ban đầu
Báo cáo kết quả đào tạo
Chọn UE đào tạo
Phát quảng bá trọng số thưa
Báo cáo kết quả đào tạo
Chọn thiết bị
Tổng hợp liên kết được nén
Phát quảng bá trọng số thưa
Bước cắt tỉa của BS
"train stop msg"
Báo cáo tài nguyên UE
Báo cáo kết quả đào tạo diff(Wk)
Đào tạo từ thưa thớt đến dầy đặc
UE 1
UE 2
UE 3
UE 4
Lần lặp học liên kết được nén thứ (l-1)
Lần lặp học liên kết được nén thứ l


Chọn UE đào tạo
Phát quảng bá mô hình thưa toàn cầu ban đầu
Báo cáo kết quả đào tạo
Chọn thiết bị
Tổng hợp liên kết được nén
Phát quảng bá trọng số thưa mới
Báo cáo tài nguyên UE
Báo cáo kết quả đào tạo diff(Wk)
Đào tạo từ thưa thớt đến dầy đặc
UE 1
UE 2
UE 3
UE 4
Lần lặp học liên kết được nén thứ (l+1)

Hình 10.31. Dòng thời gian học liên kết nén để nhận dạng hình ảnh

10.10.2.2.Quá trình diễn ra

Điều kiện ban đầu

UE có thể có khả năng phần cứng và thuật toán tính toán để đào tạo mô hình AI/ML chẳng hạn như nhận biết hình ảnh và video.

UE có thể gửi kết quả đào tạo trung gian đến máy chủ CFL.

Máy chủ CFL có thể chọn thiết bị đào tạo và xác định cấu hình đào tạo.

Máy chủ CFL có thể tổng hợp các kết quả đào tạo trung gian và tạo ra một mô hình toàn cục thưa thớt cho lần lặp lại học tập tiếp theo.

Máy chủ CFL có thể phân phối chế độ AI/ML toàn cục cho một nhóm người dùng đã chọn.

Các luồng dịch vụ

Bước 1: Máy chủ CFL chọn người dùng đào tạo từ người dùng ứng viên.

Bước 2: Máy chủ CFL gửi thông tin cấu hình đến người dùng đã chọn.

Bước 3: Máy chủ CFL phân phối mô hình toàn cục thưa thớt ban đầu (hoặc, tổng hợp) cho những người dùng được chọn thông qua mạng 5G.

Bước 4: Mỗi UE mở rộng mô hình toàn cục thưa thớt và đào tạo mô hình mở rộng bằng cách sử dụng dữ liệu cục bộ của nó. Sau đó, mỗi UE chỉ gửi gradient trọng số có giá trị quan trọng đến máy chủ CFL.

Bước 5: Máy chủ CFL tổng hợp các kết quả đào tạo nhận được từ các UE đào tạo và cập nhật mô hình toàn cục bằng cách sử dụng kết quả tổng hợp.

Bước 6: Cho đến khi mô hình AI/ML đạt được mức tăng cường hiệu suất bão hòa, quá trình sẽ chạy lặp đi lặp lại từ bước 1.

Bước 7: Nếu không, máy chủ CFL thực hiện tinh chỉnh để nén mô hình toàn cục cho mô hình toàn cục. Quá trình này có thể được áp dụng thường xuyên để cải thiện băng thông và hiệu quả tài nguyên tính toán trước khi hoàn thiện đào tạo.

Cuối cùng, máy chủ CFL phân phối mô hình toàn cục thưa thớt mới cho tất cả người dùng cần cùng một mô hình AI/ML.

Điều kiện kết thúc

Đối với một UE tiềm năng, CFL có thể giảm yêu cầu thông lượng đường lên và đường xuống cho quá trình học liên kết. Ngoài ra, độ phức tạp tính toán trong UE có thể giảm đáng kể do cho phép mô hình nén.

10.10.2.3. Các yêu cầu mới tiềm năng cần thiết để hỗ trợ trường hợp sử dụng

Hệ thống 5G sẽ hỗ trợ tải lên một gradient được đào tạo cho mỗi lần lặp lại của học liên kết nén với độ trễ tối đa là 1,05 ~ 3,25 giây.

Hệ thống 5G sẽ hỗ trợ tải xuống mô hình toàn cục cho mỗi lần lặp lại của học liên kết nén với độ trễ tối đa là 1,05 ~ 3,25 giây.

Hệ thống 5G sẽ hỗ trợ truyền đơn hướng UL với tốc độ dữ liệu UL trải nghiệm 26.13 ~ 80.88Mbit/s và tính khả dụng của dịch vụ truyền thông không thấp hơn [99.9%] để báo cáo các gradient được đào tạo cho học liên kết nén.

Hệ thống 5G sẽ hỗ trợ truyền đa hướng DL với tốc độ dữ liệu DL trải nghiệm của người dùng 26,13 ~ 80,88Mbit/s và tính khả dụng của dịch vụ truyền thông không thấp hơn [99,9%] để phân phối mô hình toàn cục cho học liên kết nén.

Chương 11

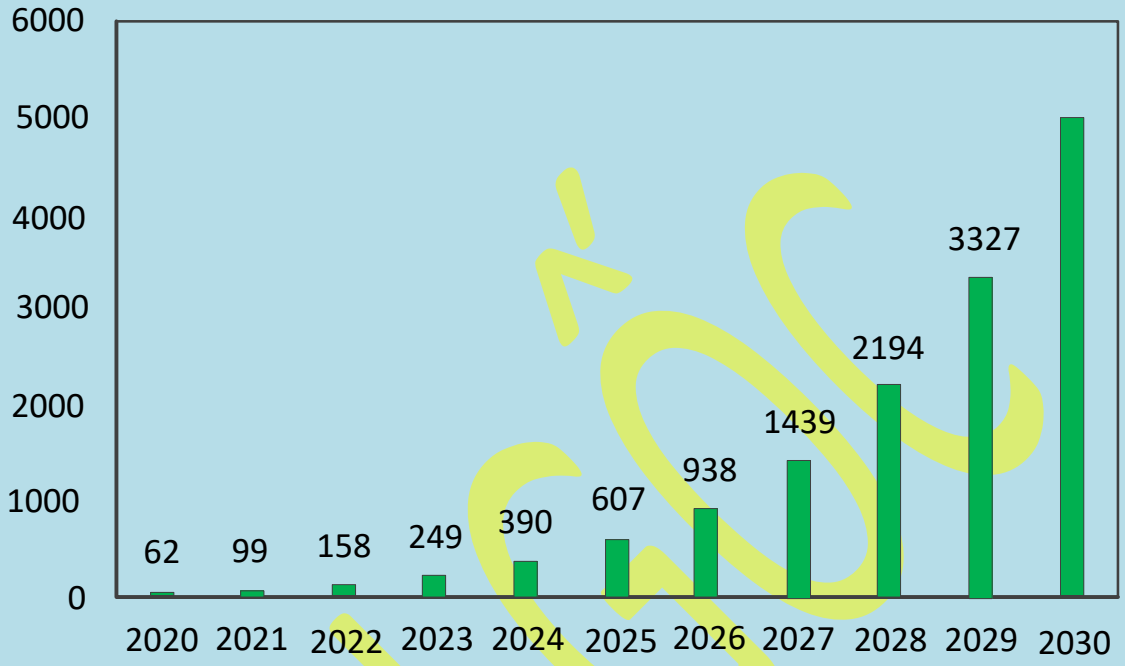
TÍCH HỢP AI/ML VÀO MẠNG TRUYỀN THÔNG DI ĐỘNG 6G

11.1. LỘ TRÌNH PHÁT TRỂN LÊN 6G

11.1.1. Mở đầu

Chúng ta đang tiến tới một xã hội dựa trên các hệ thống hoàn toàn tự động và quản lý từ xa. Sự phát triển nhanh chóng của các ứng dụng mới nổi như AI (artificial intelligence: trí tuệ nhân tạo), truyền thông 3D (3D media) và IoE (internet of everything: internert mọi thứ) đòi hỏi khối lượng lưu lượng rất lớn. Tổng khối lượng lưu lượng trong năm 2010 đã là 7,462 EB/tháng và dự báo lưu lượng này sẽ dạt đến 5016 EB/tháng. (1 Zetabyte (ZB)=10^3 Exabyte, 1 Exabyte (EB)= 10^3 Petabytes , 1 Petabyte(PB)= 10^3

Terabytes, 1 Terabyte (TB)= 10^3 Gigabytes, 1 Gigabyte (GB)= 10^3 Megabytes, 1 Megabyte (MB)= 10^3 Kilobytes, 1 Kilobyte(KB) = 10^3 bytes) . Hình 11.1 cho thấy dự báo phát triển tổng lưu lượng kết nối di động toàn cầu từ năm 2020 đến 2030.



Hình 11.1. Dự báo phát triển tổng lưu lượng kết nối di động toàn cầu từ năm 2020 đến 2030.

Dự báo này cho thấy tầm quan trọng của việc cần thiết cải thiện các hệ thống truyền thông. Các hệ thống tự động ngày càng trở nên phổ biến trong mọi lĩnh vực cuả xã hội như công nghiệp, y tế, giao thông đường bộ, hàng không. Để cung cấp các hệ thống thông minh và tự động, hàng triệu các bộ cảm biến sẽ được lắp đặt trong các thành phố, các xe ô tô, các nhà ở , các cơ sở công nghiệp, các đồ chơi và các môi trường khác. Vì thế cần có tốc độ số liệu cao với kết nối tin cậy để hỗ trợ các ứng dụng này.

Dự báo khoảng 65% dân số thế giới có thể truy nhập vào mạng 5G vào cuối năm 2025. Các mạng 5G sẽ cung cấp các tính năng mới và chất lượng dịch vụ tốt hơn so với các mạng 4G. Các mạng 5G cung cấp rất nhiều các loại dịch vụ khác nhau bao gồm: Băng rộng di động tăng cường (eMBB: enhanced Mobile Broad Band)), Truyền thông siêu tin cậy trễ thấp (URLLC), Truyền thông kiểu máy số đông (mMTC: massive Machine Type Communications). Công nghệ 5G đưa vào một số kỹ thuật bổ sung mới như các băng tần mới (chẳng hạn sóng mm và phổ quang), quản lý và sử dụng phổ nâng cao và tích hợp các băng tần cần được cấp phép và không cần cấp phép. Mặc dù các hệ thống truyền thông 5G sẽ cung cấp các cải thiện đáng kể so với các hệ thống trước nó, nhưng chúng

không thể thực hiện các yêu cầu của các hệ thống tự động và thông minh tương lai sẽ xuất hiện sau 10 năm nữa. Các mạng 5G sẽ không có đủ dung lượng cho một mạng hoàn toàn tự động và thông minh để cung cấp mọi thứ như là một dịch vụ và trải nghiệm thực tế ảo. Sự phát triển nhanh của các hệ thống tự động đặt trọng tâm vào số liệu có thể vượt quá các khả năng của các hệ thống 5G. Một số thiết bị như các thiết bị thực tế ảo sẽ vượt quá khả năng của sau 5G (B5G: Beyond 5G) vì chúng đòi hỏi tốc độ số liệu tối thiểu 10 Gbs. Vì thế cùng với việc đạt giới hạn đỉnh của 5G vào năm 2030, các tài liệu đã bắt đầu đề cập đến các mục tiêu thiết kế cho các bước tiếp theo của nó.

Để khắc phục các hạn chế của 5G nhằm hỗ trợ các yêu cầu mới, cần phát triển một hệ thống không dây thế hệ 6 (6G) với các tính năng hấp dẫn mới. Các động lực chính của 6G sẽ là sự hội tụ của tất cả các tính năng cũ như mật độ mạng cao, thông lượng cao, độ tin cậy cao, tiêu thụ công suất thấp và kết nối số đông. 6G cũng sẽ tiếp tục xu thế của các thế hệ trước như đưa ra các dịch vụ mới cùng với các công nghệ bổ sung mới. Các dịch vụ mới bao gồm AI, các thiết bị đeo thông minh, các thiết bị cấy trên người, giao thông tự động, các thiết bị thực tế ảo, cảm biến và lập bản đồ 3D. Yêu cầu quan trọng nhất đối với các mạng 6G là khả năng xử lý khối lượng lớn số liệu và kết nối tốc độ số liệu cao trên một thiết bị.

Hệ thống truyền thông không dây 6G sẽ tăng QoS nhiều lần so với 5G cùng với một số tính năng hiện có. Nó sẽ bảo vệ hệ thống và đảm bảo an ninh số liệu người dùng. Nó sẽ cung cấp các dịch vụ tiện lợi. Hệ thống truyền thông không dây 6G được kỳ vọng là một phương tiện truyền thông toàn cầu. Nó được dự kiến đạt tốc độ vào khoảng 1Tbit/s trong nhiều trường hợp. 6G được kỳ vọng đảm bảo kết nối đồng thời 1000 lần cao hơn 5G.

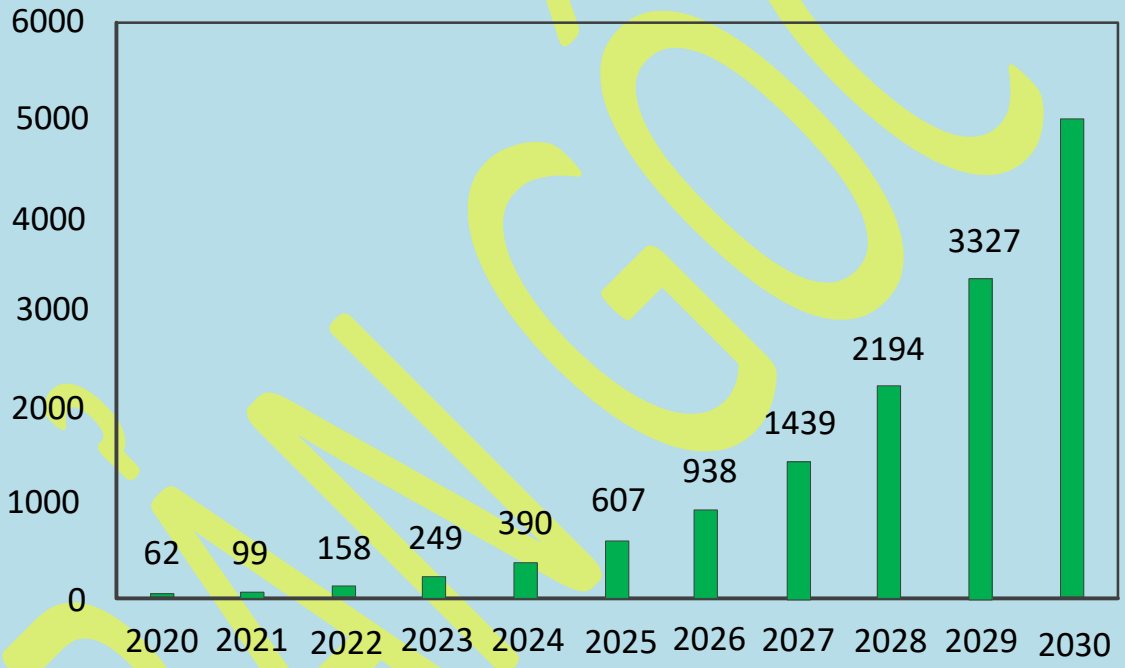
Ngoài ra nó cũng được kỳ vọng đảm bảo kết nối cự ly rất xa với độ trễ chỉ 1 ms. Tính năng hấp dẫn nhất của 6G là đảm bảo hỗ trợ hoàn toàn AI cho các hệ thống tự lái. Trong truyền thông 6G, dự tính lưu lượng kiểu video sẽ nổi trội. Các công nghệ quan trọng nhất đóng vai trò động lực cho 6G sẽ là băng tần terahertz (THz), AI, truyền thông không dây quang (OWC: optical wireless communications), nối mạng 3D, thiết bị bay không người lái (UAV: unmanned aerial vehicle) và chuyển công suất không dây.

11.1.2. Các xu thế trong truyền thông di động

Ngay từ khi bắt đầu hệ thống truyền thông di động tương tự đầu tiên vào các năm 1980, hầu như cứ 10 năm lại có một thế hệ các hệ thống truyền thông di động mới được đưa ra. Sự chuyển đổi từ thế hệ này sang thế hệ khác cải thiện các số đo QoS, bao hàm cả các dịch vụ mới và đảm bảo các tính năng mới. Trong 10 năm gần đây, lưu lượng số liệu tăng khủng khiếp do việc đưa ra các thiết bị thông minh và truyền thông M2M (máy đến

máy). Hình 12.2 cho thấy sự tăng trưởng hàm mũ của các kết nối di động. Dự báo lưu lượng năm 2030 sẽ tăng gấp 670 lần so với năm 2010.

ITU dự báo rằng đến cuối năm 2030 tổng khối lượng lưu lượng số liệu di động sẽ vượt quá 5 ZB trên tháng (1 ZB= 10^{21}byte) . Số lượng thuê bao sẽ đạt đến 17,1 tỷ so với 5,32 tỷ trong năm 2010. Khối lượng lưu lượng cho từng thuê bao cũng sẽ tăng (hình 2). Khối lượng lưu lượng cho từng thuê bao trong năm 2010 là 5,3 GB trên tháng. Tuy nhiên khối lượng này sẽ tăng 50 lần vào năm 2030. Số lượng đăng ký M2M sẽ tăng 33 lần trong năm 2020 và 455 lần trong năm 2030. Bảng11.1 cho thấy một số so sánh về sự sử dụng kết nối di động trong các năm 2010, 2020 và 2030.



Hình 11.2. Dự báo tăng trưởng khối lượng lưu lượng trên một thuê bao theo tháng.

Bảng 11.1. Các xu thế kết nối không dây toàn cầu

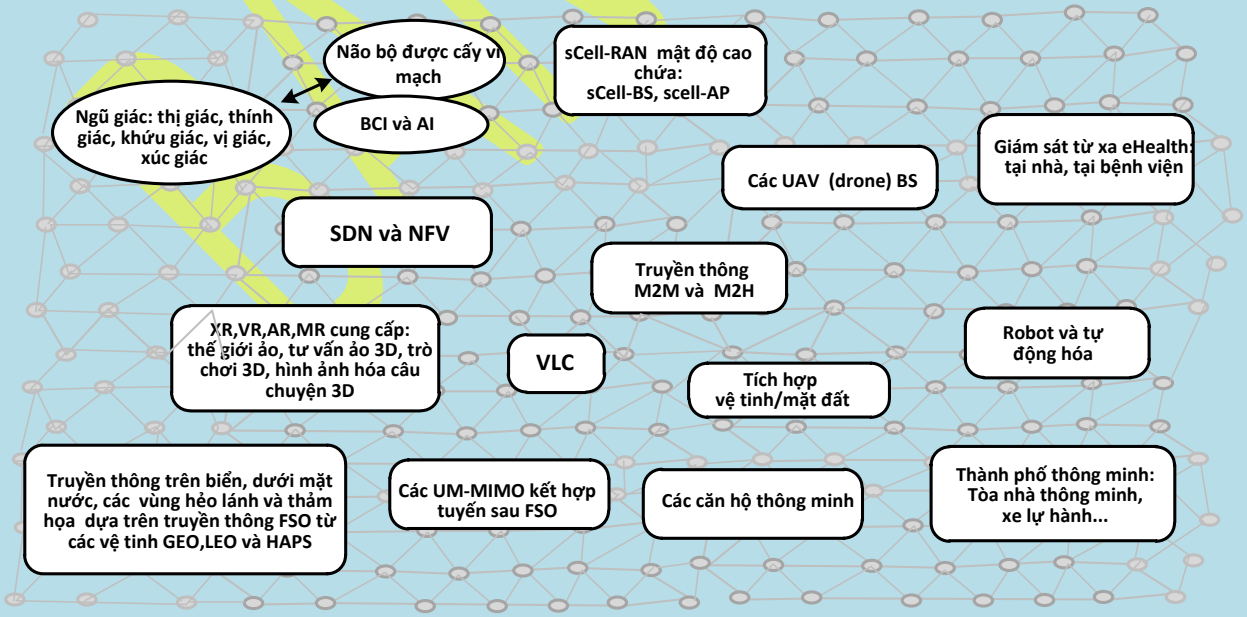
Số lượng thuê bao và khối lượng lưu lượng	**2010**	**2020 (dự báo)**	**2030 (dự báo)**	**Đơn vị**
Số lượng thuê bao di động	5,32	10,7	17,1	Tỷ
Số lượng thuê bao điện thoại thông minh	0,645	1,3	5,0	Tỷ
Số lượng thuê bao M2M	0,213	7,0	97	Tỷ
Khối lượng lưu lượng	7,462	62	5016	EB/tháng
Khối lượng lưu lượng M2M	0,256	5	622	EB/tháng
Lưu lượng trên một thuê bao	1,35	10,3	257,1	GB/tháng

BẢN GỐC

Mới đây các mối quan tâm của nghiên cứu đã chuyển đến các phương pháp thích ứng và trí tuệ với trọng tâm là số liệu. Các mạng không dây 5G tới đây sẽ xây dựng nền tảng của các mang trí tuệ nhằm đảm bảo cho các hoạt động của AI. Dự báo đến năm 2030 dung lượng 5G sẽ đạt đến giới hạn của nó. Khi này chỉ có có mạng 6G mới đảm bảo quản lý và thích ứng cho mạng hoàn toàn trí tuệ để cung cấp các dịch vụ tiên tiến. Các nhà nghiên cứu đã tiến hành nghiên cứu truyền thông 6G sẽ như thế nào vào năm 2030. Họ cũng đang xem xét các động lực có thể có đối với truyền thông không dây 6G. Một số xu thế động lực quan trọng cho phát triển 6G như sau: tốc độ bit cao, độ tin cật cao, trễ thấp, hiệu suất năng lượng, hiệu suất tần phổ, phổ tần mới, truyền thông xanh, các mạng trí tuệ, khả dụng mạng và hội tụ truyền thông, địa phương hóa, điện toán, điều khiển và cảm biến. 6G sẽ là một hệ thông toàn cầu hoàn được kết nối và số hóa hoàn toàn. Nó sẽ thúc đẩy cách mạng công nghiệp 4.0

11.1.3. Các viễn cảnh và ứng dụng

Trong tương lai, tất cả sẽ được tích hợp vào các hệ thống truyền thông không dây 6G. Tất cả các thiết bị, quản lý, xử lý tín hiệu lớp vật lý, quản lý tài nguyên và truyền thông dựa trên dịch vụ … sẽ được tích hợp bằng cách sử dụng AI**.** Nó sẽ thúc đẩy cách mạng công nghiệp 4.0 là sự chuyển đổi số của quá trình sản xuất của công nghiệp. Hình 11.3 cho thấy viễn cảnh của kịch bản kiến trúc truyền thông 6G. Dưới đây ta sẽ xét ngắn gọn các viễn cảnh và các ứng dụng của truyền thông 6G.



UM-MIMO BS (Ultra-Massive MIMO base station): trạm gốc nhiều anten phát thu mật độ siêu cao, sCell-RAN (Small cell Radom Access Network): mạng truy nhập ngẫu nhiên ô nhỏ, X/V/A/MR (Extended,Virtual, Augmented/Mixed): thực tế mở rộng/ảo/tăng cường/pha trộn, VLC (Visible Light Communication): truyền thông ánh sáng nhìn được, sCell BS (sCell Base Station): trạm gốc ô nhỏ, sCell-AP (Small Cell Access Point): điểm truy nhập ô nhỏ, SDN/NFV (Software Defined Networking/Network Function Virtualization): Nối mạng được định nghĩa bằng phần mềm/ảo hóa phần tử mạng, UAV (Unmanned Aerial Vehicle): thiết bị bay không người lái, UAV (Drone) BS : trạm gốc đặt trên UAV (Drone), BCI (Brain Commputer Interface): giao diện não- máy tính, AI (Artificial Intellegence): trí tuệ nhân tạo, M2M (Machine to Machine): máy đến máy, M2H (Machine to Human): máy đến người,FSO (Free Space Optical): quang trong không gian tự do, eHealth (Electonic Health): chăm sóc sức khỏe điện tử, GEO (Geostationary Satellite): vệ tinh địa tĩnh, LEO (low earth orbit satellite): vệ tinh quỹ đạo trái đất tầm thấp,HAPS (High altitude Pseudo Satellite): vệ tinh độ cao ngẫu nhiên.

Hình 11.3. Các kịch bản có thể xẩy ra của kiến trúc truyền thông 6G

Xã hội siêu thông minh.

Các tính năng vượt trội của 6G sẽ đẩy nhanh quá trình xây dựng các xã hội siêu thông minh dẫn đến các cải hiện chất lượng cuộc sống, giám sát môi trường và tự động hóa bằng cách sử dụng truyền thông M2M (Machine-to-Machine: máy đến máy) dựa trên AI và tự thu thập năng lượng (energy harvesting). Kết nối không dây 6G sẽ làm cho xã hội của chúng ta trở nên siêu thông minh thông qua việc sử dụng các thiết bị di động thông minh, xe cộ tự hành, ….. Ngoài ra rất nhiều thành phố sẽ triển khai các taxi bay dựa trên công nghệ không dây 6G. Các căn nhà thông minh sẽ trở thành hiện thực nhờ điều khiển từ xa bằng các lệnh phát đi từ các thiết bị thông minh.

Mở rộng môi trường truyền thông

Ngày nay, môi trường truyền thông có mặt ở khắp mọi nơi giống như không khí quanh ta và cần thiết như điện và nước. Vì thế người dùng không cần biết bối cảnh truyền thông và vùng dịch vụ truyền thông. Đòi hỏi phải có môi trường truyền thông ở mọi nơi cùng với mở rộng vùng hoạt động của con người và vật dụng. Các tòa nhà cao tầng, các drone, các ô tô bay, các máy bay và thậm chí không gian vũ trụ sẽ là các vùng hoạt động đương nhiên và không chỉ mặt đất mà cả bầu trời, không gian vũ trụ sẽ là các vùng truyền thông không thể thiếu. Nhu cầu đối với các vùng truyền thông trên biển và dưới biển không ngừng tăng. Do các nhu cầu của các mạng cảm biến, các xí nghiệp không cần con người, và các công trình xây dựng không cần con người, nên cũng cần xây dựng một vùng truyền thông trong môi trường không có con người. Vì vậy mọi nơi trên mặt đất, bầu trời , biển cả sẽ trở thành một vùng truyền thông.

Truyền thông giữa người và các vật dụng

Các chức năng tiên tiến như các thiết bị XR (VR,AR,MR: Virtual Reality, Augmented Reality, Mixied Reality: thực tế ảo, thực tế tăng cường, thực tế pha trộn), các ảnh độ phân giải cao và các ảnh toàn kỳ (holograph) lớn hơn 8K và các truyền thông ngũ giác quan trọng đó có xúc giác sẽ phát triển mạnh mẽ và truyền thông giữa con người và giữa con người và vật dụng sẽ trở nên phong phú và siêu thực. Vì thế các dịch vụ giải trí mới và các dịch vụ doanh nghiệp (enterprise service), xem các môn thể thao, … sẽ được cung cấp mọi nơi, mọi lúc. Nhờ sự phổ biến và phát triển nhanh chóng của các dịch vụ IoT,yêu cầu đối với truyền thông của các vật dụng sẽ trở nên rất lớn. Vì thế cần có truyền thông tốc độ cao, trễ thấp để đáp ứng yêu cầu xử lý khối lượng số liệu lớn (bao gồm các ảnh độ phân giải cao và điều khiển thiết bị) vượt quá khả năng của con người.

Kết hợp giữa mạng máy tính (cyber) và môi trường vật lý (CPf: cyber-physical fusion) ngày càng hoàn thiện

Rất nhiều dịch vụ sử dụng CPF sẽ được tạo ra trong thập niên 2020 và sẽ được sử dụng trong tất cả các môi trường, nhưng sẽ cần CPF tiên tiến hơn trong thập niên 2030. Bằng cách phát và xử lý khối lượng thông tin lớn không trễ giữa không gian mạng máy tính (cyberspace: không gian cyber) và không gian vật lý, hai không gian này sẽ ngày càng được kết hợp chặt chẽ hơn và cuối cùng có thể lấp kín khoảng trống giữa hai không gian này. Đối với con người, không gian cyber có thể hỗ trợ suy nghĩ và hành động của con người trong thời gian thực thông qua các thiết bị đeo và các vi thiết bị cài cắm trên cơ thể con người. Tất cả các vật thể như các thiết bị vận tải bao gồm các xe ô tô, máy xây dựng, máy công cụ, các camera giám sát và các bộ cảm biến khác nhau sẽ được liên kết vào không gian cyber. Chúng sẽ hỗ trợ an toàn và an ninh, các giải pháp cho các vấn đề xã hội và cuộc sống phông phú của con người.

Thực tế mở rộng.

Các dịch vụ thực tế mở rộng (XR: extended reality) bao gồm thực thế tăng cường (AR: augmented reality), thực tế pha trộn (MR: mixed reality) và thực tế ảo (VR: virtual reality) là các tính năng rất quan trọng của các hệ thống truyền thông 6G. Tất cả các tính năng này là các đối tượng của 3D và AI vì chúng là các phần tử động lực quan trọng. Ngoài việc đảm bảo các yêu cầu cảm nhận về tính toán, tự nhận biết, lưu trữ, giác quan, và sinh lý học, 6G sẽ đảm bảo trải nghiệm AR/MR/VR giống như thực tế bằng cách tích hợp thiết kế liên kết và kết nối không dây 6G chất lượng cao. VR là trải nghiệm 3D được phỏng tạo bởi máy tính trong đó các công nghệ máy tính sử dụng các thiết bị đeo trên đầu để tạo nên một thế giới ảo giống như thật. Các bộ đeo này tạo ra cảm giác thực và thay

thế môi trường thực hoặc tạo ra thế giới ảo. Một môi trường thực sự VR phải kết hợp được tất cả 5 giác quan. AR là quang cảnh thực của một thế giới thực tế vật lý với các chi tiết được tăng cường bởi các đầu vào cảm biến được xử lý bởi máy tính như số liệu về âm thanh, video, thị giác, và GPS (hệ thống định vị toàn cầu). Nó sử dụng thực tế hiện thời và bổ sung cho nó bằng các sử dụng một thiết bị nào đó. MR hòa nhập thế giới thực tại và thế giới ảo tạo ra một bầu không khí mới và các mường tượng mới để tương tác trong thời gian thực. MX đôi khi được gọi là thực tế lai ghép. Một đặc trưng quan trọng của MX là các nội dung của thế gới thực và thế giới nhân tạo có thể đáp ứng với nhau trong thời gian thực. XR bao trùm tất cả: các môi trường kết hợp thực và ảo, và các tương tác người -máy được tạo ra bởi công ngệ máy tính và các thiết bị đeo. Nó bao gồm tất cả các dạng để mô tả nó như AR, VR và MR. Nó kết hợp chung các thuật ngữ AV, VR và MR vào một thuật ngữ. Kết nối không dây tốc độ số liệu cao, trễ thấp và tin cậy cao trong hệ thống 6G là các nhân tố quan trọng cho một trải nghiệm XR (AR, VR và MR) đích thực.

Các hệ thống robot và tự hành được kết nối.

Hiện nay nhiều nhà nghiên cứu công nghệ ô tô đang nghiên cứu các xe ô tô tự hành và được kết nối. Các hệ thống 6G sẽ hỗ trợ phát triển các robot được kết nối và các hệ thống tự hành. Hệ thống UAV đưa hàng là một thí dụ của các hệ thống này.Vận tải và xe tự hành có thể thay đổi đáng kế cách sống hàng ngày của chúng ta. Hệ thống 6G có thể thúc đẩy triển khai thực sự các xe tự hành. Xe tự hành sẽ nhận biết môi trường xung quanh nó bằng cách kết hợp nhiều loại cảm biến khác nhau như LiDAR (Light detection and Ranging: đo khoảng cách dựa trên chiếu xạ ánh sáng), radar, sonar (đo khảng cách bằng sóng âm), odometry (đánh giá sự thay đổi vị trí dựa trên sự di chuyển của các bộ cảm biến) và các IMU (inertial measurement unit: các bộ đo gia tốc và độ xoay để xác định vị trí). Hệ thống 6G sẽ hỗ trợ kết nối tin cậy giữa xe đến mọi thứ (vehicle-to-everything) và giữa xe đến server (vehicle-to server). UAV là một kiểu của máy bay không người lái. Các mạng 6G sẽ hỗ trợ bộ điều khiển mặt đất và các cuộc truyền thông của hệ thống giữa UAV với mặt đất. Các UAV được sử dụng trong nhiều lĩnh vực như quân sự, thương mại, khoa học, nông nghiệp, vui chơi giải trí, an ninh trật tự, giao hàng, giám sát, chụp ảnh trên không trung, quản lý thảm họa và đua drone (drone racing). Ngoài ra UAV còn được sử dụng để phát quảng bá không dây và truyền dẫn tốc độ cao khi không có trạm gốc hoặc nó không hoạt động.

Tương tác không dây giữa não và máy tính.

Giao diện não-máy tính (BCI: brain-computer interface) là một phương pháp để điều khiển các đồ dùng và dụng cụ được sử dụng hàng ngày trong các xã hội thông minh, nhất

là các đồ dùng và dụng cụ được sử dụng trong gia đình và trong các hệ thống y tế. Đây là truyền thông trực tiếp giữa não bộ và các thiết bị bên ngoài. BCI bắt các tín hiệu của não bộ phát đến một thiết bị số, phân tích và chuyển đổi chúng vào các lệnh hay các hành động. Các tính năng của truyền thông 6G sẽ hỗ trợ việc các thực hiện thực sự hệ thống BCI để tạo ra một cuộc sống thông minh.

Giao tiếp xúc giác (Haptic communication).

Haptic communication là giao tiếp không dùng lời nói mà thông qua cảm ứng xúc giác (haptic hay touch: chạm, sờ). Truyền thông 6G sẽ hỗ trợ giao tiếp xúc giác; người dùng từ xa có thể nhận được trải nghiệm xúc giác thông qua một hệ thống tương tác thời gian thực. Các tính năng cao hơn của các mạng truyền thông 6G sẽ hỗ trợ thực hiện các hệ thống haptic.

Y tế thông minh.

Các hệ thông y tế cũng sẽ hưởng lợi từ các hệ thống 6G vì các sự đổi mới như AR/VR, hiển diện từ xa holographic, điện toán biên (Edge Computing) và AI sẽ hỗ trợ xây dựng các hệ thống y tế thông minh. 6G sẽ hỗ trợ hệ thống giám sát từ xa tin cậy trong các hệ thống y ế thông minh. Phẫu thuật từ xa có thể được thực hiện nhờ truyền thông 6G. Hệ thống 6G tốc độ số liệu cao, trễ thấp và siêu tin cậy sẽ hỗ trợ truyền tải các khối lượng lớn số liệu y tế nhờ vậy có thể cải thiện thực hiện y tê và chất lượng y tế.

Tự động hóa và sản xuất

6G sẽ đảm bảo tự động hóa hoàn toàn dựa trên AI. Thuật ngữ tự động hóa đề cập đến điều khển tự động các quá trình, các thiết bị và các hệ thống. Các hệ thống tự động hóa 6G sẽ đảm bảo truyền thông tin cậy, khả định cỡ và an ninh cao bằng cách sử dụng các mạng truyền thông tốc độ số liệu cao và trễ thấp. Hệ thống 6G cũng đảm bảo tính toàn vẹn vì nó đảm bảo chuyển số liệu không lỗi, không mất số liệu giữa phát và thu.

Truyền thông tin của 5 giác quan

Để trải nghiệm thế giới xung quanh, con người sử dụng 5 giác quan: thính giác, thị giác, khứu giác, vị giác và xúc giác. Các hệ thống truyền thông 6G sẽ chuyển từ xa các số liệu nhận được từ 5 giác quan này. Công nghệ này sử dụng quá trình thần kinh thông qua việc tích hợp các giác quan từ cơ thể người và môi trường xung quanh và sử dụng cơ thể một cách hiệu quả trong môi trường này và hoàn cảnh dịa phương. Công nghệ BCI sẽ hỗ trợ hiệu quả ứng dụng này.

Internet mọi thứ (IoE: Internet of Everthing)

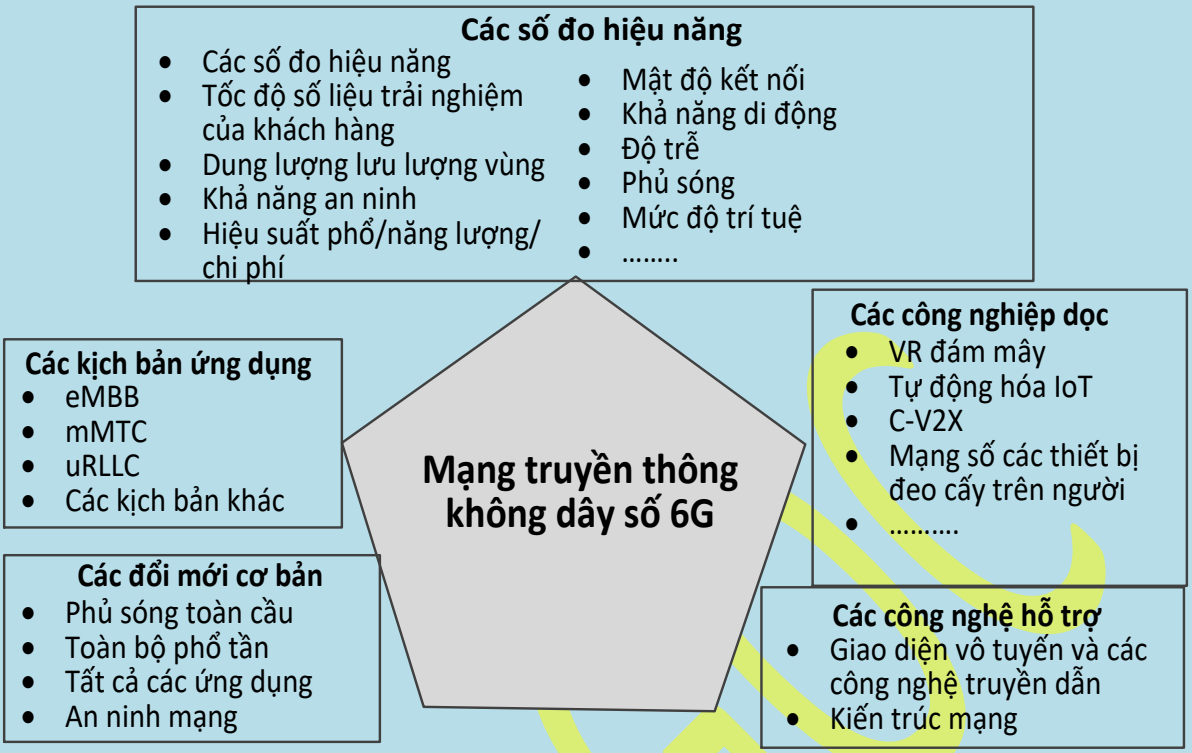
Internet mọi thứ là sự tích hợp liền mạch và kết hợp tự động giữa một số lượng rất lớn của các phần tử điện toán và các bộ cảm biến, các đối tượng hay các thiết bị, con người, các quá trình và dữ liệu bằng cách sử dụng hạ tầng internet. 6G sẽ đảm bảo hỗ trợ IoE đầy đủ. IoE là một khái niệm mở rộng của IoT, nó là thuật ngữ bao trùm tích hợp bốn thuộc tính trong cùng một khung như: dữ liệu, con người, các quá trình và các thiết bị vật lý. IoT nói chung đề cập đến các thiết bị vật lý, các đối tượng vật lý và truyền thông giữa chúng với nhau, nhưng IoE đưa ra trí tuệ để kết hợp con người, số liệu, các quá trình và các đối tượng vật lý vào một hệ thống. IoE sẽ được sử dụng cho các xã hội thông minh như các ô tô thông minh, y tế thông minh và các cơ sở công nghiệp thông minh.

11.3.4. Các đặc tả và yêu cầu

5G đã phải cân nhắc để cân đối một số vấn đề như: thông lượng, độ trễ, hiệu suất năng lượng, giá thành triển khai, độ tin cậy và độ phức tạp phần cứng. Vì thế khó có thể kỳ vọng rằng 5G sẽ đáp ứng được yêu cầu thị trường vào năm 2030. Khi này, 6G sẽ lấp kín khoảng trống giữa 5G và yêu cầu thị trường. Dự tính quá trình tiêu chuẩn hóa 6G sẽ khởi đầu vào khỏng năm 2026. Dựa trên các xu thế trước đây và các nhu cầu được dự báo, các mục tiêu chính đối với 6G sẽ là (1) tốc độ dữliệu trên một thiết bị rất cao, (2) số lượng các thiết bị được kết nối rất lớn,(3) trễ rất thấp, (4) giảm tiêu thụ năng lượng đối với các thiết bị IoT không pin, (5) kết nối tin cậy siêu cao và (6) trí tuệ được kết nối với khả năng học máy (machine learning).

6G được kỳ vọng đạt được hiệu năng cao hơn, cung cấp nhiều dịch vụ mới, cùng với nâng cấp các công nghệ hỗ trợ như trên hình 11.4.

BẢN NỘP



C-V2X (Cellullar Vehicle-to Everthings): xe cộ trong mạng tổ ong kết nối mọi thứ

Hình 11.4. Các kỳ vọng đối với mạng truyền thông không dây 6G

Mạng 6G phải đáp ứng được các kịch bản sử dụng và các yêu cầu như trong bảng 11.2.

Bảng 11.2. Các kịch bản sử dụng và các yêu cầu đối với 6G.

Các kịch bản sử dụng	feMBB, umMTC, muRLLC, MBRLLC, ERLLC, các kịch bản khác (HCS, MPS, LDHMC, ELPC)
Các ứng dụng	• Xã hội và ngành dọc đa chiều • Internet xúc giác (Haptic Internet) • Internet công nghiệp • Quan sát biển sâu • Internet các thết bị nano trong cơ thể người (IoBNT: Internet of bio nano things)
Các kiểu thiết bị	Các bộ cảm biến và và các thiết bị DLT,CRAS, XR, thiết bị WBCI, các cấy ghép thông minh
Các đặc tính mạng	Trí tuệ hóa, đám mây hóa, mềm hóa, ảo hóa,

	cắt lát
Độ lợi hiệu suất phổ và năng lương	100× tính theo bit/Hz/s//joules/ m^3 (thể tích)
Tốc độ đỉnh	≥ 1 Tbit/s
Tốc độ trải nghiệm	1 Gbit/s
Dung lượng lưu lượng vùng	1 Gbs/m^2
Mật độ kết nối	10^7/km^2
Khả năng di động	≥1000 km
Yêu cầu trễ đầu cuối đầu cuối	< 1ms
Yêu cầu trễ vô tuyến	10 ns
Yêu cầu tin cậy cuối đầu cuối	99,9999%
Các băng tần	Dưới 6 GHz, sóng mm cho truy nhập di động, khai thác tần số cao hơn và các băng THz (cao hơn 300 GHz
Kiến trúc	Các bề mặt thông minh không phụ thuộc ô, và thử nghiệm các ô THz siêu nhỏ

Ký hiệu:

feMMB (futher enhanced Mobile Broadband): di động băng rông tăng cường hơn nữa;

umMTC (ultra-mMTC: ultra-massive Machine-Type Communications): truyền thông dạng máy siêu đông;

muRLLC(massive Ultra Reliable Low Latency Communications): truyền thông s;iaau tin cậy trễ thấp số đông

MBRLLC (Mobile Broadband Realiable Low Latence communications): truyền thông di động băng rộng tin cậy trễ thấp

ERLLC (Extremely Realable Low latence Communications): truyền thông cực tin cậy trễ thấp

HCS (Humman Centric Services): các dịch vụ con người là trung tâm

3CLS (Convergence of Communications, Computing, Control, Localization and Sensing): hội tụ truyền thông, điện toán, điều khiển, định vị và cảm biến

MPS (Multi-purpose 3CLS and Energy Services): các dịch vụ năng lượng và 3CLS đa mục đích

LDHMC (Long Distance High Mobility Communications): truyền thông di động cao cự ly xa

ELPC (Extremly Power Communications): truyền thông công suất cực thấp

DLT (Distributed Ledger Technologies): Các công nghệ kế toán phân bố (giống như Blockchain)

CRAS (Connected Robotics and Autonomous Systems): các hệ thống robot cà tự quản kết nối

XR (Extended Reality): thực tế mở rộng

WBCI (Wireless Brain Computer Interactions): tương tác không dây não-máy tính

11.2. TỔNG QUAN TÍCH HỢP AI VÀ TRUYỀN THÔNG TRONG 6G

11.2.1. Các động lực tích hợp AI vào truyền thông không dây

Các hệ thống mạng không dây tương lai sẽ phát triển vào vào các hạ tầng không dây được tích hợp sâu "truyền thông", "cảm biến", "trí tuệ" và "lưu trữ". Các nền tảng này sẽ có khả năng cung cấp các dịch vụ tùy biến và cá nhân hóa theo yêu cầu, vượt trội các khả năng của các hệ thống truyền thông thông thường dựa trên các kiến trúc cô định và các quy tắc được định trước. Các hệ thống này cần sử dụng các công nghệ AI. Chúng sẽ sử dụng các cơ sở dữ liệu và hiểu biết lớn cho suy luận và đưa ra quyết định bằng cách khai thác các khả năng tổng quát hóa của các mô hình để thích ứng các môi trường và các kịch bản khác nhau và cung cấp ấn định tài nguyên tối ưu và các giải pháp quản lý. Hai động lực chính cho tích hợp truyền thông và AI như sau.

3) **AI đẩy xa giới hạn của các hệ thống truyền thông không dây**. AI có các khả năng phi thường trong xử lý dữ liệu lớn. Dự báo đến năm 2030 dữ liệu được tạo ra hàng tháng toàn cầu sẽ đạt đến 5016 EB (bảng 11.1). Dữ liệu này bao gồm dữ liệu thiết bị đầu cuối, giao diện vô tuyến, mạng và dịch vụ theo các định dạng khác nhau như văn bản, ngôn ngữ đánh dấu mở rộng (Extencible Marup Language), ngôn ngữ đánh dấu siêu văn bản (Hypertex Markup Language), đồ họa, thông tin Audio/Video. Các nhà mạng có thể sử dụng AI để đào tạo và đưa ra quyết định dựa trên suy luận trên các kiểu số liệu lớn khác nhau để nâng cao hiệu suất và tối ưu hóa mạng theo các mục tiêu và các kích thước khác nhau. AI thiếu các khả năng dự đoán trong các giải thuật thông thường. AI có thể học từ dữ liệu quá khứ thông qua các giải thuật ML và xây dựng các mô hình để dự báo các yêu cầu mạng và các vấn đề có thể xảy ra như nghẽn mạng, sự cố thiết bị hay sự thay đổi hành vi người dùng. Chẳng hạn AI có thể dự đoán trước các đỉnh lưu lượng tiềm năng trong các ngày nghỉ đặc thù hay các sự kiện diện rộng và đưa ra trước các biện pháp ấn định tài nguyên như tăng dung lượng của BS tạm thời hay tối ưu hóa các chiến lược, nhờ vậy ngăn chặn sự cố hệ thống và tăng cường trải nghiệm người dùng.

   Khả năng thích ứng là một đặc trưng nổi bật của AI. Các môi trường truyền thông không dây có tính động và chịu ảnh hưởng cao của các nhân tố khác nhau, như thời tiết, địa hình và các thay đổi tại vị trí các thiết bị người dùng. Khả năng thích ứng của AI cho phép nó thực hiện điều chỉnh nhanh chóng theo các biến động môi trường và các thăng giáng trong các điều kiện mạng. Chẳng hạn khi, khi người dùng di chuyển từ trong nhà hay trong một xe ô tô tốc độ cao, cường độ tín hiệu và nhiễu sẽ thay đổi. AI có thể cảm nhận được các thay đổi này theo thời gian thực và tự động điều chỉnh các thông số truyền thông để đảm bảo sự ổn định của chất lượng truyền thông, thích ứng các kịch bản truyền thông khác nhau và cải thiện sự tinh cậy và linh họa của các hệ thống truyền thông không dây.

   Khác với quản lý phân cấp của các mức khác nhau trong các hệ thống truyền thông thông thường, về mặt lý thuyết, AI có thể học các cấu trúc ẩn và các thông số khác nhau để phù

hợp các chức năng phức tạp tùy ý. Điều này cung cấp một cách thức hiệu quả hơn để cảm nhận môi trường không dây thay đổi, phức tạp và mô tả đặc điểm không gian trạng thái mạng. Mặt khác, các thiết kế các mo-đun hệ thống truyền thông khác nhau có thể có các mục đích xung đột và có thể có các ràng buộc hiệu năng giữa các mo-đun. Tồn tại các đánh đổi giữa các số đo hiệu năng, như dung lượng kênh và nhiễu, sự tin cậy truyền dẫn và tiêu thụ năng lượng hệ thống. Quá trình tối ưu hóa từng mo-dum riêng lẻ thường không thể đạt được hiệu năng tối ưu tổng thể. Trong các trường hợp này, AI có thể tạo điều kiện thiết kết tối ưu hoá kết hợp giữa các mo-đun.

4) **AI để giải quyết các thách thức trong truyền thông không dây.**Các thách thức chính đối với mạng truyền thông không dây là phải đáp ứng các yêu cầu sau:
    - Sử dụng tài nguyên phổ tốt hơn,
    - Vùng phủ không dây carbon thấp,
    - O&M (Operation and Maintenace: khai thác và bảo dưỡng) hiệu qủa và chi phí hiệu quả hơn,
    - Khả năng cá nhân hóa và tùy biến theo yêu cầu khách hàng tốt hơn,
    - Truyền dẫn tin cậy và an ninh tốt hơn.

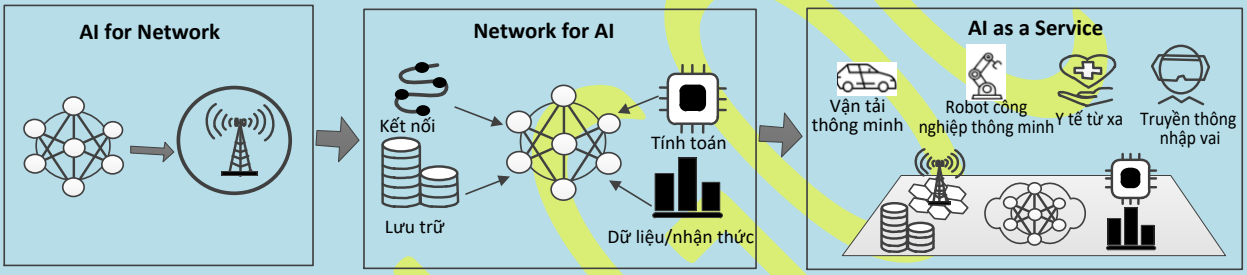
    Nguyên nhân cốt lõi hệ thống truyền dẫn không dây hiện thời không đáp ứng được các yêu cầu trên là chúng không đủ trí tuệ. Các hệ thống truyền dẫn không dây tương lai sẽ phát triển vào một hạ tầng không dây tích hợp sâu "truyền thông", "cảm biến", "trí tuệ" và "lưu trữ".

Về phía mạng truy nhập vô tuyến (RAN), AI có thể kết hợp tiến hành phân tích dữ liệu và đưa ra quyết định cho mặt phẳng đơn vị điều khiển 5G NR (5G New Radio: vô tuyến mới 5G) để thực hiện tối ưu hóa động ngay lập tức hiệu suất quản lý và lập lịch tài nguyên không dây. BS thế hệ sau (gNB) có thể đạt được tắt/bật chuyển giao thông minh dựa trên AI để đạt được tiết kiệm công suất động. Tại phía mạng, với chức năng NWDAF (Network Data Analytics Funntion: phần tử mạng phân tích dữ liệu) các thành phần mạng đạt được liên kết nối toàn diện với các thành phần CN xung quanh và thực hiện thu thập dữ liệu. AI cũng tham gia điều khiển và đưa ra quyết định của CN bao gồm đảm bảo QoS (Quality of Service: chất lượng dịch vụ), tính toán biên 5G để xử lý lưu lượng và cân bằng tải các chức năng mạng. Các tổ chức tiêu chuẩn hóa như 3GPP và IMT-2030 đã khởi đầu tiêu chuẩn hóa AI trong các hệ thống truyền thông và đạt được tiến bộ đáng kể. AI được dự kiến sẽ nâng cao hơn nữa sức mạnh cho hạ tầng trong các mạng,quản lý mạng, các hệ thống điều hành và AI trong hệ sinh thái truyền thông và các ứng dụng.

11.2.2. Tầm nhìn tích hợp AI và mạng truyền thông không dây

Tiến bộ của công nghệ AI đã tăng cường đáng kể sự phát triển của các mạng không dây. Hình 11.5 cho thấy mức độ tích hợp giữa 6G và AI từ ba góc nhìn: AI cho mạng (AI4N:

AI for Network), mạng cho AI (NET4AI: Network for AI) và AI như là một dịch vụ (AIaaS: AI as a Service). Ứng dụng AI để tăng cường mạng không dây nhìn chung được đề cập là AI4NET. Bản chất của nó là ở chỗ sử dụng AI để cải thiện hiệu suất, hiệu năng và trải nghiệm người dùng mạng không dây. Ứng dụng AI4NET trong các mạng 5G hỗ trợ đáng kể triển khai trí tuệ trong các mạng truyền thông di động và các ngành dọc. Vai trò của nó căn bản thể hiện trong trí tuệ O&M và trí tuệ truyền thông di động. Vai trò đầu nhấn mạng sử dụng AI để tối ưu các giải thuật thông thường, tự động hóa và trí tuệ hóa các nhiệm vụ trên các lớp và các hoạt động của mạng. Vai trò thứ hai tập trung lên đề suất trí tuệ dựa trên AI cho từng phân tử mạng và các thành phần chức năng, chia nhỏ các đáp ứng chủ động của mạng vào các dự đoán. Cách tiếp cận này mục đích cuối cùng là tối ưu hóa ấn định tài nguyên mạng và giảm trễ.



AI for Network: AI cho mạng, Network for AI: mạng cho AI, AI as a Service: AI như là một dịch vụ.

Hình 11.5. Phạm vi tích hợp 6G và AI.

Để đáp ứng các yêu cầu chuyển đổi và phát triển của AI trong các mạng không dây, 6G không còn chỉ là một mạng truyền thông phục vụ như là một tác nhân kết nối cho các dịch vụ trí tuệ mà thay vào đó nó phát triển vào một mạng tích hợp truyền thông, cảm ứng và các chức năng tính toán. Trong khi đó AI tăng thêm sức mạnh các mạng không dây 6G, để đạt được NET4AI nhằm cung cấp hỗ trợ toàn diện cho triển khai và các ứng dụng của các môi trường nhiều bên tham gia. NET4AI đề cập đến trường hợp mạng đóng vai trò hỗ trợ AI/ML. Ở đây, mạng không dây được sử dụng để hỗ trợ và nâng cao hoạt động và hiệu suất của các dịch vụ AI/ML. Thông qua việc tối ưu hóa kiến trúc mạng, công nghệ thu thập và truyền dữ liệu, mục tiêu là cung cấp hỗ trợ mạng mạnh mẽ cho các dịch vụ AI/ML. Hai khía cạnh này sẽ định hướng các nguyên tắc thiết kế của mạng 6G tích hợp AI/ML. Bản chất của NET4AI là cung cấp AI với các khả năng khác nhau để cho phép đào tạo/suy luận AI thời gian thực và hiệu quả hơn, và tăng cường an ninh cũng như bảo vệ tính riêng tư của dữ lieu.NHóm thúc đẩy IMT 2030 (6G) tại Trung Quốc hình dung kiến trúc mạng 6G như là một nền tảng mang tính đổi mới cho các dịch vụ để cung cấp các khả năng vượt trội kết nối thông thường. Các khả

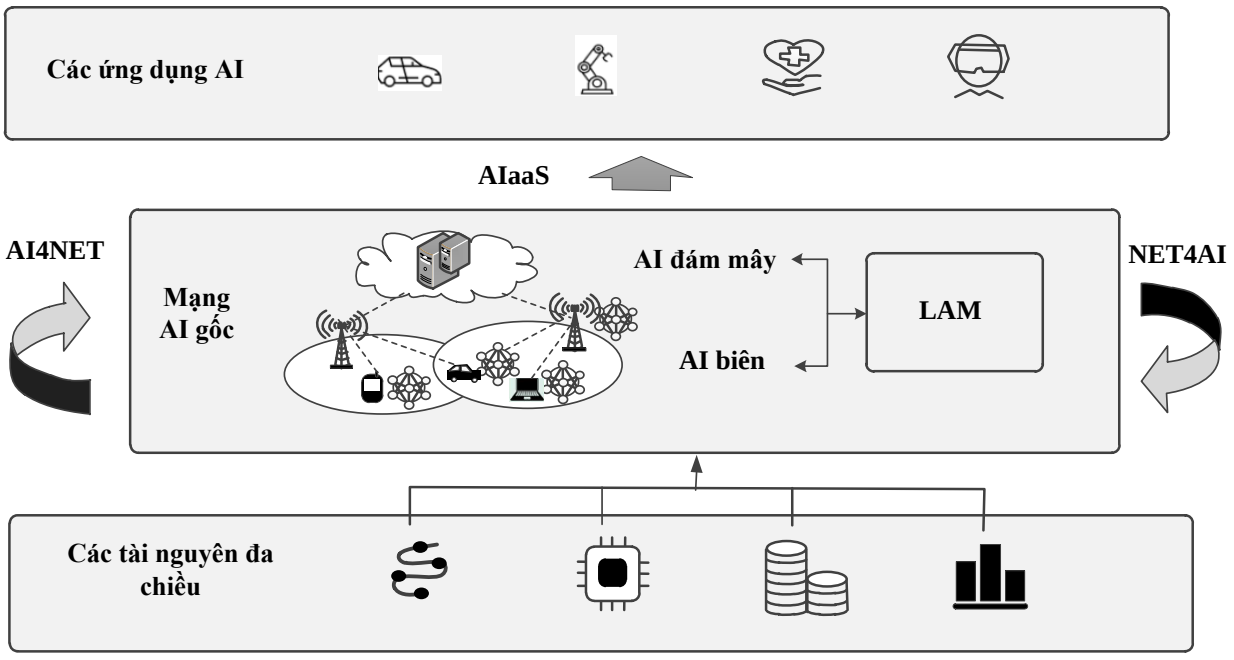
năng này bao gồm các mạng có sức mạnh tính toán, an ninh tin cậy, cảm biến, và các dịch vụ số liệu, đây là những điểm thiết yếu cho hoạt động của AI trong các mạng. Một số tác giả cũng đề xuất một kiến trúc mạng mới để cung cấp hỗ trợ gốc (Native Support) cho AI, được gọi là mạng định hướng AI, nó được trình bày như là một khung quản lý mạng với các khả năng tính toán AI và sự tham giá nhiều bên được xây dựng trong mạng.

Hơn nữa, để hỗ trợ hiệu quả " trí tuệ vốn có" (Inherent Intelligence) và đạt đến "AI gốc hay AI bẩm sinh" (Native AI), các mạng 6G sẽ tạo ra AIaaS để cung cấp và xử lý. Khái niệm này dẫn đến ý tưởng của 6G AIaaS. Cụ thể, 6G AIaaS sử dụng các tài nguyên và các chức năng trong mạng (bao gồm 6G CN, các mạng truy nhập không dây và các đầu cuối), chẳng hạn kết nối, tính toán, dữ liệu và các mô hình. 6G AIaaS nhắm đến xây dựng một hệ sinh thái dịch vụ AI phân tán, hiệu suất, hiệu suất năng lượng và an ninh., bao gồm đào tạo mô hình, suy luận AI, triển khai, và các chức năng khác trong một môi trưởng mở carbon thấp . Nó không chỉ định nghĩa lại hệ sinh thái của đám mây biên mà còn xây dựng các mô hình kinh doanh mới thông qua các mạng di động 6G, cho phép chuyển đổi từ các mạng định hướng kết nối quá khứ sang các mạng định hướng dịch vụ, cuối cùng đạt được trí tuệ mọi nơi. Các kịch bản điển hình của AIaaS bao gồm (nhưng không phải tất cả) các thành phố thông minh, nông nghiệp thông minh giáo dục toàn cầu và công nghiệp thông minh. Các ứng dụng điển hình bao gồm các dịch vụ taxi không người lái, kiểm tra lưới điện thông minh, giám sát y tế tại nhà và các lớp học ảo.

Các mô hình ngôn ngữ lớn (LLM: Large Language Model) đã thể hiện một bước đột phá đáng kể trong NLP (Natural Language Processing: xử lý ngôn ngữ tự nhiên) và có thể đóng góp tiềm năng cho phát triển 6G AI. So với các mô hình thông số nhỏ hơn truyền thống, các LLM thể hiện hiểu biết ngữ cảnh (Contextual Understanding) mạnh, tạo văn bản kết hợp (Coherent Text Generation), Lý luận logic (Logical Reasoning) và các khả năng tổng quát hóa. Các mô hình lớn mục đích chung (General Purpose Large Model) hiện có như GPT mã nguồn đóng (Closed Source Generative Pretrained Transformer: bộ chuyển đổi tiền huấn luyện tạo sinh được phát triển bởi OpenAI và mô hình Llama mã nguồn mở (Open source Llama Model) được phát triển bởi Meta (trước đây là Google) có thể xử lý các dải rộng các vấn đề chuyên biệt miền/ lĩnh vực. Mới đây, DeepSeek LLM mã nguồn mở được phát triển bởi đội ngũ DeepSeek tại Trung Quốc đã trở thành người thay đổi tiềm năng, chứng minh khả năng đào tạo các mô hình AI quy mô lớn (LAM: Large Scale AI Model) với chi phí tấp không ngờ. Các LLM có thể tiềm năng đóng vai trò then chốt trong các hệ thống truyền thông, chẳng hạn: trong quá trình xử lý nhiệm vụ và các dịch vụ thông minh.

Sự nổi lên của các LLM có thể đem đến một mô hình tích hợp 6G và AI mới. KIến trúc AI bản năng trong 6G có thể cung cấp các dịch vụ liên kết, tính toán, phân tách mô hình (chia nhỏ mô hình) và phân phối mô hình cho đào tạo và lý luận của các LLM. Các LLM có thể tăng thêm sức mạnh các lĩnh vực khác nhau của các mạng 6G (chẳng hạn: giao diện vô tuyến, phía mạng, an ninh mạng, O&M mạng, v.v..). Trong 6G, các mô hình lớn đặc thù hóa tùy biến có thể được triển khai trên các lớp mạng khác nhau. Các mô hình đặc thù hóa này có thể cộng tác xử lý các vấn đề mạng thông qua phân tách và kết hợp nhiệm vụ (Task Decomposition and Composition), cộng tác liên lớp (Cross-Layer) và cộng tác đám mây biên (Edge Cloud), để tăng cường mức trí tuệ của mạng.

Ngoài việc sử dụng các mô hình AI tùy chỉnh được thiết kế riêng cho các chức năng và các nhiệm vụ đặc thù, các mô hình lớn thể hiện hoạt động tổng quát nổi trội trên nhiều nhiệm vụ và các tính năng đột phá trên các nhiệm vụ phức tạp mà các mô hình nhỏ không thể đạt được. Các chức năng chính của các mô hình lớn ngày càng phù hợp với các đặc tính đa kịch bản và đa nhiệm vụ của 6G, giảm bớt hiệu quả khối lượng công việc trong nhận thức mô hình và cài đặt số đo hiệu suất trong các mạng 6G cho thấy các triển vọng ứng dụng rộng lớn. Một dạng ttích hợp 6G với AI có thể có được minh họa trong hình 11.6. Tận dụng dữ liệu từ đa dạng các mạng không đồng nhất trong 6G, các mô hình lớn này có thể tổng hợp các kịch bản và các dịch vụ 6G khác nhau, đặt nền tảng trọng yếu cho tăng cường AI trong các mạng 6G. Chẳng hạn đề xuất 6GGANA trong sách trắng 6GNETGPT đã xây dựng một LLM giống như ChatGPT, cung cấp một mô hình mới cho các hoạt động và quản lý nâng cao các mạng 6G.



LAM (Large Scale AI Model): mô hình AI quy mô lớn.

Hình 11.6. Minh họa tích hợp 6G và AI.

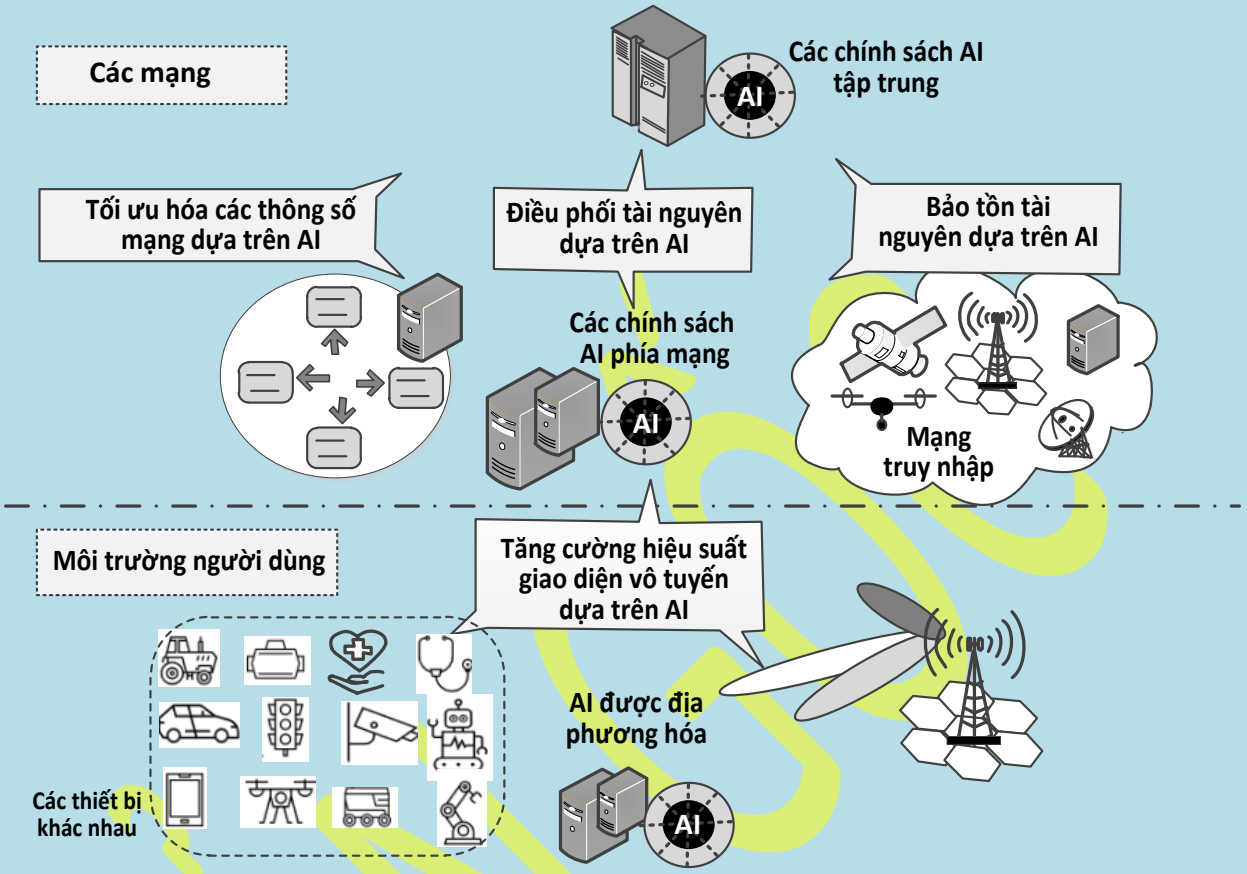
11.3. AI CHO MẠNG (AI4NET)

11.3.1. Tổng quan

AI4NET đề cập đến kịch bản trong đó trí tuệ nhân tạo/học máy (AI/ML) được sử dụng để tăng cường sức mạnh cho mạng. Trong bối cảnh này, các công nghệ AI được khai thác để nâng cao và tối ưu hóa hiệu suất và quản lý mạng không dây. Mục tiêu tổng thể của AI4NET là tận dụng trí tuệ nhân tạo để tăng cường hiệu quả, tính ổn định của mạng và cuối cùng là hiệu suất mạng và trải nghiệm người dùng.

Mục đích cơ bản nhất của tích hợp truyền thông vô tuyến và AI là ở tăng cường hiệu năng, hiệu suất mạng và trải nghiệm người dùng bởi sức mạnh của AI và điều mày được đề cập như là AI4NET. Tại giai đoạn này, chỉ tập trung là làm thế nào sử dụng các giải thuật AI để tối ưu hóa hiệu năng truyền thông và các chức năng mạng. Chẳng hạn, AI có thể tối ưu hóa quá trình điều chế và giải điều chế để truyền dẫn tín hiệu chính xác và hiệu quả hơn; với sự hỗ trợ của AI, tài nguyên mạng có thể được ấn định thông minh để đạt được cân bằng tải, và và có thể xây dựng quản lý O&M tự động để cải hiện hiệu suất O&M và giảm chi phí. Việc đưa vào AI không được kỳ vọng có ảnh hưởng đáng kể lên kiến trúc mạng gốc. Thay vào đó, nó cải thiện các vấn đề truyền thông trong việc nhận biết đặc thù bằng các đào tạo các mô hình giải huật AI. Qúa trình này tương tự như cấy ghép chính xác các miếng vá thông minh vào mạng. Nó không chỉ duy trì sự ổn định kiến trúc mạng mà còn từ từ tăng cường mức độ thông minh của mạng.

Khái niệm AI4NET được trình bày trong hình 11.7, trong đó AI không dây có các khả năng như chiết xuất tính năng, dự đoán thích ứng, tối ưu hóa, xử lý thời gian thực, đánh giá tương quan và phân cụm cảnh hiện trường (Scene Clustering). Các khả năng này tạo lập nền tảng của các mạng không dây được AI hỗ trợ và có thể được triển khai trực tiếp trên BS và CN để tạo điều kiện cho truyền thông không dây. AI tối ưu hóa các phương pháp thông thường để tăng cường hiệu năng mạng trên các thiết bị đầu cuối khác nhau. Tại phía mạng, AI tận dụng các khả năng của nó để cải thiện chất lượng của các dịch vụ đầu từ đầu đến đâu cuối, tối ưu hóa các chức năng của các phần tử mạng, tiến hành quản lý tự động mạng lớp cao hơn và tăng cường hiệu suất O&M. Sự xuất hiện của AI cũng dẫn đến các chuỗi dịch vụ mới cho mục đích cải thiện chất lượng và phục vụ người dùng tốt hơn.



Hình 11.7. Sơ đồ các khả năng AI cho mạng theo khái niệm AI4NET.

11.3.2. Tăng cường hiệu suất giao diện vô tuyến

Phần này sẽ trình bày cách thức mà công nghệ AI có thể cải thiện truyền dẫn trong phản hồi thông trạng thái kênh (CSI: Channel State Information), các máy thu OFDM (Orthogonal Frequency Division Multiplexing: ghép kênh phân chia theo tần số), quản lý búp sóng và định vị không dây.

11.3.2.1. Giải thuật phản hồi CSI dựa trên AI

Trong mạng 6G, chi phí gián tiếp cho phản hồi được kỳ vọng tăng mạnh cùng với số lượng các anten tăng đáng kể. Vì thế đảm bảo kết cấu lại kênh chính xác và giảm thiểu chi phí hồi tiếp CSI là một trong các vấn đề thắt nút cổ chai cần phải vượt qua. Sử dụng công nghệ AI trong hooifphanr hồi CSI phân tích chi tiết các đặc tính kênh tinh vi này. Thông qua đào tạo dựa trên một khối lượng lớn các mẫu được gán nhãn, mô hình AI có thể nhận được các mẫu kênh đặc trưng trong các môi trường pha-đinh đa dạng, các kịch bản suy luận và các tình huống đa đường. Trái ngược với các phương pháp đánh giá CSI

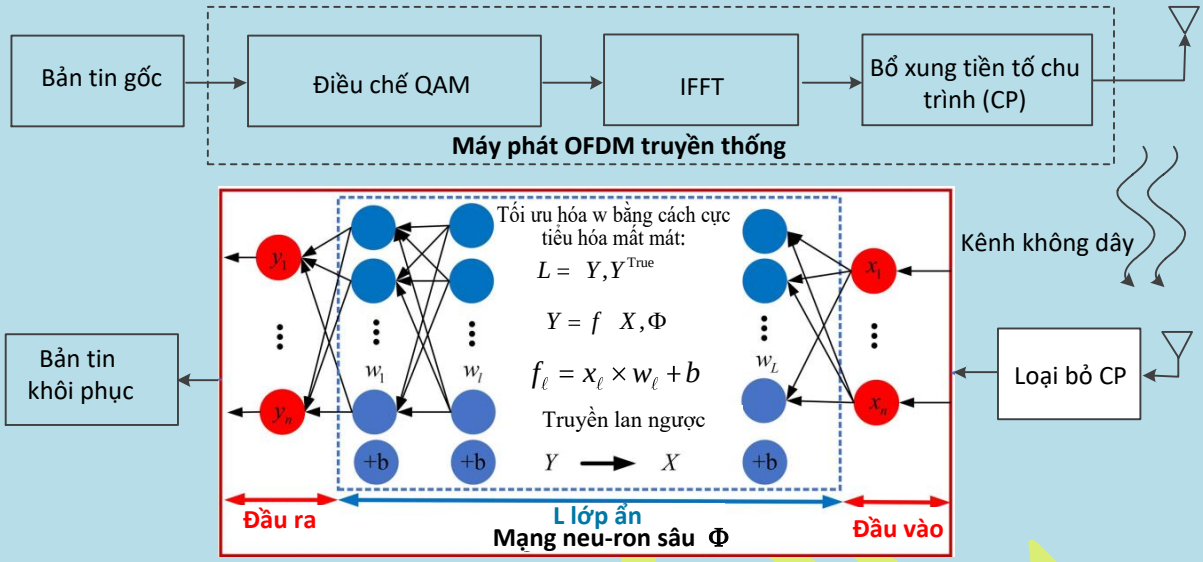
thông thường dựa trên lý thuất thống kê hay các mô hình tuyến tính đơn giản, các phương ơphaps dựa trên AI cho thấy các năng lực tổng quát hóa mạnh hơ và có thể đáng giá CSI chính xác hơn trong các môi trường truyền thông thực tế phức tạp.

Khối lượng dữ liệu CSI lớn và các đặc tính ngẫu nhiên của CSI khiến cho AI trở nên tiềm năng hữu ích cho thiết kế một cơ chế phản hồi CSI mới. Chẳng hạn các đạc tính phi tuyến của DL có thể được sử dụng cho chiết xuất và nén các tính năng của CSI. Sơ đồ CsiNET được một số tác giả đề xuất sử dụng tích cập để chiết xuất các tính năng, nén và tái tạo lại lại các kênh, nó vượt trội hơn sơ đồ phẩn hồi CSI dựa trên cảm biến nén về độ chính xác và phức tạp tính toán. Bọ mã hóa CsiNET soạn thảo các từ mã CSI kích thước cao, trong khi đó bộ giải mã chịu trách nhiệm giải mã các từ mã này về CSI nguyên gốc. Các cấu trúc bên trong của các bộ mã hóa và giải mã của các giải thuật khác nhau thay đổi dẫn đến các khác nhau trong hiệu suất tái tạo lại. Các mô phỏng cho thaatys lợi thê cực lớn của phản hồi CSI dựa trên AI so với các phương pháp bảng mã truyền thống. AI với thông tin bổ sung có thể cải thiện hiệu quả bình phương độ tương đồng Cosine tổng quát (SGCS: Square of Generalaized Cosine Similarity) và thông lượng.

11.3.2.2. Máy thu OFDMA dựa trên AI

Phương pháp AI có thể chiết xuất thông tin quan trọng trong các tần số hoa tiêu thay đổi theo thời gian thưa thớt. Bản đồ phước tạp quan hệ giữa đầu vào và đầu ra được xây dượng hiệu quả, nhờ vậy máy thu dựa trên AI có thể học và thích ứng các đặc tính kênh trong môi trường phức tạp. Từng mo-đun trong máy thu tthoong thường đòi hỏi lập mô hionhf và tính toán chính xác dẫn đến hệ thống phức tạp và gánh năng tính toán. AI có khả năng chiết xuất các tính năng trực tiếp từ tín hiệu gốc và giải điều chế nó theo cách dựa trên dữ liệu nhờ đó đơn gản hóa quá trình xử lý phức tạp của các máy thu truyền thống.

Liên quan đến các máy thu không daây OFDM, một số nhà nghiên cứu đã sử dụng mạng ne-ron sâu kết nối toàn bộ (Fully Connected DNN: Fully Connected Deep Neural Network) để cải thiện máy thu OFDM theo mo-dun. Khung điển hình của một máy thu dựa trên AI được cho trong hình 11.8.



QAM (Quuadrature Amplitude Modulation): điều chế biên độ cầu phương, IFFT (Inverse Fast Fourier Transformation): biến đổi Fourier nhanh đảo, CP (Cyclic Prefix): tiền tố chu trình.

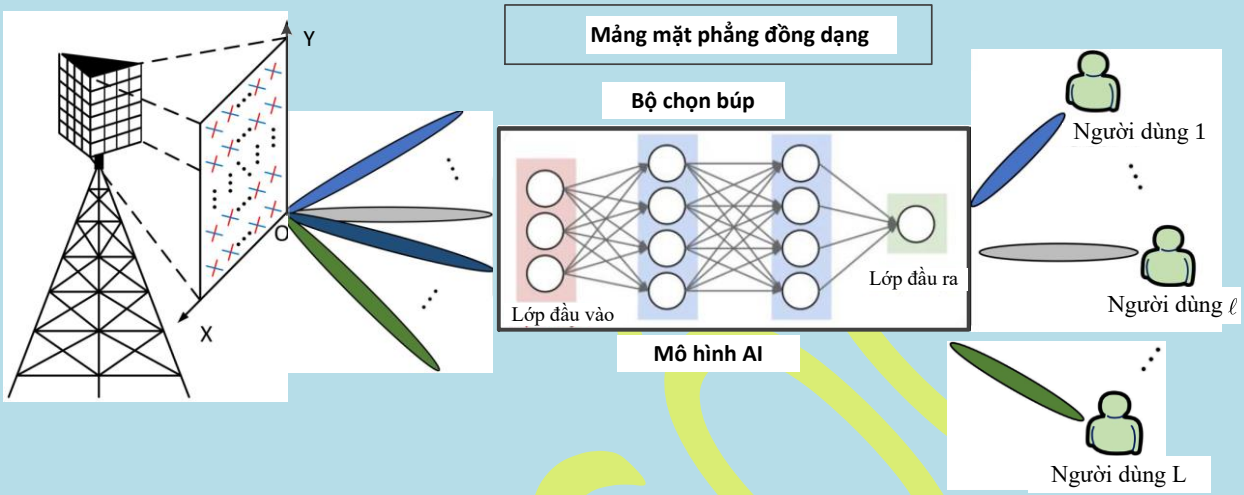
Hình 11.8. Máy thu OFDM dựa trên AI. Sử dụng các mô-dun AI để thay thế xử lý tính toán lớp vật lý. Các khả năng của AI có thể bù đắp méo tín hiệu.

Ta sử dung các thông số Φ của mạng neuron sâu (DNN) để xây dựng quan hệ giữa đầu vào X và đầu ra Y như là $Y=f(X,\Phi)$. Giả thiết rằng có L lớp ẩn, ta ký hiệu w_ℓ là trọng số thông số của lớp ℓ, trong đó ℓ nằm trong dải từ 0 đến L+1. w_0 và w_{L+1} là các trọng số của lớp đầu vào và đầu ra tương ứng. b ký hiệu cho độ lệch của từng lớp. Trong giai đoạn đào tạo, quá trình tối ưu hóa mô hình giảm thiểu mất mát giữa đầu ra Y và nhãn thực Y^{True}. Như vậy, các thông số mạng Φ được cập nhật bằng các thay đổi w_ℓ trong quá trình truyền ngược. Do đó, mô hình AI có thể mã hóa thông tin của các cảnh hiện trường thực tế vào các mạng neuron và tối ưu hóa hiệu suất của các giải thuật bằng cách điều chỉnh cấu trúc và các thông số của các mạng neuron.

11.3.2.3. Quản lý búp sóng dựa trên AI

Các phương pháp quản lý búp hiện nay được chia thành: (i) chọn búp dựa trên quét búp, (ii) chọn búp dựa trên dự báo vị trí, (iii) chọn búp dựa trên tìm kiếm phân cấp. Các phương pháp truyền thống này có một số hạn chế khi áp dụng cho các hệ thống mMIMO (massive Multi-Input Multi-Ouput: nhiều đầu vào nhiều đầu ra kích thước lớn) sóng mi-li-met. Nguyên nhân căn bản là các môi trường không dây thực tế thường có các đặc tính khác nhau. Sau khi một búp bị co hẹp, truyền dẫn sẽ nhạy cảm hơn với sự che khuất. Tăng số lượng các búp sẽ làm tăng đáng kể số lượng hoạt động tìm kiếm búp trong căn chỉnh búp. Vì thế khó chọn chính xác búp tối ưu ngay lập tức.

Với áp dụng AI, có thể chọn búp tối ưu mà không cần quét tất cả các cặp búp, nhờ giảm bớt chi phí gián tiếp tồn tại trong phương pháp quản lý búp truyền thống như được thể hiện trong hình 11.8.



Hình 11.8. Sơ đồ của quản lý búp dựa trên AI. AI chọn búp tối ưu bằng cách tìm kiến ánh xạ phức tạp quan hệ giữa đầu và đầu ra nhờ vậy có thể giảm đáng kể chi phí quản lý búp trong khi vẫn đảm bảo chính xác dự đoán.

11.3.2.4. Định vị dựa trên AI

Mạng 6G bắt buộc phải trang bị các khả năng định vị chính xác cao cho đa dạng các kịch bản. Thông tin định vị này thường trong dạng các tọa độ địa lý của người dùng di động tương đối so với một số điểm tham chiếu. Nhiều ứng dụ yêu cầu định vị chính xác cao, chẳng hạn xe ô tô tự hành có hướng dẫn (AGV: Automated Guided Vehicle) công nghiệp và theo dõi tài sản, đặc biệt là định vị chính xác trong nhà, nhưng hệ thống định vị toàn cầu (GPS) không thể sử dụng được trong nhà. Mặc dù có một tín hiệu tham chiếu định vị đặc thù, LTE không thể đáp ứng các yêu cầu cho định vị chính xác cao, do độ chính xác của nó khá thấp và khoảng cách khá lớn giữa các BS (khoảng 20 m thậm chí với băng thông 20 MHz). Bluetooth, WiFi và các công nghệ dựa trên định vị không dây khác có các chi phí triển khai cao và khó trở thành công nghệ định vị toàn cầu.

Ưng dụng công nghệ AI được đưa ra để đạt được định vị chính xác cao trong kỷ nguyên 6G. Một cách tiếp cận là sử dụng kỹ thuật rừng ngẫu nhiên (Random Forrest) có thể tạo ra một mô hình phân loại để chia nhỏ một vùng rộng lớn vào nhiều lưới nhỏ (Small Grid) và sau đó dự đoán vị trí người dùng. Cũng có thể sử dụng mô hình hối quy (Regression Model) để dự đoán các tọa độ vị trí người dùng. Trong một phương pháp khác, peceptron đa lớp có một ứng dụng giống như rừng ngẫu nhiên và có thể thiết lập các mô hình hồi

quy và phân loại để định vị các mục tiêu. Tất cả các giải thuật cũng có thể được đào tạo đến nhận biết dấu vết (Fingerptint) của tính hiệu không dây tại một vị trí cụ thể và kết hợp dữ liệu (Fusse Data) từ các bộ cảm biến khác nhau để đạt được định vị chính xác cao.

Với việc sử dụng rộng rãi mMIMO, AI đã cho thấy nhiều lợi thế đáng kể hơn trong định vị chính xác cao. Sử dụng kịch bản tồn tại đồng thơi kênh trực xạ (Light-of- Sight) và kênh không trực xạ (NLoS: Non-Line-of-Sight) dựa trên mạng neuron làm thí dụ, ML và khối lượng dữ liệu kênh đáng kể có thể hỗ trợ hiệu quả ánh xạ quan hệ giữa đáp ứng kênh và các tọa độ vị trí để giải quyết vấn đề định vị trong môi trường phức tạp và tăng độ chính xác định vị. Có thể đạt được độ chính xác định vị rất cao mà không cần chi phí bổ sung đáng kể, thậm chí chính xác vào khoảng 20 mm trong các điều kiện LoS và NLoS trong nhà. Do tính đa dạng của các chế độ dữ liệu trong các môi trường phức tạp, phát triển hơn nữa và áp dụng thực tế của định vị dựa trên AI đòi hỏi ta cần xét đến sự bền vững và tin cậy của mô hình AI để cho phép quản lý di động.

11.3.3. Cải thiện hiệu năng O&M mạng

AI có thể phân tích một khối lượng lớn dữ liệ và đưa ra quyết định thời gian thực đặc biệt hữu ích trong các mạng truyền thông không dây nơ mà các biến số và các thông số cần được liên tục giám sát và được hiệu chỉnh để tối ưu hóa hiệu suất hệ thống. Nó có thể ấn định động tài nguyên như băng thông và công suất trong các mạng không dây để tối ưu hóa hiệu suất hệ thống. Ngoài ra tích hợp AI cho phép mạng hiểu biết tốt hơn và dự đoán sự phức tạp và các biến động của mạng. Bằng các kết quả dựa trên dự báo, bảo dưỡng có thể thực hiện mạng chủ động và kịp thời. Bằng cách sử dụng học sâu của AI và các công nghệ nhận dạng mẫu, ta có thể đạt được hiểu biết sâu hơn về hành vi mạng nhơ vậy thực hiện quản lý tài nguyên mạng và các chiến lược lập lịch tối ưu hơn. Tiếp sau, vai tro quản lý O&M mạng sẽ được đưa vào thông qua các trường hợp cụ thể.

11.3.3.1. Dự báo lưu lượng dựa trên AI

Sự phát triển dần dần của các mạng 6G báo trước sự phổ biến của các kịch vản ứng dụng mới như thực tế ảo (VR) và truyền thông nhập vai (Immersive Communication). Trong các ứng dụng này, lưu lượng mạng tỏ ra thay đổi rất lớn. Bằng các dự đoán lưu lượng, có thể nhận dạng trước được các thăng giáng lưu lượng và các giai đoạn đỉnh, nhờ đó cho phép ấn định và lập biểu động các tài nguyên mạng. Chẳng hạn, một số tác giả đã sử dụng mạng neuron hồi quy (RNN: Recurrent Neural Network) để đạt được dự đoán không gian thời gian kết hợp nhằn cải hiện mô hình hóa chính xác các thay đổi lưu lượng

mạng. Một phương pháp được đề xuất dựa trên các mạng tin tưởng sâu cho phép dự đoán hiệu quả các xu thế thay đổi dài hạn lưu lượng mạng. Kết quả chứng minh rằng, so với các mô hình đường trung bình chuyển động tự hồi quy (Autoregressive Moving Avarage) thông thường, bộ nhớ dài-ngắn hạn (LSTM: Long Short-term Memory) vượt trội trong các nhiệm vụ dự báo lưu lượng, đặc biệt là trong xử lý dữ liệu chuỗi thời gian phức tạp.

Trong các phương pháp dự đoán lưu lượng, một BS thông thường cần thu thập dữ liệu từ các vị trí địa lý khác nhau, điều này không thể tránh khỏi đưa vào chi phí truyền thông gián tiếp bổ sung và các rủi do an ninh tiềm năng. Kết quả là, các phương pháp dự đoán dựa trên học liên kết (FL: Fedeerated Learning) đã nổi lên. Phương pháp FeDDA đã được đề xuất, nó cho phép giảm đáng kể trễ và chi phí băng thông gián tiếp liên kết với truyền dẫn dữ liệu vì chỉ phát các thông số mô hình thay vì dữ liệu lưu lượng thô. Phương pháp này không chỉ đảm bảo tính riêng tư mà còn cải thiện hiệu suất xử lý dữ liệu. Một số tác giả đã đề xuất dự đoán liên kết được đánh trọng số theo khoảng cách cho phép nhận được các đặc tính không gian-thời gian tốt hơn và cải thiện dự báo lưu lượng.

Để giảm thời gian dào tạo mô hình, một số tác giả đã đề xuất phương pháp dựa trên học chuyển giao (Transfer Learning) cho phép tăng tốc quá trình đào tạo bằng cách khám phá các tương quan giưã các mẫu lưu lượng trong các vùng khác nhau. Các nghiên cứu khác đã sử dụng các kỹ thuật như chia sẻ dữ liệu và học siêu cấp (Meta-Learning) để tối ưu hóa hơn nữa hiệu suất của các mô hình dự đoán lưu lượng. Trong tương lai, các phương pháp dự đoán lưu lượng mạng sẽ không chỉ chiết xuất các tính năng không gian-thời gian và các phụ thuộc dài hạn, mà còn tăng cường thiết kế trọng lượng nhẹ (Light Weight) và các khả năng bảo vệ tính riêng tư của các mô hình.

11.3.3.2. Bảo tồn năng lượng BS dựa trên AI

Công nghiệp ICT đóng góp trực tiếp khoảng 4% phát xạ khí ga nhà kính (Greenhose Gas), trong đó các mạng truyền thông di động chiếm trên 10% con số này. Để đáp ứng lại các thách thức đặt ra bởi thay đổi khi hâu, Khuyến nghị được ITU phát hành 1470 yêu cầu các nhà mạng di động phải đạt được giảm 45 % các phát xạ carbon từ 2020 đến 2030 để cho phép chuyển đổi công nghiệp đến bền vững (hoạt động liên tục mà không gây hại môi trường).

Tối ưu hóa trong các mạng 6G phô bày một tương quan cao với trễ người dùng. Các nghiên cứu tập trung lên cách thức đạt được tiết kiệm năng lượng mạng bằng cách tắt hiệu quả BS trong khi vẫn đảm bảo hiệu duy trì chất lượng dịch vụ cơ bản. Một số tác giả đề xuất chiến lược tắt BS dựa trên ngưỡng lưu lượng. Trong các giai đoạn đỉnh, phương

pháp này có thể dẫn đến chuyển giao thường xuyên dẫn đến các gián đoạn dịch vụ. Một số tác giả phân loại các trạng thái BS vào bốn chế độ nghủ khác nhau dựa trên thang thời gian kích hoạt của BS. Trong khi các chế độ ngủ sâu có thể giảm đáng kể tiêu thụ năng lượng, thì chúng cũng có thể dẫn đến tăng trễ tỉnh giấc làm ảnh hưởng dịch vụ.

Chiến lược tiết kiệm năng lượng cho BS dựa trên học tăng cường (RL: Reinforcement Learning) giữa tác nhân và môi trường khai thác, thay đổi động các thông số để thích ứng các thay đổi tải dịch vụ và các giao động trong các điều kiện kênh. Một số phương pháp sau được đề xuất. Một sơ đồ quản lý trạng thái BS dựa trên Q Learning (là một thuật toán học tăng cường) được đề xuất, trong đó tiêu thụ năng lượng của BS được điều chỉnh động để đáp ứng các thay đổi tải dựa trên khám phá khoảng thời gian tối ưu của các chế độ nghủ khác nhau. Một số tác giả xem xét nhiễu giữa nhiều BS và thiết kế một phương pháp dựa trên các chiến lược cộng tác. Bằng cách cho phép cộng tác giữa các BS, phương pháp này giảm tiêu thụ năng lượng trên toàn bộ hệ thống. Một số tác giả đề xuất nối tầng học máy tăng cường (Cacscading RL) để tối ưu hóa quá trình quyết định của quản lý người dùng và các trạng thái BS trong các kịch bản nhiều BS, cho phép cải thiện hơn nữa hiệu suất mạng.

Việc đưa vào năng lượng năng lượng tái tạo (Renewable Energy), đặc biệt các nguồn năng lượng sạch, chẳng hạn: điện mặt trời như là nguồn năng cho mạng, có thể cải thiện đáng kể hiệu suất mạng và giảm sự phụ thuộc và các lưới điện thông thường. Một số tác giả đã tích hợp RL với mạng neuron dạng đồ thị (Graph Neuron Network) để tối ưu hóa chuyển đổi năng lượng và phát xạ carbon trong các mạng không dây bằng cách khai thác công nghệ năng lượng mặt trời quang điện. Các nỗ lực trong các mạng không dây bằng sẽ được tích hợp sâu với sử dụng năng lượng tái tạo để cung cấp các giải pháp bền vững hơn.

11.3.3.3. Tối ưu hóa các thông số mạng dựa trên AI

Thiết lập hợp lý các thông số mạng là tối cần thiết, các thông số này quyết định thiết kế, tiêu thụ năng lượng và hiệu suất của mạng 6G, và phải được điều chỉnh với phân phối thời gian thực của các người dùng và môi trường điện từ trường. Các thông số mạng không dây có thể được phân loại vào hai kiểu: các thông số BS và các thông số ngưỡng. Các thông số BS bao gồm các chức năng chuyển giao cơ sở như tìm gọi (Paging) và cắt lát (Slicing). Các thông số ngưỡng thiết lập cấu hình ngưỡng cho chọn ô, đưa ra quyết định sự kiện và chuyển giao giữa các ô lân cận. Ngoài ra quy hoạch tài nguyên phổ tần trong các hệ thống truyền thông không dây cũng là một thành phần tối cần thiết của tối ưu hóa thông số. Các nghiên cứu cho thấy rằng phổ tần thường không được sử dụng hết và các công nghệ AI cho đã chứng minh rằng tiềm năng rất lớn trong vô tuyến nhận thức.

AI có thể giám sát, loại bỏ nhiễu, dự đoán xu thế sử dụng phổ tần và cải thiện đáng kể sử dụng các tài nguyển phổ tần. Tối ưu hóa các thông số truyền thống dựa trên các kiểm tra trên xe ô tô và các kinh nghiệm chuyên gia. Khi các nhà mạng xác định các khiếm khuyết đáng kể trong chất lượng dịch vụ truyền thông, họ sẽ triển khai các xe ô tô kiểm tra để thu thập và báo cáo các số đo hiệu suất mạng, như thông lượng mạng, trễ truyền dẫn video/audio. Sa đó họ tạo lập biểu đồ phân tán (Scatter Plot: một công cụ trực quan hóa dữ liệu sử dụng tọa độ Decartes để thể hiện mối quan hệ giữa hai biến định lượng). Cuối cùng kinh nghiệm chuyên gia được sử dụng để phân thích và tối ưu hóa các thông số mạng. Do số lượng các anten 6G tăng đáng keerkeenh không còn là một cấu trúc tổn hao mạch một chiều, mà được mở rộng vào một phổ công suất N chiều. Hiệu quả của điều chỉnh thông sô dựa trên chuyên gia là thấp, dẫn đến khó khăn thay đổi hàng tỷ thông số mạng. Ngoài ra, các phương pháp điều chỉnh dựa trên kinh nghiệm thông thường thường dựa trên các quy tắc cố hữu dẫn đến khó khăn nắm bắt đầy đủ và thích ứng các quan hệ phức tạp trong mạng. Bảng 11.3 cho thấy các thông số khác nhau trong các mạng hiện thời.

Bảng 11.3. Các thông số khác nhau trong các mạng hiện thời

Kiểu thông số	Tên thông số	Lĩnh vực áp dụng
Thông số thiết bị đầu cuối	Tỷ lệ lỗi khối Thông tin lưu lượng Báo cáo đo đạc	Tối ưu hóa lập lịch Giám sát lưu lượng Mã hóa thích ứng
Thông số giao diện vô tuyến	Trễ thời gian đa đường Nhiễu giữa các ô Thông tin trạng thái kênh	Lập mô hình kênh Loại bỏ nhiễu Dự báo trạng thái kênh
Thông số CN	Nhật ký hoạt động Cấu hình topo mạng Tiêu thụ năng lượng mạng	Xử lý lỗi Cân bằng tải Tiết kiệm năng lương
Thông số dịch vụ	Thời gian trực tuyến Giao thức vận tải Lịch sử tiêu thụ	Dịch vụ được cá nhân hóa Điều khiển truyền dẫn Giám sát dịch vụ

.

Các mô hình AI đã chứng tỏ các ưu điểm không thể dự đoán trước trong điều chỉnh thông số quy mô lớn, khớp phi tuyến tính (Nonlinear Fitting) và tối ưu hóa thời gian thực.Trong ngữ cảnh của kiến trúc rắc rối có nhiều chi tiết và các các đặc điểm phi tuyến tính của các mạng 6G, công nghệ AI có thể đạt được hiểu biết sâu hơn về các đặc điểm phi tuyến tính của mạng bằng cách học và phân tích chi tiết dữ liệu mạng rộng lớn và tối ưu hóa các thông số dựa trên dữ liệu và kịch bản thời gian thực. Học tăng cường sâu (DRL:Deep Reinforcement Learning) đã được triển khaiu cho tối ưu hóa các thông số mạng. Cụ thể,

từng số đo hiệu suất trong mạng truyền thông hiện thời được xem là một trạng thái RL riêng biệt và các hoạt động điều chỉnh được thực hiện trên các thông số mạng được chỉ định như là các hành động khác nhau. Trong thủ tục này, bằng cách thích ứng tức thời hành vi của mô hình RL trên các trạng thái khác nhau, mô hình sẽ nhận được thông tin phản hồi từ môi trường để đánh giá giá trị của từng hành động. Mô hình thường tăng cường các hành động này bằng cách tạo ra các phần thưởng cao hơn.

11.3.3.4. Quản lý di động dựa trên AI

Ngay từ 2015, 3GPP đã bắt đầu nghiên cứu làm sao tích hợp công nghệ AI vào mạng 5G. Chẳng hạn,các nhóm công tác 3GPP SA2 và SA5 đã khởi động các đề án về "các yếu tố chp phép tự động mạng". Sử dụng các công nghệ như DL, AI có thể dự đoán các hành vi của người dùng, bao gồm tốc độ và phương di động, bằng cách đó cho phép chọn và chuyển giao chủ động giữa các ô để giảm chi phí báo hiệu. Trong các kịch bản mạng siêu dày đặc, các thay đổi ô thường xuyên có thể dẫn đến tăng chi phí gián tiếp và các gián đoạn kết nối gây ảnh hưởng tiêu cực QoS. Một số tác giả đã đề xuất RNN dựa trên GRU (Gated Recurrent Unit: đơn vị hồi quy có cổng) để xử lý thích ứng các quyết định chuyển giao nhờ vậy giảm đáng kể trễ và cải thiện dung lượng mạng. Một số tác giả đề xuất một phương pháp dự đoán vị trí mới của các xe ô tô với sử dụng AI trong các kịch bản mạng siêu dầy đặc, phương pháp này cho phép chọn BS dựa trên các vị trí được dự đoán nhờ vậy giảm tần suất chuyển giao, tỷ lệ thất bại và tiết kiệm chi phí báo hiệu chuyển giao. Điều này chứng minh rằng quản lý di động dựa trên AI là thiết thực.

Mặt khác thông qua RL và các kỹ thuật khác, AI có thể tối ưu hóa động các chiến lược quản lý di động dựa trên môi trường mạng và hành vi người dùng, xử lý các môi trường mạng tốt hơn. Vì các mạng tại các tần số cao hơn đòi hỏi đòi hỏi triển khai BS dầy đặc hơn, một số tác giả xử lý vấn đề giảm dung lượng mạng do chuyển giao thường xuyên bằng đề xuất giải thuật mạng Q sâu đa tác nhân (Multi-Agent DQN: Multi-Agent Deep Q-Nework) và tập trung, cả hai giải thuật đều có thể tìm được ấn định người dùng-BS gần tối ưu. Giải thuật DQN giảm đáng kể hay thậm chí hoàn toàn tránh được các chuyển giao ping-pông (chuyển giao bị đập đập lại) so với giải thuật thông thường dựa trên SINR (Signal Interference plus Noise Ratio: tỷ số tín hiệu trên nhiễu công tạp âm). Một số tác giả đề xuất một khung dựa trên ML để tối ưu hóa chuyển giao gữa các băng dưới 6 GHz và sóng mi-li-met trong các mạng xe cộ, bằng cách dụng RL để cho phép xe cộ dự đoán phà khám phá mục tiêu chuyển giao theo thời gian thực. Phương pháp này giảm trễ chuyển giao, cải thiện kết nối trong các mạng xe cộ, đạt được chuyển giao liền mạch thậm trí trong các kịch bản di động cao và tăng cường trải nghiệm người dùng.

11.3.3.5. Điều phối dựa trên AI

Với hàng trăm tỷ thiết bị được kết nối, dữ liệu được tạo ra không ngừng tăng trưởng bùng nổ. Chỉ đơn giản tăng cường các khả năng truyền thông là không đủ để đáp ứng các yêu cầu xử lý dữ liệu thời gian thực trong các kịch bản ứng dụng điều khiển. Bằng cách khai thác các khả năng truyền thông và tính toán tại biên mạng và tại phía thiết bị, toàn bộ quá trình thu thập, xử lý, phân tích dữ liệu và đưa ra quyết định được thực hiện gần phía thiết bị hơn nhờ vậy tránh được nhược điểm trễ gây ra bởi nghẽn trong CN.

Lập lịch kết hợp các tài nguyên truyền thông và tính toán được thiết lập trên RL đạt được ấn định hợp tác và tối ưu các tài nguyên truyền thông và tính toán thông qua tương tác và học giữa tác nhân và môi trường mạng 6G. Đối diện với các khó khăn ấn định tài nguyên trong các mạng tích hợp không gian-mặt đất, một số tác giả đã đề xuất một giải thuật ấn định tài nguyên mạng ảo nhận thức băng thông (Bandwidth Aware Virtual Resource Allocation) được RL hỗ trợ. Giải thuật này sử dụng RL với một mạng chính sách để nhúng nút, ưu tiên xử lý các yêu cầu băng thông cao để đáp ứng các yêu cầu băng thông khắt khe của người dùng. Với giả thiết các vấn đề liên quan tồn tại trong các thiết bị di động và hệ thống truyền thông 6G, được thúc đẩy bởi khái niệm nhúng mạng ảo, một số tác giả đã tiên phong đưa ra một giải thuật lập lịch tài nguyên động nhận thức đa đối tượng (Multi-Objective Aware Dynamic Resource Scheduling Algorithm) được thiết kế cho các mạng tính toán đa lớp nhằm tăng tính linh hoạt tài nguyên. Họ đã thiết kế một mạng lập lịch được hỗ trợ bởi DRL và tinh chế quá tình chế quá trình học, nhờ vậy đưa ra một giải pháp bền vững cho các chiến lược lập lịch tài nguyên. Liên quan đến tình huống mà áp dụng các mạng truy nhập vô tuyến mở (O-RAN) trong điều khiển và tối ưu hóa các chức năng của mạng truy nhập không dây trong các mạng 6G chưa được khám phá rộng rãi, một số tác giả đã phát triển các giải thuật phức tạp thấp dựa trên các phương pháp chẳng hạn như RL để xử lý các vấn đề kết hợp tối ưu hóa phân chia và ấn định lưu lượng. Nỗ lực này cung cấp các hiểu biết sâu sắc cho thực hiện một mạng tự động hóa hoàn toàn với điều khiển và tính linh hoạt được tăng cường.

Để xử lý các thách thức của các yêu cầu tài nguyên tính toán và ấn định tài nguyên thông minh trong các dịch IoT của mạng 6G, một số tác giả đề xuất một giải pháp hiệu quả dựa trên DRL cho mở rộng tài nguyên IoT và sắp xếp dịch vụ và kiểm tra hiệu quả của ấn định tài nguyên tự quản đa ứng dụng thông qua mô phỏng tập dữ liệu. Một số tác giả đề xuất phương pháp DRL đa nhiệm (Multi-Task DRL) dựa trên các mạng tính toán đồ thị bằng cách đưa vào một cơ chế cắt lớp mạng và định tuyến kết hợp nhằm đạt được tính bền vững trong các môi trường mạng khác nhau. Một số tác giả thiết lập khung DRL không dựa trên mô hình (model-free) bởi một cấu trúc phân cấp tích hợp gradient chính sách tất định sâu (DDPG: Deep Deterministic Policy Gradient) có thay đổi và DQN kép

để hiện thực hóa một chiến lược cắt mảnh RAN với độ chi tiết hai lớp cho cực đại hóa QoS và hiệu suất phổ của lát cắt. Ngoài ra, một sơ đồ ấn định phổ động cũng dược được đề xuất, nó sử dụng các ưu điểm của DQN và giảm sự bùng nổ trạng thái bởi một khung phần thưởng và phạt đến ấn định động các khối tài nguyên cha được ấn định và nhận được giải pháp tối ưu Pareto cho các vấn đề con thông qua phân tích Chebychev (Chebychev Decomposition).

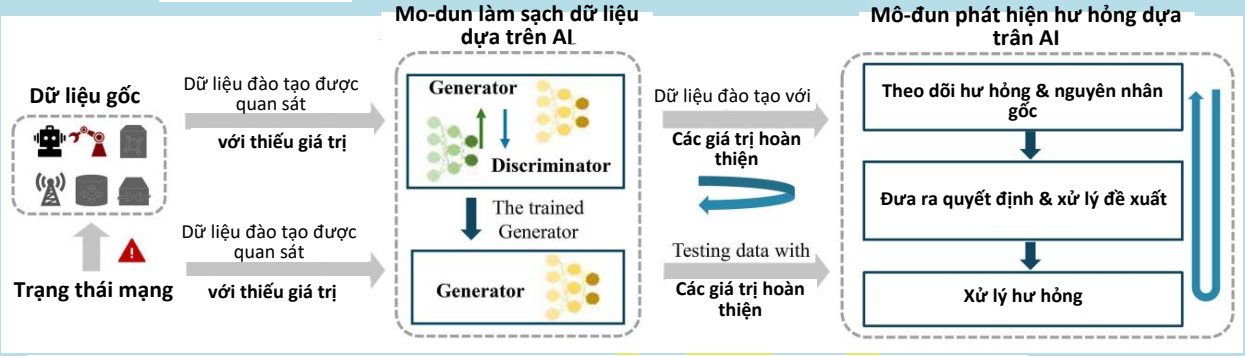
Điều phối dựa trên AI đối mặt nhiều thách thức và các cơ hội trong tương lai. Chẳng hạn, cần nhanh chóng cải thiện các giải thuật DRL để đối phó các mô trường có khả năng thay đổi. Cần ra sức tăng cường sự thích ứng với các kịch bản phức tạp để đáp ứng yêu cầu tài nguyên đa dạng trong các kich bản đặc biệt của các nghành công nghiệp khác nhau, cải thiện đánh giá hiệu suất và tăng cường tính tương tác của giải thuật.

11.3.3.6. Nnận thức tình trạng và phát hiện hư hỏng dựa trên AI

Công nghệ nhận thức tình trạng ban tặng cho các mạng truyền thông khả năng giám sát, dự đoán và các khả năng đáp ứng thời gian thực. Các ứng dụng của nó bao gồm quản lý và tối ưu hóa, an ninh mạng cũng như phát hiện và dự báo hư hỏng. Nhận thức tình trạng thu thập, lưu trữ và phân tích các khối lượng lớn dữ liệu mạng thông qua các nền tảng dữ liệu lớn và các kỹ thuật DL, qua đó cung cấp tình báo mối đe dọa (Threat Intellegence) chính xác và kiểm tra lưu lượng toàn diện. Công nghệ này sử dụng DL để nhận dạng chính xác các trạng thái của các phần tử khác nhau bao gồm lưu lượng mạng, các nhật ký và các chỉ số hiệu suất quan trọng (KPI), dự đoán các xu thế phát triển mạng và chính thức hóa các chiến lược đáp ứng chính xác thông qua thông tin phân tích thu nhận được và các trạng thái được dự đoán để thực hiện chuyển từ “phòng vệ thụ động” sang “phòng vệ chủ động”. Sự thể hiện mối đe dọa và suy luận nhân quả (Causal Reasoning) có thể nhận dạng và suy luận nguồn gốc của các tấn công, cho phép phòng ngừa sự tái diễn các sự kiện an ninh tương tự.

Phân tích hư hỏng mạng cấu thành một trong các phần tử thiết yếu của O&M. Vì số lượng các thiết bị tăng và quy mô mạng mở rộng, số lượng các cảnh báo mạng tăng bùng nổ và quan hệ giữa chúng cực kỳ phức tạp. Áp dụng AI để xây dựng một mô hình phân tích phát hiện hư hỏng cho ta mô hình có khả năng phân tích dữ liệu đa chiều cho phép nó phục vụ như là một cộng cụ thông minh và hiệu quả hơn cho phân tích hư hỏng. Bằng cách tiến hành phân tích hợp nhất các dữ liệu này, hệ thống có thể nhận dạng hiệu qủa các vấn đề phổ biến trong mạng và thực hiện chuẩn đoán hư hỏng chính xác. Các kết quả nhận được sau xử lý hư hỏng sẽ được sử dụng cho phản hồi để cập nhật và tối ưu hóa mô hình chẩn đoán, qua đó cho phép hệ thống hoạt động hiệu quả trong môi trường mạng và

các yêu cầu kinh doanh thường xuyên thay đổi. Các luồng côngm việc cụ thể này được minh họa trong hình 11.9.



Generator: bộ tạo sinh, Discriminator: bộ phân biệt, The trained Generator: bô tạo sinh được đào tạo

Hình 11.9. Phát hiện hư hỏng dựa trên AI

Bằng cách tích hợp công nghệ AI, ta có thể thiết lập phân tích hư hỏng mạng có tệ thống, thông minh và sơ đồ đào tạo ngược để có thể cung cấp các khả năng phát hiện hư hỏng tự động, chính xác, linh hoạt và theo dõi ngược, liên tục tăng cường hiệu quả chẩn đoán và xử lý. Trong khi đó, nó cũng có thể thích ứng các thay đổi trong môi trường mạng và các yêu cầu kinh doanh.

11.4. MẠNG CHO AI (NE4AI)

Một chuỗi các trường hợp sử dụng và các kịch bản AI đã được phát triển và một phần công việc tiêu chuẩn hóa đã được thực hiện dưới sự lãnh đạo của 3GPP. Điều này dẫn đến các chức năng AI "plug-in" (trình cắm) mức mô-đun khác nhau trong các mạng 5G. Mặc dù cách tiếp cận này thuận tiện, nó vẫn thiếu các xem xét tính hệ thống và tính tương tác dẫn đến hạn chế tính khả định cỡ khả năng AI và khả năng tổng quát hóa mô hình.

Các công nghệ then chốt như các các kiến trúc gốc AI, công nghệ thông tin được định nghĩa phần mềm và ảo hóa dịch vụ đã tiến bộ đáng kể trong 5G (chẳn hạn: kiến trúc hỗ trợ CN dựa trên dịch vụ và triển khai các chức năng CN dựa trên đám mây). 3GPP cũng đã chuẩn hóa NWDAF như là một phần tử hỗ trợ AI trong CN. Hỗ trợ này bao gồm các cơ chế và các quá trình liên quan, trong khi đó đưa vào các mô hình hoạt động AI được chuẩn hóa. Tuy nhiên các tiến bộ này vẫn bị ràng buộc bởi một số nhân tố bao gồm sự chín muồi công nghệ, các mối lo ngại về an ninh và sự phức tạp của khai thác và bảo dưỡng (O&M) hệ thống. Vì vậy, kiến trúc mạng truy nhập hiện thời phần lớn duy trì

"thiết kế BS chuyên môn hóa truyền thông tách biệt" truyền thống. Trong cấu trúc này các BS được thiết kế riêng cho các dịch vụ truyền thông, thiếu đi sự linh hoạt giao diện với các dịch vụ mới nổi dựa trên AI . Từ góc nhìn dịch vụ AI từ đầu đến đầu cuối, không có định nghĩa rõ ràng các phần tử chức năng AI tại phía mạng truy nhập. Tích hợp và phối hợp các chức năng dịch vụ AI giữa CN và mạng truy nhập vẫn chư được xử lý. Sự không kết nối này gây cản trở kiến trúc mạng hiện thời hỗ trợ tầm nhìn trí tuệ được hình dung trong 6G.

Để hỗ trợ "trí tuệ gốc" và đạt được trí tuệ tổng quát mọi nơi, mạng 6G đòi hỏi các đổi mới đột phá trong kiến trúc để cho phép nhúng nội sinh các khả năng AI trong mạng. Điều này miêu tả khái niệm lõi giai đoạn II (nghĩa là NET4AI) trong tích hợp sâu 6G và AI, báo hiệu sự chuyển dịch từ một mạng truyền thông chỉ cung cấp các dịch vụ kết nối sạng một mô hình mạng mới cung cấp một mảng các đa dạng các dịch vụ bao gồm kết nối, sức mạnh tính toán, dữ liệu và các giải thuật. Mô hình mới này sẽ mở rộng CN, các mạng truy nhập và các đầu cuối, để cung cấp hỗ trợ AI toàn diện.

11.4.1. Kiến trúc NET4AI trên 6G

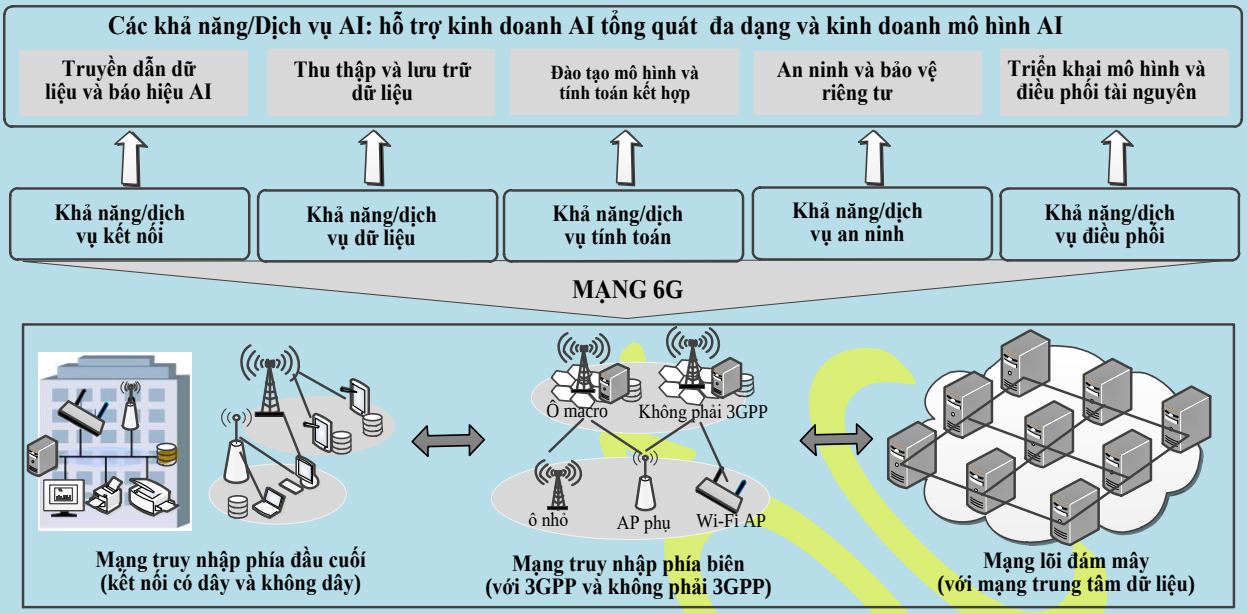
Chine Mobile đề xuất thiết kế một hệ thống 6G như là một cách hệ thống vượt qua giới hạn kết nối trong công nghiệp được tập trung trên khái niệm "ba lớp và sáu mặt phẳng" (Three Layers and Three Planes). "ba lớp" đề cập đến *lớp vật lý*, *lớp mạng* và *lớp ứng dụng* và *dịch vụ*. "sáu mặt phẳng" bao gồm *mặt phẳng truyền thông, mặt phẳng dữ liệu , mặt phẳng tính toán, mặt phẳng trí tuệ, mặt phẳng an ninh, mặt phẳng quản lý và điều phối*. Năm mặt phẳng đầu tiên là là các mặt phẳng chức năng độc lập được tạo thành bởi tách riêng mạng 6G từ đầu đến cuối (End to End) để cung cấp các khả năng và các dịch vụ. Mặt phẳng quản lý và điều phối thực hiện kết hợp linh hoạt các chức năng trên các lớp và và các mặt phẳng, thực hiện quản lý và lập lịch các tài nguyên, và các dạng thức của một chuỗi dịch vụ hoàn chỉnh. Cấu trúc này tập trung lên các phần tử khả năng đa chiều trong mạng. China Telecom đề xuất một kiến trúc mạng 6G hiển diện mọi nơi và siêu hội tụ, được xây dựng trên khái niệm "ba lớp và ba khu vực" (Three Layers and three Sectors). "ba lớp" bao gồm *lớp hạ tầng*, lớp chức năng mạng và lớp hệ điều hành, còn "ba lĩnh vực" bao gồm *tích hợp dữ liệu (Data Integration)*, *đơn giản thông minh (Intelligence Simlicity)* và *sự đáng tin cậy (Trustworthiness)*. Kiến trúc này sử dụng lĩnh vực đơn giản thông minh để booj não thông minh, các kết nối và các phương tiện trên toán bộ ba lớp, nhừ vậy cho phép tính bao trùm (Inclusiveness). China Unicom mường tượng một mạng 6G tương lai *thông minh, hội tụ, xanh* và *đáng tin cậy*. Kiến trúc của nó được cấu trúc theo chiều thẳng đứng vào các lớp truyền thông dựa trên mặt đất, dựa trên

không trung, dựa trên không gian và theo chiều ngang vào ba miền: miền *tài nguyên cảm biến (Sensing Resource), miền điều khiển chức năng (Func tion Control)* và *miền ứng dụng dịch vụ (Service Application)*. Ba lớp và ba miền này được liên kết với nhau bởi hai chuỗi: *trí thông minh gốc, an ninh và tin cậy*.

Huawei giới thiệu một kiến trúc AI lấy nhiệm vụ làm trung tâm, khẳng định rằng các mạng 6G sẽ đưa vào các chiều tài nguyên mới và hỗ trợ phối hợp các tài nguyên đa chiều. Thiết kế này cho phép các mạng 6G hỗ trợ AI một cách tự nhiên và đạt được chuyển đổi từ trí thông minh lấy phiên là trung tâm vào các hoạt động lấy nhiệm vụ làm trun tâm. CICT đưa ra khái niệm tích hợp đa mạng và trí thông minh rộng khắp với mường tượng rằng 6G sẽ thực hiện bao phủ sâu ba chiều, phát triển các mạng vùng cơ thể (Body Area Network) tập trung lên con người, tạo ra vùng phủ đa lớp với cả chiều rộng và chiều sâu và đào tạo hệ thống thống minh rộng khắp với tích hợp nhiều mạng. OPTO đề xuất rằng 6G sẽ định nghĩa lại tương tác con người và AI, bằng cách làm cho AI như là một hạ tầng rộng khắp. Hệ thống 6G sẽ nhấn mạnh sự phối hợp trí thông minh, hiệu suất và tính linh hoạt, và tích hợp sâu AI vào mặt phẳng người dùng, mặt phẳng điều khiển và mặt phẳng chức năng. Hơn nữa, có căn cứ và chính thức, nhóm xúc tiến IMT2030 cho 6G (IMT2030 Promotion Group) tai Trung Quốc mường tượng kiến trúc 6G như là mộ nền tảng mở và sáng tạo cho các dịch vụ thông tin, cung cấp các khả năng vượt qua giới hạn kết nối. Các khả năng này bao gồm các mạng có năng lực tính toán, an ninh được tin cậy, cảm biến, các dịch vụ dữ liệu và tự quản thông minh mạng dựa trên AI.

Các nghiên cứu đã đạt được hiểu biết cốt lõi của thiết kế các kiến trúc mạng 6G nhấn mạnh “cấu trúc phân tán”, “trí tuệ nội sinh” và “đơn giản tích hợp”. Các mạng tương lai sẽ phát triển vào các nền tảng tiên tiến được tích hợp vượt trên các dịch vụ kết nối viễn thông có thể cung cấp các dịch vụ đa chiều và các khả năng tương ứngvượt ra ngoài truyền thông. Chúng sẽ đáp ứng các kinh doanh đa dạng bằng cách cung cấp các dịch vụ quản lý vòng đời từ đầu đến cuối trong suốt quá trình xử lý của chúng. Ngoài ra khả năng AI của mạng có thể được để mở cho thế giới bên ngoài, đồng thời khả năng AI bên ngoài có thể được đưa vào mạng để có thể tìm kiếm nguồn lực khả năng AI từ đám đông (Crowndsourcing).

Chúng ta sẽ trình bày chi tiết về kiến trúc NET4AI, nghĩa là sự hỗ trợ được cung cấp bởi năm khả năng/ dịch vụ khác trong mạng (nghĩa là: truyền thông, kết nối, tính toán dữ liệu và quản lý và điều phối) với nhấn mạnh tương tác của chúng với AI. NET4AI sẽ cung cấp hỗ trợ kiến trúc cho thực hiện 6G AI gốc để đạt được tích hợp sâu giữa mạng và AI như minh họa trong hình 11. 10.



Hình 11.10. Bểu đồ kiến trúc của NET4AI. Mạng 6G về mặt vật lý bao gồm: CN đám mây (gồm các trung tầm dữ liệu tập trung/tính toán), mạng truy nhập phía biên (gồm các phương pháp kết nối 3GPP và không phải 3GPP), mạng truy nhập phía thiết bị đầu cuối (gồm các phương pháp kết nối có dây và không dây) được mô tả ở phần dưới của hình vẽ. NET4AI thể hiện vai trò ánh xạ và hỗ trợ kết nối truyền thông, tính toán dữ liệu, an ninh và quản lý và điều phối trong AI, trong số sáu khả năng và dịch vụ (nghĩa là: kết nối truyền thông, dữ liệu, tính toán, an ninh, AI và quản lý và điều phối) được trừu tượng hóa từ các dịch vụ chức năng của mạng 6G được chỉ thị bởi các mũi tên trong phần dưới hình vẽ.

11.4.2. Kết nối cho AI (Connection for AI)

Trong 6G, mặt phẳng điều khiển và mặt phẳng người dũng sẽ được tăng cường hơn nữa để nhắm đến bản chất định hướng theo dịch vụ và mở của các chức năng mạng, và cung cấp hỗ trợ các dịch vụ AI mới nổi. Một mặt 6G cần tăng cường và tối ưu hóa hơn nữa quản lý dịch vụ và giao tiếp các chức năng mặt phẳng điều khiển hiện có để giảm sự phức tạp của các chức năng mạng và các quá trình tương tác báo hiệu. Ngoài ra, 6G se mở rộng mặt phẳng điều khiển thông thường bằng cách tích hợp quản lý điều khiển truy nhập không gian-mặt đất rộng khắp để cho phép các chức năng mới và cung cấp quản lý phiên và điều khiển chính sách định hướng theo dịch vụ trên trên truyền thông, tính toán, dữ liệu, và các phần tử quan trọng khác. Mặt khác, mặt phẳng người dùng sẽ được tăng cường bằng các tăng cường hay đưa vào các giao thức chuyển tiếp của mặt phẳng người dùng mới, tổng quát và linh hoạt hơn để có thể thực hiện chuển tiếp và xử lý dịch vụ nhanh dựa trên truyền thông, tính toán và các phần thử khác. Sử dụng các công nghệ vi

dịch vụ và hỗ trợ của phần cứng thế hệ sau, chẳng hạn: các chip khả lập trình hiệu suất cao, các công nghệ đám mây gốc và ngôn ngữ tiên tiến P4 (Programming, Protocol-Independent Packet Processors: các bộ xử lý gói lập trình độc lập giao thức), các chức năng mặt phẳng người dùng (UPF) sẽ được cấu trúc lại vào các mô-đun định hướng theo dịch vụ. Đổi mới linh hoạt và phối hợp hiệu quả giữa giữa các mô-đun chức năng sẽ phá vỡ các rằng buộc của các giao thức phân lớp thông thường, nhờ vậy tăng cường hiệu quả chuyển tiếp dữ liệu và sử dụng băng thông.

Khi xem xét nguyên tắc phân tách mặt phẳng điều khiển, các kiểu dữ liệu AI định hướng theo dịch vụ được mang bởi các kết nối viễn thông có thể một cách tự nhiên được phân loại thành hai nhóm chính: báo hiệu AI và dữ liệu AI. Kết nối cho AI bao hàm cung cấp hỗ trợ kết nối cho báo hiệu AI và dữ liệu AI.

- Bán hiệu AI. Báo hiệu AI được sử dụng để phát đi các bản tin điều khiển liên quan đến AI, chẳng hạn: các bản tin yêu cầu /trả lời dịch vụ AI liên quan đến năng lực tính toán cho phân tích AI, v.v… Nó cũng có thể bao hàm cả thông tin bổ sung cho phân tích AI phối hợp giữa nhiều phần tử.
- Dữ liệu AI. Dữ liệu AI được sử dụng để chuyển giao dữ liệu AI đầu vào một giải thuật AI, thường được đề cập như là các vectơ để yêu cầu mô hình/giải thuật AI. Nó cũng được sử dụng để phát đi các mô hình AI, chẳng hạn phát đi một mạng neuron, thường có nghĩa là phát đi các trọng số thông số (chũng có thể được trình bày như là các vectơ) và cấu trúc mạng của nó (có thể được trình bày như là các vô hướng (Scalar)). Nó có thể được sử dụng hơn nữa cho phát đi dữ liệu đào tạo AI. Dữ liệu này thường rất lớn và dành riêng cho giai đoạn đào tạo các mô hình trí tuệ.

Đối với các kịch bản kinh doanh AI và các yêu cầu hiệu suất kết nối liên quan, 3GPP TS 26.261 cũng đã có một số kết quả cho các hệ thống 5G, chủ yếu ảnh hưởng lên các số đo hiệu suất kết nối thống như thông lượng, trễ và độ tin cậy (xem chương 9). Nói một cách khác tiêu chuẩn của 3GPP hiện thời chỉ phân tích ảnh hường của hệ thống 5G (chủ yếu là phần CN) liên quan đến làm sao nó hỗ trợ các ứng dụng/dịch vụ AI từ các góc nhìn chức năng và hiệu suất. Từ góc nhìn phân tích xử lý từ đầu đến cuối, báo hiệu AI và dữ liệu AI tham gia và các dịch vụ/ứng dụng cần truyền dẫn qua giao diện vô tuyến. Tuy nhiên trong các tiêu chuẩn và các kiến trúc hiện thời hiện thời, khong có mô-đun nào chỉ dẫn quá trình truyền dẫn này hay tiêu chuẩn hóa nội dung truyền dẫn.

11.4.3. Dữ liệu cho AI (Data for AI)

Mạng 6G cần một khả năng dữ liệu (Data Capacity) mới khác với mặt phẳng người dùng thông thường. khả năng này phải tuân thủ quy định dữ liệu và các yêu cầu giám sát, trong khi vẫn tăng cường hiệu quả phân tích và xử lý dữ liệu để giải quyết các thách thức quản lý và hỗ trợ các dịch vụ AI. Khả năng dữ liệu này hỗ trợ thực hiện thu thập, làm sạch, lưu trữ dữ liệu và cho phép quản lý toàn bộ vòng đời của các xử lý dịch vụ dữ liệu, trong khi vẫn đảm bảo an ninh và tính riêng tư của dữ liệu. Bằng cách sử dụng các mô hình lưu trữ và xử lý dữ liệu lai ghép dựa trên kết hợp các phương pháp tập trung và phân tán, mạng 6G có thể giải quyết lịnh hoạt các yêu cầu kịch bản khác nhau: (i) cho dữ liệu nhạy cảm hay mức riêng tư cao, xử lý và lưu trữ cục bộ có thể được thực hiện tại RAN/CN và phía thiết bị đầu cuối để đảm bảo không dò rỉ dữ liệu, (ii) mặt khác, dữ liệu có phân tích toàn cục và toàn diện được xử lý và lưu trữ tập trung trong trung tâm dữ liệu. Thông qua hợp tác chặt chẽ giữa trung tâm dữ liệu và các nút cuối, 6G có thể quản lý hiệu quả các khối lượng dữ liệu lớn bắt nguồn từ các lĩnh vực kinh doanh và công nghệ khác nhau, trong khi vẫn đảm bảo bảo vệ an toàn tính riêng tư.

Khả năng dữ liệu là một nền tảng chắc chắn cho các dịch vụ AI thông minh nội sinh. Một số tập dữ liệu có thể truy cập công khai đã được sử dụng rộng rãi như DeepMIMO và RadioML. Tập dữ liệu DeepMIMO (dựa trên các mô phỏng dò tia (Race-Tracing) cho các môi trường truyền thông vô tuyến thực) cung cấp dữ liệu kênh MIMIO và đã được ứng dụng rộng rãi trong các kịch bản như lập mô hình kênh, tạo búp, định vị người dùng trong các mạng 6G. Ngược lại, tập dữ liệu RadioML bao gồm dữ liệu tín hiệu vô tuyến với nhiều sơ đồ điều chế và chủ yếu được sử dụng cho nhận biết điều chế tín hiệu, phát hiện nhiễu và quản lý phổ tần. Ngoài ra tập dữ liệu "AI/ML trong thách thức 5G" (AI/ML in 5G Challenge) do ITU phát hành bao phủ đa dạng các kịch bản truyền thông trong các mạng 5G, cung cấp hỗ trợ cực kỳ quan trọng cho nghiên cứu về tối ưu hóa, dự đoán lưu lượng và ấn định tài nguyên. Các tập dữ liệu này không chỉ đặt nền tảng vững chắc cho nghiên cứu AI trong lĩnh vực viễn thông mà còn cung cấp các điểm chuẩn thống nhất cho thẩm định và so sánh các hiệu suất của giải thuật.

Dữ liệu cho AI (Data for AI) nhắm đến hỗ trợ hiệu quả thu thập, truyền dẫn, lưu trữ và chia sẽ dữ liệu từ đầu đến cuối cho phép giải quyết làm sao tạo điều kiện, đẩy nhanh và cung cấp an ninh dữ liệu cho các chức năng AI bên trong hoặc bên ngoài mạng 6G. Phụ thuộc vào phạm vi tiềm năng, hỗ trợ dữ liệu được cung cấp bởi dữ liệu cho AI phải bao gồm năm thành phần: (i) thu thập/phân phối dữ liệu. Thành phần này cung cấp cơ chế phát hàng-đăng ký (Publish-Subscribe) cơ sở cho các người tạo dữ liệu và các người tiêu thụ để tăng cường hiệu quả thu thập/phân phối dữ liệu và cung cấp các mẫu cho đào tạo quy mô lớn được yêu cầu bởi các mô hình AI để đạt được các kết quả tối ưu. (ii) An ninh và quyền riêng tư. Thành phần này sử dụng các công nghệ bảo vệ an ninh và quyền riêng

tư để cung cấp các dịch vụ tin cậy chất lượng cao được thiết kế riêng cho các nhu cầu của người dùng và của mạng, để đảm bảo sự bảo vệ quyền riêng tư của người dùng và của mạng, tính bất biến và khả năng truy xuất nguồn gốc. (iii) Phân tích dữ liệu. Thành phần này sử dụng các mô hình AI, các giải thuật, và hiểu biết để cung cấp thông tin thống kê, thông tin chuyển sâu dự đoán, phân tích sự dị thường của mạng và các đề xuất tối ưu hóa để tăng cường kinh nghiệm tiêu thụ dữ liệu cho các chức năng bên trong và bên ngoài và các ứng dụng thông minh liên quan. (iv) Xử lý dữ liệu. Thành phần này cung cấp các công cụ di truyền (Genetic Tool) cho chuyển đổi định dạng, chiết suất tính năng và các nhiệm vụ xử lý khác cho dữ liệu được thu thập, để thực hiện các yêu cầu đào tạo cho dữ liệu đầu vào và tạo điều kiện cho quản lý dữ liệu phức tạp đa chiều. (v) Lưu trữ dữ liệu. Thành phần này lưu trữ và tìm lại dữ liệu đã được thu thập, cung cấp hỗ trợ lưu trữ cho các dịch vụ liên quan đến an ninh và riêng tư dữ liệu, phân tích dữ liệu hay xử lý dữ liệu và thiết lập hơn nữa một nền tảng dữ liệu để hỗ trợ các dịch vụ AI.

Thông qua các chức năng dịch vụ số liệu được chuẩn hóa, có thể thiết lập một xử lý toàn diện trong mạng 6G. Vì các nhà nghiên cứu thường cần xây dựng các tập dữ liệu dành riêng cho các kịch bản ứng dụng đặc thù, dữ liệu cho AI có thể cung cấp hướng dẫn khả thay đổi. Ta xét một quá trình xây dựng tập dữ liệu làm thí dụ. Thứ nhất, các nguồn dữ liệu có thể được chọn lựa từ dữ liệu được thu thập trong các mạng truyền thông thực tế hay dữ liệu mô phỏng được tạo ra bằng cách sử dụng các công cụ mô phỏng có độ trung thực cao (chẳng hạn NS-3, OMNeT++). Số liệu thực tế có thể phản ảnh các kịch bản thực tế, nhưng có thể kéo theo các vấn đề an ninh và quyền riêng tư, trong khi đó dữ liệu mô phỏng cung cấp sự linh hoạt và có thể được tùy chỉnh cho các kịch bản đặc thù tùy theo các yêu cầu cụ thể. Thứ hai, xử lý và gán nhãn dữ liệu là hai bước thiết yếu. Dữ liệu truyền thông thường chứa một lượng lớn tạp âm, vì thế cần các hoạt động xử lý như làm sách và chuẩn hóa, và gán nhãn cẩn thận dựa trên các mục tiêu nghiên cứu (chẳng hạn: dự đoán trạng thái kênh hay phát hiện nhiễu). Khi sử dụng các dữ liệu thực tế bắt buộc phải tuân thủ chặt chẽ các quy định bảo vệ quyền riêng tư dữ liệu. Các kỹ thuật như ẩn danh (Anonymization) hay quyền riêng tư vi sai (Differential Privacy) phải được sử dụng. Cuối cùng nhưng không kém phần quan trọng, các nhà nghiên cứu được khuyến khích tạo ra các tập dữ liệu có thể truy nhập công khai dựa trên cơ sở đề quyền riêng tư được bảo vệ để thúc đẩy nghiên cứu hợp tác và so sánh giải thuật trong lĩnh vực này. Nó được khuyến nghị sử dụng một định dạng và tiêu chuẩn thống nhất để tạo điều kiện chia sẻ dữ liệu và khả năng tái tạo các kết quả nghiên cứu.

11.4.4. Tính toán cho AI (Computation for AI)

Sự phát triển của 6G sẽ đem lại các tiến bộ đáng kể trong tính tóan, bao gồm cảm biến tính toán (Computational Sensing), điều khiển tính toán (Computational Control) và thực

hiện tính toán (Computational Execution). Cảm biến tính toán sẽ thu thập thông tin về các tài nguyên tính toán phân tán rộng khắp trên phạm vi rộng lớn từ các đám mây đến các thiết bị đầu cuối. Điều khiển tính toán sẽ đáp ứng các yêu cầu dịch vụ tính toán bằng cách cung cấp các chiến lược thông minh điều khiển năng lực tính toán có mặt mọi nơi. Thực hiện tính toán (được hướng dẫn bởi các chiến lược) sẽ thực hiện các dịch vụ, như quản lý phiên, điều khiển lưu lượng và thực hiện chính sách.

Các khả năng tính toán phải kết hợp với kết nối truyền thông để tối ưu hóa hiệu quả tiêu thụ năng lượng của các dịch vụ mới nổi như AI. Tích hợp này cần thiết phát triển một môi trường thống nhất để đảm bảo hoạt động liên tục của các dịch vụ tính toánn trên các thiết bị, các nút biên và các đám mây. Môi trường này sẽ cho phép chọn động các nút tính toán biên cho các nhiệm vụ tính toán có xét đến các nhân tố như trễ, băng thông, năng lực tính toán và hiệu quả năng lượng. Khung thống nhất này sẽ cho phép các nhiệm vụ tính tóan chuyển tiếp giữa các đầu cuối, các mạng, các nút mạng và các đám mây. Năng lực tính toán này cùng với các giải thuật AI của các chức năng sẽ phục vụ mạng hay các thiết bị để cải thiện các hoạt động của mạng , nhưng nó cũng sẽ tiềm năng được khai thác thông qua các giaio diện thống nhất để phục vụ các ứng dụng của lớp cao hơn. Năng lực tính toán có thể được phân loại vào ba kiểu: năng lực tính toán được dành riêng cho các phần tử mạng, năng lực tính toán phân tán bên ngoài và năng lực tính toán nội sinh bên trong mạng.

1) **Năng lực tính toán dành riêng cho AI của phần tử mạng (Network Element Dedicated for AI)**. Kiểu năng lực tính toán này thông thường bao gồm các đơn vị tính toán và lưu trữ được cấu thành từ các bộ xử lý chuyên dụng và các thiết bị khả lập trình. Nó phục vụ các phần tử trong các mạng truyền thông di động (chẳng hạn: BS hay CN). Nó chủ yếu được sử dụng cho các dịch vụ ứng dụng AI tùy chỉnh (chẳng hạn: AI4NET) để tăng cường hiệu suất tính toán hoặc tối ưu hóa các hoạt động mạng, như phản hồi CSI, ước tính kênh và tạo búp. Do các hạn chê tài nguyên, nó không thể hỗ trợ các dịch vụ và các ứng dụng AI đòi hỏi các tính toán và đào tạo quy mô lớn. Nó cũng gây thách thức cho hỗ trợ các ứng dụng ba.
2) **Năng lực tính toán phân tán bên ngoài cho AI (External Computation Power for AI).** KIểu năng lực tính toán này thông thường tồn tại trong dạng điện toán biên /MEC (Mobile Edge Combuting: điện toán biên di động) phân tán., chủ yếu sử dụng các đơn vị xử lý trung tâm đa dụng (CPU), các đơn vị xử lý đồ họa hiệu xuất cao (GPU) và các card tăng tốc khả lập trình. Năng lực tính tính toán bên ngoài chuyển dịch các tính toán từ các trung tâm dữ liệu tập trung đến biên của mạng truy nhập để cho phép tối ưu hóa và hỗ trợ các ứng dụng công nghiệp có các yêu cầu tính toán cao và trễ chặt chẽ, như tăng tốc video và các kịch bản thực tế ảo

(VR) và thực tế tăng cường (AR). Do bản chất bên ngoài của mình, nó đòi hỏi quản lý thông qua các chức năng mạng thống nhất dẫn đến đưa vào trễ một lượng trễ nhất định có thể gây ảnh hưởng lên các dịch vụ và các ứng dụng AI rất nhậy cảm trễ. Ngoài ra phương pháp chồng lấn bên ngoài có thể không tối ưu hóa sử dụng tài nguyên gây cản trở xây dựng, triển khai và sử dụng hiệu quả các dịch vụ AI.

3) **Năng lực tính toán gốc mạng phân tán cho AI (Distributed Neiwork-Native Computation Power)**. Trong tầm nhìn của mạng tương lai, mỗi phần tử mạng sẽ sở hữu các khả năng điều khiển, chuyển tiếp và tính toán với vô số nút tính toán được triển khai trên toàn bộ mạng. Khả năng tính toán (được biết như là khả năng tính toán nội sinh) sẽ thúc đẩy phát triển và triển khai AI nội sinh cho phép hợp tác trí tuệ phạm vi rộng được phân tán cho các dịch vụ hợp tác. Nó bù đắp cho các thiếu sót của năng lực tính toán bên ngoài, cho phép đáp ứng tức thời các thay đổi trong di động và các mạng, trong khi vẫn thúc đẩy sự xuất hiện và phát triển của các ứng dụng AI tương lai như XR đám mây nhập vai (Immersive Cloud XR), giao tiếp ba chiều (Hologram Communication), tích hợp cảm giác (Sensory Interconnection), tích hợp cảm biến và truyền thông (Communication-Sensing Integration) và DT (Digital Twin: bản sao số hay song sinh số).

11.4.5. An ninh cho AI (Security for AI)

Một số lỗ hổng an ninh vẫn còn lại sau thết kế các hệ thống 2G-5G và các biện pháp an ninh thông thường thường bị thất bại. Mạng 6G cần thiết lập một hệ thống an ninh tự lực và có tính tin cậy cố hữu tại lõi của hệ thống. Các tiến bộ nhanh chóng của điện toán đam mây, dữ liệu lớn và các công nghệ AI cung cấp hỗ trợ kỹ thuật để xây dựng một hệ thống an ninh mạng 6G. Trong những năm gần đây, an ninh nội sinh yin cậy đã nổi lên như là một cách tiếp cận mới cho 6G, được đặc trưng bởi bốn tính năng: hợp tác, phòng vệ chủ động thông minh, sự đáng tin cậy và bảo vệ quyền riêng tư. Khái niệm an ninh nội sinh có tin cậy cũng được áp dụng cho AI.

1) **Hợp tác mọi nơi (Ubiquitious Collaboration)**. Hợp tác đã trở thành một cách tiếp cận công nghệ để tăng cường các khả năng truyền thông. Hợp tác thông minh mọi nơi sẽ cho phép NET4AI thích ứng với các môi trường động và phức tạp và các kịch bản khác nhau đồng thời cải thiện sự bền vững của mặt phẳng AI để nhận dạng chính xác các tấn công hiểm độc tiềm năng hay các hành vi bất thường. Các phòng vệ an ninh tương lai sẽ dịch chuyển từ bị cách ly đến hợp tác hiệu quả cao để đạt được các các phòng vệ hợp tác cá nhân giữa các thực thể hay các tác nhân thông minh, các phòng vệ liên lớp hay phòng thủ đa lớp (Inter-Layer

Defense) giữa các lớp của giao thức hay các lớp của kiến trúc và các phòng vệ liên miền giữa các miền không gian của mạng.

2) **Phòng vệ tích cực thông minh (Intellegence Proactive Defense)**. Sự triển khai nhanh chóng các mạng 6G và các công nghệ AI tạo ra các cơ hội cho các phòng vệ an ninh thông minh. Phồng vẹ chủ động thông minh sẽ cho phép NET4AI học và thích ứng liên tục đồng thời tự động điều chỉnh các chiến lược phòng vệ dựa trên các thay đổi trong mạng và các mối đe dọa an ninh để kịp thời chống lại các tấn công.
3) **Sự đáng tin cậy (Trustworthiness)**. 6G sẽ phát trên hơn nữa về phía các khả năng đáng tin cây để đạt được một hệ thống an ninh nội sinh toàn diện bao gồm an ninh (chẳng hạn: bảo mật, toàn vẹn và khả dụng), tin cậy, khả năng phục hồi và an toàn. Điều này có thể một hệ sinh thái được tin tưởng cho NET4AI. Ngoài ra, điều khiển truy nhập tin cậy, linh hoạt và xác thực nhận dạng hiệu quả có thể (đến một mức độ nào đó) ngăn chặn truy nhập trái phép ảnh hưởng đến môi trường được tin tưởng nội sinh.
4) **Bảo vệ quyền riêng tư (Pricacy Protection).** Sự tích hợp sâu của AI, dữ liệu lớn và các mạng 6G sẽ làm trầm trọng hơn các thách thức bảo vệ an ninh thông minh. NET4AI phải xử lý dự liệu kinh doanh và cá nhân, đồng thời ngăn chặn dò rỉ thông tin cá nhân. ML phân tán mới nổi, trong khi đạt được trí tuê mọi nơi phân tán và một phần tránh được dò rỉ thông tin cá nhân gây ra bởi đào tạo mô hình AI, vẫn có các rủi do an ninh. Vì thế, NET4AI tích hợp ML phân tán với các phương pháp bảo vệ quyền riêng thư hiện thời và xây dựng một hệ sinh thái dữ liệu an ninh, hiệu quả.

11.4.6. Điều phối cho AI (Orchestration for AI): 6G được xem xét đề xử lý một mức độ tự quản mạng cực cao, nghĩa là khả năng quàn lý và điều phối mạng có thể tiến hành điều phối và lập lịch cho các thành phần chức năng khác nhau của các yêu cầu dịch vụ, nhờ đó cho phép tự tối ưu, tự phát triển và tự chữa lành của mạng.

6G được xem xét đề xử lý một mức độ tự quản mạng cực cao, nghĩa là khả năng quàn lý và điều phối mạng có thể tiến hành điều phối và lập lịch cho các thành phần chức năng khác nhau của các yêu cầu dịch vụ, nhờ đó cho phép tự tối ưu, tự phát triển và tự chữa lành của mạng. Hệ thống 6G sẽ đưa vào các yêu cầu dịch vụ và ứng dụng bên ngoài đồng thời để lộ các tài nguyên và các chức năng đa dạng bên trong. Điều này đòi hỏi luồng chức năng chéo linh hoạt và lập lịch cho các tài nguyên đa chiều, như tính toán, truyền thông và lưu trữ trong mạng để cung cấp các dịch vụ ứng dụng tùy chỉnh và cá nhân hóa dựa trên các công nghệ ảo hóa và vi dịch vụ. Chẳng hạn, các khả năng AI và dữ liệu dược phân tích trong mạng có thể được để mở cho các bên thứ ba để cung cấp sự hỗ trợ cần

thiết , hành vi tìm kiếm nguồn lực từ đám đông (Crowdsourcing) cần thiết để hỗ trợ các chức năng quản lý và điều phối. Hơn nữa, thông qua phân tích và suy luận các tài nguyên tính toán và lưu lượng mạng kết hợp, các gNB được con-ten-nơ hóa (Containerized gNBs: chức năng của các gNB được chia nhỏ thành các vi dịch vụ và đóng gói vào các con-te-nơ) có thể tự động định cỡ tăng hoặc giảm dựa trên tải lưu lượng dự đoán được, cho phép ấn định tài nguyên động và tiết kiệm năng lượng. Vì thế các khả năng quản lý và điều phối liên quan mật thiết một cách hiệu quả và linh hoạt đến tất cả các khả năng, các tài nguyên quản lý, điều phối va các chức năng khác trên tất cả các lớp chức năng của mạng 6G, cho phép đạt được tối ưu hóa kết hợp.

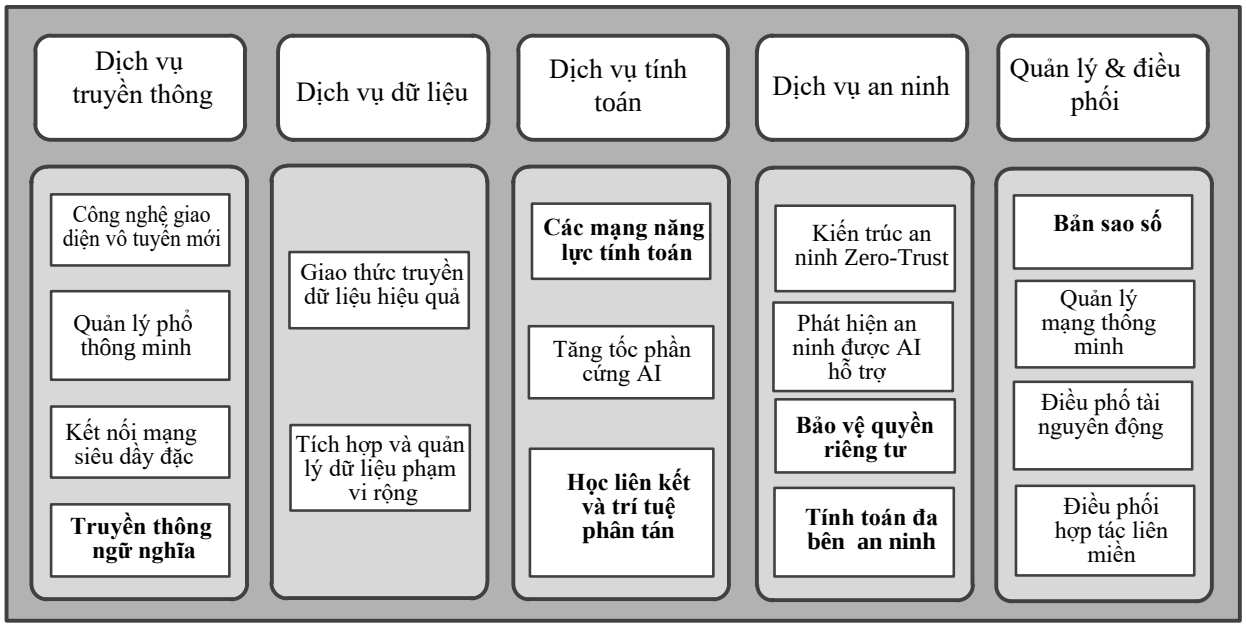
Điều phối cho AI có nghĩa là hỗ trợ hiệu quả cho triển khai, hoạt động và tối ưu hóa các dịch vụ AI trong mạng thông qua điều phối. Nó lập cấu hình tự động và điều phối các tài nguyên trong mạng dựa trên các yêu cầu cụ thể của các dịch vụ AI để đảm bảo các ứng dụng AI nhận được năng lực tính toán, lưu trữ, băng thông mạng cần thiết và các tài nguyên khác. Ngoài ra, khả năng điều phối cho phép giám sát hiệu suất AI thời gian thực bằng cách điều chỉnh động ấn định tài nguyên để đáp ứng các thăng giáng tải dịch vụ, qua đó đảm bảo sự ổn định và hiệu quả trong mạng và cung cấp các đảm bảo mạng vững bền cho các ứng dụng thông minh khác nhau. Đặc biệt, điều này thể hiện trong các nét sau:

1) **Triển khai và ấn định tài nguyên tự động**. Khả năng điều phối đáp ứng nhanh chóng các yêu cầu tài nguyên mạng của các dịch vụ AI thông qua tự động hóa. Trong khi triển khai dịch vụ AI, bộ điều phối phân tích thông minh sự sử dụng tài nguyên mạng hiện thời, tự động lập cấu hình các thiết bị mạng, thiết lập các kết nối và triển khai dịch vụ. Xử lý tự động này tăng cường hiệu quả triển khai, giảm thiểu các sai lỗi và đảm bảo khởi động dịch vụ nhanh chóng và hoạt động ổn đinh.
2) **Tối ưu hóa động và định cỡ đàn hồi**. Khi tất cả các dịch vụ hoạt động, các yêu cầu tài nguyên có thể thăng giáng. Khả năng điều phối có tính năng tối ưu hóa động và giám sát liên tục hiệu suất dịch vụ AI, bao gồm sự sử dụng CPU, tiêu thụ bộ nhớ, băng thông mạng và các số đo quan trọng khác. Khi phát hiện thiếu hụt và tặng dư tài nguyên, hệ thống điều phối tự động điều chỉnh ấn định tài nguyên, chẳng hạn bổ sung thêm các nút tính toán, mở rộng dung lượng lưu trữ hoặc tối ưu hóa băng thông để đáp ứng các yêu cầu động của các dịch vụ. Ngoài ra, hệ thống điều phối hỗ trợ định cỡ đàn hồi, tự động thay đổi kích thước các tài nguyên dựa trên các tải dịch vụ AI để đảm bảo sự ổn định và hiệu quả.
3) **Tự chữa lành hư hỏng và O&M**. Bằng cách tích hợp giám sát, cảnh cáo và các chức năng khắc phục sự cố, bộ điều phối kịp thời nhận diện và định vị các hư hỏng, tự động khởi động cơ chế khôi phục. Khả năng tự chữa lành hư hỏng này

giảm thiểu các gián đoạn đối với các dịch vụ AI, tăng cường sự khả dụng và và tin cậy dịch vụ. Ngoài ra, điều phối tạo điều kiện cho O&M mạng thông minh, cho phép các nhà quản trị mạng quản lý các tài nguyên mạng và các dịch vụ AI hiệu quả hơn, giảm chi phí O&M và cải thiện hiệu suất.

11.5. CÁC CÔNG NGHỆ HỖ TRỢ CHỦ CHỐT

Phần này sẽ trình bày các công nghệ hỗ trợ chủ chốt cho NET4AI. Chine Mobile đề xuất thiết kế một hệ thống 6G một cách hệ thống vượt qua giới hạn kết nối trong công nghiệp được tập trung trên khái niệm "ba lớp và năm mặt phẳng". "ba lớp" đề cập đến lớp vật lý, lớp mạng và lớp ứng dụng và dịch vụ. "năm mặt phẳng" bao gồm mặt phẳng truyền thông, mặt phẳng dữ liệu , mặt phẳng tính toán, mặt phẳng trí tuệ, mặt phẳng an ninh, mặt phẳng quản lý và điều phối. Bốn mặt phẳng đầu tiên là là các mặt phẳng chức năng độc lập được tạo thành bởi tách riêng mạng 6G từ đầu đến cuối (End to End) để cung cấp các khả năng và các dịch vụ. Mặt phẳng quản lý và điều phối thực hiện kết hợp linh hoạt các chức năng trên các lớp và và các mặt phẳng, thực hiện quản lý và lập lịch các tài nguyên, và các dạng thức của một chuỗi dịch vụ hoàn chỉnh. Cấu trúc này tập trung lên các phần tử khả năng đa chiều trong mạng. Năm mặt phẳng khả năng/dịch vụ trong kiến trúc mạng 6G tương lai được trình bày trong hình 11.11.



Hình 11.11. Công nghệ cho phép NET4AI

Dưới đây ta sẽ xét các công nghệ liên quan chặt chẽ với các sự cần thiết AI.

11.5.1. Trí tuệ phân tán và FL

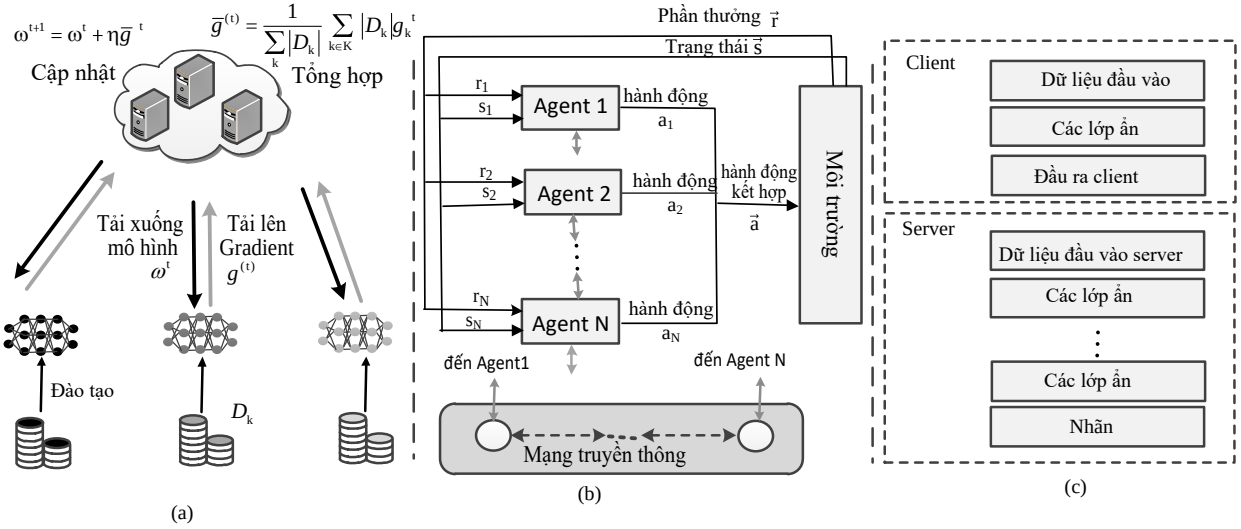
NET4AI phải xử lý các yêu cầu kép của trễ và sự tin cậy của truyền thông và tính toán cũng như các lo lắng ngày càng tăng về an ninh. Trí tuệ phân tán và FL (Federated Learning: học liên kết) là các công nghệ chủ chốt có thể giải quyết các thách thức này. Các kiến trúc tập trung truyền thông thường không còn đáp ứng được các yêu cầu trễ thấp, truyền thông và tính toán tin cậy cao. Ngoài ra các kiến trúc tập trung thông thường không hiệu quả trong việc hỗ trợ trí mọi nơi cho yêu cầu của các mạng không dây tương lai. Khuynh hướng hướng đến triển khai các phần tử trí tuệ phân tán trở nên ngày càng hiển nhiên, với dịch chuyển dần dần trí tuệ từ các hệ thống tập trung vào các hệ thống phân bố. Vì thế cần dưa vào các kiến trúc tính toán thông minh phân tán để sử dụng hoàn toàn dữ liệu đa chiều và các tài nguyên tính toán mà các đầu cuối người dùng và các nút có được để cho phép kết nối thông minh.

Trí tuệ phân tán không dây đề cập đến tổ chức các nhiệm vụ AI theo cách phân tán trong các mạng không dây và kết hợp AI và ML hợp tác mà không cập nhật toàn bộ dữ liệu thô đến đám mây trung tâm. Quan điểm này giảm bớt áp lực băng thông mạng và giảm gánh nặng tính toán của đám mây trung tâm. Ngoài ra, vì năng lực tính toán của điện thoại thông minh và các thiết bị IoT tăng và người dùng trở nên lo lắng hơn về quyên riêng tư dữ liệu, các thiết bị di động có thể xử lý tính toán và đào tạo AI nhiều hơn. Sự gần gũi của AI với các người dùng và dữ liệu cho phép các vòng lặp và cập nhật nhanh hơn dẫn đến hiệu quả hỗ trợ trí tuệ nội tại rộng khắp.

Trí tuệ phân tán là công nghệ nền tảng cho thực thiện trí tuệ nội tại được mường tượng cho các mạng 6G. Với việc tiếp nhận kiến trúc phân phán, các chức năng và các tài nguyên mạng được phân tán trên nhiều nút và nhiều thiết bị, nhờ vậy loại bỏ sự phụ thuộc vào một nút trung tâm đơn lẻ để quản lý toàn bộ mạng. Mỗi nút trong một mạng phân tán hoạt động với một mức độ độc lập cho phép đưa ra quyết định và thực hiện nhiệm vụ độc lập mà không cần truyền thông liên tục và phối hợp với các nút khác. Sự tự chủ này tăng cường sự linh hoạt, làm cho nó có khả năng thích ứng tốt hơn các điều kiên môi trường động và phát triển các mẫu yêu cầu.

Trong các mạng phân tán, dữ liệu được lưu trữ trên nhiều nút và các nhiệm vụ tính toán được xử lý hợp tác giữa chúng. Cách tiếp cận lưu trữ và tính toán không tập trung cải thiện sự khả định cỡ mạng, tăng tốc đào tạo và suy luận của các LAM (Large-Scale AI Model: mô hình AI quy mô lớn) và tăng cường hiệu suất toàn mạng. Ngoài ra, trí tuệ phân tán tạo điều kiện học trực tuyến và tối ưu hóa liên tục bằng cách cho phép thu thập dữ liệu, xử lý và cập nhật mô hình thời gian thực, nhờ đó cải thiện hiệu suất và sự chính xác của hệ thống AI. Công nghệ AI phân tán giải quyết các hạn chế của các hệ thống AI

tập trung, như các nút thắt cổ chai truyền thống và các lo lắng quyền riêng tư dữ liệu. Nó giảm chi phí gian tiếp và đảm bảo an ninh dữ liệu bằng cách giữ thông tin nhạy cảm gần nguồn của nó hơn. AI phân tán sẵn sàng để tạo ra một hệ sinh thái thông minh mới cho 6G, thúc đẩy một mô hình mạng khả định cỡ, hiệu quả, bảo vệ quyền riêng tư hơn. NHư được thể hiện trong hình 11.12, Kiến trúc trí tuệ phân tán bao gồm các thành phần sau.



Hình 11.12. Minh họa các về (a) FL (Federating Learning): học liên kết, (b) MARL (Multi Agent Reinforocement Learning): học tăng cường đa tác nhân, (c) SL (Sharing Learning): học chia sẻ trong trí tuệ phân tán.

1) **FL**. FL là một mô hình của trí tuệ phân tán cho phép tăng cường tính riêng tư bằng cách đào tạo các mô hình AI tại các thiết bị cục bộ và chỉ chia sẻ các thông số mô hình thay vì dữ liệu thô. Trong quá trình đào tạo, từng nút client đào tạo cục bộ và tải lên các trọng số mô hình đều đặn. Một nút trung tâm tổng hợp các trọng số này và hồi tiếp các trọng số được tổng hợp đến client cho vòng đào tạo tiếp theo hay cho suy luận. Thí dụ điển hình của FL trong mạng truyền thông không dây được minh họa trong hình 11.12(a). Qúa trình bắt đầu với tải xuống mô hình toàn cục $\omega^{(t)}$ cho vòng t đến các thiết cục bộ. Từng thiết bị k đào tạo mô hình trên tập dữ liệu D_k riêng của mình, tạo ra một cập nhật gradient $g_k^{(t)}$ và sau đó nó được cập nhật đến server thông số. Server này áp dụng giải thuật trung bình bình hóa liên kết (FedAvg: Federated Averaging) để tổng hợp các cập nhật này và gradient toàn cục $\bar{g}^{(t)} = \frac{1}{\sum_k |D_k|} \sum_k |D_k| g_k^{(t)}$, sau đó cập nhật mô hình toàn cục ω^{t+1} cho vòng tiếp theo. Qúa trình này tiếp diễn cho đến khi hội tụ mô hình. FL có thể

được chia thành FL theo chiều ngang (gọi tắt là FL ngang) và FL theo chiều dọc (gọi tắt là FL dọc). FL ngang nhấn mạnh hợp tác liên nguồn và chia sẻ dữ liệu, trong khi đó FL dọc tạo điều kiện cho hợp tác và trao đổi dữ liệu xuyên suốt các cấp độ khác nhau trong cùng một nguồn dữ liệu.

- **FL chiều ngang (Horizotal FL)**. Cũng còn được biết là FL dựa trên mẫu, chiều ngang có thể được áp dụng khi các tập dữ liệu của các người tham gia khác nhau có cùng một không gian tính năng nhưng các không gian mẫu khác nhau. Thông qua FL ngang, các người tham gia khác nhau có thể kết hợp đào tạo một mô hình dưa trên các tập dữ liệu nội bộ trong khi vẫn bảo vệ quyền riêng tư và cải thiện sự chính xác và tổng quát hóa mô hình. FL ngang cho phép học thời gian thực trên các thiết bị biên phân tán, cho phép các cập nhật thông số kịp thời cho các môi trường và dữ liệu thay đổi liên tục. Nó hỗ trợ hợp tác trên khắp các thiết bị với các khả năng tính toán thay đổi và phân tán dữ liệu, đáp ứng đa dạng các nhu cầu được cá nhân hóa của các thiết bị bị đầu cuối của mạng 6G. Nó cũng hỗ trợ tính toán trên không trung (OAC: Over-the-Air Computation) để đạt được khả định cỡ tuyệt vời, chấp nhận các khiếm khuyết truyền dẫn và có thể tiềm năng được thực hiện một cách phân tán hoàn toàn.
- **FL chiều dọc (Vertical FL)**. FL dọc đề cập đến một thiết lập trong đó các tập dữ liệu có các tính năng khác nhau, nhưng cùng các mẫu/các người dùng. Nó phù hợp cho các kịch bản trong đó có chồng lấn cao (High Overlay) giữa các người tham gia, nhưng chồng lấn thấp (Low Overlay) hoặc không có chồng lấn các tính năng. Trong quá trình đào tạo, dữ liệu của người tham gia phải đầu tiên được điều chỉnh thông qua lấy mẫu để chồng lấn các tập dữ liệu trên tất cả các người dùng. Sau đó, mô hình được đào tạo nội bộ, và các thông số nhận được được cập nhật đến server trung tâm. FL dọc sử dụng các tương quan giữa các nguyền dữ liệu đa dạng để cho phép các tập dữ liểu bổ sung và được làm giầu nhờ vậy tăng cường sự tổng quát hóa và bền vững của mô hình AI. Bằng cách chia sẻ các thông số, FL dọc tạo điều kiện cho học đa cấp độ (Cross-Level Learning) và truyền kiến thức, cho phép các nguồn dữ liệu trao đổi các kinh nghiệm học và cải thiện thiện hiệu suất tổng thể. Để giải quyết vấn đề da dạng người dùng, một cách tiếp cận mới cho một hệ thống khuyến nghị cộng tác theo phương đứng dựa trên giải thuật phân cụm (Clustering Algorithm) và mô hình yếu tố tiềm ẩm (Latent Factor Model) được đề xuất. Phương pháp này tăng cường hiệu quả sự chính xác của các kết quả khuyến nghị. Một số tác giả đề xuất ý tưởng FL đứng ẩn ý định (Inten-Hidden) cho đào tạo hợp tác đa phía của các mô hình dự báo căn bệnh trong chẩn đóa y khoa và xây dựng một giao thức an ninh sử dụng mật mã hóa đồng hình (HE: Homoomorphic Encription) cho bảo vệ quyền riêng tư dữ liệu. Một số tác giả đề xuất học sâu liên

kết (Federated DL)) dựa trên các mạng không gian-thời gian dài-ngắn hạn cho giao thông thành phố thông minh trong các môi trường MEC. Sử dụng các tính năng mẫu đa dạng (chẳng hạn các vùng khác nhau) trong cùng một không gian mẫu (chẳng hạn tình hình giao thông và khoảng thời gian cụ thể) cho phép xây dựng mô hình liên kết cho dự đoán luồng giao thông. FL đứng có thể làm lộ ra các quan hệ giữa các tính năng khác nhau của các mẫu dữ liệu làm cho nó có giá trị cao hơn cho các ứng dụng như các hệ thống khuyến nghị và dự báo giao thông.

2) Học tăng cường đa tác nhân (MARL: Multi-Agent Reinforcement Learning)

Trong môi trường chia sẻ, các tác nhân học tìm kiếm các giải thưởng riêng cho mình. Mạng 6G hỗ trợ các ứng dụng AI đa dạng như các thành phố thông minh, giao thông thông minh, có thể hưởng lợi từ MARL.

Hệ thống đa tác nhân (MAS) bao gồm một tập hợp các tác nhân ra quyết định tồn tại trong một môi trường chung, như minh họa trong hnh 11.12 (b). Các tác nhân này quan sát môi trường của chúng và giao tiếp với nhau để thực hiện các hành động phù hợp với mục tiêu của chúng. Thiết lập đa tác nhân liên quan đến việc quản lý các quá trình ra quyết định và học tập của nhiều thực thể có thể tương tác với nhau và môi trường chung của chúng.

Với việc đưa ra quyết định thích ứng của mình, MARL cho phép các tác nhân tự lập tối ưu hóa và điều chỉnh để tăng cường trải ngiệm người dùng trên khắp các ứng dụng khác nhau. Một số tác giả đã đưa ra một giải thuật MARL phi tập trung dựa trên giải thuật diễn viên-nhà phê bình lợi thế (Advantage Actor-Critic) trong đó diễn viên: Actor (chọn hành động), nhà phê bình: Critic (đánh giá hành động bằng cách cho điểm) để cải thiện hiệu suất học bằng các sử dụng hàm Advatage (lợi thế) để đánh giá một hành động tốt hơn như thế nào so với trung bình. Giải thuật này giải quyết hiệu quả vấn đề tắc nghẽn tong các hệ thống giao thông vận tải thông minh. Một số tác giải đề xuất giải thuật DDPG hồi quy đa tác nhân (Multi Agent Recurrent Deep Determitistic Plolicy Gradient: gradient chính sách tất định sâu hồi quy đa tác nhân) để quản lý các giai đoạn của đèn giao thông theo thời gian thực bằng cách sử dụng dữ liệu của interrnet phương tiện, để giảm thiểu tắc nghẽn trong môi trường đường bộ phức tạp và có thể quan sát một phần. Trò chơi Markov quan sát được một phần cũng được sử dụng để lập mô hình trong lái xe tự hành kết nối. Mọt số tác giả tập trung lên tính toán hợp tác trong các vệ tinh LEO (vệ tinh quỹ đạo thấp) để phát triển các mô hình quyết định giảm tải (Offloading) được tối ưu hóa thông qua MARL với một hàm phạt. Ngoài ra, một phương pháp điều độ sản cuất (Production Scheduling) trực tuyến dựa trên MARL cũng đã được đề xuất cho các nhà máy, trong khi đó học tăng cường sâu đa tác nhân cũng đa được áp dụng cho truy nhập đa kênh và nhiệm vụ giảm tải (Offloard) trong công nghệp được hỗ trợ bởi MEC.

3) Học học phân chia (Split learning)

Do các hạn chế của các thiết bị bị ràng buộc tài nguyên không thể hỗ trợ các mô hình học sâu hay cộng tác FL. SL (Split Learning: học phân chia) mỏi lên như là một giải pháp khả thi. SL phân chia các mô hình ML bằng cách phân tán các phần khác nhau giữa client và server để đảm bảo rằng dữ liệu vẫn được an ninh. Trong khi các nhiệm vụ tính toán nặng được trao cho các nút trung tâm mạnh mẽ, thì các lớp xử lý dữ liệu nhẹ hơn được vẫn được giữ nguyên tại đầu cuối nơi lưu trữ dữ liệu này. Như là một giải pháp thay thế hay bổ sung cho FL, SL giảm gánh nặng xử lý trên các thiết bị bị ràng buộc bởi tài nguyên và nắm giữ tiềm năng cho phát triển trí tuệ phân tán trong 6G. Một số tác giả đã đưa SL và các ứng dụng y tế cho phép nhiều thực thể đào tạo công tác các mô hình DL mà không chia sẻ dữ liệu thô nhạy cảm, đạt được hiệu suất hứa hẹn trên các tập dữ liệu y khoa. SL đã được nghiên cứu cho đào tạo cộng tác nhiều cơ sở, cho thấy hiệu suất client được cải thiện và bảo vệ tính riêng tư tin cậy hơn. Để xử lý các vấn đề như quá khớp và hội tụ chậm trong phương pháp đào tạo lần lượt gốc, một số tác giả đã đề xuất mộtphương pháp đào tạo song song. Phương pháp này giảm thiểu quá khớp thông qua điều chỉnh lô đào tạo thích ứng và đồng bộ lớp nút nhờ vậy tăng cường hiệu quả đào tạo trong khi vẫn đảm bảo quyền riêng tư. SL cũng đã được áp dụng cho các mạng không dây để sử dụng dữ liệu riêng tư của thiết bị đầu cuối trong IoT.

Một phương pháp đào tạo song song dựa trên phân cụm đã được đề xuất để tối ưu hóa chọn lựa lớp cắt (Cuting layer), phân cụm thiết bị và ấn định phổ để giảm hiệu qua các trễ đào tạo. Tuy nhiên, SL vẫn tiếp tục dễ bị tổn thương đối với đa tạp dữ liệu, các phương pháp đào tạo phức tạp, hội tụ chậm và thất bại hội tụ tiềm năng.

Các công nghệ được đề cập ở trên tất cả đều có thể đóng góp cho trí tuệ phân bố. FL phù hợp cho các thiết bị AI trong các kich bản đòi hỏi tính riêng tư dữ liệu và có thể áp dụng được cho các ứng dụng đòi hỏi trí tuệ mới nổi của 6G do nó sở hữu tính tương thích với các mô hình học thông minh khác. MARL phù hợp với các nhu cầu học trực tuyến trong các môi trường phức tạp và hoàn thành các nhiệm vụ AI trong môi trường khả quan sát và khả tương tác. SL phù hợp cho học cộng tác của các mô hình DL trong các thiết bị bị ràng buộc bởi tài nguyên và có thể thính ứng đến các mức ràng buộc tài nguyên khác nhau thông qua phân chia mô hình đa mức. Trí tuệ phân tán có thể mở rộng đến các hướng nghiên cứu bổ sung. Các hướng này bao gồm: (i) tiếp nhận sửa lỗi và cơ chế phát lại như các mã sửa lỗi tiên tiến và các chiến lược phát lại thích ứng để đảm bảo sự tin cậy trong các môi trường truyền thông không tin cậy; (ii) thực hiện các giao thức an ninh và các biện pháp bảo vệ quyền riêng tư bao gồm mật mã hóa dữ liệu, xác nhận nhận dạng và kiểm soát truy nhập để bảo vệ an toàn dữ liệu khỏi truy nhập và sự can thiệp trái phép; (iii) hiệu quả trao đổi dữ liệu/ thông số mô hình để tăng cường hiệu quả năng lượng của

các giải thuật học và (iv) công bằng thông qua cơ chế khuyến kích (Incentive Mechanisms) cho các thiết bị tham gia .

11.5.2. Các bản sao số cho AI gốc của mạng 6G

Tài liệu của ITU định nghĩa các yêu cầu và các kiến trúc của các mạng DT, nhấn mạnh vai trò cốt lõi của nó là trình bày ảo các mạng vật lý. Các mạng DT cho phép phân tích, hội chẩn, mô phỏng và điều khiển mạng vật lý. ITU đề xuất thiết kế kiến trúc "ba lớp", "ba miền" và "hai vòng kín", đồng thời nhấn mạnh các ứng dụng trong các hoạt động, tối ưu hóa, đổi mới và diễn tập chiến thuật an ninh. Chẳng hạn, DT tối ưu hóa giảm tải tính toán trong công nghiệp IoT trong các liên kết uRLLC. Hình 11.13 cho thấy kiến trúc DT tham chiếu theo định nghĩa của ITU.

Ba lớp đề cập đến lớp mạng vật lý (Physical Network Layer), lớp bản sao số mạng (Network Digital Tween Layer) và lớp ứng dụng mạng (Network Application Layer) của các hệ thống mạng DT. Ba miền đề cập đến miền dữ liệu (Data Domain), miền mô hình (Model Domain) và miền quản lý (Management Domain) của lớp mạng DT tương ứng với ba phân hệ kho dữ liệu thống nhất (Uniffied Repossitory), các mô hình dữ liệu thống nhất (Uniffied Data Models) và quản lý thực thể DT (DT Entity Management). Hai vòng đề cập đến giả lậpvà tối ưu hóa (Emulation and Optimalization) của vòng kín trong dựa trên các mô hình dữ liệu DT, và tối ưu hóa, phản hồi và điều khiển của vòng kín ngoài của các ứng dụng mạng dựa trên kiến trúc ba lớp. Các lớp trong kiến trúc DT tham chiếu như sau.

1) Lớp dưới cùng là lớp mạng vật lý. Tất cả các phần tử mạng trình bày trong các mạng vật lý trao đổi khối lượng lớn dữ liệu và điều khiển mạng bằng một thực thể DT của mạng thông qua các giao diện hướng nam (SBI: South Bound Interrfaces). Mạng vật lý có thể là một mạng của nhà mạng viễn thông hay một mạng của trung tâm dữ liệu, mạng khuôn viên, IoTcông nghiệp hay các kểu mạng khác.
    a) Kho dữ liệu thống nhất: Bằng cách thu thập và cập nhật dữ liệu hoạt động của các thực thể mạng khác nhau thông qua giao diện SBI, một nguồn duy nhất dữ liệu thực cho mạng DT được tạo ra, cung cấp thông tin đầy đủ và chính xác cho các ứng dụng mạng khác nhau.
    b) Các mô hình dữ liệu thống nhất: Đối với các kịch bản ứng dụng điển hình, phân hệ mô hình dữ liệu thống nhất hoàn thiện lập mô hình dựa trên dữ liệu, cung cấp các thực thể mô hình cho các ứng dụng mạng khác nhau và cực đại hóa khả năng thích ứng và lập trình các dịch vụ mạng. Các thực thể mạng cần hoàn thiện đầy đủ giả lập và kiểm chứng, dự đoán, lập lịch, lập cập cấu hình,

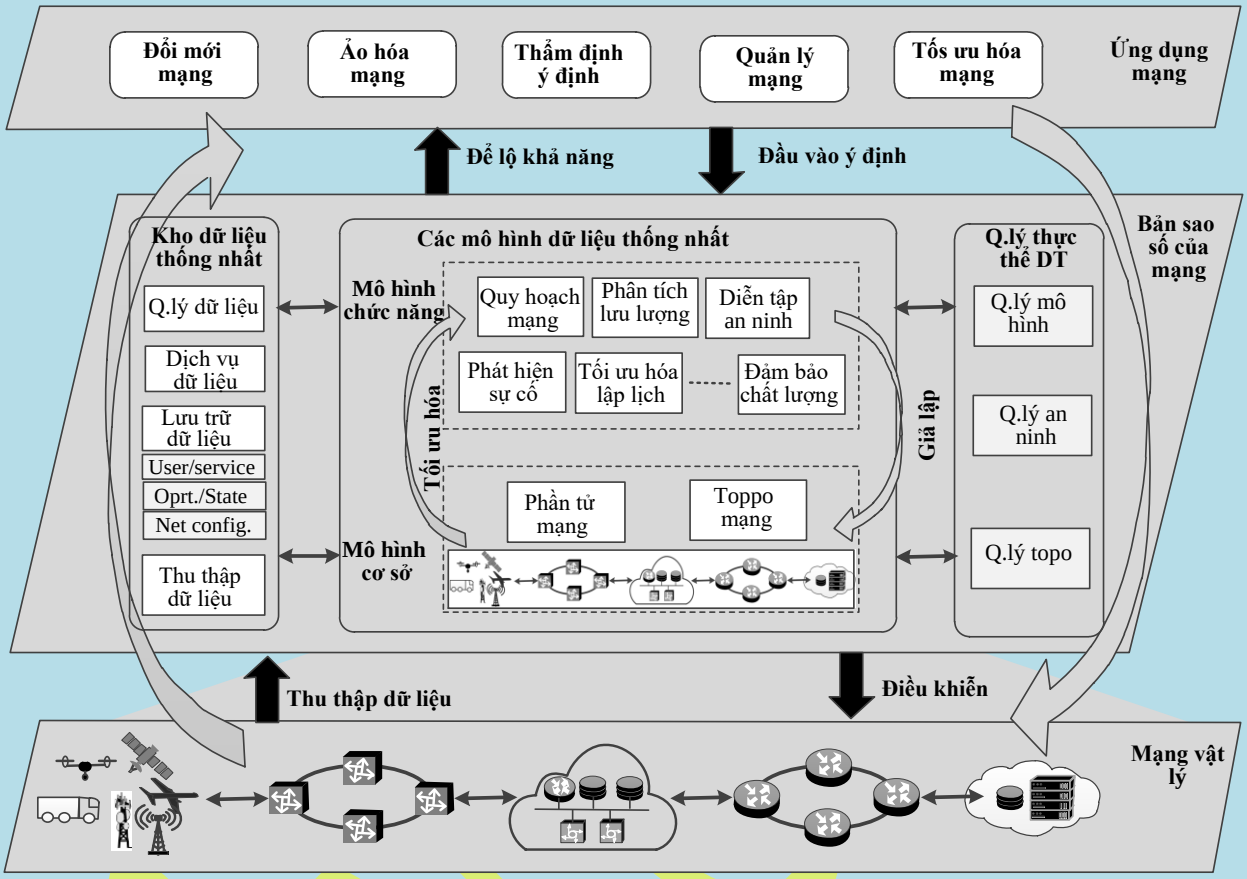
tói ưu hóa và các mục tiêu khác trong mạng DT để đảm bảo sự hiệu qủa và tin cậy của các của các điều khiển thay đổi (Change Control) trước khi chúng được phân tán đến mạng vật lý.

c) Quản lý thực thể DT hoàn thiện chức năng quản lý mạng DT, ghi lại vòng đời thực thể, ảo hóa và điều khiển các phần tử khác nhau của DT mạng, bao gồm quản lý cấu hình topo, quản lý mô hình và quản lý an ninh.

2) Lớp giữa là lớp DT của mạng, đây là lõi cả hệ thống DTN. Lớp này chứa các phân hệ then chốt; kho dữ liệu thống nhất, các mô hình dữ liệu thống nhất và quản lý thực thể DT.

3) Lớp trên cùng là lớp ứng dụng của mạng. Các ứng dụng của mạng đưa vào các yêu cầu đến lớp DT của mạng thông qua các giao diện hướng bắc (NBI: Northbound Interface) và khai các dịch vụ thông qua lập mô hình các thực thể. Sau khi thẩm định đầy đủ, lớp DT mạng gửi các cập nhật điều khiển đến lớp các thực thể mạng thông qua các giao diện hương nam (SBI: Southbound Interface). Vận hành, bảo dưỡng mạng và tối ưu mạng có thể được triển khai nhanh với chi phí thấp hơn và hiệu suất cao hơn và gây ít ảnh hưởng hơn lên vận hành của một kinh doanh mạng.

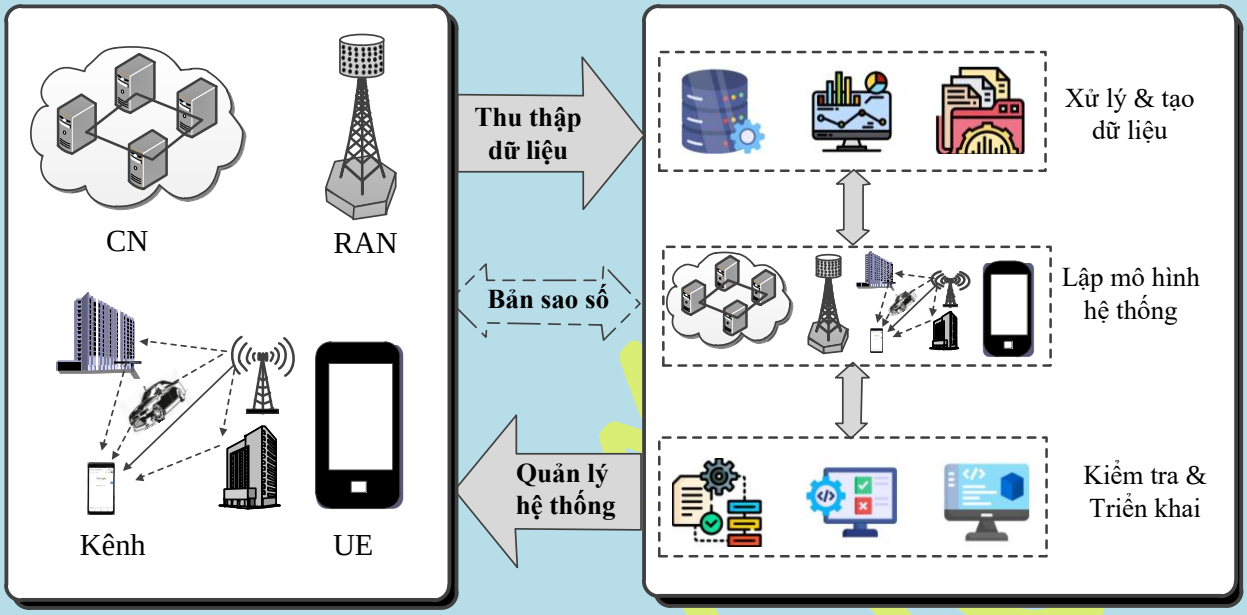
BẢN GỐC



User/Service (User/Service Data) dữ liệu người dùng/dịch vụ; Oprt./state (Operatioanal/Status/Log Data): dữ liệu hoạt động/trạng thái/nhật ký; Net.Config (Network Environement/Configuaration Data): dữ liệu mô trường, cấu hình mạng.

Hình 11.13. Kiến trúc DT tham chiếu theo định nghĩa của ITU.

Các hệ thống truyền thống thông thường dựa trên dữ liệu lịch sử và các mô hình thông kê gập khó khăn để thích ứng nhanh chóng các thay đổi mạng động dẫn đến các điều chỉnh rất khó khăn. Các hạn chế này cản trở khả năng thích ứng, tin cậy và linh hoạt của hệ thống. Công nghệ bản sao số (DT: Digital Tween) được đưa vào lĩnh vực truyền thông để giải quyết các thách thức này cho phép cung cấp một cách tiếp cận linh hoạt, hiệu quả và chính xác đến tối ưu hóa tài nguyên. Tiếp nhận DT không chỉ tăng cường khả năng phản ứng nhanh chóng dễ dàng trước những thay đổi mà còn đưa ra các mô hình đổi mới cho hoạt động của mạng, giảm đáng kể các chi phí và các rủi do nghiên cứu và phát triển cho AI. Hình 11.14 mô tả kiến trúc vật lý của 6G và DT cho AI.



Hình 11.14. DT cho AI

Bằng cách tạo ra một bản sao ảo của một đối tượng vật lý, các DT thiết lập một kết nối hai chiều giữa các thế giới vật lý và ảo, cho phép phản ảnh trạng thái hoạt động của thực thể vật lý và mô phỏng hành vi tương lai của nó. Trong các hệ thống truyền thông 6G, công nghệ DT tạo điều kiện cho xây dựng các bản sao mạng toàn diện để giám sát các thiết bị , dự đoán trạng thái mạng và tối ưu hóa ấn định tài nguyên.

Thực hiện DT dựa trên sự kết hợp giữa một số công nghệ then chốt bao gồm thu thập dữ liệu chính xác cao, lập mô hình thời gian thực và mô phỏng, tương tác dữ liệu hai chiều, phân tích và tối ưu hóa thông minh. Trong các hệ thống truyền thông, các bộ cảm biến phân tán liên tục thu thập dữ liệu đa dạng về các trạng thái của các thiết bị mạng, các biến động lưu lượng, các thông số môi trường và các mẫu hành vi. Các dữ liệu này được phát đến nền tảng DT thông qua các liên kết truyền thông trễ thấp tốc độ cao. Nền tảng DT sử dụng các mô hình vật lý và các giải thuật ML dựa trên dữ liệu để xây dựng các bản sao có độ trung thực cao để mô phỏng toàn diện trạng thái hoạt động và các thay đổi động của các thực thể vật lý. Bằng cách tương tác dữ liệu hai chiều, các hệ thông vật lý và các bản sao ảo đạt được đồng bộ thời gian thực. Khi một sự tăng vọt lưu lượng được dự báo, DT có thể điều chỉnh một cách thực tế các chiến lược phân phối lưu lượng, tối ưu hóa các tài nguyên phổ và công suất và giảm bớt áp lực lên các thiết bị mạng. Tích hợp tính toán biên sẽ tăng cường hơn nữa hiệu quả của công nghệ này nhờ việc phân tán lập mô hình, mô phỏng và tối ưu hóa các nhiệm vụ đến các nút biên gần các thực thể vật lý hơn. DT cho phép quản lý thông minh và tối ưu hóa động các mạng phức tạp trong các hệ thống

truyền thông 6G, nhờ vậy hỗ trợ bền vững cho ấn định tài nguyên hiệu quả ,ổn định mạng và các nâng cấp trí tuệ.

Mạng 6G nhắm đến đạt được dung lượng siêu cao, truyền thông cự ly cực ngắn, dịch vụ hơn cả nỗ lực tốt nhất (Beyond Best Effort Service), truyền thông chính xác cao và tích hợp nhiều kiểu truyền thông. Tuy nhiên tầm nhìn này đưa ra các thách an ninh, hiệu suất phổ, hiệu suất năng lượng và khả năng chi trả. Sự xuất hiện của công nghệ DT cung cấp các cơ hội hứa hẹn cho giải quyết các thác thức này. DT cung cấp một bản sao ảo hóa cho các mạng 6G, cho phép giám sát lưu lượng và phân tích tích mạng toàn diện. Sử dụng phản hồi từ mạng ảo hóa, các hệ thống 6G có thể tăng cường an ninh của chúng bằng cách chuẩn bị trước các mối đe dọa. Ngoài ra, DT cho phép nhận dạng yêu cầu và cung cấp dịch vụ bằng cách phân tích dữ liệu truyền thông để phân biệt các mẫu và các quy tắc sử dụng. Hiểu biết dự đoán các nhu cầu truyền thông cho phép các mạng 6G dành trước tài nguyên như phổ, dự báo trước các nhu cầu tương lai. Ngoài ra tích hợp công nghệ DT cho phép 6G hỗ trợ các dịch vụ đổi mới bao gồm AR/VR và xe tự hành. Với việc giải quyết các vấn đề an ninh, hiệu suất phổ, trí tuệ và tùy chỉnh, DT định nghĩa lại và tăng tốc phát triển các mạng 6G.

Nghiên cứu quan trọng tích hợp công nghệ DT với các mạng 6G để tăng cường hiệu suất mạng. Chẳng hạn, DT đã được đề xuất cho quản lý các bộ phản xạ siêu mặt phẳng trong truyền thông terahertz 6G. Bằng cách lập mô hình, dự đoán và điều khiển các đặc tính truyền sóng của các tín hiệu trong nhà, DT cực đại hóa tỷ số tín hiệu trên tạp âm (SNR) terahertz của hệ thống. Ngoài ra, một số tác giả đưa ra khái nhiệm các mạng không dây DT bằng cách tích hợp DT và các mạng không dây. Cách tiếp cận này chuyển dịch xử lý và tính toán dữ liệu thời gian thực đến mặt phẳng biên và sử dụng DT để giảm thiểu sự không tin cậy của truyền thông đường dài giữa các người sử dụng đầu cuối và các server biên trong các mạng 6G.

Một trong các khả năng quan trọng của các mạng 6G sẽ là hỗ trợ khối lượng lớn các thiết bị đầu cuối dẫn đến tạo ra lưu lượng lớn. Chẳng hạn, một số tác giả đã thiết kế tính toán biên trong các mạng 6G. Bằng cách kết hợp trạng thái mạng động của người dùng di động và DT, nghiên cứu này đã giải quyết các vấn đề liên quan đến biên với việc sử dụng học tăng cường sâu (DRL) và học chuyển giao (Transfer Learning) dẫn đến thành công giảm trễ hệ thống trung bình và cải thiện sử dụng tài nguyên. Tương tự một số tác giả đã đề xuất một phương pháp giảm tải (Offload) nhiệm vụ được DT hỗ trợ, bằng cách tạo lập các DT cho tất các trạng thái của thiết bị để đạt được trễ và tiêu thụ công suất thấp hơn trong mạng. Ngoài ra một số tác giả đã nghiên cứu cơ chế đấn định tài nguyên động cho các dịch vụ dựa trên DT trong 6G IoT. Nghiên cứu này đã cải thiện hiệu suất ấn định tài

nguyên bằng cách thiết lập chuỗi chức năng dịch vụ với các DT và sử dụng một phương pháp RL tập thể.

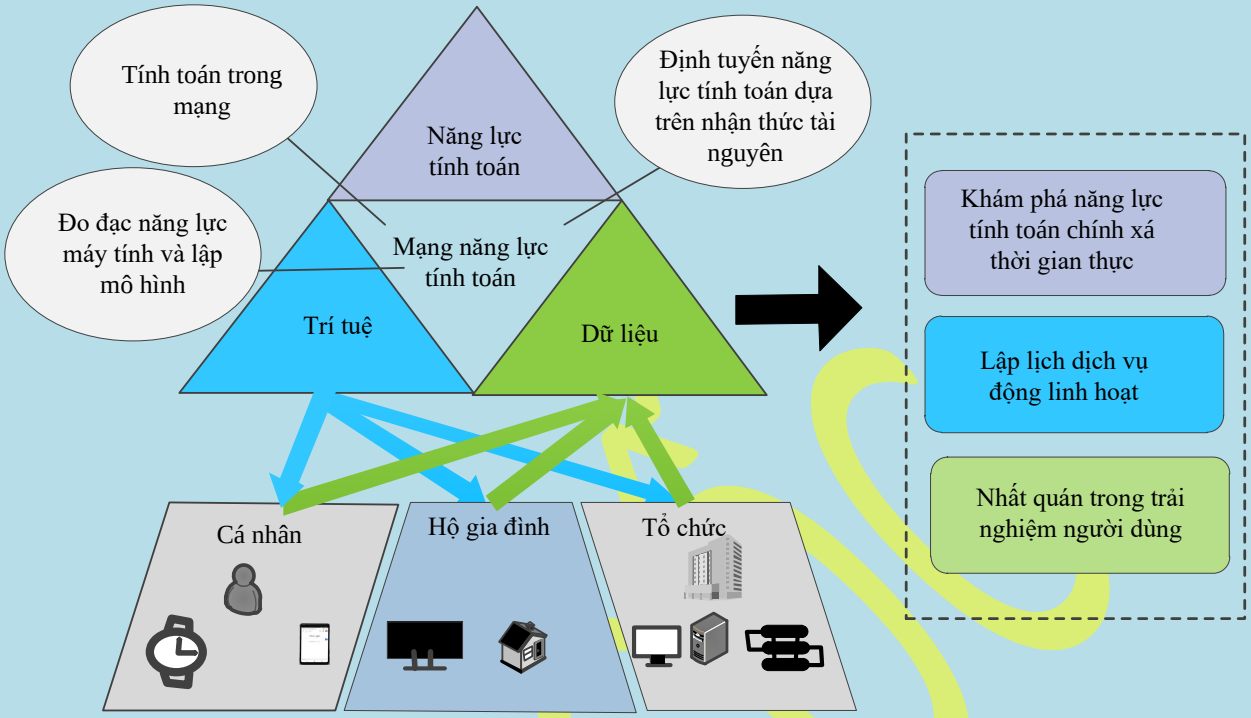
Ngoài ra, kênh DT (DT Channel) đã được đề xuất như là một ánh xạ ảo của một kênh không dây phản ánh toàn bộ quá trình của các trạng thái pha đinh và bieens thiên của kênh trong thế giới vật lý. DTC cso thể được áp dụng cho trong các kịch bản sử dụng khác nhau của 6G, cung cấp các tăng cừng hiệu năng đáng kể trong tốc độ dữ liêu, trễ, hiệu suất phổ và tin cây liên kết. Một nền tảng thực hiện DTC mới đã được đề cuất bao gồm thu thập dữ liêu, cảm biến môi trường mô-đun tái tạo để xây dựng quan hệ giữa môi trường và kênh cho dự đoán kênh, quyết định truyền thông và tương tác. Một mô hình kênh dựa trên tạo cụm hat nhân cũng đã được đề xuất để phân tích lập bản đồ mối quan hệ giữa các cụm và các vật tán xạ trong môi trường truyền sóng nhằm cho phép tái tạo kênh trên các kịch bản khấc nhau. Ngoài ra một hồ kiến thức về môi trường vô tuyến đã dượcđề xuất để phục vụ như là một yếu tố đặc thù cho DTC.

Tích hợp DT với các công nghệ chuỗi khối, FL trong truyền thông không dây sẽ bảo vệ an ninh và quyền riêng tư mạnh mẽ hơn.Trong các môi trường IoT, một số tác giả giảm thiểu trễ giảm tải tính toán trong các liên kết RRLLC bằng cách sử dụng các mạng không dây biên.

11.5.3. Mạng năng lực tính toán mẽ và AI

Kiến trúc NET4AI đòi hỏi các khả năng dịch vụ tính toán mạnh mẽ, vì các mạng không dây rất quan tâm đến các nhiệm vụ tính toán liên quan đến tính toán hiệu suất cao như xử lý dữ liệu phạm vi siêu lớn và DL. Mạng năng lực tính toán (Computing Power Network) là công nghệ hỗ trợ quan trọng cho các dịch vụ tính toán. Nó đạt được liên kết nối tính toán mọi nơi thông qua cảm biến và hợp tác tương hỗ giữa mạng và các tài nguyên tính toán, hòa hợp các tài nguyên đám mây, biên, đầu cuối và thực hiện các nhiệm vụ tính toán trong mạng, đặc biệt là các dịch vụ tính toán liên quan đến AI. Mạng năng lực tính toán là một kiểu hạ tầng thông tin mới. Hình 11.15 cho thấy, nó hỗ trợ phát hiện năng lực tính toán, quản lý năng lực tính toán, giảm tải nhiệm vụ và tính toán của mạng. Nó kết nối các nút trung tâm năng lực tính toán phân tán về mặt địa lý thông qua các topo mạng và thu thập cũng như chia sẻ năng lực tính toán, dữ liệu và các tài nguyên ứng dụng.

Các nguyên lý kỹ thuật của mạng năng lực tính toán bao gồm ba khía cạnh: (i) khám phá năng lực tính toán thời gian thực, (ii) lập lịch dịch vụ linh hoạt và động, (iii) nhất quán với trải nghiệm người dùng.



Hình 11.15. Mạng năng lực tính toán và AI

Mạng năng lực tính toán được phân thành ba thể loại.

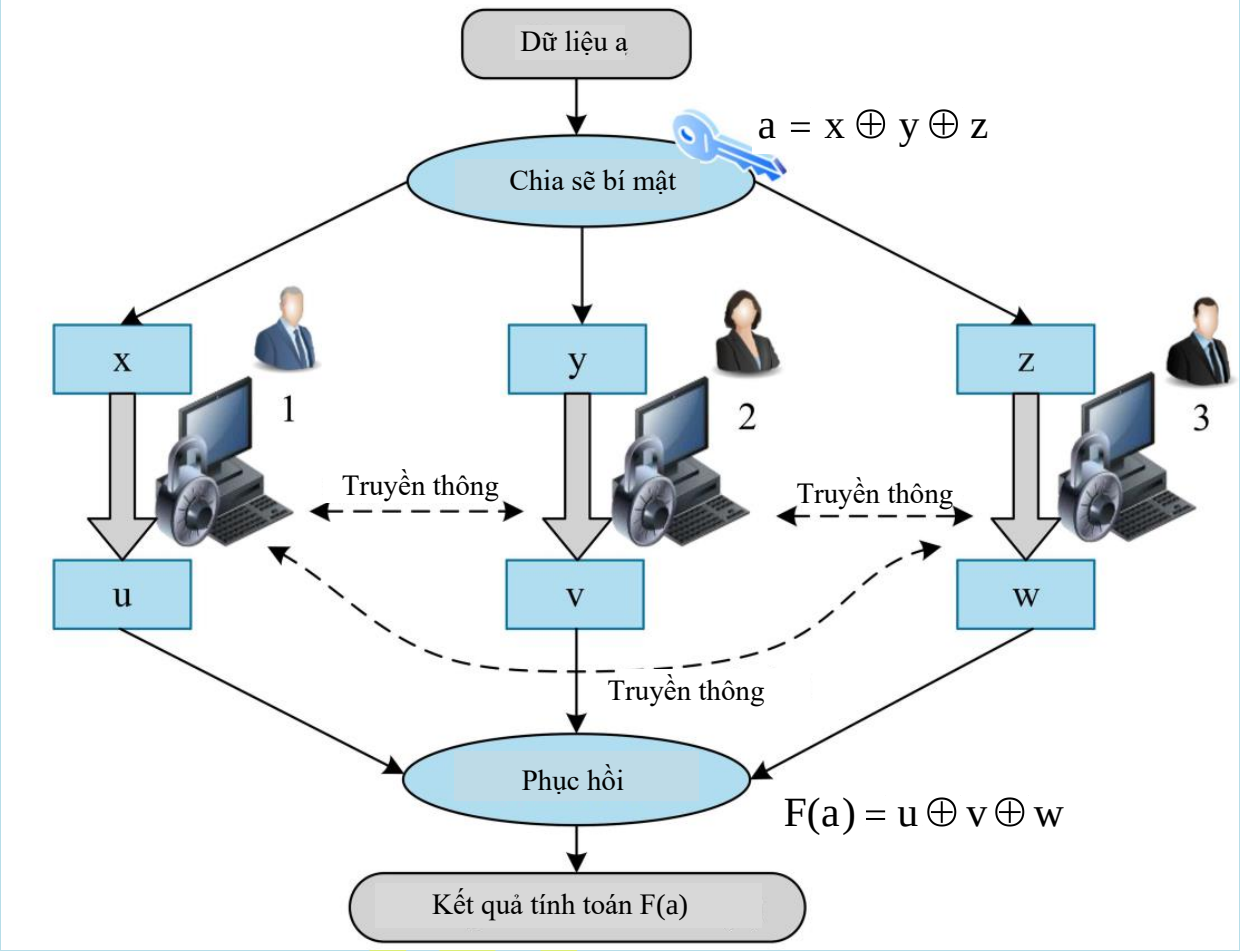
1) **Đo năng lực tính toán và lập mô hình.** Đây là một công nghệ nền tảng để cung cấp các dịch vụ năng lực tính toán. Trong tương lai, các nhà cung cấp dịch vụ năng lực tính toán trong các mạng năng lực tính toán sẽ không bị giới hạn đến các trung tâm dữ liệu đặc thù hay các cụm tính toán. Họ có năng lực tính toán mọi nơi từ đám mây, biên các các thiết bị đầu cuối. Chia sẻ hiệu quả năng lực tính toán này thông qua các kết nối mạng đòi hỏi cảm nhận chính xác dung lượng tính toán của các chip đa tạp, các kiểu kinh doanh phù hợp cho các chip hiện thời và vị trí của chúng trong mạng, cũng như quản lý và giám sát hiệu quả.
2) **Định tuyến năng lực tính toán dựa trên nhận thức.** Trong mạng năng lực tính toán, sau khi đo và lập mô hình các tài nguyên tính toán, thông tin này được mã hóa và được đóng vào các gói của mạng dựa trên thông tin tài nguyên tính toán được chia sẻ, hướng dẫn định tuyến kinh doanh đến các tổ hợp tài nguyên (Resource pool) khác nhau thông qua cộng tác giữa các tổ hợp tài nguyên tính toán để xử lý kinh doanh, điều này cho phép nhận thức của mạng về các tài nguyên tính toán để hướng dẫn định tuyến toàn cục.

3) **Tính toán trong mạng (INC: In-Network Computing).** Sử dụng việc triển khai các công nghệ mạng khả lập trình, INC xử lý các gói trong mạng. Chia xẻ năng lực INC với việc sử dụng các tài nguyên đa tạp mở và khả lập trình sẽ tăng tốc xử lý dữ liệu gần nguồn mà không cần thay đổi chế độ hoạt động kinh doanh gốc, giảm trễ trễ phản hồi của ứng dụng và đơn giản hóa các xử lý triển khai ứng dụng. Các mạng năng lực tính toán hỗ trợ các người dùng điều chỉnh định cỡ tài nguyên theo yêu cầu để thích ứng đến các ứng dụng AI có các kích cỡ và các mức độ phức tạp khác nhau.

11.5.4. Tính toán nhiều bên an ninh (Sercure Multi-Party Computation)

Trong các môi trường phức tạp cao, đa tạp và phát triển động, bắt buộc phải đảm bảo an ninh dữ liệu trong khi vẫn phải đạt đượcsự công tác hiệu quả. Vì công nghệ mạng phát triển không ngừng, nên các phương pháp tấn công mạng cũng không ngừng trở nên tinh vi hơn. Các cơ chế an ninh thông thường cũng thường xuyên không phù hợp với các mối đe dọa mới nổi này. Hình 11.15 cho thấy, tính toán an ninh nhiều bên (SMPC: Security Multi-Party Computation) là một giao thức mật mã hóa cho phép nhiều bên kết hợp tính toán một hàm trong khi đó vẫn duy quyền riêng tư dữ liệu đầu vào của họ. Mỗi bên nhận được kết quả tính toán đầu ra đúng mà không biết các đầu vào riêng tư của các bên khác. SMPC có các đặc trưng như phi tập trung hóa, an ninh dữ liệu đầu vào và các kết quả tính toán chính xác. Nó cho phép AI xử lý các môi trường mạng phức tạp, bảo vệ quyền riêng tư của người dùng, hỗ trợ các kịch bản ứng dụng mời nổi và thúc đẩy chia sẻ và sử dụng dữ liệu.

BẢN NHÁP



Hình 11.15. Tính toán nhiều bên an ninh

Các công nghệ hỗ trợ chính của SMPC bao gồm các mạch điệnlộn xộn (Garbled Circuits), chuyển giao vô thức (Oblivious Transfer), chia sẻ bí mật (Sharing Secret) và mật mã hóa đồng hình (Hormomorphic Encryption).

1) Kỹ thuật các mạch điện lộn xộn (Garbled Circuits) bao gồm biên dịch logic tính toán vào một mạch và mật mã hóa từng cổng (chẳng hạn: AND, XOR, v.v…) trong mạch điện này. Các bên tham giá có thể hoàn thành tính toán bằng cách tương tác với thông tin đươc mã hóa mà không biết logic đặc thù của mạch.
2) Kỹ thuật chuyển giao vô thức (Oblivious Transfer) là một giao thức cho phép một bên tham gia (chẳng hạn bên gửi) gửi đi một trong số nhiều mẩu thông tin, trong đó phía thu chọn chỉ một một mẩu thông tin. Phía phát không biết mẩu thông tin nào đã được phía thu chọn.

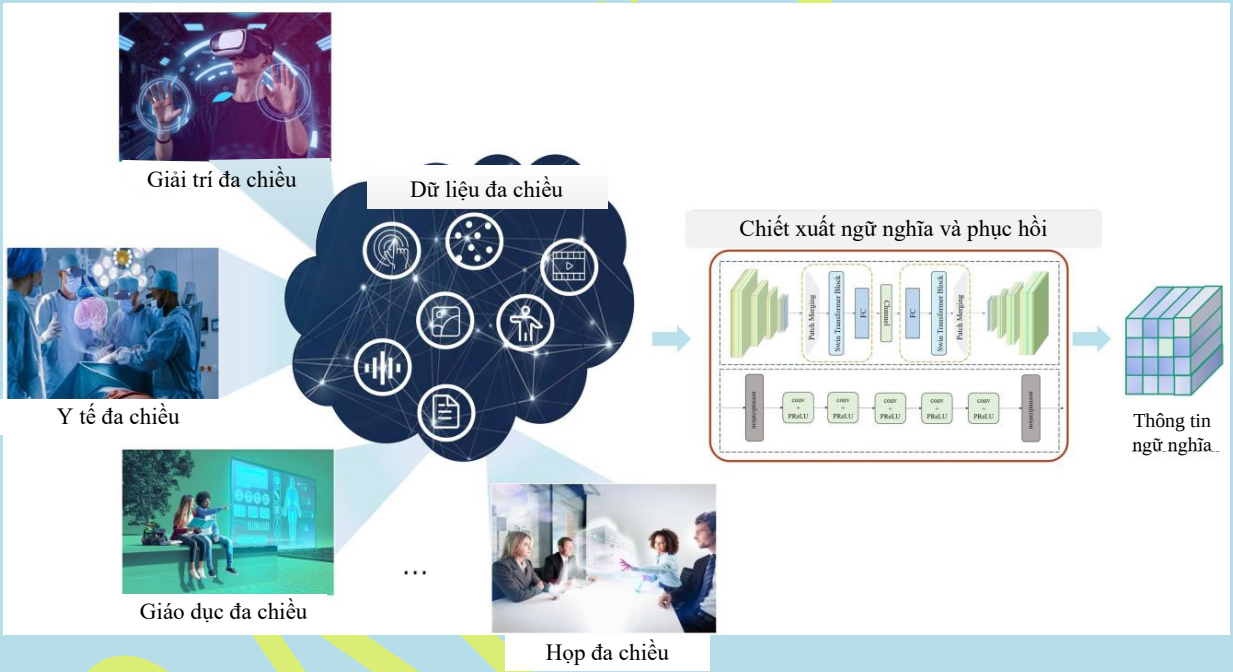
3) Chia sẻ bí mật (Sharing Secret) bao gồm phân chia một bí mật thành nhiều phần và phân phối chúng giữa các bên tham gia. Chỉ có thể khôi phục lại bí mật gốc khi một số lượng cần thiết các phần được kết hợp.
4) Mật mã hóa hóa đồng hình (Hormomorphic Encryption) là một phương pháp mật mã hóa đặc biệt cho phép cá ctính toán đặc thù (chẳng hạn cộng và nhân) có thể được thực hiện trên dữ liệu đã được mật mã hóa. Kết quả của các tính toán này, khi được giải mã, tương đương với kết quả của cùng các tính toán được thực hiện trên số liệu gốc.

Nghiên cứu gần đây sử dụng SMPC để đạt được tổng hợp mô hình an ninh trong FL, liên quan đến đào tạo các mô hình ML trên các thiết bị phi tập trung trong khi vẫn duy trì cục bộ hóa dữ liệu đào tạo. Để chống lại các tấn công đảo ngược (Reverse Attack) lên dữ liệu, một cơ chế tổng hợp bản tồn tính riêng tư dựa trên các kỹ thuật chia sẻ bí mật của SMPC có thể bảo vệ hệu quả FL chống lại các tấn công như vậy. Để giải quyết chi phí truyền thông gián tiếp và khả định cỡ kém trong các SMPC, một SMPC hai gia đoạn dựa trên khung FL đã được đề xuất. Khung này cho phép nhiều client hợp tác đào tạo các mô hình AI trong khi vẫn bảo vệ quyền riêng tư của chúng. Ngoài ra, một kiến trúc FL phân cấp, sử dụng SMPC, đã được phát triển để giảm bới các chi phí truyền thông và các vấn đề khả định cỡ liên quan đến tổng hợp các mô hình toàn cục chỉ tại một vài nút, cho phép nâng cao hiệu quả truyền thông và khả định cỡ mà không xâm phạm quyền riêng tư. Một kiến trúc FL lai ghép đã được đề xuất, nó kết hợp FL với các môi trường thực hiện tin tưởng, SMPC, và các vệ tinh bắc đẩu (Beidau Satellites). Kiến truc này đạt được phân phối khóa, mật mã hóa, giải mật mã an ninh và cung cấp các cơ chế kiểm chứng cho từng bên tham gia để đảm bảo an ninh dữ liệu cục bộ. Ngoài ra, SMPC có thể kết hợp với công nghệ chuỗi khối để theo dõi và giảm thiểu hành vi gây hại, nhằm đạt được một khung học cộng tác biên an ninh và đáng tin cậy.

Việc SMPC bảo vệ quyền riêng tư dữ liệu trong khi vẫn bảo vệ tính toán hợp tác trên nhiều phía làm cho nó trở thành tài sản giá trị cho các ứng dụng 6G. Nó có thể đóng góp cho FL, cho ấn định và tối ưu hóa tài nguyên, kết hợp và phân tích dữ liệu an ninh, kiểm chứng nhận dạng, quản lý khóa và các hệ thống phát hiện gian lận thanh toán (Anti-Fraud Billing System) phân tán. Bằng cách sử dụng tính toán trên dữ liệu mã hóa (Encrytioned Computation), SMPC đảm bảo xử lý an ninh thông tin nhạy cảm. Trong khi SMPC tăng cường quyền riêng tư và an ninh chia khi sẻ dữ liệu trong các mạng 6G, vẫn còn các thách thức, bao gồm chi phí gián tiếp tính toán và truyền thông, tiêu chuẩn hóa giao thức, và tìm được cân bằng giữa bảo vệ quyền riêng và hiệu quả.

11.5.5. Truyền thông ngữ nghĩa (Semantic Communication)

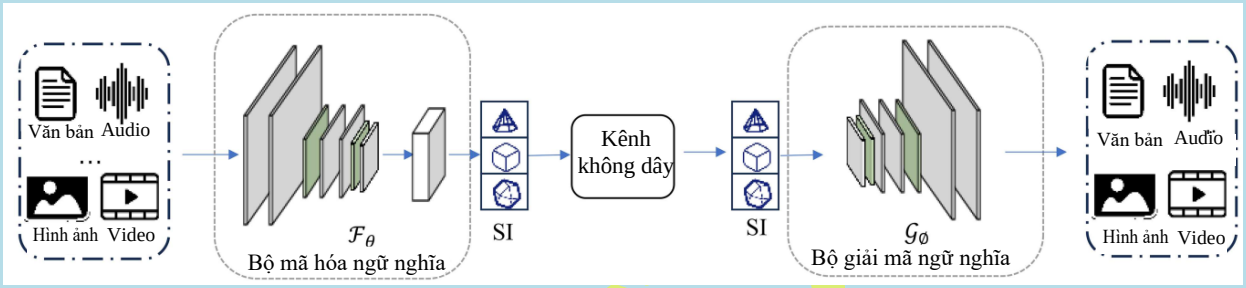
Truyền thông ngữ nghĩa, thường dựa trên các mạng DNN, được coi là một công nghệ hứa hẹn trong 6G. So với truyền thông cú pháp (Syntactic Communication) tập trung lên truyền dẫn chĩnh xác dữ liệu theo các bit, truyền thông ngữ nghĩa chỉ phát đi các ngữ nghĩa của bản tin. Nó có thể bảo toàn các quan hệ ngữ nghĩa khi phát thông tin, nhờ vậy hỗ trợ tải xuống các nhiệm vụ. Các đặc tính này hỗ trợ đáp ứng các yêu cầu của các ứng dụng 6G tiêu tốn dữ liệu, chẳng hạn truyền thông đa chiều (Holographic Communication) và XR như được thể hiện trong hình 11.16.



Hình 11.16. Minh họa các ứng dụng đòi hỏi tương tác trễ siêu thấp trong các mạng 6G và các truyền thông ngữ nghĩa dựa trên DL có thể hưởng lợi rất lớn từ các ứng dụng này.

Hình 11.17 cho thấy một mô hình ngữ nghĩa cơ sở, bao gồm các phần tử thiết yếu: bộ mã hóa ngữ nghĩa, kênh và bộ giải mã ngữ nghĩa. Tại phía máy phát, bộ mã hóa ngữ nghĩa chiết xuất và mã hóa SI (Semantic Information) của dữ liệu nguồn. Quá trình này bao gồm chiết suất SI và nén hoặc loại bỏ thông tin không không quan trọng. Để đạt được điều này dữ liệu thô được mã hóa bởi mạng nơ-rôn ký hiệu là $\mathcal{F}_\theta$, đầu ra duy trì được ý nghĩa quan trọng trong khi xóa bỏ các chi tiết không quan trọng. Các ngữ nghĩa súc tích được phát đi trên kênh một vật lý bị tạp âm, chẳng hạn kênh không dây. Tại máy thu, bộ giải mã ngữ nghĩa giải mã dữ liệu thu. Quá trình này bao gồm "sự hiểu biết" và "suy luận" nội dung ngữ nghĩa được gửi đến từ máy phát. Để hoàn thành điều này, bộ giải mã

ngữ nghĩa , được kỹ hiệu là $\mathcal{G}_\theta$, xử lý dữ liệu thu với mục đích ánh xạ nó ngược về dữ liệu nguồn.



SI (Semantic Information): thông tin ngữ nghĩa

Hình 11.17. Mô hình truyền thông ngữ nghĩa cơ sở

Hầu hết các nghiên cứu truyền thông ngữ nghĩa tập trung lên truyền dẫn văn bản, truyền dẫn và nhận biết audio, truyền dẫn và nhận biết video, hình ảnh từ góc độ phương thức dữ liệu (Data Modality) và phân loại nhiệm vụ.

1) Một số tác giả đã phát triển một hệ thống truyền thông ngữ nghĩa văn bản dựa trên DL được đặt tên là DeepSC, nó thực hiện mã hóa kênh ngữ nghĩa kết hợp để đạt được độ lợi hiệu suất vượt trội so với các công nghệ hiện có đặc biệt là tỷ số tín hiệu trên tạp âm thấp. Nhận thấy rằng hầu hết các số đo ngữ nghĩa không khả vi và mô hình tối ưu hóa dựa trên RL truyền dẫn ngữ nghĩa phức tạp và phạm vi rộng đã được đưa ra. Kỹ thuật này sử dụng các cập nhật lặp ngẫu nhiên tự phê bình (Self Critical Random Iterative Update) để đào tạo các bộ phát thu ngữ nghĩa tách rời (Decouped Semantic Transceiver), cho phép xử lý vấn đề tối ưu hóa kênh ngữ nghĩa không khả vi. Một số tác giả đề xuất một kỹ thuật truyền thông ngữ nghĩa định hướng văn bản được đặt tên là Reed Solomon HARQ (Reed-Solomon Hybrid Automatic Repeat Request: Yêu cầu phát lại tự động Reed Solomon) bằng cách kết hợp HARQ và mã hóa kênh Reed Solomon bằng các mã hóa ngữ nghĩa để xử lý vấn đề không hiệu quả và không linh hoạt với các độ dài câu thay đổi.
2) **Truyền dẫn và nhận dạng Audio**. Hệ thống truyền thông ngữ nghĩa audio dựa trên DL được đặt tên là DeepSC dã được phát triển bởi một số tác giả. Sơ đồ truyền thông ngữ nghĩa audio này áp dụng bộ mã hóa/giải mã ngữ nghĩa và bộ mã hóa/giải mã kênh kết hợp để giảm bớt méo ngữ nghĩa gây ra bởi tạp âm và nhiễu từ các kênh vô tuyến. Một số tác giả đã khám phá truyền thông ngữ nghĩa dựa trên FL để phát triển bộ tự mã hóa dựa trên Wave2sec cấu thành từ các mạng CNN. Hệ thống này mã hóa, phát và giải mã Audio SI hiệu quả đồng thời giảm chi phí truyền thông gián tiếp.

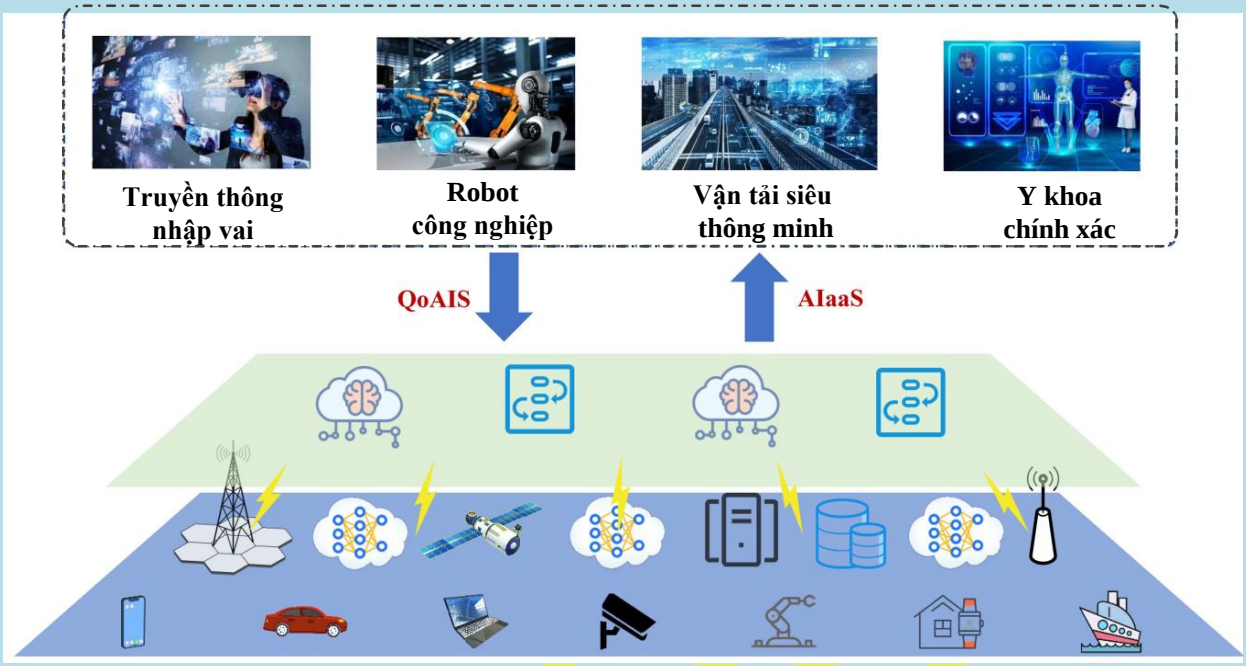
3) **Truyền dẫn và nhận dạng hình ảnh**. Một số tác giả đã tích hợp một kiến trúc tuyền thông ngữ nghĩa hình ảnh DeepSCC, trong đó các chức năng của bộ mã hóa và giải mã được thông số hóa bởi CNN và đào tạo kết hợp trên cùng một tập dữ liệu để giảm thiểu sai sô bình phương trung bình của hình ảnh được tái tạo lại. Đào tạo kết hợp cho phép DeepSCC tránh được hiệu ứng vách đá (Cliff Effect) và thíc ứng với các SNR thay đổi.
4) **Truyền dẫn và nhận dạng video.** Một số tác giả đã phát triển một hệ thống DeepJSCC độ dài thay đổi sử dụng các kiến truc truyền dẫn phi tuyến tính và mã hó có điều kiện để thích ứng chiết xuất các tính năng ngữ nghĩa của cá khung video. Hệ thống này vượt trội độ chính xác cua truyền dẫn mã hóa không dây và nhận dang video truyền thống.

   Truyền thông ngữ nghĩa có các ưu điểm duy nhất trong sự hiểu ngữ nghĩa, thúc đẩy sự tuến bộ trong các ứng dụng như internet xúc giác (Tacile Internet), truyền thông đa chiều, XR và các mạng cộng sinh người máy (Human-Machine Symbiolic Network), thúc đẩy phát triển một mạng hiệu quả và có trí tuệ không ngừng tăng.

11.6. AI NHƯ LÀ MỘT DỊCH VỤ (AIaaS)

Khái niệm AaaS mường tượng một nền tảng rộng khắp mạng để chuyển phát liền mạch các dịch vụ AI đến các người dùng và các thiết bị được kết nối. Để hỗ trợ hiệu quả hơn "trí tuệ gốc" (Native Intellegence) và đạt được "trí tuệ nhân tạo toàn năng" (Artificial Universal Intellegence) mọi nơi, các mạng 6G sẽ xử lý trí tuệ nhân tạo như là một dịch vụ (AIaaS: AI as a Service), dẫn đến xuất hiện khái niệm 6G AIaaS như được trình bày trong hình 11.15. 6G sẽ cung cấp hỗ trợ hiệu quả từ đầu đến cuối cho cho các công việc kinh doanh và các ứng dụng liên quan đến AI, kết nối thông minh các tác nhân phân tán để triển khai AI quy mô lớn trên tất cả các nghành nghề. Các dịch vụ 6G AI sẽ xử lý các nhu cầu cho hiệu suất thời gian thực cao, an ninh và quyền riêng tư cao hoặc tiêu thụ công suất tổng thể thấp bằng cách tiến hành đào tạo AI hay suy luận trong mạng, cung cấp các khả năng được thiết kế cho các kịch bản ứng dụng khác nhau.

Các mạng 6G, như là các kiến trúc thông minh vốn có, sẽ có khả năng đào tạo hiệu quả hoặc suy luận hiệu quả AI phân tán phạm vi rộng thông các tài nguyên và các chức năng trong mạng bao gồm truyền thông, tính toán, các tập dữ liệu và các nô hình nền táng]j chuyển từ AI4NET trong các mạng 5G cho AI vào 6G sẽ cuối cùng cung cấp trí tuệ vượt xa hơn dữ liệu (Go beyond Data), trực tiếp hướng đến các người dùng. Sự chuyển đổi này sẽ định nghĩa lại hệ sinh tái của các đám may biên và tạo ra các mô hình kinh doanh mới thông qua các mạng di động 6G, chuyển dịch các mạng tập trng theo kết nối sang các mạng định hướng theo dịch vụ và rốt cuộc đạt được trí tuệ mọi nơi.



Hình 11.15. AI như là một dịch vụ (AIaaS).

11.6.1. Các kịch bản điển hình của AIaaS

Các kịch vả điển hình cho 6G AIaaA là ngữ cảnh và tình trạng mà tại đó AI trong các mạng 6G được áp dụng bao gồm nhưng không giới hạn ở các ngành nghề và các lĩnh vực chẳng hạn nông nghiệp, giáo dục và y tế.

11.6.1.1. Truyền thông nhập vai (Immersive Communication)

Trong truyền thông tương lai, các công nghệ truyền thông nhập vai như VR, AR và các báo cáo đo đạc sẽ tăng cường các trải nghiệm tương tác. AaaS sẽ hỗ trợ các công nghệ này với phiên dịch ngôn ngữ thời gian thực, phân tích cảm xúc và các đề xuất được cả nhân hóa trong các môi trường ảo. Các mô hình AI được đào tạo sẽ được thẩm định để đảm bảo các tương tác thực tế và đáng tin cậy. 6G AIaaS sẽ cung cấp đào tạo AI toàn diện, thu thập dữ liệu người dùng cho học đa chế độ và cho phép đào tạo , cập nhật các mô hình nhanh. Nó sẽ hỗ trợ liên kết đa thiết bị, cho phép các thiết bị khác nhau tương tác với các môi trường ảo. Vùng phủ được mở rộng của 6G sẽ kết nối liền mạch các thế giới thực và ảo, tạo điều kiện cho chia sẻ thông tin hệ thống chéo. Công nghệ DT sẽ tăng cường thẩm định AI, tạo ra dữ liệu mẫu giầu hơn các môi trường vật lý, cải thiện sự ổn định và hiệu suất của mô hình. Do số lượng các đầu cuối thông minh tăng, kết nối siêu

rộng của 6G và trễ thấp sẽ cực kỳ quan trọng cho đào tạo AI và thẩm định, hỗ trợ định vị chính xác trong các môi trường ảo và thu thập dữ liệu khối lượng lớn. 6G AIaaS phải cung cấp các dịch vụ tự quản để đảm bảo các tương tác trong các môi trường ảo, với các tùy chọn trải dài từ các cịch vụ hoàn toàn tự quản đến các dịch vụ do con người điều khiển. Hỗ trợ này sẽ dẫn đến truyền thông nhập vai có tính thực thực tế và hiệu quả hơn, cải thiện cộng tác và truyền thông từ xa.

11.6.1.2. Các robot công nghiệp

Các robot công nghiệp sẽ được sử dụng rộng rãi trong các kịch bản sản xuất và vận tải tương lai. 6GaaaS cung cấp các dịch vụ đào tạo AI cho các robot, chẳng hạn thu thập dữ liệu thông qua các robot, áp dụng đào tạo mô hình cho học đa tác nhân và phân tán các mô hình đã được đào tạo cho các robot. Các khả năng truyền dẫn cực độ của 6G sẽ hỗ trợ đào tạo mô hình nhanh và trao đổi thông số nhanh giữa các robot và mạng. 6GaaS cũng có khả năng cung cấp liên kết đa hệ thống, cho phép các thiết bị đầu cuối khác nhau liên kết với mạng bằng các robot một cách linh hoat và tương tác với môi trường thông qua công nghệ 6G AIaaS. Chúng thậm chí có thể tham gia vào học môi trường chéo và hệ thống ché ovới các robot từ các bộ phận khác nhau, chia sẻ các kinh nghiệm. Ngoài ra, mạng 6G có thể sử dụng các mạng DT để cung cấp các dịch vụ kiểm chứng AI, đào tạo lặp các mô hình AI với hiệu suất bền vững và tốt hơn thông qua thẩm định trước hiệu năng. 6GAIaaS cung cấp các dịch vụ kiểm chứng cho các mô hình đã được đào tạo, yêu cầu mạng DT tạo ra dữ liệu mẫu trong nhiều kịch bản hơn môi trường thực tế, giảm chi phí gián tiếp và ảnh hưởng hiệu suất của thu thập dữ liệu từ môi trường vật lý.

11.6.1.3. Y khoa chính xác

Y tế thông minh sẽ bao gồm các khía cạnh khác nhau của phòng ngừa, dự báo, hội chẩn, suy luận, giám sát bệnh tật, phẫu thuật lâm sàng, chăm sóc bệnh nhân, phát triển thuốc và và vắc -xin trong suốt vòng đời đời. Hệ thống 6G mới sẽ hỗ trợ tốt hơn truyền dẫn khối lượng lớn thông tin và đồng bộ cần thiết cho y tế thông minh và trực tiếp cho phép xử lý và đưa ra quyết định về thông tin. Bằng cách sử dụng AI trên các mạng 6G, các cơ sở y khoa phân tán về mặt địa lý và các chuyên gia y tế có thể kết nối trên phạm vi rộng hơn, tổng hợp nhiều mô hình và các mẫu dữ liệu tình huống nhiều hơn và nhanh chóng đồng bộ thông tin giữa các bác sĩ và các bệnh nhân. Điều này tăng cường học nhóm và FL, cải thiện liên tục sự chính xác, tin cậy và hiệu suất thời gian thực của các hội chẩn mang tính dự đoán và các hành động thời gian thực. Y tế thông minh cũng có thể đạt được quy

hoạch hợp lý và ấn định các tài nguyên y tế khác nhau tối ưu hơn thông qua AI của mạng 6G, giảm đáng kể áp lực lên các chuyên gia ý tế và các bệnh nhân.

11.6.1.4. Giao thông vận tải siêu thông minh

Giao thông vận tải thông minh sẽ bao gồm các giai đoan khác nhau trên toàn bộ chiều thời gian bao gồm tải xuống bản đồ phân giải cao, cảm biến môi trường xe cộ, dự đoán môi trường và lập kế hoạch tuyến đường. Lấy lái xe tự hành làm thí dụ, hệ thống 6G có thể cung cấp các dịch vụ dữ liệu dựa trên AI, các dịch vụ giảm tải tính toán AI, dự đóan môi trường và lập kế hoạch tuyến dựa trên AI. Các dịch vụ AI dựa trên dữ liệu đề cập đến hệ thống 6G cung cấp các xe tự hành các kết quả cảm biến môi trường được vận hành bởi AI. Chẳng hạn, khi một xe tự hành gập phải một cảm biến điểm mù, hệ thống 6G có thể thu thập dữ liệu cảm biến thông qua các bộ cảm biến hay các tín hiệu không dây, sau đó sử dụng các mô hình AI để suy luận môi trường có điểm mù và phản hồi các kết quả đến xe ô tô, nhờ vậy tăng cường sự an toàn của lái xe. Giảm tải tính toán Ai đề cập đến xử lý mà tại đó đó các xe tự hành chuyển một phần tải suy luận của chúng hoặc các tínhtoán đào tạo đến hệ thống 6G. Chẳng hạn, khi một xe lái qua một đoạm đường phức tạp, năng lực tính toán cần thiết cho AI có thể tăng đáng kể dẫn đến trễ không thể chấp nhận. Trong các trường hợp này, ô tô có thể giảm tải một phần các nhiệm vụ tính toán AI đến hệ thống 6G, như vậy giảm gánh nặng xử lý trên các chip của xe ô tô.

Vói các khả năng DL và hợp nhất đa cảm biến (Multi Sensor Fusion), một kiến trúc tích hợp cảm biến và đưa ra quyết định được thiết lập cho các xe ô tô, hạ tầng ven đường và các nền tảng đám mây. Kiến trúc này tiến đến một thời đại mới của vận tải thông minh được đặc trưng bởi các dịch vụ đi lại mọi nơi và quản lý, điều khiển toàn diện thông minh.

11.6.2. Chất lượng dịch vụ AI (QoAIS)

Chất lượng dịch vụ AI (QoAIS: Quality of AI Service) đề cập đến khung các số đo toàn diện để đánh giá các dịch vụ AI trong mạng. Nó là một khung đa chiều bao gồm các chiều (hay các đại lượng) then chốt như hiệu suất, kết nối, tính toán, dữ liệu , an ninh và điều phối. Đặc biệt, các số đo hiệu suất trong QoAIS khác với các chỉ số hiệu suất quan trọng KPI của mô hình truyền thống được sử dụng trong ML. Trong khi các KPI của mô hình chủ yếu đánh giá các số đo đánh giá nội tại của các mục đích phục vụ mô hình như tối ưu hóa và tinh chỉnh thì QoAIS mở rộng khái niệm này. Nó kết hợp các nhân tố như sự tổng quát và bền vững của các mô hình hay các giải thuật, cung cấp một cái nhìn tổng thể hiệu suất của toàn bộ hệ thống. Các số đo cho các chiều khác của QoAIS là mới, vì

chúng chịu trách nhiệm đặc biệt cho các đặc trưng và các yêu cầu duy nhất của các mạng 6G. Bảng 11.3 cho thấy các chỉ báo cmà một hệ thống QoAIS cần có.

Chiều chỉ báo	Các chỉ báo của QoAIS
Hiệu suất	Biên các số đo hiệu suất, thời gian đào tạo, tổng quát hóa, khả năng tái sử dụng, tính bền vững, khả năng tương tác, tính nhất quán giữa hàm mất mát và các đối tượng tối ưu hóa, sự công bằng, v.v..
Kết nối	Băng thông và jitter, trễ liên kết và jitter, tỷ số bit lỗi và jitter, v.v…
Dữ liệu	Sự dư thùa tính năng, sự hoàn thiện, sự chính xác dữ liệu, chuẩn bị dữ liệu tốn thời gian, cân bằng không gian mẫu, động lực học phân phối dữ liệu, v.v…
Tính toán	Sự chính xác tính toán, khoảng thời gian, hiệu quả, v.v….
An ninh	Bảo mật thông tin, các mức độ quyền riêng tư dữ liệu/giải thuật,
Điều phối	Hoàn toàn tự quản, có khả năng điều khiển một phần bởi con người, hoàn toàn có khả năng điều khiển bởi con người, v.v…

- **Các chỉ báo liên quan đến hiệu suất**. Các khía cạnh quan trọng để đánh giá và cải thiện các mô hình AI bao gồm biên các số đo hiệu suất (Performance Metrics Boundary), thời gian đào tạo, tổng quát hóa, khả năng tái sử dụng, tính bền vững, khả năng tương tác, tính nhất quán giữa hàm tổn thất và các đối tượng tối ưu hóa, sự công bằng. Biên các số đo hiệu suất định nghĩa các giới hạn trên và dưới của đánh giá chất lượng của các chỉ số hiệu năng của mô hình như tỷ số bit lỗi, sự chính xác và sự thu hồi. Tổng quát hóa phản ảnh khả năng của mô hình đưa ra các dự đoán chính xác về dữ liệu mới chưa gặp, trong khi đó sự tái dử dụng nhấn mạng khả năng của nó vẫn còn hiệu dụng trên đa dạng các kịch bản. Tính bền vững đảm bảo mô hình duy trì hiệu suất nhất quán bất chấp các nhiễu loạn và các tấn công đối nghịch. Khả năng tương tác tập trung lên mức độ mà ở đó có thể hiểu và giải thích được hoạt động bên trong mô hình và các đầu ra, thúc đẩy tính minh bạch và lòng tin. Sự nhất quán giữa hàm tổn thất và các đối tượng tối ưu hóa đảm bảo sự căn chỉnh giữa thiết kế hàm mất mát trong quá trình đào tạo và và các mục tiêu của hệ thống AI, tính đến tất cả các biến để đạt được kết cục mong muốn.
- **Các chỉ báo liên quan đến kết nối**. Các dịch vụ AI tin cậy và hiệu quả phụ thuộc lớn vào các đặc tính của mạng bao gồm băng thông, jiter, trễ liên kết, tỷ số bit lỗi và sự tin cậy kết nối tổng thể. Băng thông thể hiện dung lượng truyền dẫn của mạng, ảnh hưởng trực tiếp lên tốc độ và chất lượng của các ứng dụng AI. Jitter đề cập tơi sự thay đổi thời gian tới của gói, nó có thể dẫn đến sự không nhất quán trong hiệu suất dịch vụ, đặc biệt trong các nhiệm vụ nhạy cảm trễ. Trễ liên kết định lượng thời gian cần cho dữ liệu để di chuyển qua mạng, các trễ thấp hơn giúp

cho hệ thống AI hoạt động xuôn sẻ và đáp ứng nhanh hơn. Tỷ số bit lỗi đo tốc độ mà tại đó các lỗi bit xẩy ra trong quá trình truyền dẫn, nhấn mạnh tầm quan trọng của trao đổi dữ liệu chính xác và tin cậy. Cùng nhau, các yếu tố định nghĩa sự ổn định và độc lập của kết nối, ảnh hưởng trực tiếp lên các QoAIS.

- **Các chỉ báo liên quan đến dữ liệu**. Chất lượng và các đặc tính của dữ liệu đóng vai trò sống còn trong việc đảm bảo thành công của các hệ thống AI, tập trung lên sự dư thưa tính năng, sự hoàn thiện, sự chính xác, và thời gian chuẩn bị. Sự dư thừa tính năng nhấn mạnh sự hiển diện của chồng lấn hay các tính năng không liên quan, có thể gây phức tạp cho đào tạo mô hình và giảm hiệu suất. Sự hoàn thiện đảm bảo rằng tất cả các thuộc tính cần thiết của dữ liệu đều có sẵn để đạt được thấu hiểu sâu xắc và dự đoán toàn diện. Sự chính xác dữ liệu nhấn mạnh tầm qua trọng của dữ liệu khả dụng không lỗi để duy trì sự hợp lệ của các đầu ra AI. Chuẩn bị dữ liệu tiêu tốn thời gian phản ảnh các thách thức liên quan đến làm sạch, chuyển đổi và tổ chức dữ liệu cho các mục đính đào tạo. Cân bằng không gian mẫu đảm bảo trình bày công bằng các loại khác nhau hay các thể loại khác nhau trong tập dữ liệu, quan trọng cho giảm thiểu độ lệch (Bias). Động lực học phân phối đơn giản theo dõi các thay đổi trong các phân phối dữ liệu theo thời gian, cho phép các mô hình thích ứng và duy trì hiệu quả trong các quá trình tiến hóa môi trường. Kết hợp lại, các khía cạnh này củng cố nền móng cho xây dựng các mô hình AI bền vững và đáng tín cậy.
- **Các chỉ báo liên quan đến tính toán**. HIệu suất và sự chính xác trong tính toán mang tính quyết định để đảm bảo các hệ thống AI tin cậy và hiệu suất cao, bao gồm các nhân tố như tính toán chính xác, khoảng thời gian, và hiệu suất. Khả năng tính toán phản ánh sự chính xác của các kết quả tính toán đảm bảo sự tin cậy các đầu ra của các nhiệm vụ. Chiều này nhấn mạnh hiệu quả và sự chính xác của các tính toán AI, bao gồm sự chính xác tính toán, khoảng thời gian và hiệu quả. Khoảng thời gian đo thời gian cần thiết để hoàn thành các xử lý tính toán, với tính toán nhanh hơn cho phép các hoạt động thời gian thực và phi thời gian thực trong các hệ thống AI. Hiệu suất đánh giá sự sử dụng tài nguyên tính toán bao gồm bộ nhớ, công suất xử lý và tiêu thụ năng lượng. Tối ưu hóa các khía cạnh này đảm bảo rằng các dịch vụ AI không chỉ chính xác mà còn chi phí hiệu quả và bCaền vững.
- **Các chỉ báo liên quan đến an ninh**. Bảo vệ dữ liệu, các giải thuật và hệ thống là một yêu cầu cơ bản cho các dịch vụ AI, với các xem xét quan trọng bao gồm bảo mật, quyền riêng tư, xác thực và khả năng theo dõi. Bảo mật thông tin đảm bảo các thực thể không được phép không thể truy nhập dữ liệu nhạy cảm, bảo vệ niềm tin của người dùng và tuân thủ các quy định. Các mức độ quyền riêng tư của dữ

liệu/giải thuật nhấn mạnh các biện pháp được thực hiện để bảo vệ các giải thuật bản quyền và dữ liệu cá nhân, giảm các rủi do để lộ hoặc lạm dụng sai. Xác thực đầu ra (Output Autnenticity) đảm bảo rằng dữ liệu được sử dụng trong các xử lý AI là thật và không bị thay đổi, đảm bảo đầu ra tin cậy. Khả năng theo dõi đề cập đến khả năng theo dõi nguồn gốc, luồng và các thay đổi của dữ liệu và các giải thuật, cho phép giải trình và minh bạch dịch vụ.

- **Các chỉ báo liên quan đến điều phối**. Mức độ tự động hóa và sự can thiệp của con người vào các dịch vụ AI đóng một vai trò quan trọng trong việc xác định cách thức quản lý và điều khiển các dòng công việc. Tự chủ trong các dịch vụ AI định nghĩa mức tự động hóa và phạm vi can thiệp của con người cần có lên các dòng công việc số liệu, đào tạo, thẩm định và suy luận. Nó phản ảnh các kỳ vọng của người dùng liên quan đến tự động hóa và được phân loại thành ba mức. Tại mức cao nhất, tự chủ hoàn toàn cho phép các dịch vụ AI hoạt động động lập không có sự can thiệp của con người. Trái lại, điều khiển một phần của con người bao gồm các luồng công việc mà tại đó một số giai đoạn được tự động hóa, nhưng một số giai đoạn khác đòi hỏi sự hỗ trợ của con người. Cuối cùng, điều khiển hoàn tòa bởi con người cần có sự tham dự của cong người tại mọi giai đoạn của luồng công việc, hoàn toàn không dựa vào tự động hóa.

11.6.3. Các khả năng quan trọng cho hỗ trợ AaaS

Việc đạt được các ảnh thị giác dựa vào một số khả năng then chốt có thể được nhóm vào ba khía cạnh chính.

11.6.3.1. Hỗ trợ dữ liệu và khả năng tiếp cận mô hình

Một yêu cầu căn bản đối với một nền tảng AIaaS là hỗ trợ dữ liệu bền vững. Có thể đạt được điều này bằng cách sử dụng các khả năng cảm biến của mạng và các chức năng thu thập dữ liệu của các thiết bị IoT và các đầu cuối người dùng để cho phép thu thập và lưu trữ dữ liệu thời gian thực. Việc đảm bảo quyền riêng tư dữ liệu trong khi vẫn duy trì kết nối trực tiếp các người dùng với dữ liệu là rất quan trọng, vì kết nối này là nền tảng cho các dịch vụ đào tạo mô hình và suy luận. Ngoài ra các dịch vụ AI cũng dựa trên đa dạng các mô hình được thiết kế cho các nhiệm vụ khác nhau.Nền tảng này phải cho phép người dùng tiếp cận được các mô hình tồn tại từ trước hay tạo ra các mô hình tùy chỉnh phù hợp với các nhu cầu của họ. Một giao diện dành riêng kết nối người dùng đến các mô bình này sẽ tạo điều kiện cho tùy chỉnh và tăng cường tính linh hoạt của các dịch vụ AI.

11.6.3.2. Tính toán phân tán và tối ưu hóa tài nguyên

Năng lực tính toán được phân tán vốn có trong các CPU, GPU trải rộng khắp toàn mạng 6G và các tài nguyên nhớ đệm trên khắp các nút CN. Các BS, các thiết bị người dùng các hệ thống bên thứ ba tạo tành xương sống của AIaaS. Bằng cách sử dụng mạng được định nghĩa bằng phần mềm, có thể phát hiện, đánh nhãn và ảo hóa các tài nguyên này cho quản lý thống nhất. Các giao thức như các bảng Hash (bảng băm) phân tán được áp dụng để lưu trữ, tìm lại và cập nhạt các thông tin này một cách hiệu quả. Nhờ khả năng nhìn thấy tài nguyên tại chỗ, có thể ấn định động các tài nguyên tính toán, nhơ đệm và băng thông để đáp ứng nhu cầu người dùng. Các kỹ thuật như DRL, lý huyết trò chơi điều khiển dự báo mô hình cho phép ấn định thích ứng tài nguyên và lập lịh dịch vụ, đảm bảo hoạt động hiệu quả thậm chí trong điều kiện khối lượng công việc thay đổi.

11.6.3.3. An ninh và các các biện pháp bảo vệ bảo vệ vòng đời

Viêc phân tán các nút nút tính toán biên đám mây đưa ra các thách thức mới trong việc bảo vệ các kết nối động và chéo miền. Cấn phải có các biện pháp an ninh toàn diện để bảo vệ các dịch vụ AI trong suốt vòng đời của chúng. Các biện pháp này bao gồm thu thập dữ liệu an ninh và nhận thực để đảm bảo tính toàn vẹn của các đầu ra. Các giải thuật hòa nhập dữ liệu an ninh có thể xác thực và làm sạch dữ liệu từ nhiều thiết bị khác nhau, trong khi đó mã hóa đồng hình (HE: Hommorphic Encryption) cung cập kết hợp dữ liệu mà vẫn bảo toàn tính riêng tư. Các mô hình bảo vệ vòng đời với thuộc tính thông minh đang kể cũng quan trọng không kém. Các biện pháp như kiểm soát tiếp cận, quyền riêng tư vi sai (Differential Privacy), FL và giám sát thời gian thực các luồng dữ liệu có thể bảo vệ chống lại sự sử dụng không được phép, đỏa ngược mô hình và các mối đe dọa mới nổi khác.

Việc tích hợp các khả năng này vào các nền tảng AIaaS một các tiếp cận gắn kết để xử lý đồng thời các vấn đề quản lý lý dữ liệu, tính toán, lập lịch và an ninh. Trải nghiệm dịch vụ đa chiều được cung cấp bởi các nền tảng này phải thống nhất với các nhu cầu đặc thù người dùng. Cùng với sự phát triển của các mạng 6G, các hạn chế của mạng truyền thống sẽ được khắc phục, cho phép AIaaS phát triển mạnh mẽ trong một kiến trúc mạng thông minh, thích ứng và hiệu quả.

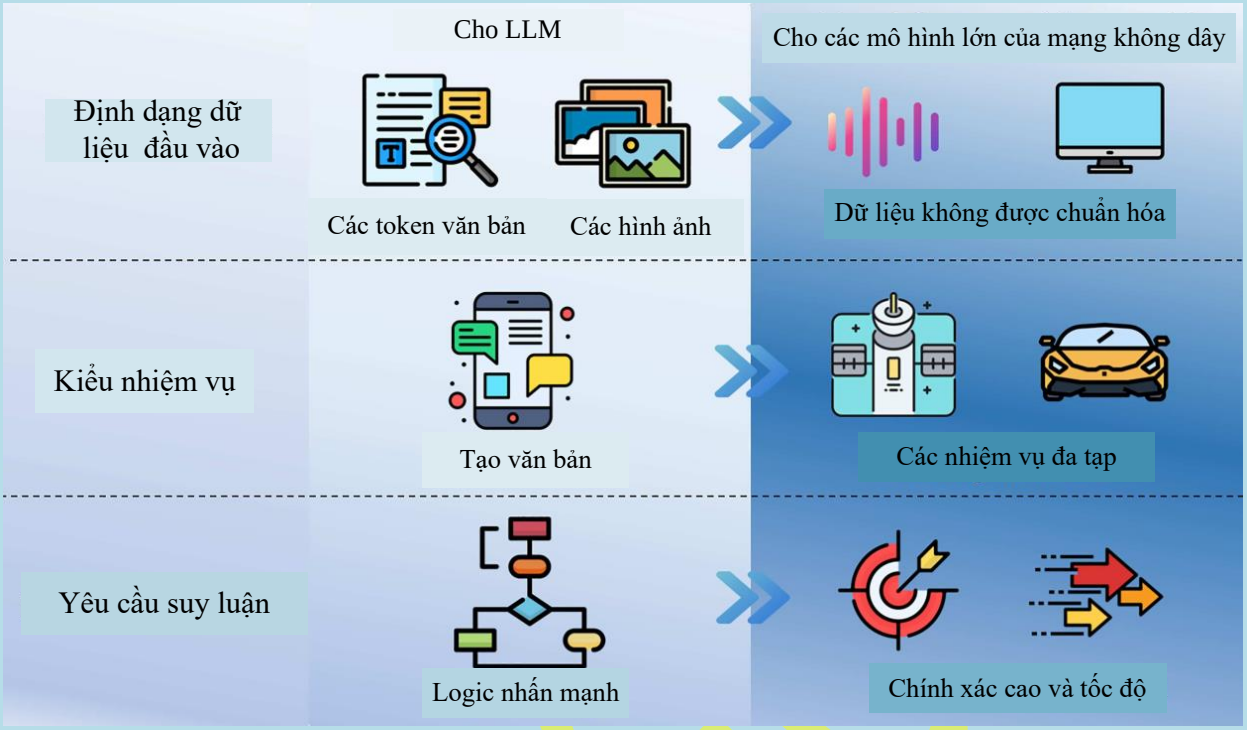
11.7. MÔ HÌNH LỚN CỦA MẠNG KHÔNG DÂY

LAM (Large Scale AI Model: mô hình AI quy mô lớn) thể hiện các khả năng tuyệt vời và tiềm năng lớn trong học thuật và các ngành nghề. Tuy nhiên, khuynh hướng này cũng tạo

ra các thách thức mới cho cơ sở hạ tầng mạng nền tảng. Các mạng không dây đóng một vai tròcực kỳ quan trọng hạ tầng này và trực tiếp liên quan đến hiệu suất của LAM (chẳng hạn: truyền dẫn, tốc độ xử lý dữ liệu v.v…). Phần này nhắm đến khám phá vai trò then chốt của các mạng không dây trong việc hỗ trợ hoạt động của các LAM này và tiến hành phân tích ngữ nghĩa và phản ảnh các vấn đề liên quan đến các mạng không dây trong ngữ cảnh xây dựng các LAM.

11.7.1. So sánh giữa LLM và mô hình lớn của mạng không dây

LLM (Large Language Model: mô hình ngôn ngữ lớn) đã được chứng minh sự hứa hẹn của chúng trong các nhiệm vụ văn bản và hình ảnh. Áp dụng trực tiếp các LLM truyền thống cho các hệ thống truyền thông truyền thống là một thách thức, vì dữ liệu trong hệ thống này bao gồm dữ liệu của giao thức khác nhau trong các mạng truy nhập, CN và dữ liệu lớp ứng dụng từ lớp cao hơn. Ngoài ra, các dữ liệu này hoặc có cấu trúc hoặc không có cấu trúc và có thể đa tạp với các tính năng dựa theo thời gian và có quan hệ. Đặc biệt là giao diện vô tuyến liên quan đến truyền dẫn của kênh vô tuyến và quản lý lập lịch bao gồm đa dạng thông tin được tạo ra bởi các tiêu chuẩn truyền thông vô tuyến không dây khác nhau như LTE và 5G. CN xử lý định tuyến, xác thực và điều khiển dịch vụ của dữ liệu người dùng bao gồm nội dung nhạy cảm như hành vi người dùng và thông tin vị trí, đòi hỏi các mức độ cao của quyền riêng tư và an ninh. Dữ liệu quản lý hoạt dộng và giám sát trạng thái, hội chẩn sự cố và tối ưu hóa hiệu suất của thiết bị mạng. Các tập dữ liệu này thông thường là quy mô lớn, đa chiều, thiếu các thiết kế hiệu quả để thống nhất hóa dữ liệu trên khắp các tiêu chuẩn khác nhau. Các đặc trưng này cẫn xem xét các yêu tố khác nhau như các quan hệ không gian thời giam, các yêu cầu thời gian thực và sự ổn định hệ thống khi xử lý và phân tích dữ liệu truyền thông không dây, trình bày các khác biệt và các thách thức đáng kể so với các cách tiếp cận xử lý dữ liệu LLM truyền thống, như được thể hiện trong hình 11.18.



Hình 11.18. Các khác biệt của LLM và mô hình lớn của mạng không dây

Mặc dù LLM đã phát triển đáng kể xử lý văn bản, nhưng áp dụng trực tiếp chúng để xây dựng mô hình lớn của mạng không dây là không thể. Trong các ứng dụng cụ thể, việc tích hợp chuyên môn trong một lĩnh vực cụ thể với các kỹ thuật xử lý dữ liệu tiên tiến là rất quan trọng cho xử lý và phân tích hiệu quả dữ liệu truyền thông không dây, xử lý các phức tạp của nó và các thách thức biến động.

11.7.2. Các cách tiếp cận tiềm năng để xây dựng một mô hình lớn của mạng không dây

Mô hình lớn của mạng không dây được đề cập như là một giải pháp thông minh và hiệu quả cho các hoạt động mạng truyền thông, đặt được thông qua kiến thức và công nghệ từ lĩnh vực truyền thông. Một mô hình lớn của mạng không dây chung cấp quản lý mạng, phát hiện sự cố của các hoạt động và các dịch vụ hội chẩn. Các hỏng hóc trong mạng có thể dẫn đến giảm cấp hiệu suất và gián đoạn dịch vụ. Bằng cách phân tích dữ liệu mạng khối lượng lớn, mô mô hình lớn của mạng không dây có thể giám sát mạng thời gian thực, nhanh chóng xác định và định vị các nguyên nhân tiềm năng và cung cấp các chiến lược chẩn đoán chính xác.

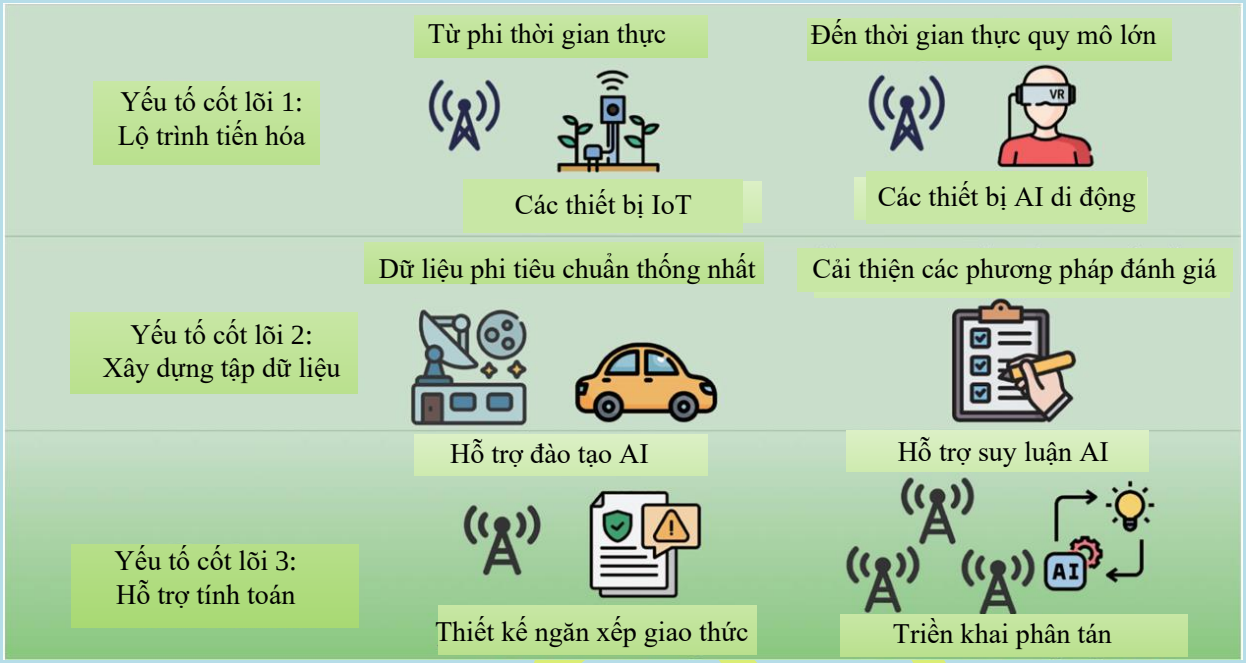
Mộ mô hình lớn của mạng truyền thông không dây tối ưu hóa điều phối nhiệm vụ và lập lịch. Các mạng truyền thông hiện dại liên quan đến đa dạng các nhiệm vụ và xử lý kinh

doanh phức tạp, đòi hỏi quản lý và lập lịch hiệu quả để đảm bảo các hoạt động của mạng. Thông qua một phân tích có chiều sâu về topo mạng và các yêu cầu kinh doanh, một mô hình lớn của mạng không dây tối ưu hóa các chuỗi thực hiện nhiệm vụ và các ấn định tài nguyên, tăng cường sử dụng tài nguyên mạng và xử lý kinh doanh hiệu quả.

Liên quan đến tối ưu hóa hiệu suất và ấn định tài nguyên, mô hình lớn của mạng không dây áp dụng các chiến lược và các gải thuật thông minh. Bằng cách giám sát liên tục tải mạng, các tình huống lưu lượng và tích hợp các cơ chế phân tích dự báo và điều chỉnh động, một mô hình được đề xuất bởi một số tác giả đã đạt được tối ưu hóa hiệu suất mạng và ấn định tài nguyên hợp lý, đảm bảo sự ổn định và sự tin cậy của mạng. Quan điểm công nghiệp phổ biến đề xuất rằng mô hình lớn của mạng không dây phải tiếp nhận một cấu trúc ba lớp bao gồm L0, L1 và L2.

- Lớp L0 trình bày mô hình lớn toàn năng cho toàn bộ mạng. Thiết kế của lớp này cần xem xét các cơ chế kiến trúc và hoạt động tổng thể của các hệ thống truyền thông để đảm bảo nó có các khả năng tổng quát hóa mạng. Thông qua việc phân tích dữ liệu toàn diện và dưa ra quyết định thông minh, nó nhắm đến cung cấp các dịch vụ tối ưu hóa và quản lý cho tất cả các nhiệm vụ truyền thông.
- Lớp L1 bao gồm các mô hình lớn đặc thù lĩnh vực. Thiết kế của L1 nhấn mạnh quá trình lập mô hình và tối ưu hóa cho các thách thức và các vấn đề của lĩnh vực. Chẳng hạn, mô hình lớp L1 có thể phân tích các điều kiện kênh không dây và hành vi người dùng trên giao diện vô tuyến để cải thiện tốc độ truyền dẫn và vùng phủ. Mô hình lớp L1 có thể sử dụng topo mạng và thông tin trạng thái thiết bị cho hội chẩn sai lỗi và tối ưu hóa lập lịch trong CN và các hoạt động.
- Lớp L2 có thể cung cấp cá giải pháp tùy chỉnh được thiết kế cho các kịch bản đặc thù trong khung này. Chẳng hạn, mô hình lớp L2 có thể tối ưu hóa hiệu suất sử dụng năng lượng trong việc bảo tồn năng lượng trong mạng bằng cách phân tích lưu lượng dữ liệu và các tải thiết bị. Liên quan đến dự đoán năng lượng và giám sát sai lỗi, mô hình lớp L2 có thể sử dụng các kỹ thuật DL để nhận dạng các dị thường và phát đi các cảnh báo sớm, nhờ vậy tăng cường sự ổn định và tin cậy của mạng.

Hình 11.19. cho thấy chiết lý xây dựng cho vệc xây dựng một mô hình lớn của mạng không dây trong các mạng truyền thông, bao gồm lộ trình tiến hóa, xây dựng bô dữ liệu và hỗ trợ tính toán, bổ sung lẫn nhau để tạo thành khung nền tảng của các mô hình lớn của mạng không dây.



Hình 11.19. Triết lý xây dựng mô hình lớn của mạng không dây

Yếu tố cốt lõi 1: lộ trình tiến hóa. Xây dựng một mô hình lớn có khả năng áp dụng phổ quát cho toàn bộ mạng đối mặt nhiều thách thức. Mặt khác, dữ liệu trong các mạng truyền thông thể hiện các đặc trưng có cấu trúc cao, bao gồm thông tin toàn diện, mở rộng và bao quát từ các thiết bị và các bộ cảm biến khác nhau, gồm có lưu lượng mạng, trạng thái thiết bị và hành vi người dùng. Sự đa dạng và phức tạp của các dữ liệu này thể hiện nhiều thách thách thức cho việc tối ưu hóa mô hình. Mặt khác, các nhiệm vụ trong các mạng truyền thông có thể phức tạp và độc lập. Chẳng hạn các đặc tính của lưu lượng mạng có thể thay đổi đáng kể trên các kịch bản kinh doanh khác nhau làm cho các cách tiếp cận lập mô hình thông nhất truyền thống kém khả thi. Cách tiếp cận theo từng giai đoạn để lập xây dựng mô hình trở nên quan trọng cho xử lý hiệu quả các thách thức nói trên. Tập trung đầu tiên có thể là quy mô nhỏ, các mô hình lớp 2 ngoại tuyến. Thông qua xác nhận thực tế các kịch bản của từng công việc quan trọng khác nhau như hiệu suất năng lượng mang, dự đoán lưu lượng và giám sát sai lỗi, có thể đạt được sử dụng hiệu quả các tài nguyên mạng. Khi thiết kế các mô hình mức độ L2, nhấn mạnh sẽ đặc đặt lên xử lý các thách đặc thù lĩnh vực, sử dụng các cách tiếp cận lập mô hình và tối ưu hóa theo mục tiêu để đạt được các khả năng tổng quát hóa lĩnh vực trên giao diện vô tuyến, các ứng dụng và phía CN. Để xây dựng một mô hình mức độ L0 có thể áp dụng phổ quát bao gồm toán mạng, việc xem xét cẩn thận kiến trúc tổng thể của hệ thống truyền thông và

các cơ chế hoạt động là vấn đề thiết yếu để đảm bảo khả năng của mô hình hỗ trợ các dịch vụ và các chiến lược cho các nhiệm vụ truyền thông.

Yếu tố cốt lõi 2: Xây dựng tập dữ liệu. Xây dựng tập dữ liệu là vấn đề thiết yếu cho các mô hình mạng quy mô lớn, ảnh hưởng trực tiếp lên công hiệu của đào tạo mô hình AI và ứng dụng. Các thách thức xoay quanh các khó khăn thu thập và chất lượng dữ liệu. Để xử lý các khó khăn trong việc thu thập dữ liệu, phải khám phá nhiều cách thức, bao gồm cộng tác chủ động với các đối tác ngành để thúc đẩy chia sẽ và cởi mở, liên quan đến các nhà mạng viễn thông, các nhà sản xuất thiết bị và các đối tác khác để mở rộng vùng phủ của các tập dữ liệu. Ngoài ra việc thiết lập các tiêu chuẩn và các hướng dẫn ngành có thể hướng dẫn các bên liên quan tuân thủ các tiêu chuẩn thu thập và xử lý dữ liệu thống nhất, tạo điều kiện chia sẻ và trao đổi dữ liệu để khắc phục các thách thức thu thập dữ liệu. Một khung đánh giá chất lượng dữ liệu toàn diện cũng không thể thiếu. Sử dụng các số đo chất lượng dữ liệu như sự chính xác, sự hoàn thiện và sự nhất quán, dữ liệu phải được đánh giá và giám sát để phát hiện mọi sự bất thường của chất lượng dữ liệu và đưa ra các biện pháp tương ứng cho cải thiện và tối ưu hóa. Áp dụng các công cụ và các công nghệ quản lý chất lượng dữ liệu có thể tự động kiểm tra và xác thực chất lượng . Trong quá trình thu thập dữ liệu, điều quan trong là phải tăng cường giám sát và quả lý các nguồn dữ liệu để đảm bảo sự toàn diện và tin cậy của thu thập dữ liệu. Các kỹ thuật làm sạch và xử lý tước dữ liệu cần được áp dụng để loại bỏ tạp âm và xử lý các giá trị thiếu trong dữ liệu thô. Ngoài ra, việc thiết lập các thủ tục tiêu chuẩn hóa dữ liệu bền vứng để thống nhất các định dạng sẽ giảm sự hỗn tạp và các không nhất quán của tập dữ liệu.

Yếu tố cốt lõi 3: Hỗ trợ tính toán. Liên quan đến các yêu cầu tính toán của truyền thông 6G để hỗ trợ các mô hình mạng quy mô lớn, các nỗ lực tối ưu hóa cần có cả trong đào tạo và triển khai AI. Điều này nhắm đến truyền dẫn dữ liệu và sử dụng các tài nguyên tính toán hiệu quả, tin cậy ổn định. Trong các dịch vụ AI, các yếu tố quan trọng gồm có thu thập dữ liệu đào tạo hiệu quả, tin cậy, các xử lý truyền dẫn, thiết kế giao thức độc quyền, hỗ trợ từ các thành phần mạng và các giải thuật lập lịch lưu lượng tùy chỉnh. Khi thiết kế cá mạng truy nhập và lõi, các giao thức độc quyền được thiết kế cho cho truyền dẫn nhiệm vụ AI cần được phát triển. Các giao thức này cần xét đến các đặc trưng độc nhất của các nhiệm vụ AI, như khối lượng dữ liệu và cường độ tính toán lớn để đảm bảo hiệu quả và hiệu suất truyền dẫn tối ưu. Khi thiết kế các phần tử mạng để hỗ trợ truyền dẫn nhiệm vụ AI, cần xem xét các vấn đề liên quan đến kiến trúc phần cứng và chức năng phần mềm để xử lý và phát các nhiệm vụ đào tạo AI hiệu qun. Khi lập bịch lưu lượng, việc chọn lựa các chiến lược phù hợp là vấn đề then chốt để hỗ trợ đào tạo các LAM. Các chiến lược lập lịch lưu lượng thông thường không đủ cho các nhiệm vụ đào tạo AI, cần

có thiết kế các giải thuật lập lịch tùy chỉnh được thiết kế riêng cho các đặc trưng của các nhiệm vụ AI. Các giải thuật này cần tính đến các thuộc tính của nhiệm vụ, các điều kiện của tải mạng và tính khả dụng tài nguyên của thiết bị để đảm bảo các nhiệm vụ đào tạo AI nhận được các tài nguyên mạng và tính toán kịp thời.

Trong các dịch vụ suy luận AI, các yiêu cầu cao của LAM về không gian lưu trữ và các khả năng tính toán thường vượt quá yêu cầu mà một BS có thể cung cấp. Cáhcch tiếp cận khả thi là thiết kế một kiến trúc hiệu quả sử dụng hợp tác các nút phân tán để cùng nhau xử lý lưu trữ mô hình và các nhiệm vụ suy luận. Bên cạnh việc thiết kế kiến trúc hợp tác các nút phân tán, quản lý và lập lịch hiệu qủa các nhiệm vụ suy luận là các vấn đề then chốt để cực đại hóa sử dụng các tài nguyên tính toán của từng nút trong khi vẫn đảm bảo hiệu quả và ổn định hệ thống. Một xem xét quan trọng là thiết kế các giải thuật ấn định và lập lịch nhiệm vụ. Các tiếp cận này cho phép ấn định và lập lịch hợp lý các nhiệm vụ suy luận. Ngoài ra, cần chú ý tăng cường giám sát và quản lý tự động hệ thống. Các công cụ và cơ chế quản lý tự động, như lập cấu hình và triển khai tự động, chẩn đoán sự cố và khôi phục có thể được đưa vào để phát hiện và xử lý kịp thời các sự cố và các sự bất thường. Điều này giảm các chi phí quản lý và bảo dưỡng, cải thiện khă năng bảo dưỡng và khả năng quản lý. Các biện pháp tối ưu này cùng nhau cung cấp hỗ trợ kỹ thuật và sự đảm bảo tốt hơn cho các khả năng tính toán mô hình AI lớn trong các mạng 6G.

11.7.3 Các thách thức triển khai mô hình lớn của mạng không dây

- **Sự đa tạp của dữ liệu**. Trong các mạng không dây, sự đa dạng và phức tạp của dữ liệu đặt ra các thách thức đáng kể trong việc xây dựng corpus (kho ngữ liệu: một tập hợp lớn có hệ thống các văn các văn bản hoặc âm thanh được tổng hợp và sắp xếp theo một quy tắc nhất định) đào tạo và kiểm tra quy mô lớn toàn diện. Các mạng chứa các kiểu dữ liệu có cấu trúc và không có cấu trúc khác nhau, bao gồm thông tin kênh đường lên và đường xuống, các báo cáo đo đạc, cực tiểu hóa dữ liệu, kiểm tra trên xe, dữ liệu giám sát KPI, các chỉ số hiệu suất topo mạng, các nhật ký chi phí và tiêu thụ năng lượng. Các nguồn dữ liệu có tính đa tạp rất cao và các dữ liệu từ các nguồn gốc khác nhau thường thể hiện sự khác bệt rõ rệt về độ phân giải chi tiết và thời gian. Cần xem xét cách thức căn chỉnh hiệu quả các dữ liệu đa tạp này để xây dựng một tập dữ liệu nhất quán và đại diên. Việc khám phá các kỹ thật như chuẩn hóa dữ liệu, căn chỉnh chuỗi thời gian theo thời gian và hòa nhập dữ liệu có thể đóng góp cho việc xây dựng corpus đào tạo chất lượng cao. Ngoài ra, cần chú ý chuyển đổi dữ liệu đầu vào đã xử lý vào các định dạng đầu vào phù hợp cho các giải thuật suy luận. Điều này có thể bao gồm chiết xuất và lựa chọn tính năng, giảm chiều dữ liệu và tạo ra các trình bày đầu và phù hợp để cho phép mô hình thực hiện suy luận và đưa ra quyết định hiệu quả.

- **Suy luận thời gian thực**. Mô hình lớn của mạng không dây có thể được áp dụng cho các nhiệm vụ đưa ra quyết định thời gian thực, như ấn định phổ động, điều khiển công suất và quản lý truy nhập người dùng. Điều này đặc biệt quan trong cho lập lịch tài nguyên mạng, nơi mà yêu cầu các ràng bộc thời gian chặt chẽ với suy luận hoàn tất trong vài mi-li giây để quản lý và ấn định hiệu quả các tài nguyên mạng truyền thông. Điều này trực tiếp ảnh hưởng đến sự đáp ứng thời gian thực và chất lượng dịch vụ của mạng. Để đáp ứng các yêu cầu này, thiết kế và triển khai các mô hình quy mô lớn trong các mạng không dây phải ưu tiên tối ưu hóa tốc độ suy luận, bao gồm tăng tốc phần cứng, tối ưu hóa giải thuật và điều chỉnh kiến trúc mạng. Ngoài ra, việc đạt được cân bằng sự chính xác của mô hình và hiệu quả tính toán là vấn đề then chốt cho quản lý và lập lịch tài nguyên mạng hiệu quả và tin cậy.
- **Định tuyến mơ hồ**. Việc xây dựng một khung mô hình quy mô lớn toàn năng cho truyền thông không dây là một nhiệm vụ đầy thách thức. Thực tiễn công nghiệp hiện thời có xu hướng sử dụng các mô hình ngôn ngữ lớn được đào tạo trước như là các phần tử nền tảng, sau đó tinh chỉh chúng cho các ứng dụng công nghiệp đặc thù. Tuy nhiên sự phức tạp và tính động của các mạng không dây đòi hỏi tính khả dụng và thích ứng cao hơn. Các nhà nghiên cứu đã đạt được một số tiến bộ trong việc khám phá các mô hình dựa trên truyền thông ngữ nghĩa cho phép đạt được tiến bộ trong việc hiểu biết và xử lý các luồng dữ liệu trong các mạng không dây. Tuy nhiên, tính khả thi và hiệu quả của các mô hình này trong các triển khai thực tế bao gồm khả năng ổn định và hiệu suất trong các môi trường và điều kiện thay đổi vẫn chưa được giải quyết và đòi hỏi tiếp tục xác thực bằng thực nghiệm.
- **Khả năng tương thích giao thức.** Việc tich hợp và triển khai mô hình quy mô lớn của mạng truyền thông không tránh khỏi dựa trên ngăn xếp giao thức của mạng truyền thông. Tuy nhiên các công nghệ mô hình quy mô lớn chỉ mới xuất hiện trong những năm gần đây, các vấn đề tương thích với các ngăn xếp giao thức truyền thông hiện có vẫn còn cần được xử lý. Việc liệu kiến trúc ngăn xếp giao thức hiện thời có thể đáp ứng các yêu cầu tương lai hay không và làm sao để tăng cường và tối ưu hóa giao thức hiện có để hỗ trợ tốt hơn ứng dụng của các mô hình này trong các mạng không dây đặt ra các thách thức kỹ thuật lớn. Điều này đòi hỏi phân tích ngăn xếp giao thức hiện có để nhận dạng các rào cản tiềm năng cho việc tích hợp mô hình quy mô lớn và khám phá các nguyên tắc thiết kế giao thức mới để đảm bảo hoạt động hiệu quả và ổn định của các mô hình quy mô lớn của mạng không dây.
- **Các mối đe dọa an ninh**. Mô hình lớn của mạng không dây đối mặt với các thách thức an ninh. Thứ nhất, nó có thể gặp các tấn công quyền riêng tư. Các tập dữ liệu từ các mạng 6G có thể liên quan đến thông tin nhạy cảm như vị trí, mẫu di động của người dùng và nội dung truyền thông . Không có sự bảo vệ tương xứng, các dữ liệu

này dễ bị tổn thương đối với các mối đe dọa quyền riêng tư, có thể làm mất niềm tin của người dùng và các vấn đề pháp lý. Thứ hai, sự giả mạo dữ liệu là một mối đe dọa đáng kể khác đối với các mô hình quy mô lớn. Các kẻ tấn công có thể gây ảnh hưởng lên quá trình đào tạo và các kết cục đưa ra quyết định bằng cách làm sai lệch hoặc chỉnh sửa các nguồn dữ liệu. Chẳng hạn, dữ liệu đầu vào sai có thể dẫn đến điều chỉnh sai các yêu cầu của người dùng hay các tình trạng nghẽn mạng. Ngoài ra các tấn công mạng nhắm đến các mô hình của mạng viễn thông quy mô lớn cũng cũng độc hại. Các tác nhân độc hại có thể làm gián đoạn các hoạt động của mô hình hay các yêu cầu thường xuyên thông qua các tấn công lũ lụt (Flooding Attack) hay các tấn công làm cạn kiệt tài nguyên, ảnh hưởng nghiêm trọng sự khả dụng dịch vụ và các khả năng đưa ra quyết định thời gian thực. Cuối cùng, đảm bảo an ninh và tính toàn vẹn của các kiến trúc mô hình quy mô lớn là vấn đề hết sức quan trọng. Kỹ thuật đảo ngược (Reverse Techhology)và sự rò rỉ thông số mô hình có thể cho phép các kẻ tấn công lấy được thông tin nhạy cảm về thuộc tính trí tuệ hay cơ chế hoạt động, nhờ đó gây nguy hiểm cho sự ổn định và tin cậy của các dịch vụ.

11.8. TỔNG KẾT

Chương này trước hết trình bày tổng quan tầm nhìn hệ thống truyền thông di động 6G. Tiếp theo chương trình bày nhấn mạnh những điểm quan trong nhất của phát triển 6G đên một hạ tầng không dây tích hợp truyền thông, cảm biến tính toán trí tuệ và lưu trữ dựa trên tích hợp sâu rộng AI và truyền thông. Sau đó chương trình bày ba giai đoạn tích hợp của AI với truyền thông, cụ thể là "AI4NET", NET4AI" và "AIaaService". Ba giai đoạn này thể hiện sự dịch chuyển chiến lược từ việc sử dụng AI cho tối ưu hóa mạng, đến việc tạo điều kiện cho hỗ trợ của AI cho mạng để cung cấp AIaaService, qua đó nhấn mạnh quan hệ độc lập và cộng tác giữa truyền thông và AI. Mạng 6G sẽ đưa ra các mô hình dịch vụ mới, tích hợp liền mạch các khả năng của AI vào các kịch bản ứng dụng khác nhau, qua đó phục vụ một nền tảng mạnh mẽ hỗ trợ sự tiến bộ của AI. Mô hình quy mô lớn của mạng không dây đang sẵn sàng phục vụ như là các nhân tố then chốt trong các mạng 6G cho phép tích hợp hiệu quả AI với các công nghệ truyền thông để phát triển các hoạt động mạng thông minh, tự quản và hiệu quả cao. Chương này cũng chỉ ra các thách thức phải đối mặt khi tích hợp AI và truyền thông trong kỷ nguyên 6G, bao gồm các cân nhắc về tính bền vững, các yêu cầu AI thời gian thực, các vấn đề an ninh và quyền riêng tư, cũng như tương tác và hợp tác giữa con người và AI. Chương cũng vạch ra các cơ hội nghiên cứu tương lai đầy triển vọng, được kỳ vọng để phát triển triển khai, tôi ưu hóa AI và truyền thông 6G.

Chương 12

RAN MỞ (OPEN- RAN) VÀ HOẠT ĐỘNG AI/ML TRONG O-RAN

12.1. TỔNG QUAN

RAN là phần tử cốt lõi của hạ tầng truyền thông di động tổ ong. RAN tạo thành một liên kết thiết yếu giữa thiết bị người dùng (UE) và mạng lõi (CN). Khi yêu cầu thông lượng cao hơn và mật độ thiết bị tăng cao, các kiến trúc RAN phát triển đáng kể, chuyển đổi qua nhiều thế hệ các mô hình triển khai.

Quá trình phát triển của của các kiến trúc RAN như sau.

1) **RAN phân tán (D-RAN: Distributed RAN)**. Trong mô hình D-RAN, các đơn vị xử lý băng gốc (BBU: Baseband Processing Unit) và vô tuyến (RU: Radio Units) được đặt cùng vị trí tại từng site, được tích hợp chặt chẽ trong một hệ sinh thái của nhà cung cấp . Mặc dù tin cậy, kiến trúc này dẫn đến sử dụng tài nguyên không hiệu quả và các chi phí triển khai cao.
2) **RAN tập trung (C-RAN: Centralized RAN).** C-RAN tập trung các BBU trong một vị trí dùng chung như là một trung tâm dữ liệu, cho phép chúng phục vụ niều RU. Điều này cải thiện hiệu suất năng lượng, phổ tần giảm các chi phí khai thác nhưng đưa ra các yêu cầu chặt chẽ về trễ và băng thông lên các liên kết tuyến trước.
3) **RAN ảo hóa (vRAN: Virtualized RAN).** vRAN sử dụng các nguyên lý ảo hóa chức năng mạng (NFV: Network Fucntion Virtualization) và nối mạng được định nghĩa bằng phần mềm (SDN: Software Defined Networking) để ảo hóa các BBU trên các phần cứng COTS (Commerchial off-the Shelf: phần cứng thương mại có sẵn). Trong khi vRAN tăng khả năng linh hoạt, thì nó thường dựa trên các RU bản quyền và các giao diện không chuẩn hóa dẫn đến hạn chế tính trung lập hoàn toàn của nhà cung cấp.
4) **RAN mở (O**-RAN: Open RAN). O-RAN xây dựng trên vRAN băng việc đưa vào các giao diện hoàn toàn mở và được chuẩn hóa, cho phép phân nhỏ phần cứng và phần mềm. Các nhà mạng có thể hòa trộn và khớp nối các thành phần , giảm các chi phí và thực hiện các khả năng tiên tiến như cắt lát mạng và điều phối đám mây gốc.

Mức độ phức tạp của các mạng di động tổ ong không ngừng tăng cùng với các hệ thống không dây thế hệ tiếp theo được xây dựng trên nền tảng của các công nghệ và các băng tần không đồng nhất. Các nghiên cứu phát triển mới bao gồm MIMO (Multi Input Multi

Output: nhiều đầu vào nhiều đầu ra), sóng milimet, sub-tetrahez và tetrrahez, cảm biến dựa trên mạng, cắt lát mạng và các xử lý tín hiệu dựa trên AI/ML cùng với nhiều nghiên cứu phát triển khác. Điều này dẫn đến các chi phí đầu tư và khai thác không ngừng tăng đối với cá nhà mạng, họ phải liên tục nâng cấp và duy trì hạ tầng của mình để bắt kịp xu thế thị trường, công nghệ mới và các yêu cầu của khách hàng.

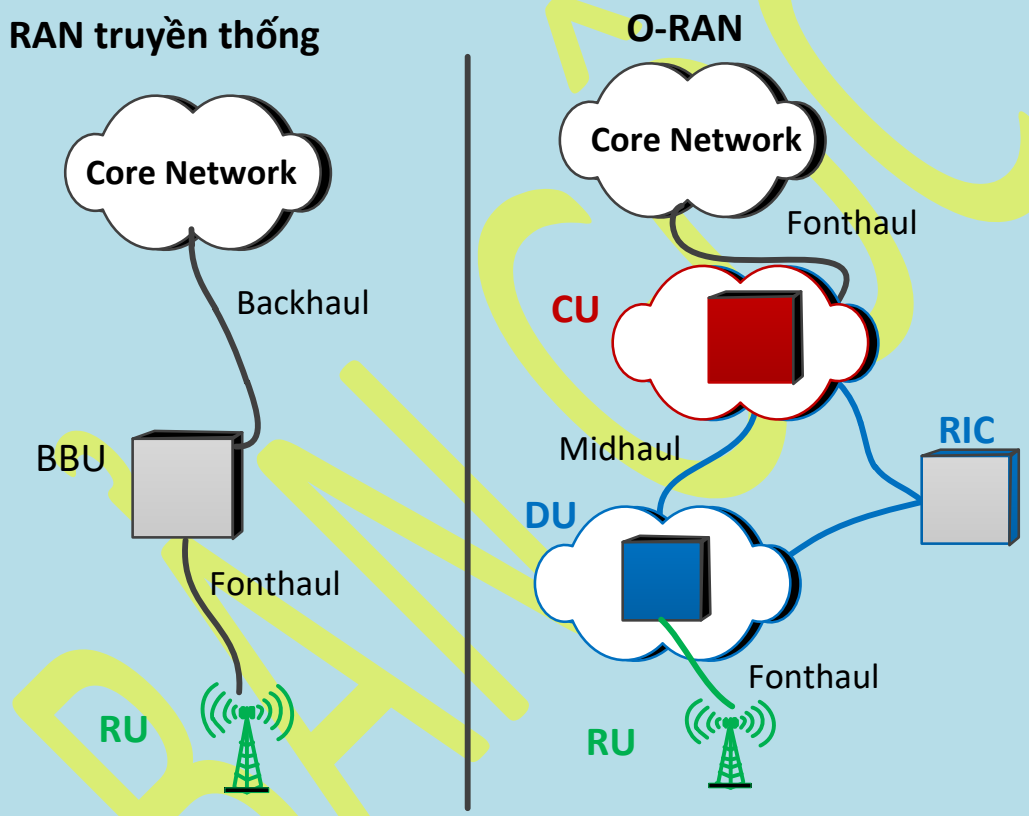
Quản lý và tối ưu hóa các hệ thống mạng này đòi hỏi các giải pháp cho phép để mở mạng truy nhập vô tuyến (RAN). Điều này cho phép để lộ số liệu và phân tích và cho phép tối ưu hóa được vận hành bởi số liệu, điều khiển vòng kín và tự động hóa. Các cách tiếp cận hiên này còn rất xa với để mở. Ngày nay các thành phần RAN là các đơn vị đơn thể, các giải pháp tất cả trong một (All-in-One Solution) ,chúng thực hiện từng lớp và mọi lớp của ngăn xếp giao thức tổ ong. Chúng được cung cấp bởi một số lượng hữu hạn các nhà cung cấp và được các nhà mạng nhìn nhận như là các hộp đen. Sự tin cậy vào các giải pháp hộp đen dẫn đến: (i) khả năng lập lại cấu hình RAN bị hạn chế, do không thể vi chỉnh hoạt động của thế bị đến hộ trợ các triển khai đa dạng và các hồ sơ lưu lượng khác nhau; (ii) điều phối giữa các nút mạng khác nhau bị hạn chế dẫn đến cản trở tối ưu hóa và điều khiển RAN kết hợp, (iii) đóng kín của nhà cung cấp với các tùy chọn hạn chế đối với các nhà khai thác triển khai và giao diện thiết bị RAM từ nhiều nhà cung cấp. Trong các hoàn cảnh này, quả lý tài nguyên vô tuyn tối ưu và sử dụng phổ tần hiệu quả thông qua thích ứng thời gian thực trở nên vô cùng thách thức.

Để khắc phục các hạn chế này, trong những năm gần đây một số nỗ lực nghiêu cứu và chuẩn hóa đã đề xuất RAN mở (O-RAN: Open RAN), một mô hình mới đối với RAN tương lai. Phát triển Open RAN dựa trên các thành phần được chia nhỏ, được ảo hóa và dựa trên phần mềm, được kết nối thông qua các giao diện được định nghĩa mở và rõ ràng, có phả năng tương tác giữa các nhà cung cấp khác nhau. Chia nhỏ và ảo hóa cho phép thực hiện các triển khai linh hoạt dựa trên các nguyên lý điện toán đám mây gốc (Cloud Native). Điều này làm tăng khả năng phục hồi và khả năng lập lại cấu hình của RAN. Các giao diện mở và khả tương tác cũng cho phép các nhà khai thác hòa nhập các nhà cung cấp thiết bị khác nhau, mở ra hệ sinh thái cho các người chơi nhỏ hơn. Cuối cùng, các giao diện mở và các ngăn xếp giao thức định nghĩa bằng phần mềm cho phép tích hợp điều khiển thông minh vòng kín được vận hành bởi dưa dữ liệu đối với RAN.

O-RAN Alliance (liên minh RAN mở) là một tổ chức công nghiệp được tạo lập năm 2018 có mục đích thực hiện các nguyên lý này trên 3GPP LTE và 5G RAN. Đặc biệt O-RAN Alliance mở rộng 3GPP NR 7.2 phân chia trạm gốc thành CU (Central UInit), DU (Distributed Unit) và RU (Radio Unit.) O-RAN kết nối chúng bằng các bộ điều khiển

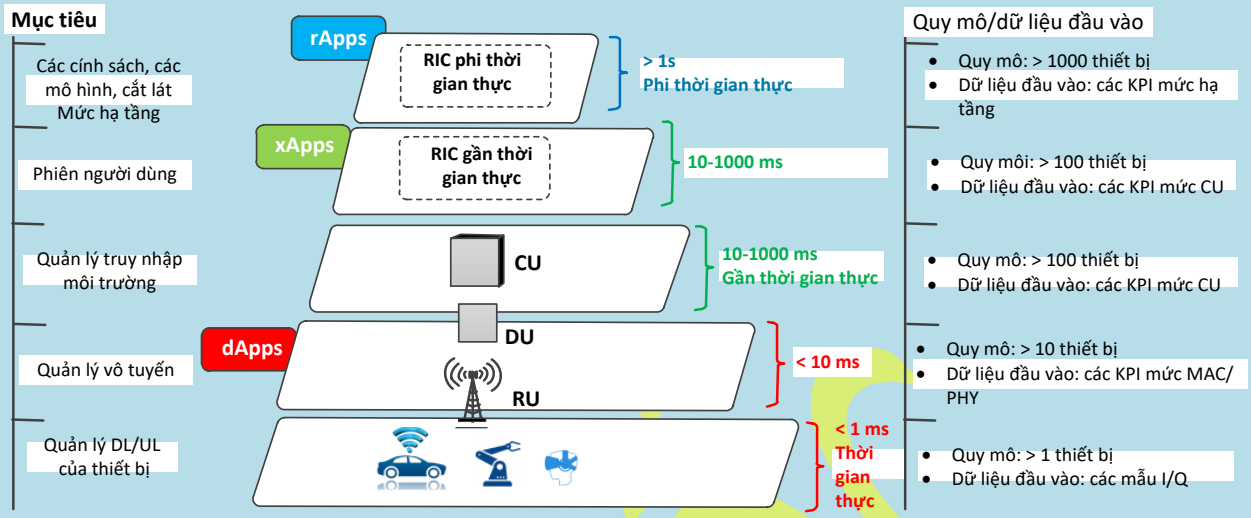
thông minh (Intellegence Controller) thông qua các giao diện cho phép truyền trực tuyến đo từ xa từ RAN và triển khai các hành động điều khiển và các chính sách đến nó. Kiến trúc O-RAN bao gồm hai bộ điều khiển trí tuê RAN (RIC: Radio Intellegence Controller) thực hiện quản lý và điều khiển mạng gần thời gian thực (10ms đến 1 s) và phi thời gian thực (hơn 1 s). Cuối cùng, O-RAN Alliance định nghĩa nền tảng ảo hóa cho RAN và mở rộng định nghĩa của 3GPP và các giao diện eCPRI để kết nối các nút RAN.

Hình 12.1và hình 12.2 cho thấy sự phát triển từ RAN truyền thống đến O-RAN và các vòng kín được tối ưu hóa trong O-RAN thông minh.



Core Network: mạng lõi; RAN (Radio Access Network): mạng truy nhập vô tuyến; O-RAN (Open Radio Access Network): mạng truy nhập vô tuyển mở; BBU (Baseband Unit: đơn vị băng gốc; CU (Central Unit): đơn vị trung tâm; (DU (Distributed Unit): đơn vị phân tán; RU (Radio Unit): đơn vị vô tuyến; RIC (Radio Intellegencce Control): điều khiển vô tuyến thông minh; Backhaul: tuyến sau; Midhaul: tuyến giữa; Fronthaul: tuyến trước.

Hình 12.1. Sự phát triển từ RAN truyền thống đến O-RAN.



Hình 12.2. Các vòng điều khiển được tối ưu hóa trong O-RAN thông minh, cho thấy các mức chức năng từ các Apps thiết bị đế các Apps mức hạ tầng, quy mô và kiểu dữ liệu đầu vào của từng lớp.

Mô hình O-RAN đưa ra sử dụng một khung điều khiển mới để hỗ trợ trí tuệ theo mô-dun và theo nhiều thang thời gian thông qua các bộ điều khiển vô tuyến thông minh (Radio Intellegence Controler) như:

- **RIC phi thời gian thực (Non-RT RIC: Non Real Timfe Radio Intellegence Controler):** RIC phi thời gian thực được đặt bên trong lớp Quản lý và Điều phối dịch vụ (SMO: Service Management and Orchestration) và hoạt động tại các phạm vi thời gian lớn hơn 1 s. Nó thực hiện điều khiển chính sách mức cao, đào tạo mô hình AI/ML và tối ưu hóa dài hạn dựa trên các KPI (Key Performance Indicator: chỉ số hiệu suất chính) mức hạ tầng trên quy mô hơn 1000 thiết bị;
- **RIC gần thời gian thực (Near-RT RIC: Near Real Time Radio Intellegence Controler)**: hoạt động trong phạm vi thời gian từ 10 ms đến 1s. Nó quản lý các hoạt động RAN động, bao gồm di động, loại bỏ nhiễu và lập lịch. Nó sử dụng các KPI mức CU từ hàng trăm thiết bị để dáp ứng các thay đổi mạng thời gian thực.

Các ứng dụng thông minh được sử dụng trong O-RAN bao gồm:

- **Các rApps**: Chạy trên RIC phi thời gian thực. Các rApps thực hiện các nhiệêm vụ dài hạn như: thực thi chính sách, căt lát mạng và tối ưu hóa tài nguyên dưới sự hương dẫn của AI.
- **Các xAppx**: Các xApps thực hiện trên RIC gần thời gian thực để cung cấp điều khiển ngắn hạn các chức năng RAN, bao gồm lập lịch thời gian thực, điều phối nhiễu và quản lý chuyển giao .

- **Các dApps**: Các dApps được triển khai thại mức CU/DU và hoạt động với độ trễ dưới ms. Chúng phân tích các MAC/PHY KPI, nhớ đệm trạng thái và các mẫu I/Q làm cho chúng phù hợp cho các nhiệm vụ điều khiển trễ cực thấp.

Các thành phần của O-RAN trong các hình 12.1 và 12.2 sẽ được xét cụ thể trong các phần sau của chương.

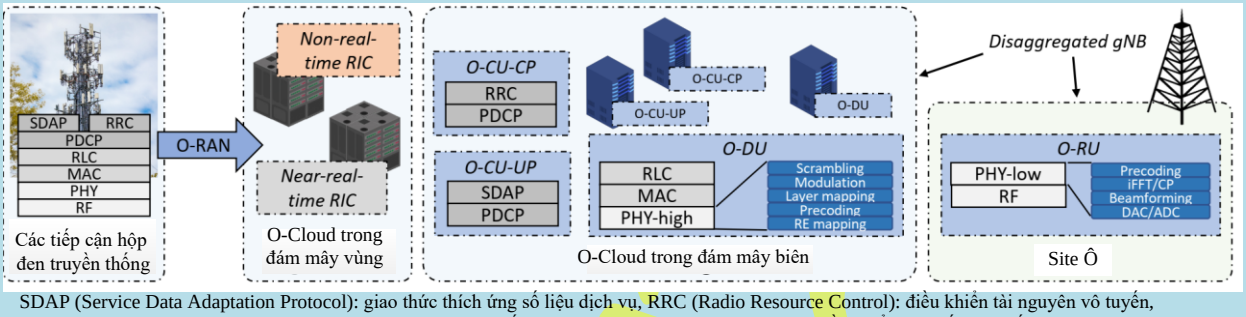
12.2 CÁC NGUYÊN LÝ KIẾN TRÚC THEN CHỐT CỦA O-RAN

Tầm nhìn của Open-RAN được xây dựng trên nhiều năm nghiên cứu các mạng mở và khả lập trình. Các nghiên cứu này là trung tâm của sự chuyển đổi kết nối mạng được định nghĩa bằng phần mềm (SDN: Software-defined Networking) và đã được đưa vào các mạng hữu tuyế trước đây. Chẳng hạn xRAN Forum do các nhà mạng khởi xướng đã đề xuất một giao diện tuyến trước (Fronthaul Interrface) chuẩn hóa và đưa ra ý tưởng các giao diện mở chuẩn hóa để tích hợp các bộ điều khiển bên ngoài vào RAN. Đồng thời hiến trúc RAN đám mây (C-RAN: Cloud RAN) (được đề xuất bởi các nhà mạng được dẫn dăt bởi CRAN Alliance) đã nổi lên như một giải pháp tập trung hầu hết các xử lý băng gốc cho RAN vào các trung tâm dữ liệu đám mây được kết nối đến các dơn vị vô tuyế đặt xa thông qua các giao diện tyến trước tốc độ cao hơn. C-RAN cho phép xử lý tín hiệu chi tiết hơn, các kỹ thuận cân bằng cách sử dụng dữ liệu tập trung và các tuyến điều khiển, giảm chi phí nhờ việc ghép chung các tài nguyên tính toán.

12.2.1. Chia nhỏ.

Hình 12.3 cho thấy sự chia nhỏ RAN phân chia các trạm gốc vào các đơn vị chức năng khác nhau, vì thể bao trọn hiệu quả và mở rộng mô hình cha nhỏ chức năng do 3GPP đề xuất cho các trạm gốc thế hệ sau của NR (gNB). gNB được chia thành CU (Central Unit: đơn vị trung tâm), DU (Distributed Unit: đơn vị phân tán) và RU (Radio Unt: đơn vị vô tuyến), (được gọi là O-CU, O-DU và O-RU trong các đặc tả của O-RAN). CU được chia tiếp vào hai thành phần logic, một cho mặt phẳng điều khiển (CP: Control Plane) và một cho mặt phẳng người dùng (UP: User Plane). Phân chia logic này cho phép triển khai các chức năng khác nhau tại các vị trí khác nhau của mạng cũng như trên các nền tàng phần cứng khác nhau. Chẳng hạn, các CU và DU có thể được ảo hóa trên các hộp máy chủ biên (với tăng tốc phần ứng đối với một số chức năng lớp vật lý), trong khi các RU thường được thực trên các FPGA (Field Programable Gate Array: mảng cổng khả lập

trình theo hiện trường) và các bo mạch ASIC (Applicatio-specific Integrated Circuit: mạch tích hợp đặc thù ứng dụng), được triển khai gần các anten RF.



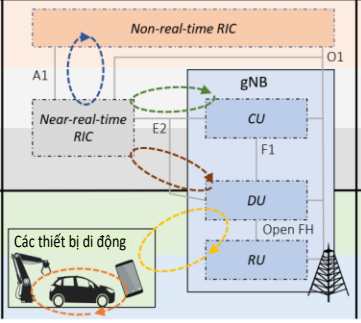
SDAP (Service Data Adaptation Protocol): giao thức thích ứng số liệu dịch vụ, RRC (Radio Resource Control): điều khiển tài nguyên vô tuyến, PDCP (Packet Data Convergence Protocol): giao thức hội tụ số liệu gói, RLC (Radio Link Control): điều khiển liên kết vô tuyến, MAC (Medium Access Contrrol): điều khiển truy nhập môi trường, PHY (Physical Layer): lớp vật lý, RF (Radio Frequency) : tần số vô tuyến, PHY-high: c lớp vật lý cao, PHY-low: lớp vật lý thấp.
O-RAN (Open Radio Acess Network): mạng truy nhập vô tuyến mở, O-CU-CP (Open -Central Unit- Control Plane): mặp phẳng điều khiển của đơn vị trung tâm mở, O-CU-UP (Open-Central Unit- User Plane): mặt phẳng người dùng của đơn vị trung tâm mở, O-DU-CP (Open -Distributed Unit-Control Plane): mặp phẳng điều khiển của đơn vị phân tán mở, O-DU (Open -Distributed Unit): mặp phẳngngười dùng của đơn vị phân tán mở
RIC (Radio Intellegence Controller): bộ điều khiển vô tuyến thông minh, Non real-time RIC: RIC phi thời gian thực, Near real-time RIC: RIC gần thời gian thực, Scrambling: ngẫu nhiên hóa, Modulation: điều chế, Layer Mapping: sắp xếp lớp, Precoding: tiền mã hóa, RE Mapping: sắp xếp phần tử tài nguyên, iFFT/CP (Inverse Fast Fourier Tansfomation/Cyclic Preamble): biến đổi Fourrier nhanh ngươc/tiền tố chu trình, Beamfforming: tạo búp, DAC/ADC (Digital to Analog Convesion/ Analog to Digital Conversion): chuyển đổi số thành tương tự/ chuyển đổi tương tự thành số. Disaggregated gBB: gNB được chia nhỏ.

Hình 12.3. Chia nhỏ trạm gốc gNB vào các đơn vị chức năng khác nhau

Trong phân chia 7.2x, RU thực hiện các chức năng miền thời gian gồm tiền mã hóa, biến đổi Fuorier nhanh (FFT), bổ sung /loại bỏ tiền tố chu trình và các hoạt động tần số vô tuyến, nhờ vậy RU rẻ hơn và dễ triển khai. Khi này DU nhận các chức năng lại của lớp vật lý bao gồm ngẫu nhiên hóa, điều chế, sắp xếp lớp, một phần tiền mã hóa, sắp xếp lên các khối tài nguyên và các lớp MAC (Medium Access Control: điều khiển truy nhập môi trường), RLC (Radio Link Control: điều khiển liên kết vô tuyến). Nói chung hoạt động của ba lớp này đồng bộ chặt chẽ với nhau. Lớp MAC tao ra các khối truyền tải (TB: Transport Block) cho lớp vật lý bằng các sử dụng sữ liệu được nhớ đệm tại lớp RLC. Cuối cùng các đơn vị CU (CP và UP) thực hiện các lớp cao hơn của ngăn xếp 3GPP, nghĩa là, lớp RRC (Radio Resource Control: điều khiển tài nguyên vô tuyến) quản lý vòng đời của kết nối; lớp ; lớp SDAP (Service Data Adaptation Protocol: giao thức thích ứng dữ liệu dịch vụ) để quản lý QoS (Quality of Service: chất lượng dịch vụ) của ccas dòng lưu lượng (còn được đề cập là các Bearer (kênh mang)); và lớp PDCP (Packet Data Convergence Protocol: giao thức hội tụ số liệu gói có nhiệm vụ sắp xếp lại thứ tự, sao chép gói tin (Packet Duplicate) và mật mã cho giao diện vô tuyến và các nhiệm vụ khác.

12. 2.2. Các bộ điều khiển trí tuệ RAN (RIC) và điều khiển vòng kín

RIC là thành phần khả lập trình có thể chạy các quy trình tối uu hóa với điều khiển vòng kín (Closed-loop Conrol) và điều phối RAN (Orchestrate RAN). Đặc biệt, tầm nhìn O-RAN có hai RIC với quan điểm trưu tượng và tập trung về mạng như các ống dữ liệu truyền trực tuyến và kết hợp hàng trăm KPM (Key Perfomance Measurement: đo đạc chỉ số hiệu suất chính) trên trạng thái hạ tầng mạng (chẳng hạn, số lượng người dùng, tải, thông lượng, sự sử dụng tài nguyên), cũng như các thông tin ngữ cảnh bổ sung từ các nguồn bên ngoài RAN. Hai RIC này xử lý dữ liệu và sử dụng các giải thuật AI, ML để quyết định và áp dụng các chính sách và các hành động trên RAN. Một các hiệu quả, điều này đưa vào dử dụng điều khiển vòng kín được vận hành bởi dữ liệu có thể tối hưu hóa tự động , chẳng hạn: mạng và cắt lát RAN, cân bằng tải, chuyển giao, các chính sách lập lịch, trong số nhiều vấn đề khác không được đề cập. Liên minh O-RAN đã dự hảo các chức năng cho một RIC phi thời gian thực (Non-RT RIC: Non-Real-Time RIC) tích hợp với bộ điều phối mạng và hoạt động trên phạm vi thời gian dài hơn 1 giây, và một RIC gần thời gian thực Near-RT RIC: (Near-Real-Time RIC) vận hành vòng điều khển kín với các nút RAN với phạm vi thời gian giữa 10 ms và 1 giây. Hình 12.4 cung cấp tổng quan điều khiển vòng kín được các RIC hỗ trợ thông qua hạ tầng O-RAN được chia nhỏ. Hình này cũng cho thấy các mở rộng thời gian thực của nó (ngoài các vấn đề khác không được đề cập) đươc xem xét bởi nGRG (next Generation Research Group: nhóm nghiên cứu thế hệ sau) là một nhóm công tác đặc biệt được thành lập bởi liên minh O-RAN với mục đích định dạngvà điều hướng sự phát trên của các đặc tả RAN để hỗ trợ 6G và sau đó. Dưới đây ta sẽ xét từng RIC và các vòng điều khiển liên quan.

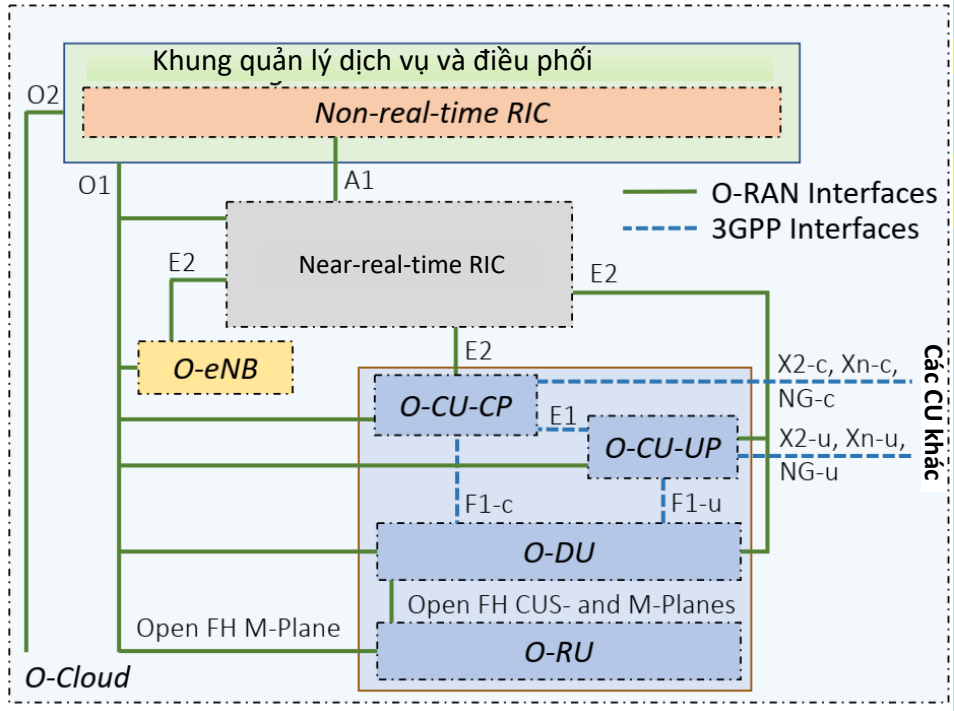
Đối tượng điều khiển và học	Quy mô (các thiết bị)	Dữ liệu đầu vào	Phạm vi thời gian	Kiến trúc	Thách thức và các hạn chế	
Các chính sách, mô hình và cắt lát	> 1000	Các KPM hạ tầng	Non-real-time > 1 s		Điều phối các triển khai quy mô lớn	Được hỗ trợ bởi O-RAN
Quản lý phiên người dùng, chẳng hạn, cân bằng tải, chuyển giao	> 100	Các CU KPM, chẳng hạn, số lượng các phiên, lưu lượng PDCP	Near-real-time 10-1000 ms		Xử lý các luồng nhiều CU và nhiều phiên	
Quản lý truy nhập môi trường, chẳng hạn, chính sách lập lịch, cắt lát RAN	> 100	Các MAC KPM, chẳng hạn, sử dụng PRB, nhớ đệm	Near-real-time 10-1000 ms		Phạm vi thời gian nhỏ, điều khiển nhiều DU/UE	
Quản lý tvô tuyến chẳng hạn, lập lịch, tạo búp	~10	Các MAC/PHY KPM, chẳng hạn, sử dụng PRB, ước tính kênh	Real-time < 10 ms		Không hỗ trợ tùy biến vòng kín thời gian thực	Nghiên cứu thêm
Quản lý thiết bị DL/UL chẳng hạn,điều chế	1	Các mẫu I/Q	Real-time < 1 ms		Tiêu chuẩn hóa thiết bị và cấp độ RU	

Non-Real-Time RIC: RIC phi thời gian thực, Near-Real-Time RIC; RIC gần thời gian thực, Open FH (Open Fronthaul: tuyến trước mở.

Hình 12.4. Điều khiển vòng kín được hỗ bởi kiến trúc O-RAN và các mở rộng có thể có. Các mũi tên với đường đứt nét trên biểu đồ kiến trúc biểu thị các vòng điều kiển khiển.

12.2.2.1. RIC phi thời gian thực và vòng điều khiển kín

RIC phi thời gian thực là một thành phần của khung quản lý và điều phối dịch vụ (SMO: Service Management and Orchestrration) như được minh họa trong hình 12.5 và bổ sung cho RIC gần thời gian thực cho hoạt động của RAN thông minh và tối ưu hóa trong phạm vi thời gian 1 giây. Với sử dụng vòng điều khiển phi thời gian thực, RIC phi thời gian thực cung cấp hướng dẫn, làm giầu thông tin và quản lý các mô hình cho RIC gần thời gian thực. Ngoài ra, RIC phi thời gian thực có thể ảnh hưởng lên các hoạt động SMO cho phép RIC phi thời gian thực khả năng điều hành gián tiếp tất cả các thành phần của kiến trúc O-RAN được kết nối đến SMO, nhờ vậy đưa ra các quyết định và áp dụng các chính sách ảnh hưởng đến hàng nghìn thiết bị. Điều này thể hiện các thách thức định cỡ cần được giải quyết thông qua xử lý hiệu quả và thiết kết phần mềm. Chi tiết hơn về RIC phi thời gian thực sẽ được xét trong phần sau.



O-RAN Interfaces: các giao diện O-RAN, 3GPP Interfaces: các giao diện 3GPP, Open FH M-Plane (Open Fronthaul M-Plane): mặt phẳng M tuyến trước mở, Open FH CUS (Open Fronthaul Control User Synchronization Plane): mặt phẳng đồng bộ người dùng điều khiển tuyến trước mở.

Hình 12.5. Kiến trúc O-RAN với các thành phần và các giao diện từ ORAN và 3GPP. Các giao diện O-RAN được vẽ ở dạng các đường đậm, các giao diện 3GPP là các đường không đứt nét

12.2.2.2. RIC gần thời gian thực và vòng điều khiển

RIC gần thời gian thực được triển khai tại biên mạng và vận hàn vòng điều khiển với chu kỳ giưa 10 mili giây và 1 giây. Như thấy trong các hình 10.4 và 10.5, RIC bấn thời gian thực tương tác với các DU và các CU trong RAN cũng như với các trạm gốc 3G LTE tương thích với O-RAN (eNB). RIC gần thới gian thực thường liên kết với nhiều nút RA, vì thế điều khiển vòng kin gân thời gian thực có thể tác động lên QoS cuat hàng trăm hay hàng nghìn các thiết bị người dùng (UE).

RIC gần thời gian thực bao gồm các ứng dụng hỗ trợ logic tùy biến, được gọi là xApps và các dịch vụ cần để hỗ trợ thực hiện xApps. Một xApp là một vi dịch vụ (Microservice) có thể sử dụng để thực hiện quản lý tài nguyên thông qua các giao diện đặc thù và các mô hình dịch vụ. Nó nhận dữ liệu từ RAN (chẳng hạn, ô của người dùng hay các KPM của lát cắt như thấy trong hình 12.5) và (nếu cần) tính toán và gửi trở lại các hành động điều khiển. Để hỗ trợ xApps, RIC gần thời gian thực có: (i) một cơ sở dữ liệu chứa thông tin về RAN (chẳng hạn: danh sách cac nút RAN được kết nối, các người dùng, v.v…) và phục vụ như là lớp chia sẻ dữ liệu giữa các xApps, (ii) hạ tầng nhắn tin trên các thành phần khác nhau của nền tảng, cũng hỗ trợ cả đăng ký các thành phần RAN với xApps; (iii) kết cuối các các giao diện mở và các giao diện lập trình ứng dụng (API: Application Programming Interface); và (IV) cơ chế phân giải xung đột để điều phối điều khiển cùng một chức năng RAN bởi nhiều xApps. Ta sẽ xét chi tiết ơn các đặc tính và các chức năng của xApps trong phần sau.

12.2.2.3. Các mở rộng đến các vòng điều khiển thời gian thực

Hình 12.4 cũng bao hàm các vòng hoạt động trong miền thời gian thực, nghĩa là dưới 10 ms, đối với đo đạc tài nguyên vô tuyến tại mức nút RAN hay thậm chí tối ưu hóa. Thí dụ điển hình của điều khiển thời gian thực bao gồm lập lịch, quản lý búp sóng và phát hiện phản hồi ít hơn các thông số lớp vật lý (chẳng hạn: sơ đồ điều chế và mã hóa, nhận biết nhiễu). Các vòng này, có số lượng thiết bị hữu hạn cần được tối ưu hóa, không phải là một phần của kiến trúc O-RAN hiện thời, nhưng đã được đề cập trong một số đặc tả để nghiên cứu sâu hơn.

`12.2.3. Ảo hóa

Nguyên lý thứ ba của kiến trúc O-RAN là đưa vào sử dụng các thành phần bổ sung cho quản lý và tối ưu hóa hạ tầng mạng và khai thác, bằng cách mở rộng các hệ thống biên vào các nền tảng ảo hóa. Tất cả các thành phần của kiến trúc O-RAN trong hình 12.3 có thể được triển khai trên nền tảng điện toán đám mây lai được gọi là O-Cloud. Đặc biệt, là một bộ các tài nguyên điện toán và hạ tầng ảo được kết hợp với nhau trong một hoặc

nhiều trung tâm dữ liệu vật lý. Nền tảng này kết hợp các nút vật lý, các thành phần phần mềm (chẳng hạn: hệ điều hành, phần mềm giám sát máy ảo (Virtual Machine Hypervisor), v.v…), các chức năng quản lý và điều phối, và đặc tả mô hình ảo hóa cho O-RAN. Nó cho phép : (i) tách riêng các thành phần phần cứng và phần mềm , (ii) hài hòa và định nghĩa các khả năng phần cứng cho hạ tầng O-RAN, (iii) chia sẻ phần cứngs giữa các người thuê khác nhau và (iv) triển khai tự đông và khởi tạo các chức năng RAN.

Nhóm công tác 6 của liên minh O-RAN cũng đã triển khai các trừu tượng tăng tốc phần cứng (được gọi là lớp trừu tượng tăng tốc (AAL: Accelaration Absstract Layer) để định ngĩa các API chung giưa các bộ xử lý logic dựa trên phần cứng dành riêng và kiến trúc O-RA được mềm hóa (O-RAN Softwaerized), chẳng hạn: cho mã hóa/giải mã kênh và sửa lỗi trước (FEC: Forward Error Corection). Điều này tác động lên các thực hiện RAN ảo hóa được tăng tốc bởi phần cứng để hỗ trợ các yêu cầu của các trường hợp sử dụng 3GPP NR (ví dụ: URLLC (Ultra reliable and Low lattency Communication: truyền thông siêu tin cậy và trễ thấp)) lưu thông trên cả phần cứng thương mại (chẳng hạn: Nvidea Aerial Platform, NEC Nuberu và từ Intel). Một số nghiên cứu cũng đề cập tăng tốc giải mã lớp vật lý dựa trên FPGA với giao diện Open AirInterface dựa trên nguyên mẫu.

Đồng thời, WG 7 đang định nghĩa các đặc trưng mà phần cứng hộp trắng (White Box Hardware) cần để thỏa mãn thực hiện một bộ phận thiết bị hợp chuẩn O-RAN, chẳng hạn: các ô pico trong nhà, các ô micro và macro ngoài trời (tại tần số dưới 6GHZ và các sóng mi li mét), truy nhập tích hợp và các nút tuyến sau, các cổng tuyến trước. Điều này bao phủ các thành phần kiến trúc khác nhau trong hình 12.5, bao gồm các nút RAN (CU, DU, RU) và các bộ hỗ trợ giao diện tuyến trước. Cac đặc tả này làm sáng tỏ các thông số chức năng tương ứng với các kịch bản được quan tâm, (chẳng hạn: các băng tần, băng thông, khoảng cách giữa các site, các cấu hình MIMO) và các đặc tính phần cứng (chẳng hạn: các bộ tăng tốc, tính toán, kết nối) của các nút.

Trong số các giao diện đặc thù O-RAN, giao diện E2 kết nối RIC gần thời gian thực với các nút RAN. E2 cho phép các vòng gần thời gian thực được thể hiện trên hình 12.2 thông qua truyền trực tuyến đo từ xa từ RAN và phản hồi điều khiển từ RIC gần thời gian thực. RIC gần thời gian thực được kết nối đến RIC phi thời gian thực thông qua giao diện A1 để cho phép vòng điều khiển phi thời gian thực và triển khai chính sách, hướng dẫn và các mô hình thông minh trong RIC gần thời gian thực. RIC phi thời gian thực cũng kết cuối giao diện O1 để kết nối đến mọi mô hình và các mô hình thông minh trongRIC gần thời gian thực. RIC phi thời gian thực kết nối đến tất cả các thành phần RAN khác để quản lý và điều phối các chức năng mạng. Sau cùng, RIC phi thời gian thực và SMO cũng kết nối đến đám mây mở (O-Cloud) của O-RAN. Giao diện tuyến trước (FH Interface) kết nối các DU và các RU. Liên minh O-RAN cũng đã định nghĩa các bài kiểm

tra chính xác và rõ ràng để thúc đẩy tương tác trrên các thực hiện giao diện khác nhau., với tập trung ban đầu lên giao diện tuyến trước E2. Ta sẽ xét chi tiếp từng giao diện trong phần sau.

Nhờ các giao diện mở, kiến trúc O-RAN được mô tả trong hình 10.6 có thể được triển khai bằng cách chon lựa các vị trí mạng khác nhau (biên đám mây, các site ô) cho các phần thiết bị khác nhau với nhiều cấu hình. Một thí dụ triển khai (được gọi là kịch bản B) được thể hiện trong hình 10.6 với các RIC được triển khai trong đám mây, còn các CU và các DU được triển khai tại biên, và RU tại biên ô. Cũng có thể có các chiến lược triển khai khác với các RIC và các nút RAN đặt cùng vị trí, chẳng hạn để hỗ trợ các 5G mạng cục bộ và riêng.

Ảo hóa các thành phần RAN và các thành phần tính toán của O-RAN được kỳ vọng sẽ đem đến tiết kiệm và tối ưu hóa tiêu thụ công suất liên quan đến RAN. Ảo hóa cho phép RAN dễ ràng và động tăng và giảm quy mô cánhu cầu các tài nguyên tính toán cần thiết để đáp ứng các yêu cầu của người dùng, nhơ vậy hạn chế tiêu thụ công suất đối với các chức năng mạng thực thực tế.

12.2.4. Các giao diện mở

Liên minh O-RAN đã đưa ra các đặc tả kỹ thuật mô tả các giao diện mở kết nối các một số các thành phần khác nhau của kiến trúc O-RAN. Hình 12.5 cho thấy các giao diện mớ được định nghĩa bởi O-RAN, cũng như các giao diện bên trong RAN từ các đặc tả của 3GPP. Các giao diện từ các đặc tả 3GPP này là một phần yếu tố hỗ trợ kiến trúc gNB được phân nhỏ, tuy nhiên được bổ sung bởi giao diện O-FH (Open Fronthaul: tuyến trước mở) của O-AN giữa DU và RU. Các giao diện O-RAN hỗ trợ việc khắc phục các tiếp cận hộp đen của RAN truyền thống, vì chúng để lộ phân tích dữ liệu và đo từ xa cho các RIC và hỗ trợ các kiểu khác nhau của các hành động điều khiển và tự động hóa, từ điêu khiển của RAN cho ảo hóa và tối ưu triển khai.

Nếu không có O-RAN, quản lý tài nguyên vô tuyến và tối ưu hóa các chức năng mạng ảo/vật lý sẽ bị đóng kín và không linh hoạt, nghĩa là, các nhà mạng không có cùng cấp độ truy nhập thiết bị trong RAN của họ hoặc sẽ phải đươc được thực hiện thông qua cách tiếp cận từng phần nhỏ tùy biến. Tiêu chuẩn hóa các giao diện này vì thế là một bước then chốt tiến đến phá vỡ việc nhà cung cấp bị khóa chặn trong RAN, chẳng hạn: cho phép RIC gần thời gian thực của một nhà cung cấp tương tác với trạm gốc của nhà cung cấp khác hay cho phép tương tác các CU, DU và RU từ các nhà sản xuất khác nhau. Điều này cũng thúc đẩy sự cạnh tranh, đổi mới , đẩy nhanh các chu kỳ cập nhật/nâng cấp và việc thiết kế, đưa đưa các thành phần được mềm hóa vào hệ sinh thái RAN dễ ràng hơn.

Liên minh O-RAN cũng liên kết với một SDO (Standard-Development Organization: tổ chức phát triển tiêu chuẩn), chẳng hạn 3GPP hay nhóm công tác 5 của ITU (International Telecommunication Union: liên minh viễn thông quốc tế) để tiêu chuẩn hóa các giao diện (sẽ xét trong phần 5).

Trong số các giao diện đặc thù O-RAN, giao diện E2 kết nối RIC gần thời gian thực với các nút RAN. E2 hỗ trợ các vòng gần thời gian thực như thấy trong hình 12.4 thông qua truyền trực tuyến đo lường từ xa từ RAN và phản hồi điều khiển từ RIC gần thời gan thực. RIC gần thời gian thực được kết nối đến RIC thời gian thực thông qua giao diện A1 hỗ trợ vòng điều khiển phi thời gian thực và triển khai chính sách, hướng dẫn, và các mô hình thông minh trong RIC gần thời gian thực. RIC phi thời gian thực cũng kết cuối giao diện O1 để kết nối đến từng thành phần RAN khác để quản lý và điều phối các chức năng mạng. Cuối cùng, RIC phi thời gian thực và SMO kết nối đến đám mây O-RAN của O-RAN thông qua giao diện O2, và giao diện FH (tuyến trước) của O-RAN kết nối với các DU và các RU. Liên minh O-RAN cũng đã định nghĩa rõ ràng và chụ thể một tập các bài kiểm tra để thúc đẩy tương tác trên các thực hiện khác nhau với tập trung ban đầu lên giao diện FH và E2. Chúng ta sẽ xét chi tiết tương giao diện trong phần 5.

Nhờ các giao diện mở, kiến trúc O-RAN được trình bày trong hình 12.7 có thể được tiển khai bằng cách chọn các địa điểm khác nhau (đám mây, biên, các site ô) cho các phần thiết bị khác nhau với nhiều cấu hình. Mọt thí dụ triển khai được thể hiện trong hình 12.3, với các RIC được triển khai trong đám mây, các CU và DU, tại biên và các site ô RU. Các chiến lược triển khai khác có thể là các RIC và các nút RAN được đặt cùng vị trí, chẳng hạn để hỗ trợ các mạng 5G cục bộ và tư nhân.

12.3. RIC GẦN THỜI GIAN THỰC, XAPPS VÀ ĐIỀU KHIỂN RAN

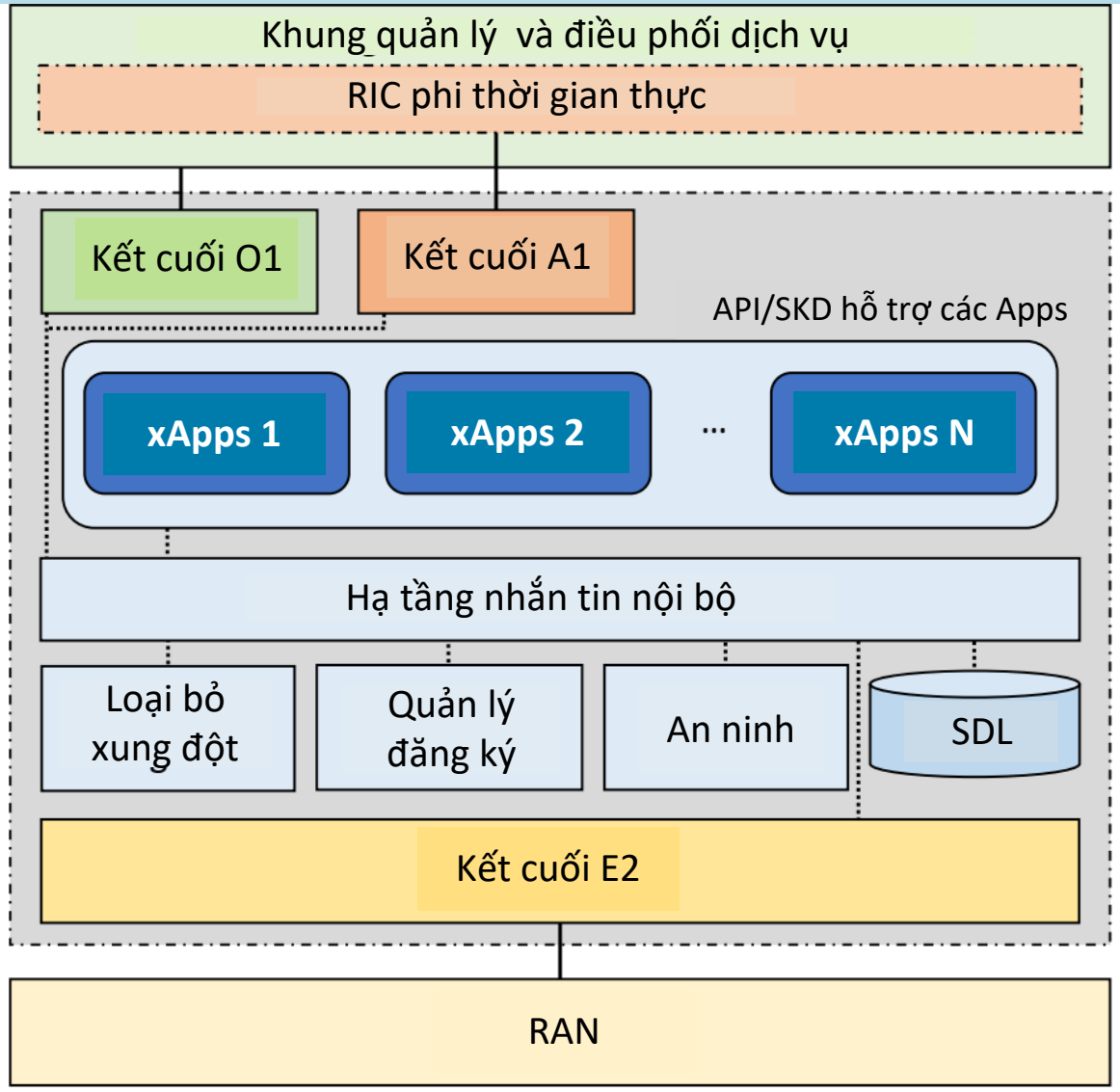
RIC gần thời gian thực (Near RT-RIC) là lõi điều khiển và tối ưu hóa RAN thông qua khả năng do giao diện E2. Trong phần này ta sẽ xét các chức năng của RIC gần thời gian thực và các thực hiện RIC gần thời gian thực khả dụng trong miền mã nguồn mở.

Như đã trình bày trong phần 2, nền tảng RIC gần thời gian thực chứa các kết cuối của các giao diện (O1, A1 và E2), các xApps và các thành phần cần thiết để thực hiện và quản lý các xApps. Các đặc tả O-RAN mô tả các yêu cầu và các chức năng của các thành phần khác nhau của RIC để có thể kỳ vọng các thực hiên phù hợp đặc tả có thể cung cấp cùng tập các dịch vụ, nhưng không đưa ra thêm các yêu cầu thực hiện. Tuy nhiên, liên minh O-RAN thông qua cộng đồng phần mềm O-RAN (OSC: O-RAN Software Community) cũng cung cấp một thực hiện tham chiếu của một RIC gần thời gian thực tuân theo các đặc tả trong [50] và có thể được sử dụng làm mẫu thử nghiệm cho các giải pháp O-RAN.

RIC gần thời gian thực của OSC được xây dựng dựa trên cơ sở nhiều thành phần vận hành như là các vi dịch vụ trên cụm Kubernetes.

12.3.1. Các thành phần bên trong RIC gần thời gian thực

Hình 12.6 cho thấy tổng quan kiến trúc của một RIC gần thời gian thực điển hình.



SDL (Shared Data Layer): lớp dữ liệu chia sẻ

Hình 12.6. Kiến trúc RIC gần thời gian thực. RIC gần thời gian thực kết nối đến RAN thông qua giao diện, tại đáy hình vẽ (mầu vàng) và đến RIC phi thời gian thực/SMO thông qua các giao diện A1 và O1 tại đỉnh của các vẽ (mầu da cam và xanh lục). Truyền thông giữa các thành phần RIC (mầu xanh sáng) được thực hiện

bởi hạ tầng nhắn tin nội bộ. RIC gần thời gian thực có thể hòa nhập logic tùy biến như là các xApps (xanh xẫm).

Ca thành phần nền tảng chính của kiến trúc RIC gần thời gian thực bao gồm:

- **Hạ tầng nhắn tin bên nội bộ.** Hạ tầng nhắn tin nội tại kết nối các xApps, các dịch vụ nền tảng và các kết cuối giao diện với nhau. Các đặc ta không bắt buộc bất cứ một công nghệ đặc thù nào (OSC sử dụng thư viện tùy biến được gọi là RMR (RIC Message Router: Router nhắn tin của RIC), nhưng họ liệt kê danh mục các yêu cấu và các chức năng mà hệ thông con này cần để hỗ trợ đăng ký, chuyển phát và xóa các điểm cuối (Chẳng hạn: các thành phần nội bộ RIC và các xApps) và cung cấp cá API để gửi và nhận các tin nhắn thông qua truyền thông điểm đến điểm hay cơ chế phát hành/đăng ký. Nó cũng cung cấp địnhj tuyến và độ bền để tránh mất mát dữ liệu nội bộ;
- **Loại bỏ xung đột.** Thành phần này xử lý các xung đột có thể phát sing giữa các xApps khác nhau. Điều này cần thiết vì các xApps độc lập khác nhau có thể áp dung các cấu hình xung khắc trong khi vẫn tìm
-
- cách đạt được mục tiêu tối ưu hóa độc lập, dẫn đến giả cấp hiệu suất. Miền xung đột có thể là một người dùng, một kênh mang hay một ô và có thể liên can đến mọi hành đông điều khiển được thực hiện bởi RIC. Các đặc tả của O-RAN nhấn mạnh ba loại xung đột. Xung đột trực tiếp có thể được phát hiện trực tiếp bởi thành phần nội bộ này. Chẳng hạn, nhiều xApps có thể áp dụng các thiết lập khác nhau cho cùng các thông số trong cùng mục tiêu điều khiển hay các xApps có thể yêu cầu nhiều tài nguyên hơn sự khả dụng. Loại bỏ xung đội có thể giải quyết bởi thành phần loại bỏ sung đột để quyết định xApp thắng thế hoặc hạn chế phạm vị của một hành động điều khiển (phân giải hành động trước). Trong khi đó, các xung đột gián tiếp hay ẩn tàng không thể quan sát được trực tiếp và có thể hoặc không thể phụ thuộc và quan hệ giữa các xApps khác nhau. Chẳng hạn, casccaaus hình để tối ưu hóa hiệu suất của các loại người dùng nào đó có thể làm giảm cấp các người dùng khác theo các cách không rõ ràng. Các xung đột này có thể bị phát hiện và được loại bỏ thông qua kiểm tra hậu hành động, chẳng hạn: bằng ccách giám sát hiệu suất của hệ thống sau khi áp dụng các chính sách điều khiển khác nhau. Tổng thể, loại bỏ xung đột là một thành phần quan trọng của RIC đang được nghiên cứu đưa vào RIC gần thời gian thực của OSC;
- **Bộ quản lý đăng ký thuê bao.** Chức năng quản lý đăng ký thuê bao này cho phép các xApps kết nối đến các chức năng được để lộ trên giao diện E2. Nó cũng điều khiển việc từng xApps truy nhập đến các bản tin E2và có thể hội nhập nhiều yêu cầu đăng ký thuê bao giống nhau đến E2 vào một yêu cầu duy nhất;

- **Hệ thống con an ninh.** Theo [50], thành phần này có mục đích mức cao ngăn chặn các xApps gây hại là dò rỉ dữ liệu RAN hay gây ảnh hưởng hiệu suất RAN;
- **Cơ sở dữ liệu thông tin mạng (NIB) và API của lớp dữ liệu chia sẻ.** R-NIB (RAN Network Information Base: cơ sở thông tin mạng RAN) lưu trữ thông tin về các nút E2 và UE-NIB (UE Network Information Base: cơ sở thông tin mạng UE) chứa các các bản ghi danh sách UE và nhận dạng chúng. Nhận dạng UE (chẳng hạn: UE-ID) là một khóa và phần thông tin nhậy cảm trong RIC vì nó cho phép điều khiển đặc thù UE, nhưng đồng thời nó có thể để lộ thông tin nhậy cảm về các người dùng. UE-NIB cho phép có thể theo dõi và liên kết nhận dạng của cùng một người dùng trong các nút E2 khác nhau. Cơ sở dữ liệu có thể được truy vấn bởi các thành phần khác nhau của nền tảng RIC (gồm các xApps) thông qua ccuarDL (Shared Data Layer: lớp dữ liệu chia sẻ). OSC RIC cung cấp một thực hiện thư viện SDL có thể biên dịch được trong các xApps, cũng như một cơ sở dữ liệu dựa trên Redis.
- **Quản lý xApp.** RIC gần thời gian thực có các dịch vụ và các API để quản lý tự động vòng đời các xApps từ hòa nhập (Ondoard) đến triển khai và kết cuối (khởi động bởi SMO), cũng như theo dõi ghi lai laithông tin sự cố, lập cấu hình, thanh toán, giêu suất, an ninh FCAPS: Fualt. Configuration Accauting, Performance).

12.3.2. Các xApps của RIC gần thời gian thực

Các thành phần chính của RIC gần thời gian thực là các xApps. Một xApp là một thành phần cắm là chạy (Plug-and-Play) thực hiện logic tùy biến, chẳng hạn cho phân tích dữ liệu RAN và điều khiển RAN. xApps có thể nhận dữ liệu và đo lường từ xa từ RAN và phản hồi điều khiển bằng cách sử dụng giao diện E2 (sẽ xét trong phần 5).

Theo các đặc tả RAN, xApp được định nghĩa một bộ mô tả (Descriptor) và bởi một ảnh phần mềm xApp (xApp Software Image) (nghĩa là một tập các file cần thiết để triển khai xApp đầy đủ chức năng). Bộ mô tả xApp (chẳng hạn: một YAML hay file JSON) bao gồm thông tin về các thông số cần thiết để quản lý xApp, thí dụ, chẳng hạn: các chính sách tự điều chỉnh quy mô, triển khai, xóa và nâng cấp thông tin. Ngoài ra nó có thể mô tả kiểu dữ liệu được xApp tiêu thụ cũng như các khả năng điều khiển. Đặc biệt là, trong RIC gần thời gian thực của OSC, RIC được định nghĩa như là môt ảnh Docker có thể triển khai được trên hạ tầng Kubernetes bằng cách áp dụng Descriptor Schema (mô hình mô tả), là một file đặc tả các thuộc tính của con-te-nơ.

Hiện nay, các đặc tả O-RAN chỉ bắt buộc một tập hạn chế các API mà nền tảng RIC gần thời gian thực cần thiết để cung cấp cho xApp (bao gồm các SDL API và cácAPI đăng ký/khám phá/ đăng ký thuê bao).

12.3.3. Các thực hiện RIC gần thời gian thực mã nguồn mở

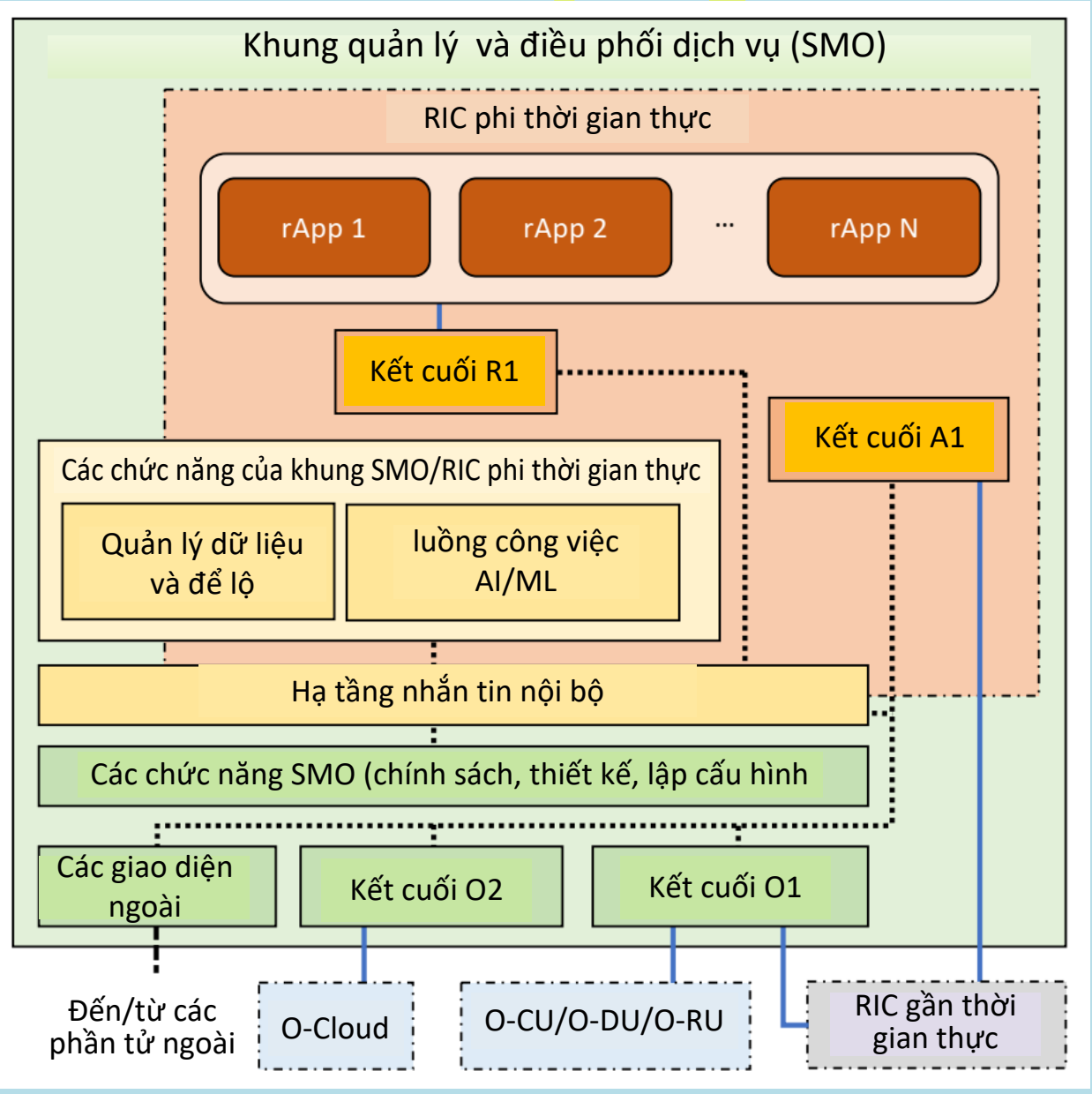
Ngoài thực hiện được OSC cung cấp, cộng đồng mã nguồn mở cũng có cả các thực hiện RIC gần thời gian thực từ phía thứ ba để làm giầu hệ sinh thái O-RAN. CoIO-RAN là một thí dụ, đây là một thực hiện tập trung lên thí nghiệm O–RAN dựa trên RIC gần thời gian thực của OSC. Đề án SD-RAN bởi ONF (Open Networking Foundation) được phát triển trên các thực hiện mã nguồn mở và điện toán đám mây gốc (Cloud-Native) của RIC gần thời gian thực của O-RAN, cùng với xApp để điều khiển RAN và một SDK (Software Development Kit) để tạo điều kiện thiết kế xApp mới. Các xApp này sử dụng cả các mô hình E2 phù hợp tiêu chuẩn và các mô hình dịch vụ tùy biến được phát triển bởi cộng đồng SD-RAN. Các vi dịch vụ của RIC này dựa trên bộ điều khiển ONOS (Open Networking Operation System: hệ điều hành nối mạng mở), bao gồm các dịch vụ đăng ký thuê bao, các dịch vụ thông tin dựa trên mạng và người dùng, các dịch vụ lưu trữ dữ liệu phân tán cho tính khả dụng cao và các dịch vụ của nhà mạng. Do thực hiện mở này, ONF là chía khóa để phát trển các đặc tả RIC, cũng như là chủ thể của các trường hợp sử dụng đối với hệ sinh thái O-RAN (chẳng hạn: cho chuyển giao và cân bằng tải) và các kế hoạch để kiểm tra chúng từ các nhà cung cấp viễn thông. Trong khi đó, FlexRIC lại cung cấp một RIC gần thời gian thực đơn thể để giao tiếp i ngăn xếp vô tuyến giao diện mở với RIC. Nó bao gồm các mô hình dịch vụ (SM: Service Model) cho giám sát và cắt lát các trường hợp sử dụng có thể lập trình. Cuối cùng, 5G-EnPOWER là một RIC gần thời gian thực cho các RAN không đồng nhất. Nó bao gồm các chức năng không phù hợp tiêu chuẩn như quảm lý di động cho Wi-Fi và các mạng tổ ong, hỗ trợ nhiều người thuê và phát trển mô hình ấn định tài nguyên tùy chỉnh trong các lát cát mạng. .

12.4. KHUNG RIC PHI THỜI GIAN THỰC (Non RT-RIC) VÀ ĐIỀU PHỐI

Thành phần then chốt thứ hai của kiến truc O-RAN là khung SMO (Service Management and Orchestration: quản lý và điều phối dịch vụ). Thành phần này chịu trách nhiệm xử lý tất cả các thủ tục điều phối, quản lý và tự động hóa để giám sát và điều khiển các thành phần RAN. Trước hết, SMO chứa RIC phi thời gian thực và cung cấp một tập các giao diện để hỗ trợ tương tác giữa các thành phần mạng khác nhau cũng như các khả năng thu thập dữ liệu để tạo điều kiện giám sát và điều khiển mạng thông qua AI/ML.

Hình 12.7 cho thấy kiến trúc của RIC phi thời gian thực và SMO. Các khối cơ sở của kiến trúc và các chức năng của các khối này sẽ được xét chi tiết trong phần còn lại của

phần này. Tập đầu tiên (các khối mầu da cam trong hình 12.7) định dạng các chức năng và các giao diện gắn với RIC phi thời gian thực. Tập thừ hai (mầu xanh lục trong hình 12.7) định dạng các chức năng gắn với bên ngoài RIC thời gian thực. Các chức năng của tập còn lại (các khôi mầu vàng) gắn với hoặc chưa gắn với bất kỳ thành phần đặc thù SMO nào hoặc chúng trải dài nhiều thành phần. Tương tự như hình 12.6, kiến trúc RIC phi thời gian thực tích hợp các giao diện đặc thù thực hiện để liên kết nối và điều chỉnh tương tác giữa các chức năng và các thành phần bên trong RIC phi thời gian thực và các miền của SMO. Kiến trúc này được mô tả như là hạ tầng nhắn tin trong hình 12.5.



Hình 12.7. Kiến trúc RIC phi thời gian thực và SMO. Các chức năng SMO (mầu xanh) hỗ trợ O-Cloud (thông qua giao diện O2) và các thành phần RAN khác (thông qua giao diện O1) để quản lý và điều phối. RIC phi thời gian thực có logi tùy biến (rApps mầu đỏ) và kết cuối giao diện A1 đến RIC gần thời gian thực mầu cam). Các chức năng được chia sẻ giữa RIC phi thời gian thực và SIMO có mầu vàng

12.4.1. RIC phi thời gian thực

RIC phi thời gian thực là một trong các thành phần lõi của kiến truc O-RAN. Tương tự như RIC gần thời gian thực, nó hỗ trợ vòng điều khiển kín của RAN nhưng với phạm vi thời gian lớn hơn 1s. Hơn nữa, nó cũng hỗ trợ thực hiện của các ứng dụng bên thứ ba, nghĩa là, các rApps được sử dụng để cung cấp các dịch vụ giá trị gia tăng để hỗ trợ và tạo điều kiện tổ chức và các hoạt động RAN, như: hương dẫn chính sách, làm giầu thông tin, quản lý cấu hình và phân tích dữ liệu. Như thấy trong hình 12.5, RIC phi thời gian thực có kết cuối R1 để giao tiếp các rApps với RIC phi thời gian thực. Điều này cho phép chúng truy nhập đến quản lý dữ liệu và các dịch vụ để lộ, các chức năng AI/ML cũng như các giao diện A1, O1 và O2 thông qua hạ tầng nhắn tin nội bộ. Cũng cần lưu ý rằng, mặc dù các rApps có thể hỗ trợ cùng các chức năng điều khiển được xApps cung cấp (Chẳng hạn: điều khiển lưu lượng, điều khiển lập lịch, quản lý chuyển giao) tại phạm vi thời gian lớn hơn, chúng được thiết kế để rút ra các chính sách điều khiển cho phép vận hành tại mức cao hơn và ảnh hưởng đến một số lượng lớn các người dùng và các nút mạng. Các thí dụ của rApps cho các ứng dụng của RIC phi thời gian thực bao gồm quản lý tần số và nhiễu, chia sẻ RAN, dự báo hiệu suất, đảm bảo thỏa thuận mức dịch vụ đầu cuối-đầu cuối (End-to-End SLA: End-to-End Service Level Agreeement) và cắt lát mạng.

Để đảm bảo một kiến trúc linh hoạt, theo đó hành vi và chức năng của từng thành phần mạng có thể điều chỉnh được theo thời gian thực để đáp ứng ý định và các mục tiêu của các nhà mạng, RIC phi thời gian thực cung cấp hai dịch vụ quản lý mạng và điều phối mức cao: quản lý mạng dựa trên ý đinh (Intent Based Management) và điều phối thông minh (Intellegence Orchestration).

Quản lý mạng dựa trên ý định. Chức năng này cho phép nhà mạng đặc tả các ý định của mình thông qua ngôn ngữ bậc cao thông qua giao diện ngườ-máy (kỳ vong rằng các ý định này sẽ tuân theo các thực tế được sử dụng trong các SMO khả dụng và được chính thức hóa như là các file YAML hoặc XML). Các ý định sau đó được phân tích tự động bởi RIC phi thời gian thực để quyết định chính sách và tập các rApps và xApps cần được triển khai và được thực hiện để thỏa mãn chúng. Trong ngữ cảnh này, cần lưu ý rằng

OSC (Open Sourrce Community: cộng đồng mã nguồn mở) đã có một giao diện người-máy (chẳng hạn: giao diện ý định trong thiết kết của RIC phi thời gian thực để cho phép các nhà mạng đặc tả ý định tại mức cao).

Điều phối thông minh. Tất nhiên kiến trúc O-RA hỗ trợ và tạo điều kiện triển khai, thực hiện và bảo trì trí tuệ mạng. Tuy nhiên không tránh khỏi là điều này làm cho mạng phức tạp hơn do tăng số lượng các xApps và rApps để thực hiện tại các RIC và các vị trí khác nhau trong mạng. Điều này dẫn đến cần các giải pháp có khả năng kết hợp và điều phối các ứng dụng này. Đặc biệt là, RIC phi thời gian thực đóng vai trò là trí tuệ của mạng điều phối để đảm bảo rằng các xApps và rApps được lựa chọn là: (i) phù hợp tốt để thỏa mãn ý định nhà mạng, (ii) được khởi tạo ại vị trí RIC tương ứng để đảm bảo điều khiển trên các thành phần RAN đặc thù được đặc tả trong ý định, (iii) được cung cấp dữ liệu tương ứng và (iv) đủ bền vững để không tạo ra cácxung đột do nhiều ứng dụng bằng cách điều khiển đồng thời cùng các chức năng và (hoặc) các thông số. Chẳng hạn, nếu nhà mạng khởi tạo nhiều lát cắt mạng và muốn điều khiển và tối ưu hóa các chính sách lập lịch cho từng lát cắt theo gần thời gian thực để chọ lựa một tập các trạm gốc gần với địa điểm quan tâm, RIC phi thời gian thực phải có khả năng tự động xác định rằng chỉ xApps thực hiệntại một RIC gần thời gian thực đặc thù có thể thỏa mãn yêu cầu thời gian. Hơn nữa, RIC phi thời gian thực phải có thể điều khiển các quyết định lập biểu và gửi chúng đến các RIC gần thời gian thực để điều khiển các trạm gốc được triển khai trong vùng quan tâm và như vậy có khả năng tạo ra dữ liệu cung cấp cho xApps.

12.4.2. Các chức năng khác của SMO/RIC phi thời gian thực

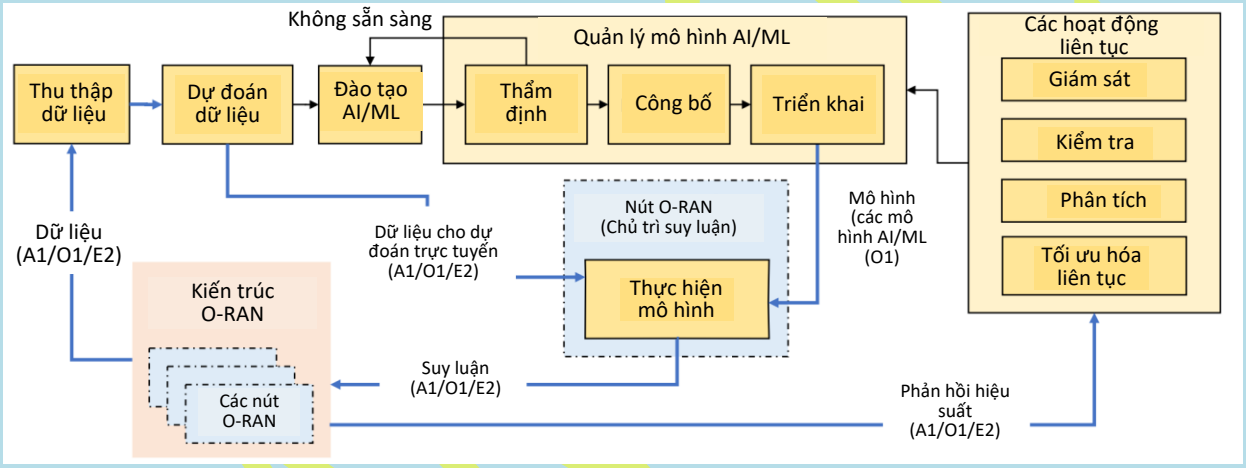
Trong phần này ta sẽ xet các chức năng và các hần tử của kiến trúcs trúc có thể nằm tại cả SMO và RIC phi thời gian thực.

Hạ tầng nhắn tin nội bộ là kết hợp của một số chức năng SMO để cho phép các thành phần bên trong SMO (thậm chí cả các phàn tử trong RIC phi thời gian thực) truy nhập và sử dụng các giao diện, dữ liệu và các chức năng do cả hai SMO và RIC thời gian thực cung cấp. Chẳng hạn, tất cả các kết cuối giao diện liên kết với các chức năng đặc thù giao diện có trong ha tầng nhắn tin nội bộ được thiế kế để tạo điều kiện trao đổi các bản tin giữa các kết cuối. Bằng cách này, các chính sách được tính toán bởi rApps có thể đạt đến RIC phi thời gian thực thông qua giao diện A1.

Quản lý dữ liệu và các dịch vụ để lộ. Các đặc tả O-RAN cũng bao gồm quản lý dữ liệu và các dịch vụ để lộ liên quan đến khung SMO/RIC phi thời gian thực. Đối với mục đich này, O-AN tuân theo giao thức người tiêu thụ/người sản suất thoe đó các người sản xuất dữ liệu trong SMO /RIC phi thời gian thực có thể quảng cáo dữ liệu công bố (chẳng hạn:

các báo cáo hiệu suất hay dựa đoán dựa trên AI về các KPM và tải). Mặt khác, các người tiêu dùng dữ liệu (chẳng hạn: rApps xác định các chính sách mức cao) có thể phát hiện, đăng ký, thu nhận và tiêu thụ dữ liệu liên quan từ một số nút được chọn trong miền SMO/RIC phi thời gian thực. Để hỗ trợ hoàn toàn các giải pháp AI/ML, SMO/RIC phi thời gian thực có thể thu thập tất cả các dữ liệu được tạo ra (được sản xuất) cũng như các hoạt động tiền xử lý AI/ML liên qua đến phân tích dữ liệu (chẳng hạn: phân tích tương quan (Corellation Analysis)) , gán nhãn và chuẩn hóa.

Luồng công việc của AI/ML. Một khả năng quan trọng khác được RIC phi thời gian thực cung cấp là có thể nhìn trước toàn bộ vòng đời AI và bao phủ tất cả các khía cạnh của phát triển AI/ML bao gồm thu thập dữ liệu, đào tạo, kiểm chứng thẩm định, triển khai và thực hiện. Hình 12.8 minh họa luồng công vệc AI/ML trong kiến trúc O-RAN.



Hình 12.8. Luồng công vệc Ai/ML trong kiến trúc O-RAN. Kiến trúc O-RAN cung cấp dữ liệu thông qua các giao diện đến các khối logic thu thập và thao tác dữ liệu. Các mô hình AI/ML sau đó được đào tạo, được thẩm định, được triển khai và thực hiện trên các nút O-RAN (chẳng hạn: các RIC). Các mô hình có thể được tinh chỉnh tiếp thông qua giám sát và tối ưu hóa liên tục dựa trên hiệu suất phản hồi từ RAN

12.4.3. Khung SMO và các thực hiện SMO mã nguồn mở

Ngoài việc chứa RIC phi thời gian thực, SMO còn cung cấp các chức năng và các giao diện bổ sung được tổng kết trong hình 12.5 (hình 6) bao gồm cả quản lý và tương tác với O-Cloud qua giao diện O1, cũng như các thành phần O-RAN khác qua giao diện O1. SMO chịu trách nhiệm quản lý các thủ tục FCAPS (Fault, Configuration, Accounting, Performance and Security: sự cố, cấu hình, thanh toán, hiệu suất và an ninh), cũng như

dịch vụ và kiểm kê tài nguyên, cấu hình topo và cấu hình mạng, bổ sung cho quản lý chính sách đối với các dịch vụ điều phối mạng.

Mặc dù một số thành viên của liên minh O-RAN tuyên bố rằng sự phát trển và tính sẵn sàng của các SMO độc quyền hợp chuẩn với các phát hành mới nhất của O-RAN, các SMO này vẫn là các giải pháp đóng kín đối với công chúng và các chi tiết thực hiện cũng như các chức năng được cung cấp của họ không có sẵn nói chung . Vì thế ta sẽ chỉ tập trung lên các giải pháp nguồn mở với các mã và các chức năng có sẵn công khai hiện đang được tích hợp với kiến trúc O-RAN: ONAP (Open Network Automation Platform: nền tảng tự đông hóa của mạng mở) va OSM (Open Service Management and Orchestration: quản lý và điều phối dịch vụ mở).

Cả ONAP và OSM đều là các nền tảng toàn diện hỗ trợ tự động hóa và điều phối trong các mạng được ảo hóa và mềm hóa. ONAP là một trong các đề án chính được phát triển và duy trì bởi Quỹ Linux (Linux Foudation) trong khi OSM được tổ chức bởi ETSI (Europeen Telecommunications Standars Institute: Viện tiêu chuẩn viễn thông Châu Âu) và tuân theo các đặc tả tiêu chuẩn chức năng mạng ảo (Virtual Network Function) của ETSI. Được duy trì bởi Quỹ Linux, ONAP cung cấp tích hợp gốc với các đề án chính như Kubernetes, Akraino, Acomos và OpenDayLight. ĐIều này làm cho ONAP là một môi trường khá phức tạp so với OSM cung cấp các dịch vụ quen thuộc trong một khung nhẹ ký hơn. Mặc dù phức tạp, ONAP đã được sử dụng tb[ỉ OSC (là nỗ lực chung giữa Quỹ Linux và liên minh O-RAN) để phát hành một nền tảng SMO mã nguồn mở cho phép tích hợp nhắn tin và các chức năng cần thiết để giám sát trạng thái của các thành phần O-RAN khác nhau cũng như nhận và xử lý các bản tin và các sự kiện (chẳng hạn: sự cố tín hiệu kiểm tra hệ thống (Heartbeat).

12.5. CÁC GIAO DIỆN MỞ CỦA O-RAN

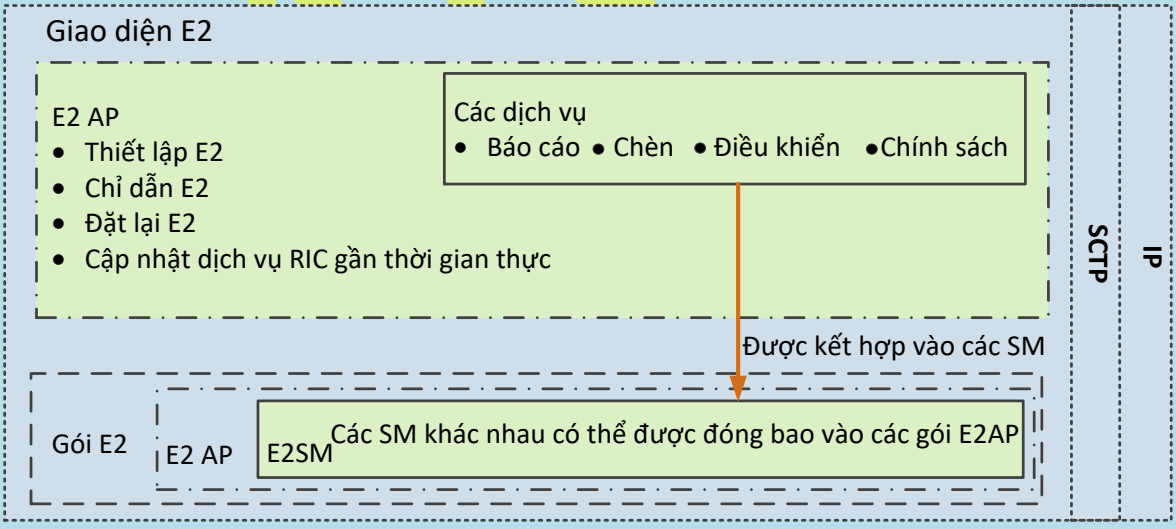
Như đã xét trong phần 2, các vòng điều khiển do RIC gần thời gian thực và các RIC phi thời gian thực hỗ trợ đươc cho phép bởi một tập các giao diện được định nghĩa bởi O-RAN. Mỗi giao diện cho phép các dịch vụ (chẳng hạn: báo cáo đo từ xa từ RAN) thông qua kết hợp các thủ tục được định nghĩa rõ ràng (chẳng hạn các thủ tục đăng ký thuê bao và chỉ dẫn cho E2). Phần này se tổng qua các trừu tượng logic và các thủ tục để định nghĩa các giao diện này và cung cấp thông tin chi tiết về vai trò của các giao diện này trong hệ sinh thái O-RAN. Các giao diện E2, O1, AI và FH sẽ lần lượt mô tả trong phần này. Cuối cùng phần này sẽ tổng quan các giao diện O-RAN còn lại và các giao diện 3GPP.

12.5.1. Giao diện E2

Giao diện E2 là một giao diện mở giữa hai điểm cuối, giữa RIC gần thời gian thực và các nút được gọi là E2, nghĩa là các DU, các CU và các LTE dNB hợp chuẩn O-RAN. E2 cho phép RIC điều khiển quản lý tài nguyên vô tuyến và các chức năng khác của các nút E2. Ngoài ra giao diện này cũng cho phép thu thập các số đo từ RAN đến RIC gần thời gian thực, hoặc định kỳ hoặc đsau khi khởi động các sự kiện được định nghĩa trước.Cả các thủ tục thu thập dữ liệu và điều khiển đều có thể liên quan đến một hay nhiều ô, các lắt cát, các loại QoS hay các UE đặc thù.

Để hỗ trợ các hoạt động nói trên, liên minh O-RAN sử dụng đa dạng mã nhận dang duy nhất. Đặc biệt, O-RAN sử dụng các mã nhận dạng dựa trên các đặc tả 3GPP cho gNB, lát cắt và loại QoS. Liên quan đến các UE đặc thù, liên minh O-RAN thay thế đưa ra một mã nhận dạng người dùng chung (chẳng hạn: UE-ID) trên các đặc tả của mình. Điều này đảm bảo một nhận dạng người dùng nhất quan và đồng nhất trên toàn bộ hệ thống mà không để lộ thông tin dịch vụ liên quan đến người dùng.

Giao diện E2 được tổ chức một các log vào hai giao thức: Giao thức E2 (E2 AP) và E2 SM (E2 Service Model). E2 AP là một giao thức cơ sở cho phép điều phối cách thức truyền thông giữa RIC gần thời gian thực và các nút E2 và cung cấp một tập các dịch vụ như mô tả trong hình 12.9 . Các bản tin E2 AP có thể nhúng các E2 SM khác nhau để thực hiện các chưc năng đặc thù (chẳng hạn: báo cáo các số đo RAN hay điều khiển các thông số RAN). Giao diện E2 chạy trên đỉnh giao thức SCTP.



Hình 12.9. Mô tả một gói O-EAN E2AP (dưới cùng bên trái) bao gồm một E2SM payload (tải tin E2SM) (trên cùng bên trái). Tải tin E2SM được đóng bao vào các

tiêu đề SCTP và IP (bên phải). Phần trên cùng của hình vẽ cũng tổng kết các dịch vụ được giao diện E2 cung cấp.

Từng nút E2 để lộ một số các chức năng RAN, nghĩa là các dịch vụ và các khả năng mà nó hỗ trợ. Chẳng hạn, các DU từ các nhà cung cấp khác nhau có thể để lộ các núm điều khiển (Control Knob) mà phụ thuộc và các núm này các thông số và các chức năng có thể được điều chỉnh, cũng như khả năng của chúng thu thập và báo cáo các số đo hiệu suất khác nhau. Bằng cách sử dụng các chơ chế công bố-đăng ký (Publish-Sunscribe), các nút E2 có thể công bố dữ liệu của chúng và các xApps trong RIC gần thời gian thực có thể đăng ký đến một hay nhiều chức năng RAN thông qua giao diện E2. Điều này cho phép phân tách rõ ràng các khả năng của từng nút và quyết định việc xApp tương tác với RAN như thế nào.

Tại mức thấp nhất, ESAP tự mình xử lý quản lý giao diện, thiết lập, báo cáo các sai lỗi cho giao diện E2) và các cập nhật dịch vụ RIC thời gian thực.

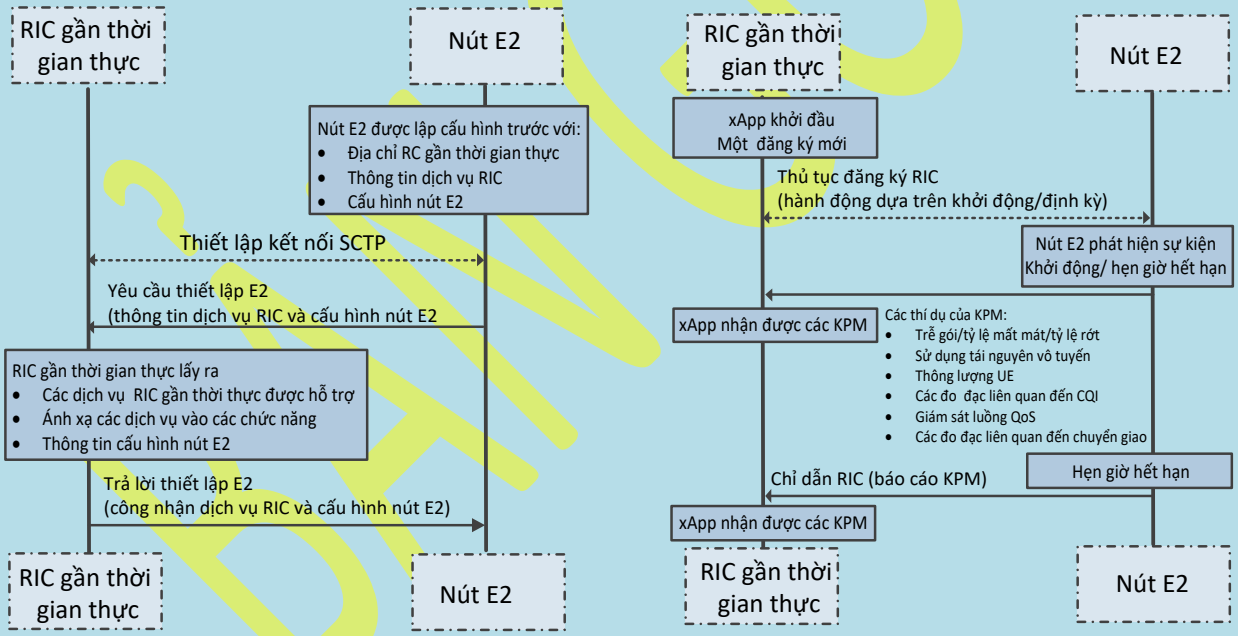
Thủ tục thiết lập E2 được thể hiện trên hình 12.10a. Đầu tiên, kết nối SCTP được thiết lập giữa RIC gần thời gian thực và nút E2 (nó biết được địa chỉ IP và cửa kết cuối E2 của RIC gần thời gian thực). Sau đó, nút E2 phát yêu cầu thiết lập E2 (E2 setup request), trong đó nó liệt kê các chức năng và cấu hình RAN mà nó hỗ trợ cùng với các mã nhận dạng nút. RIC gần thời gian thực xử lý thông tin này và trả lời bằng một trả lời thết lập E2 (E2 setup Response).

Sau khi kết nối E2 được thiết lập, E2AP cung cấp bốn dịch vụ và các dịch vụ này có thể được kết hợp theo các cách khác nhau để thực hiện một E2SM (E2 Service Model: mô hình dịch vụ E2), Các dịch vụ này (cũng được được thể hiện trên hình 12.7) là:

- Báo cáo E2 (E2 Report). Dịch vụ báo cáo E2 bao gồm các bản tin chỉ dẫn E2 RIC chứa dữ liệu và đo từ xa từ một nút E2. Dịch vụ E2 được kích hoạt dựa trên đăng ký từ một xApp đến một chức năng được cung cấp bởi nút E2. Trong quá trình đàm phán đăng ký, xApp trong RIC gần thời gian thực có thể đặc tả kích hoạt định kỳ hay theo các sự kiện khởi động (Trigger Event) theo đó nút E2 phải gửi đi các bản tin báo cáo. Dựa trên định kỳ này, một hẹn giờ được thiết lập trong nút E2 và báo cáo được gửi đi khi hẹn giời hết hạn. Chỉ dẫn RIC là một kiểu báo cáo;
- Chèn E2 (E2 Insert). Chèn dịch vụ liên quan đến bản tin được gửi đi từ nút E2 đến xApp trong RIC gần thời gian thực để thông báo xApp về một sự kiện đặc thù trong nút E2 (chẳng hạn: một UE báo hiệu khả năng thực hiện chuyển giao). Nó được kích hoạt dựa trên đăng ký thuê bao từ một xApp và liên quan đến một bản

tin chỉ dẫn RIC (của kiểu chèn). Trong trường hợp này, khởi động (Trigger) liên quan đến một thủ tục quản lý tài nguyên RAN bị treo khi bản tin chèn được gửi. Bộ định thời đợi được khởi đầu, nếu RIC không trả lời trước khi hết giờ, thủ tục trong nút E2 có thể bắt đầu lại hoặc dừng dứt khoát.

- Điều khiển E2 (E2 Control). Dịch vụ điều khiển có thể được RIC tự khởi xướng hoặc nó có thể là kết quả của việc nhận được một bản tin chèn tại RIC gần thời gian thực. Dịch vụ này dựa trên một thủ tục bao gồm hai bản tin. Một yêu cầu điều khiển RIC (RIC Control Request) từ RIC đến nútt E2 và một công nhận điều khiển RIC (RIC Control Acknowledge) theo chiều ngược lại. Các dịch vụ điều khiển có thể ảnh hưởng lên các thông số bị để lộ bởi các chức năng RAN của nút E2.
- Chính sách E2 (E2 Policy). Dịch vụ này liên quan đến một thủ tục đăng ký thuê bao để đặc tả: (i) một khởi động sự kiện, và (ii) một chính sách buộc nút E2 tự động phải tuân theo để thực hiện quản lý tài nguyên vô tuyến.



(a) Thủ tục thiết lập một phiên E2 giữa RIC gần thời gian thực và út E2. Thủ tục này được khởi xướng bởi nút E2 để tương tác với RIC gần thời gian thực.

(b) Các thủ tục liên quan đến phát trực tuyến các KPM từ nút E2 đến RIC gần thời gian thực. Thủ tụcđăng ký đươc bắt đầu bởi một xApp trong RIC gần thời gian thực. Sau đó RIC này nhận được các báo cáo

Hình 12.10. Các thủ tục thiết lập E2 và và KPM của E2SM. Đường thẳng đứng thể hiện tiến hóa hời gian của quá trình, còn các đường ngang là các bản tin được trao đổi giữa RIC gần thời gian thực và nút E2.

Các dịch vụ này sau đó được kết hợp để tạo lập một mô hình dịch vụ bản tin . Mô hình dịch vụ được chèn vào tải trọng của một trong các bản tin E2AP như thấy trong hình 12.9. Nội dung thực tế được mã hóa sử dụng ký hiệu ASN.1, chẳng hạn: thông qua các kiểu số được định nghĩa rõ ràng và cặp khóa-giá trị.

Các mô hình dịch vụ E2 (E2 SM: E2 Service Model). Hiện nay nhóm công tác WG3 của Liên minh O-RAN đã phát triển bốn mô hình dịch vụ: (i) E2SM số đo hiệu suất (E2SM KPM); (ii) E2SM các giao diện mạng (E2SM NI); (iii) E2SM cấu hình và điều khiển ô (E2SM CCC: E2SM Cell Configuration and Control); (iv) E2SM điều khiển RAN ((E2SM RC: E2SM RAN Control).

E2SM KPM báo cáo các số đo hiệu suất từ RAN với sử dụng dịch vụ báo cáo E2. Các thủ tục này được liên kết với mô hình dịch vụ KPM được thể hiện trên hình 12.8b. Trong quá trình thiết lập E2, nút E2 quảng cáo các số đo mà nó có thể để lộ. Một xApp trong RIC gần thời gian thực khi này có thể gửi đi bản tin đăng ký đặc tả các KPM nào được quan tâm, và báo cáo là định kỳ hay dựa trên khởi động. Cuối cùng, nút E2 sử dụng các bản tin chỉ dẫn kiểu báo cáo để phát trực tuyến các KPM được chọn. Các bản tin KPM khác nhau được tạo ra từ các nút E2 khác nhau, chẳng hạn: các đặc tả định nghĩa các con-te-nơ số đo hiệu suất (Performance Metric Container) cùng với các trường khác nhau tồn tại cho DU, CU-UP và CU-CP. Ngoài ra phiên bản 2 của đặc tả KPM cũng đưa ra các con-ten-nơ hiệu suất đặc thù ô và đặc thù người dùng dựa trên các số đo được định nghĩa trong các đặc tả kỹ thuật cho LTE và NR.

E2SM NI được sử dụng để nhận các bản tin thu được bởi nút E2 trên các giao điện mạng đặc thù và chuyển chúng đến miền RC gần thời gian thực thông qua các bản tin báo cáo E2 trên giao diện E2. Nút E2 quản bá giao diện nào được nó hỗ trợ trong quá khoảng thời gian thủ tục đăng ký và các giao diện này bao gồm cả X2 (két nối các LTE eNB), Xn (kết nối các NR gNB, F1 kết nối các DU và các CU.

2SM CCC (được giới thiệu 10/2022) thực hiện điều khiển và lập lại cấu hình trên các nút E2 tại ô hay tai mức nút, chẳng hạn: cần hình đoạn băng thông (Bandwith Part). Nó trước hết sử dụng báo cáo E2 và các bản tin điều khiển. Phiên bản hiện thời của đặc tả kỹ thuật cho CCC bao gồm các mục tiêu điều khiển như chọn các láng giềng X2 và Xn, cắt lát RAN, các thông số liên quan đến đoạn băng thông và các tín hiệu đồng bộ của một ô.

Điều khiển vòng kín bởi E2SM điều khiển RAN (E2SM RC)

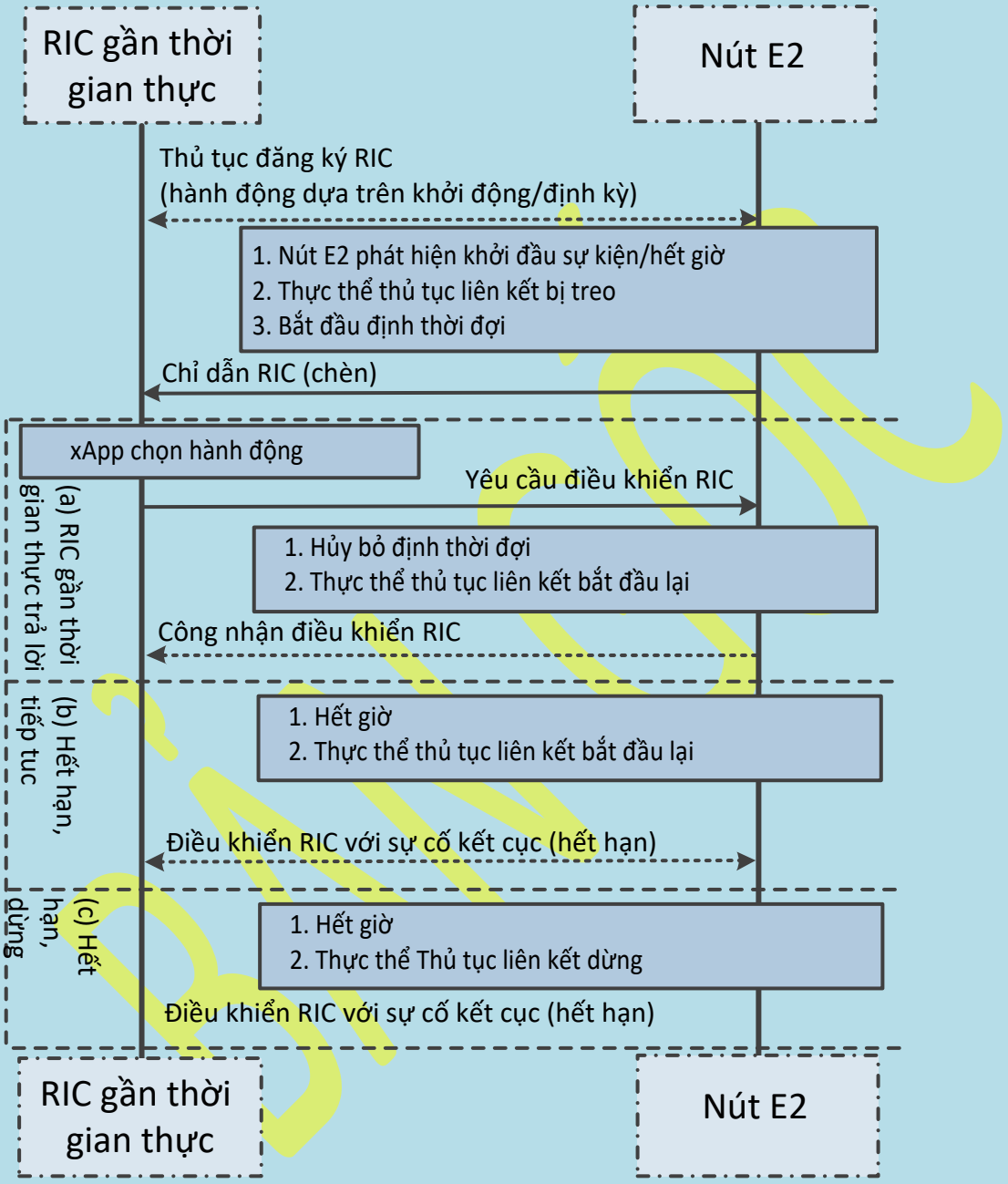
Một trong các mục tiêu của các xApps là tối ưu hóa quản lý tài nguyên vô tuyến trong các nút E2. E2SM RC thực hiện các chức năng điều khiển thông qua các dịch vụ điều khiển E2. So với E2SM CCC, nó tập trung lên điều khiển thường xuyên hơn (đến mức UE hoặc mức kênh mang). Phát hành đầu tiên cho E2SM bao gồm một tập rộng các vùng điều khiển:

- Điều khiển kênh mang (Bear Control), để thay đổi các thông số cho các kênh mang trên các nút E2 hay một UE đặc thù, chẳng hạn liên quan đến các thông số QoS, điều khiển cho phép kênh mang, phân chia kênh mang và điều khiển sao chép PDCP (PDCP Duplication: truyền kép PDCP));
- Điều khiển ấn định tài nguyên vô tuyến, để thay đổi thu không liên tục, lập lịch bán vĩnh cửu, cho các khối tài nguyên vật lý (PRB) mức lát cắt cho một nút E2, một ô, một lát cắt, môt UE hay các loại QoS;
- Điều khiển chế độ kết nối (Connected Mode) để khởi xướng một thủ tục di động cho các UE trong trạng thái kết nối RRC, chẳng hạn: chuyển giao đến một ô đặc thù hay chuyển giao có điều kiện đến một tập các ô;
- Điều khiển truy nhập vô tuyến, để thiết lập các thông số lùi truy nhập ngẫu nhiên (Random Access Backoff), cho phép UE đến một ô, và các loại khác;
- Điều khiển kết nối kép (DC: Dual Connectivity) , để lập cấu hình và khởi động chuyển giao một UE đến các ô thứ cấp đích được chọn, cập nhật các ô thứ cấp hoặc giải phóng DC;
- Kết hợp sóng mạng (CA: Carrier Aggregation), để khởi xướng CA và thay đổi các sóng mang thành phần cho một UE đặc thù;
- Điều khiển chế độ di động rỗi (Idle Mobility Control), để thay đổi các chức năng cho các thủ tục di động của UE trong trạng thái RRC rỗi (RRC Idle) bao gồm các ưu tiên chọn lại ô và các bộ định thời nhàn rỗi.

RSM RC cũng cung cấp các khả năng báo cáo nhận dạng UE và thông tin UE. Các hành động và chính sác được ESM đặc tarlieen quan đến các thông số đặc thù được tiêu chuẩn hóa bở 3GPP, chẳng hạn: hành động điều khiển thường mang một giá trị cho thành phần thông tin (IE: Information Eleement) 3GPP được định nghĩa trong đặc tả E@SM. CHẳng hạn IE điều khiển chuyển giao bao gồm (trong số các thông tin khác nhau) một ID ô đích để chuyển giao được mã hóa bởi một thành phần thông tin nhận dạng toàn cầu ô đích của 3GPP (3GPP Target Cell Global ID IE).

Để thực hiện hiệu quả các hành động điều khiển và (hoặc) thực thi chính sách, E2SM sử dụng một tổ hợp các dịch vụ E2. Hinh 12.11 cho thấy một thí dụ về các bản tin E2 được

trao đổi giữa một nút E2 và một xApp trong RC gần thời gian thực trong thời gian một thủ tục điều khiển điển hình.



Hình 12.11. Dịch vụ chèn E2 (E2 Insert) với trả lời dịch vụ chèn E2 (E2 Insert Response) tiếp theo. Đường hẳng đứng thể hiện ttiến hóa thời cửa quá trình, còn các đường ngang là các bản tin được trao đổi giữa RIC gần thời gian thực và nút E2

Trước hết, xApp đăng ký đến nút E2 để nó để lộ một chức năng RAN đặc thù được gọi là điều khiển RAN (RAN Control). Trong giai đoạn đăng ký, xApp có thể đặc tả một sự kiện khởi động hay một định thời. Khi này, nếu định thời hết hạn hay điều kiện được được định nghĩa cho khởi động được xác minh, nút E2 gửi một bản tin chèn E2 (E2 Insert) đến RIC. Chẳng hạn, điều này có thể xẩy ra khi một số đo được gNB giám sát vượt quá một ngưỡng nhất định, chẳng hạn: khi công suất thu được của tín hiệu tham chiếu (RSRP: Reference Signal Received Power) từ một ô láng giềng hay ô đích trở nên tốt hơn RSCP của ô hiện thời cộng với một khoảng dịch (còn được gọi là sự kiện A3 trong các đặc tả 3GPP). Khi này, nút E2 gửi một bản tin chèn yêu cầu chuyển giao (Handover Request) (trong đó nó có thể đặc tả cả ô đích được chính nút E2 định nghĩa), treo thủ tục quản lý tài nguyên và khởi đầu định thời. Các kết cục khác nhau có thể tiếp sau sự kiện này. RIC gần thời gian thực có thể trả lời bằng một bản tin điều khiển E2 để hoặc từ chối yêu cầu điều khiển hoặc chấp thuận nó và trả lời bằng một hàn động điều khiển (chẳng hạn: một ID ô đích được chọn bởi xApp). Một sự lưa chọn khác, nếu định thời nút E2 hết hạn trước khi nhận được bản tin từ RIC, thủ tục này có thể tự động tiếp tục hay có thể dừng tùy vào thủ tục và cấu hình đặc thù của nút E2.

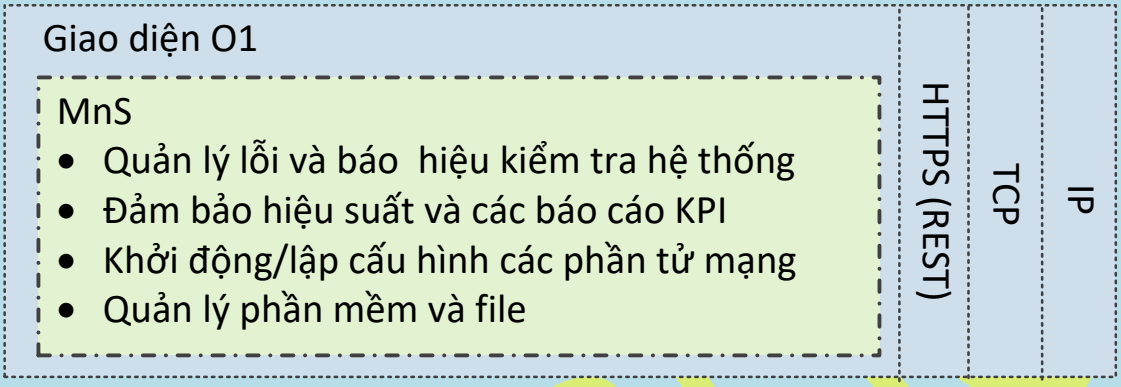
Hành động điều khiển được xApp gửi đi cũng có thể không đồng bộ, nghĩa là nó không phụ thuộc vào thu bản tin chèn từ nút E2. Ngoài ra, E2SM cũng có thể được sử dụng để đặc tả các chính sách, nghĩa là thay đổi các hành vi được quy định trước trong các nút E2. Các chính sác này có thể có hai kiểu khác nhau: (i) các chính sách điều khiển cho phép nút E2 thực hiện quản lý tài nguyên vô tuyến không cần tương tác với RIC, khi các điều kiện nhất định đươch thỏa mãn; (ii) ccs chính sách bù đắp (Offset Policy) cho phép thay đổi các ngưỡng được quy định trước bởi 3GPP hay nhà cung cấp bằng cách cộng hoặc loại bỏ các khoảng bù đắp và nhờ vậy thay đổi cách thức nút E2 thực hiện các chức năng đặc thù.

12.5.2. Giao diện O1

Ngoài giao diện E2, giao diện khác kết nối các thành phần đặc thù O-RAN với các nút RAN là giao diện O1. Nói chung, các thành phần được O-RAN quản lý (bao gồm RIC gần thời gian thực, các nút RAN) được kết nối qua O1 đến SMO và RIC gần thời gian thực.Như vậy, O1 là một giao diện mở tiếp nhận và mở rộng các thực tế được chuẩn hóa cho các vận hành và bảo dưỡng.

Giao diện O1 hỗ trợ các dịch vụ quản lý (MnS: Management Service), bao gồm quản lý vòng đời các thành phần O-RAN (từ khởi động và lập cấu hình đến khả năng chịu lỗi và các dịch vụ báo hiệu kiểm tra hệ thống (Hearrtbeat)), đảm bảo hiệu suất và theo dõi thu thập thông qua các báo cáo về các chỉ số hiệu suất chính (KPI: Key Performance

Indicator), quản lý phần mềm và file (hình 12.12). Nói chung, O1 kết nối một người cung cấp MnS (nghĩa là: nút được quản lý bởi SMO) đến một người tiêu thụ (nghĩa là: SMO).



HTTPS (Hypertext Transfer Protocol Secure): giao thức truyền siêu văn ảnh an ninh;
REST (Representational State Transfer): chuyển trạng thái đại diện

Hình 12.12. Giao diện O-RAN O1 và tải trong các dịch vụ quản lý (trái), được đong bao trong các gói HTTPS/TCP/IP (phải).

Cung cấp các dịch vụ quản lý (Provisioning Management Services) cho phép SMO đẩy các cấu hình xuống các nút được quản lý và báo cáo cập nhật cấu hình bên ngoài từ các nút được quản lý cho SMO. Để vậy O1 sử dụng một kết hợp của REST/HTTPS API và NETCONF là một giao thức được chuẩn hóa bởi IETF (Internet Engineering Task Force) cho quản lý vòng đời các chức năng được nối mạng.Cung cấp các dịch vụ được hỗ trợ bao gồm các cung cấp dịch vụ được định nghĩa trong các đặc tả kỹ thuật của 3GPP. Một dịch vụ quản lý giám sát lỗi (Fault Supervision MnS) bổ sung được sử dụng dể báo cáo các sai lỗi và các sự kiện cho SMO. Nó cũng được xây dựng trên các sự kện lỗi được 3GPP định nghĩa và có thể được các nút RAN sử dụng để báo cáo lỗi (thông qua các tải trọng JSON được chuẩn hóa) với sử dụng các RESTAPI (Representational State Transfer Application Programming Inteface: giao diện lập trình của chuyển giao trạng thái đại diện). Đối với từng nút, SMO cũng có thể truy vấn một danh các các cảnh báo (nghĩa là: các thăm dò cho phép giám sát trạng thái của các phần tử và các thành phần trong nút) trong trường hợp công nhận (Acknowledge) hay xóa (Clear) chúng. Sau cùng, SMO có thể cung cấp các báo hiệu kiểm tra hệ thống (Heartbeats) trên các thiết bị để truyền đi các bản tin thông qua dịch vụ quản lý Hearbeat Mns và quản lý không chỉ các chức năng mạng ảo mà cả các chức năng mạng vật lý (FNF: Physical Network Function). Các bản

tin Hearrtbeat được sử dụng để giám sát trạng thái và sự khả dụng của các dịch vụ và các nút.

Dịch vụ quản lý đảm bảo hiệu suất (Performance Assurance MnS) có thể được sử dụng để phát trực tuyến (theo thời gian thực) hay báo cáo khối lượng lớn (thông quan chuyển giao file) dữ liệu hiệu suất đến SMO để cho phép, chẳng hạn phân tích dữ liệu và thu thập dữ liệu cho AI/ML. SMO có thể lựa chọn các KPI (còn được đề cập là các bộ đếm) để báo cáo và tần suất báo cáo. Lựa chọn này dựa trên các trường hợp sử dụng và định dạng cho KPI được 3GPP định nghĩa hay bởi đồ án VES (VNF Even Steam: Phát luồng sự kiện phần tử mạng ảo). Các số đo hiệu suất cũng hoặc được xayt dựng trên cơ sở các tài liệu 3GPP, đặc thù nhà cung cấp hoặc được đặc tả bởi các nhóm công tác (WG) khác nhau của liên minh O-RAN. Để chuyển giao khối lượng lớn, một thông báo file sẵn sàng tước tiên được gửi từ người cung cấp MnS (chẳng hạn: nút đặc thù của RAN) đến SMO thông qua các HTTP API (Hypertext Transfer Protocol Application Programming Interface: giao diện lập trình của giao thức chuyển giao siêu văn bản). Sau đó, một chuyển giao file được thực hiện thông qua SFTP (Secure File Transfer Protocol: giao thức chuyển giao file an ninh). SMO cũng có thể tải xuống các file mà nó đã nhận được thông báo chuyển giao file trước đó. Ngoài ra một WebSocket được sử dụng cho phát trức tuyến thời gian thực sau một giao dịch bắt tay. SMO có thể giám sát các sự kiện dựa trên theo dõi thông qua dịch vụ quản lý theo dõi (Trace MnS), chẳng hạn: để lập hồ sơ các cuộc gọi, thiết lập RRC, các sự cố liên kết vô tuyến.

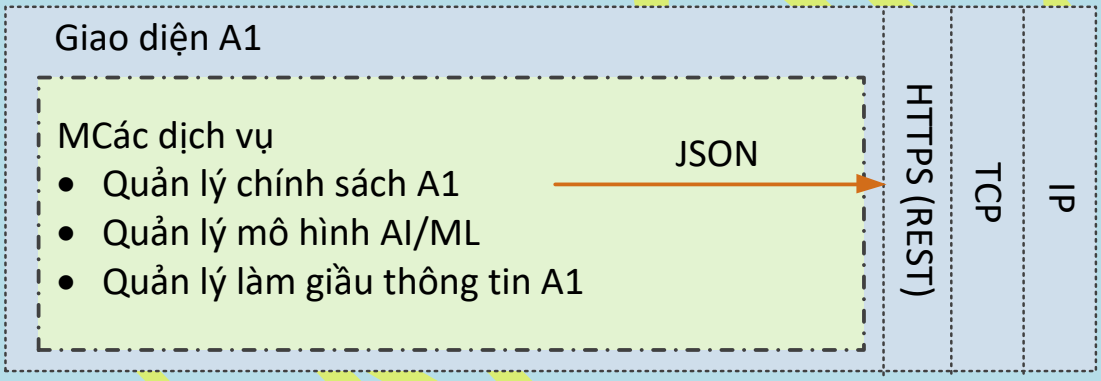
Giao diện O1 cũng có thể được sử dụng để đẩy và (hoặc) tải xuống các file đến các nút được quản lý bởi SMO. Điều này cho phép, chẳng hạn, các cập nhật phần mềm, các file lập cấu hình tạo búp mới cho các RU và triển khai các mô hình AI/ML và các chứng chỉ an ninh.

12.5.3. Giao diện A1

Giao diện A1 kết nối hai thành phần đặc thù O-RAN, nghĩa là RIC phi thời gian thực (hay SMO) và RIC gần thời gian thực như được thể hiện trong hình 12.6. Nó cho phép RIC phi thời gian thực trển khai các hướng dẫn được xây dựng trên cơ sở chính sách cho RIC gần thời gian thực (chẳng hạn: thiết lập các mục tiêu tối ưu hoa mức cao), để quản lý các mô hình ML được sử dụng, chẳng hạn, trong các xApp hoặc để đàm phán và điều phối chuyển giao làm giầu thông tin cho RIC gần thời gian thực. Điều này được thực hiện thông qua thông qua các cơ chế được định nghĩa rõ ràng dựa trên một cú pháp đặc

thù (dựa trên JSSON Schema) cho phép biểu diễn các chính sách và ý định mức cao. Các chính sách, các mô hình và làm giầu thông tin có thể liên quan đến một nhóm các UE hoặc thậm chí đến một UE đặc thù. Lưu ý rằng, hiện thời mới chỉ quản lý mô hình ML dựa trên A1 được xem xét.

Giao diện A1 được minh họa trong hình 12.13 dựa trên giao thức ứng dụng A1AP với các chức năng đang được đặc tả bổ sung cho từng dịch vụ mà nó hỗ trợ. A1AP được xây dựng trên cơ sở 3GPP cho triển khai các phần tử mạng để kết hợp các REST API trên HTTP cho truyền các đối tượng JSON. Đối với từng dichgj vụ, cả hai RIC phi thời gian thực và RIC gần thời gian thực đều có một cặt máy khác (Client) và máy chủ (Server) HTTP được sử dụng luân phiên cho quản lý dịch vụ và cho quản lý truyền dữ liệu và (hoặc) các thông báo.



HTTPS (Hypertext Transfer Protocol Secure): giao thức truyền siêu văn ảnh an ninh;
REST (Representational State Transfer): chuyển trạng thái đại diện

Hình 12.13.Tải trọng O-RA A1 JSON (tái) được đóng bao trong các gói HTTPS/TCP/IP (phải). Hình vẽ cũng tổng kết các dịch vụ có thể được cung cấp trên giao diện A1

Quản lý chính sách A1 được sử dụng bởi RIC phi thời gian thực để rút ra các chức năng của RIC gần thời gian thực để đạt được ý định mức cao (sẽ xét trong phần 4). Nói chúng, ý định này được định nghĩa thông qua QoS hay các mục đích của KPM cho tất cả các người dùng hoặc một tập con các người dùng (chẳng hạn: lát cắt) và được giám sát bằng cách sử dụng các chức năng báo cáo của giao diện O1 và phản hồi trên giao diện A1. ROC phi thời gian thực cũng có nhiệm vụ giám sát và quản lý vòng đời của các chính sách nhờ các API cho xóa, cập nhật và truy vấn các chính sách trong RIC gần thời gian thực.

Từng chính sách được xây dựng trên cơ sở JSON Schema đặc thù được nhóm lại tùy theo các kiểu chính sách khác nhau. Nói chung, tất cả các JSON schema đều có một định

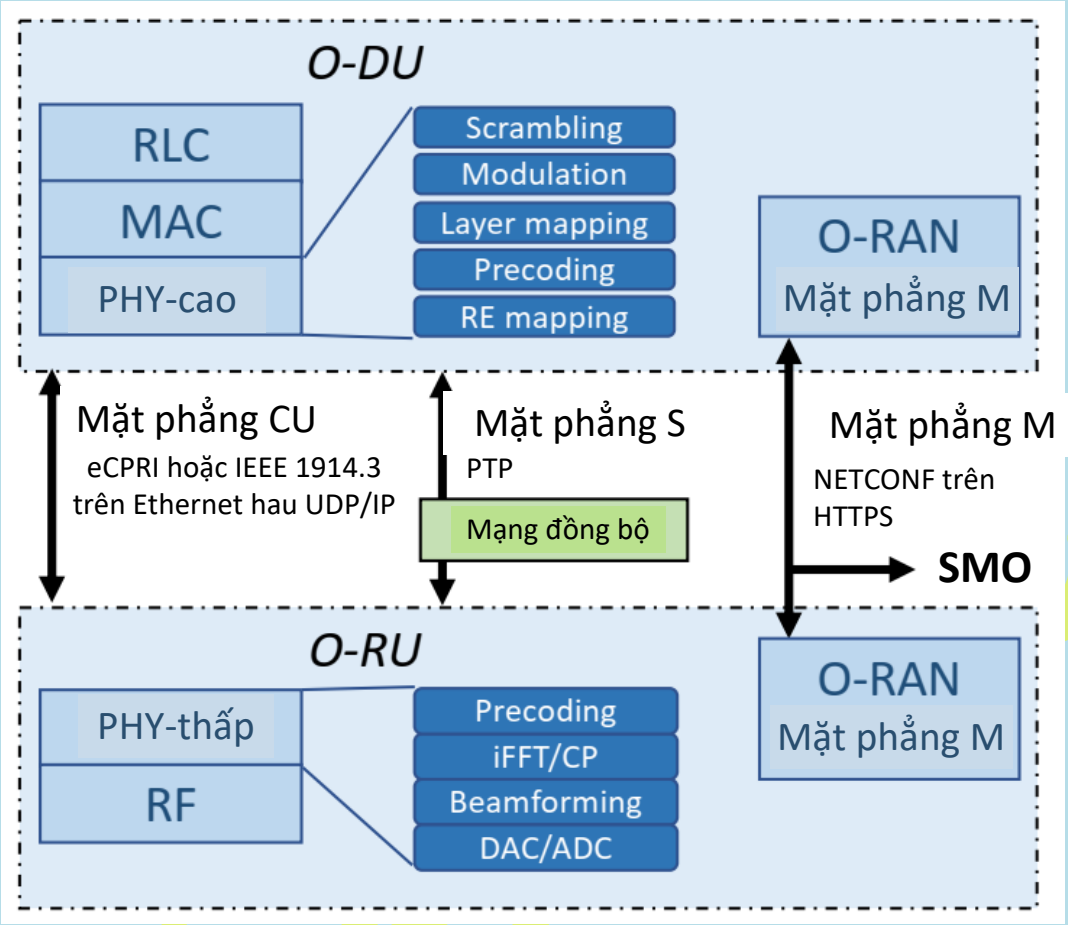
danh chính sách duy nhất cho RIC phi thời gian thực,một định danh phạm vi và một hoặc nhiều trình bày chính sách (Policy Statement). Phạm vi có thể là một UE, một nhóm UE, các lát cắt, các kênh mang (Bearer) và các loại ứng dụng. Bản thân chính sách được biểu diễn thông qua một chuỗi các trình bày chính sách có thể bao phủ các tài nguyên chính sách (chẳng hạn: các điều kiện cho sử dụng tài nguyên cho chính sách) và các mục tiêu chính sách (liên quan đến QoS hoặc các mục tiêu KPM). Đặc tả kỹ thuật của O-RAN liệt kê một số kiểu cho trình bày chính sách phụ thuộc vào các trường hợp sử dụng cụ thể (chẳng hạn: cực đại hóa thông lượng, ưu tiên điều hướng lưu lượng, các mục tiêu QoS và các trình bày khác).

Cuối cùng, làm giầu thông tin A1 (A1 EI: A1 Enrichement Information) nhắm đến cải thiện hiệu suất RAN bằng cách cung cấp thông tin thông thương không có sẵn cho bản tận RAN (chẳng hạn: các dự báo dụng lượng, các phần tử thông tin từ các nguồn bên ngoài RAN, tổng hợp các phân tích).Thực vậy, RIC phi thời gian thực và SMO có tầm nhìn toàn cục về mạng và truy nhập đến các tài nguyên thông tin bên ngoài, và có thể chuyển tiếp nó đến các xAPPx trong RIC gần thời gian thực bằng cách sử dụng A1 EI. Luồng thông tin cũng có thể vượt qua RIC phi thời gian thực và RIC này có thể chỉ dẫn RIC gần thời gian thực kết nối trực tiếp đến các nguồn EI.

12.5.4. Tuyến trước O-RAN (O-RAN FH)

Giao diện tuyến trước O-RAN (O-RAN FH: O-RAN Fronthaul) kết nối một DU hay nhiều RU đến cùng một gNB. Giao diện O-RAN FH cho phép phân tán nhiều chức năng lớp vật lý giữa RU và DU, và điều khiển các hoạt động của RU từ DU. Như đã xét trong phần 2, liên minh O-RAN đã chọn một cấu hình đặc thù (phân chia 7.2x) để phân chia lớp vật lý trong số các phân chia được đề xuất bởi 3GPP. Như thấy trong hình 12.14, phần thấp của lớp vật lý (PHY thấp) nằm trong RU thực hiện bù pha OFDM, biến đổi Fourrier nhanh ngược (iFFT: Inverse Fast Fourier Transformation) và chèn tiến tố chu trình (CP: Cyclic Prefix) cho chuyển đổi tần số-thời gian trên đường xuống, và FFTloại bỏ CP trên đường lên.

Các RU có khả năng nhiều hơn (chẳng hạn các RU loại B) cũng có thể thực hiện tiền mã hóa. Chức năng này được thực hiện tại DU cho các RU có khả năng kém hơn (các RU loại A). Để bổ xung cho các khả năng DU, các RU loại A cần hỗ trợ PHY thấp xử lý cho ít nhất 8 luồng dữ liệu. Lớp vật lý trong DU (PHY cao) thực hiện ngẫu nhiên hóa, điều chế, ánh xạ lớp và ánh xạ phần tử tài nguyên.



eCPRI (Enhanced Common Public Radio Interface): Giao diện vô tuyến công cộng chung tăng cường; PTP (Precision Time Protocol: giao thức thời gian chính xác); HTTPS (Hypertext Transfer Protocol Secure): giao thức truyền siêu văn bản an ninh;
Scrambling: ngẫu nhiên hóa; Modulation: điều chế; Layer mapping: ánh xạ lớp; Precoding: tiền mã hóa; RE mapping: ánh xạ phần tử tài nguyên; iFFT (Inverse Fast Fourier Transfer): biến đổi Fourier nhanh ngược; CP (Cyclic Prefix): tiền tố chu trình; Beamforming: tạo búp; DAC/ ADC (Digital to Analog/Analog to Digital): chuyển đổi số sang tương tự/ chuyển đổi tương tự sang số

Hình 12.14. Giao diện tuyến trước của O-RAN và các mặt phẳng tuyến trước. Giao diện này cho phép phân chía 7.2x các chức năng lớp vật lý trên DU và RU với dữ liệu và điều khiển giữa PHY cao và PHY thấp được vận chuyển trên các mặt phẳng điều khiển và người dùng (CU: Coltrol-User). Mặt phẳng S (S-Plane) cung cấp đồng bộ, quản lý mặt phẳng M (M-Plane) và điều phối các chức năng.

Theo các đặc tả của O-RAN, phân chia 7.2x dung hòa gữa sự đơn giản của giao diện và thiết kế RU, tiềm năng cho tương tác (ít hơn các thông số để lập cấu hình hơn các phân tích tách cao hơn và tốc độ dữ liệu cần thiết để vận tải tuyến tước (thấp hơn so với các

cấu hình phân tách lớp vật lý thậm chí sâu hơn nữa). Giao diện phân chia 7.2x có thể được xây dựng trên cơ sở các gói Ethernet hay UPD/IP mang tải trọng eCPRI (enhanced Common Public Radio Interface: giao diện vô tuyến công cộng chung tăng cường) hay IEEE 1914.3. Lưu ý rằng DU có thể hỗ trợ nhiều RU khác nhau hay xử lý một DU và nhiều RU. Để làm điều này, các đặc tả O-RAN FH dự kiến trước một thành phần bổ sung để ghép một luồng FH đến nhiều RU, chẳng hạn để phục vụ các sóng mang của cùng các ô từ các RU khác nhau hay kết nối nối tếp (Daisy Chaining) các RU. Ngoài ra, giao diện O-RAN FH được thiết kế để hỗ trợ truyền thông siêu tin cậy, trễ thấp giữa các DU và các RU với định thời phù hợp các yêu cầu của các luồng URLLC (Ultra Reliable Low Latency Communication: truyền thông siêu tin cậy trễ thấp) . Chẳng hạn, giao diện FH bao gồm các kỹ thuật nén điều chế khác nhau và các mạng FH có thể được thiết kế để hỗ trợ URLLC với jitter tối thiểu.

Giao thức O-RAN FH bao gồm bốn chức năng khác nhau (hay bốn mặt phẳng). Ngoài các mặt phẳng người dùng U-plane) và mặt phẳng điều khiển (C-plane) (để vận chuyển dữ liệu và các lệnh điều khiển của lớp PHY), O-RAN FH còn có một mặt phẳng đồng bộ (S-plane) cho quản lý định thời giữa các DU và các RU, mặt phẳng điều khiển (M-plane) để lập cấu hình các chức năng RU từ chính DU. Dưới đây ta sẽ xét các chức năng của các mặt phẳng khác nhau này.

Mặt phẳng điều khiển (C-plane). Mặt phẳng điều khiển chịu trách nhiệm truyền các lệnh từ PHY-cao (phần cao của lớp vật lý) trong DU đến PHY-thấp (phần thấp của lớp vật lý), bao gồm các lập cấu hình cho lập lịch và tạo búp, quản lý các NR numerlogy khác nhau trong các khung con khác nhau, lập cấu hình tiền mã hóa đường xuống và điều khiển chia sẻ phổ tần. Đối với chia sẻ phổ tần, các đặc tả bao gồm các thủ tục truy nhập được cấp phép và được hỗ trợ (LAA: Licensed-Assisted Access), chẳng hạn cấp phép thực thiện nghe trước khi nói (LBT: Listen-Before- Talk) trong cặp RU và DU và các khai thác chia sẻ phổ động cho phép đặc tả các PRB nào được dành riêng cho phổ chia sẻ giữa LTE và NR.

Các bản tin mặt phẳng C được đóng bao vào các tiêu đề và tải trọng eCPRI hay IEEE 1914.2cùng với các trường đặc thù và các lệnh cho các thủ tục điều khiển khác nhau. Các đặc tả O-RAN FH cũng cung cấp các chi tiết về cách có thể liên kết các hướng dẫn của mặt phẳng điều khiển (chẳng hạn liên quan đến sử dụng một vectơ tạo búp đặc thù) với các gói dữ liệu mặt phẳng U (và theo cách đó các kỹ hiệu sẽ được phát).

Kết hợp mặt phẳng C và mặt phẳng M có thể được sử dụng để lập cấu hình và quản lý các khả năng tạo búp của RU, một tính năng quan trọng trong các mạng 5G, đặc biệt tại FR2 (vùng tần số vô tuyến 2 theo quy định của 3GPP). Cụ thể là, mỗi RU có thể hỗ trợ

các panel nhiều anten, mỗi panel có các mảng nhiều máy phát (TX) hoặc nhiều máy thu (RX). Mỗi mảng có thể được ánh xạ đến một hay nhiều luồng dữ liệu. Giao diện FH cho phép điều khiển biên và pha của các phần tử phát xạ trong các mảng pha (Phased Array) của RU (để tạo búp trong miền thời gian) hoặc các trọng số số mã hóa trước (để tạo búp trong miền tần số), với bốn tùy chọn tạo búp khác nhau. Với tạo búp cho búp được quy định trước, DU chọn động (thông qua mặt phẳng C) các vectơ tạo búp thời gian và (hoặc tần số) mà RU quảng cáo là khả dụng khi khởi đầu (thông quan mặt phẳng M). Một lựa chọn khác, với tạo búp dựa trên thuộc tính (Attribute Based Beamforming), RU có thể chọn các búp dựa trên các thuộc tính đặc thù, chẳng hạn các góc phương vị và nâng cao. Với tạo búp dựa trên trọng số, DU cũng có thể đặc tả các trọng số cho các vectơ tạo búp miền thời gian và (hoặc) tần số đặc thù. Tùy chọn cuối cùng là tạo búp dựa trên thông tin kênh, trong đó DU cung cấp RU thông tin trạng thái kênh (CSI :Channel State Information) đối với một người dùng đặc thù và RU tính toán các trong số tạo búp. O-RAN FH hỗ trợ một mô hình được định nghĩa rõ ràng cho các anten RU để DU có thể chắc chắn xác dịnh các phần tử anten, phân cực của chúng và định hướng panel.

Mặt phẳng U (U-plane). Chức năng chính của mặt phẳng U là truyền các mẫu I/Q trong miền tần số giữa RU và DU. Thông thươngf, môt bản tin mặt phằng C đặc tả cấu hình lập lịc và tạo búp và tiếp theo đặc tả này là một hay nhiều bản tin mặt phẳng U với các mẫu I/Q cần được phát theo các cơ hội truyền dẫn tương ứng. Mặt phẳng U cũng chịu trách nhiệm định thời truyền dẫn các bản tin này để xử lý trước khi phát và có thể nén chúng để truyền dữ liệu hiệu quả hơn.

Mặt phẳng S (S-plane). Mặt phẳng S chịu trách nhiệm đồng bộ thời gian, tần số và pha giữa các động hồ của các DU và các RU. Đây là chìa khóa cho hoạt động đúng của một hệ thống có phân chia vị trí thời gian và tần số được phân tán trên nhiều đơn vị. Nhờ tham chiếu đồng hồ chia sẻ, DU và RU có thể căn chỉnh đúng các tài nguyên thời gian và tần số cho phát và thu các kênh điều khển và dữ liệu của LTE và NR.

Mặt phẳng S của O-RAN FH có thể được triển khai với các cấu hình topo khác nhau tùy theo có liên kết DU-RU trực tiếp hay hai phần tử này được nối đến một kiến trúc các chuyển mạch Ethernet. Ngoài ra, các đặc tả FH cũng chứa các hồ sơ đồng bộ khác nhau dựa trên các giao thức khác nhau như PLFS (Physical Layer FrequencySignal: báo hiệu tần số lớp vật lý) hay PTP (Precision Time Protocol: giao thức thời gian chính xác) có thể đạt được độ chính xác thời gian nhỏ hơn micro giây.

Mặt phẳng M (M-plane). Mặt phẳng M của O-RAN FH là một giao thức chạy song song với các mặt phẳng C và M với các điểm cuối được dành riêng bên trong DU và RU để thiết lập các tunnel IPv4 và IPv6. Nó cho phép khởi xướng và quản lý kết nối giữa DU

và RU, và lập cấu hình RU. Trong ngữ cảnh này, các đặc tả dự đoán trước hai tùy chọn kiến trúc, chẳng hạn: kiến trúc phân cấp trong đó SMO quản lý DU và DU quản lý RU; kiến trúc lai trong đó SMO cũng có thể tương tác trực tiếp với RU. Như vậy, mặt phẳng M của O-RAN FH có thể hoạt động như là giao diện O1 của RU. Về phần O1, các hướng dẫn quản lý được xây dựng trên cơ sở NETCONF. Cuối cùng, trái ngược với các mặt phẳng C-, U-, S, mặt phẳng M được mật mã hóa đầu cuối đến đầu cuối thông qua SSH (Secure Shell: vỏ bọc an ninh) và (hoặc) TLS (Transport Layer Security: an ninh lớp vận tải).

Mặt phẳng M chịu trách nhiệm cho một số hoạt động liên quan đến vòng đời của RU. Trước tiên, nó quản lý khởi động, trong trong đó RU thiết lập quản lý với DU và (hoặc) SMO nhờ các địa chỉ IP được định nghĩa trước hay các cấu hình DHCP (Dynamic Host Configuaration Protocol: giao thức cấu hình máy chủ động). Sau đó, nó thiết lập các cập nhật phần mềm, quản lý cấu hình, hiệu suất và giám sát lỗi, và quản lý file cho truyền dữ liệu khối lượng lớn. Ngoài ra, mặt phẳng M còn quản lý đăng ký RU và PNF (Physical Network Function: chức năng mạng vật lý), các thông số của liên kết RU-đến DU (gồm cả định thời) và cập nhật các vectơ tạo búp (từ việc trển khai các bảng mã mới đến tạo độ nghiêng (tilting) và hiệu chỉnh các anten.

Ngoài việc đặc tả giao diện FH, Liên minh O-RAN còn phát triển một bộ các đặc tả để định rõ đặc trưng các khả năng vận tải và đồng bộ của tuyến trước hoặc tuyến chéo (Crossshaul: kết hợp cả tuyến trước và tuyến sau) để hỗ trợ kết nối giữa DU và RU. Các lĩnh vực được quan tâm khác liên quan đến quản lý mạng tuyến trước là: các mạng dựa trên ghép kênh phân chia theo sóng (Wave-Division- Multiplexing Network) và các kiến trúc mạng chuyển mạch gói với các tính năng hiện đại (chẳng han cắt lát). Trong ngữ cảnh này cần lưu ý rằng ETSI (European Telecommunications Standards Institute: viện tiêu chuẩn viễn thông châu Âu) đã công nhận tài liệu mặt phẳng đồng bộ, điều khiển và người dùng của giao diện FH của liên minh O-RAN như là một tiêu chuẩn ETSI.

12.5.5. Các giao diện khác

Giao diện O2 kết nối SMO đến O-RAN Cloud (đám mây O-RAN) cho phép quản lý và cung cấp các chức năng mạng theo cách lập trình. Nó tuân theo định nghĩa của danh mục thiết bị được quản lý bởi O-Cloud, thực hiện giám sát, cung cấp, khả năng chịu lỗi và các cập nhật. Do các chức năng này là chung cho một số vấn đề liên quan đến ảo hóa chức năng mạng (NFV: Network Func tion Virtualization), các hạ tầng và các tiêu chuẩn, nhóm WG6 của Liên minh O-RAN xem xét các ứng cử có thể cho thực hiện giao diện

này từ một tập các tiêu chuẩn đã được biết rõ và các giải pháp mã nguồn mở , chẳng hạn: các tiêu chuẩn EISI NFV liên quan, các giao diện 3GPP dựa trên dịch vụ, ngăn xếp mở Kubernetes (Kubernetes Open Stack và các đề án ONAP/OSM.

Cuối cùng như thấy trong hình 12.6, Kiến trúc phân nhỏ của O-RA cũng sử dụng các giao diện bổ sung được định nghĩa bởi 3GPP. Nổi bật là, giao diện E1 kết nối CU đến DU với các giao diện con cho các mặt phẳng người dùng và điều khiển. Giao diện Xn (X2) kết nối các gNB (các eNB) khác nhau, chẳng hạn: thực hiện chuyển giao hoặc cho phép kết nối kép. Giao diện Uu tồn tại giữa UE và các gNB và giao diện NG kết nối gNB với mạng lõi 5G, chẳng hạn: UPF (User Plane Function: chức năng mặt phẳng người dùng) cho mặt phẳng người dùng và AMF (Access Management Function: chức năng quản lý mạng) cho mặt phẳng điều khiển.

12.6. HOẠT ĐỘNG AI/ML TRONG O-RAN

Phần này sẽ trình bày chi tiết tổng quan các thủ tục và các bước điều chỉnh các luồng công việc trong kiến trúc O-RAN. Tiếp sau, phần này sẽ trình bày hướng dẫn từng bước vòng đời của luồng công việc này : từ thu tập dữ liệu đến triển khai và thực hiện thực tế trí tuệ mạng theo thời gian thực dựa trên các thành phần kiến trúc và các giao diện được mô tả trong các phần 2,3 và 4, và các giao diện được trình bày trong phần 5.

Luồng công việc AI/ML được nhóm nghiên cứu O-RAN WG2 cùng với các đặc tả trong [49]. Luồng công việc này bao gồm sáu bước chính: (i) thu thập và xử lý dữ liệu, (ii) đào tạo, (iii) thẩm định, (iv) triển khai, (v)thực hiện và suy luận AI/ML và (vi) hoạt động liên tuc. Các bước khác nhau của luồng công việc AI/ML đầu cuối-đầu cuối hỗ trợ các nhà thiết kế mạng tránh được các vấn đề AI/ML trong mạng, bao gồm, chẳng hạn: độ khả dụng hay tính tổng quát hạn chế của dữ liệu, đào tạo không đúng, định nghĩa không gian hành động và thiết kế tác nhân, kiểm tra và thẩm định bị giới hạn và thiếu hạ tầng cho khối lượng công việc của AI/ML bền vững (Resilent AI: AI có khả năng chịu đựng và phuục hồi).

Để minh họa và làm rõ, dưới đây ta sẽ mô tả các thủ tục liên quan đến vòng đời AI/ML trong các hệ thống O-RAN. Để làm thí dụ ta xét trường hợp trong đó nhà mạng nhắm đến phát triển, đào tạo và triển khai một xApp để điều khiển các chính sáchcắt lát RAN bằng cách thính ứng chúng theo gần thời gian thực với tải mạng hiện thời và yêu cầu lưu lượng thông qua các giải thuật dựa trên AI. Trong thí dụ này trạm gốc làm chủ ba lát cắt, cụ thể là một lát cắt URLLC (Ultra-Reliable Low Latency Communications: truyền thông trễ thấp siêu tin cậy) cho các dịch vụ trễ thấp siêu tin cậy, một lát cắt eMBB

(Enhanced Mobile Broadband: băng rộng di động tăng cường) cho lưu lượng thông lượng cao (chẳng hạn: truyền video trực tuyến và truyền giao file) và mMTC (massive Machine-Type Communications: truyền thông kiểu máy số đông) để xử lý lưu lượng được tạo ra, chẳng hạn, bởi các bộ cảm biến nhỏ và các thiết bị IoT. Mục đích của xApp là điều khiển các cnính sách cắt lát RAN bằng cách ấn định các PRB (Physical Resource Block: khối tài nguyên vật lý) khả dụng cho từng lát cắt để thỏa mãn các yêu cầu hiệu xuất đa dạng.

Thu thập và xử lý dữ liệu. Trước hết, dữ liệu được thu thập trên các giao diện O1, A1 và E2 và được lưu trữ trong các tập dữ liệu lớn (chẳng hạn: các kho tập trung của hồ dữ liệu (Data Lake)), nơi mà nó có thể được lấy ra theo yêu cầu. Thiết kế kỹ lưỡng các giải pháp báo cáo dữ liệu cho các giao diện này sẽ thúc đẩy thu tập các tập dữ liệu lớn, có thể hỗ trợ các giải pháp AI/ML tổng quát hóa tốt hơn.

Ngoài ra, do các giải pháp AI/ML khác nhau có thể sử dụng các kiểu KPM khác nhau được thu thập trong các chu kỳ thời gian khác nhau và với độ chi tiết khác nhau (chẳng hạn: thông lượng, độ trễ, sơ đồ điều chế và mã hóa (MCS: Modulation and Coding Scheme), thông tin chất lượng kênh (CQI: Channel Quality Index: chỉ số chất lượng kênh), yêu cầu dữ liệu, jitter, v.v…), các đặc tả O-RAN cũng xem xét bước tiền xử lý sơ tập dữ liệu (hay chuẩn bị trước). Tại đây, dữ liệu cho cả đào tạo lẫn suy luận trực tuyến được tạo dạng và định dạng theo kích thước đầu vào của mô hình AI/ML đặc thù được xét. Bước này có thể bao gồm sử dụng các bộ tự mã hóa (Autocoder) để giảm kích thước cũng như các thủ tục xử lý dữ liệu AI/ML được định nghĩa rõ ràng, chẳng hạn chuẩn hóa, đóng kín và tạo lại dạng. Tiền xử lý này giúp tránh được các vấn đề kích thước hoặc các vấn đề cung cấp dữ liệu bị định dạng kém cho các tác nhân AI/ML.

Thí dụ: liên quan đến xApp điều khiển các chính sách cắt lát RAN, bước này bao gồm thu thập dữ liệu và các số đo hiệu suất trên giao diện O1 để tạo ra tập dữ liệu đào tạo sử dụng cho bước tiếp theo (nghĩa là, giai đoạn đào tạo). Chẳng hạn, vì xApp phảicó khả năng thích ứng các chính sách cắt lát RAN cho các lát cắt khác nhau theo yêu cầu dữ liệu và các mức hiệu suất đòi hỏi hiện thời, dữ liệu được thu thập phải gồm khối lượng các PRB cần thiết để phát dữ liều được yêu cần đến từng người sử dụng của ba lát cắt này, cũng như các kết quả đo thông lượng (eMBB), số lượng các gói (mMTC) và trễ (URLLC).

Đào tạo. Các đặc tả O-RAN không cho phép tiển khai bất kỳ một giải pháp được điều khiển bởi dữ liệu (Data Driven Solution) không được đào tạo: tất cả các mô hình AI/ML đều phải trải qua giai đoạn đào tạo ngoại tuyến. Điều này đảm bảo độ tin cậy của trí tuệ và tránh các vấn đề điển hình của các mô hình AI/ML bị đào tạo kém hoặc không được

đào tạo có thể dẫn đến sự cố hoặc sự kém hiệu quả trong mạng, chẳng hạn: các dự đoán, phân loại và (hoặc) hành động không chính xác. Tuy nhiên điều này không loại trừ O-RAN vẫn hỗ trợ đào tạo trực tuyến, nếu nó chỉ được sử dụng để vi chỉnh và cập nhật mô hình được đào tạo ngoại tuyến trước đó. Thí dụ: nhà mạng có thể đào tạo các các dạng giải thuật khác nhau để điều khiển số lượng các PRB được ấn định cho từng lát cắt nhưng khác nhau theo chi tiết thực hiện của chúng. Thí dụ, nhà mạng có thể đào tạo một tập các tác nhân học tăng cường sâu (DRL: Deep Reinforcement Learning), các cây quyết định, các khuôn dạng kết hợp và đầu vào khác nhau (chẳng hạn: tập con các KPM đặc thù và số lượng của chúng), các kiến trúc khác nhau (chẳng hạn: độ sâu và độ rộng của một tác nhân DRL, số lượng cacsneuron, v.v…). Mục đích của thủ tục này là đào tạo số lượng lớn các giải thuật AI và nhận diện các giải thuật nào phù hợp nhất để thực hiện một nhiệm vụ đặc thù.

Thẩm định và công bố. Một khi các mô hình đã được đào tạo, chúng phải trải qua giai đoạn thẩm định để đảm bảo rằng chúng tin cậy, bền vững và hiệu qủa khi thực hiện phân loại, dự đoán hay các nhiệm vụ điều khiển. Nếu thẩm định thành công và các mô hình dường như đã sẵn sàng cho triển khai, chúng được công bố và được lưu trữ trong AI/ML catalog trên SMO/RIC phi thời gian thực. Nếu không, chúng phải trải qua thiết kế lại bổ sung cho đến khi thẩm định thành công. Thông thường, các tác nhân không đáp ứng các yêu cầu thẩm định khi các không gian đầu vào và đầu ra không được thiết kế hợp lý hoặc khi quá trình mô hình hóa được thực hiện bởi các mạng neuron của tác nhân thất bại trong việc nhận được các quan hệ quan trọng giữa đầu và đầu ra. Thí dụ: sau khi đào tạo hoàn tất, các giải thuật AI khác nhau được so sánh với nhau và đối chiếu với các tập dữ liệu thẩm định khác nhau bao gồm cả dữ liệu chưa từng được thấy để xác định các mô hình hiệu quả nhất trong diều khiển các chính sách cắt lát RAN. Thí dụ, kiểm tra thẩm định điển hình bao gồm đánh giá các giải pháp AI đa dạng thực hiện trên các mẫu lưu lượng đa dạng và yêu cầu, số lượng và phân phối người dùng, các tần số công tác và băng htông khả dụng. Thủ tục này có thể hoặc chỉ ra giải pháp AI không thực hiện tốt và cần được đào tạo lại, cũng như xác định tập con các giải thuật AI có thể công bố cho AI/ML catalog, cũng như cung cấp thông tin bổ sung về các điều kiện mạng (chẳng hạn: tải mạng, mẫu di động, kích thước triển khai) , nhờ đó giải pháAI đặc thù đem đến hiệu suất tốt nhất để nhà mạng có thể triển khai giải pháp AI phù hợp nhất cho một triển khai đặc thù.

Triển khai. Các mô hình lưu trữ trong AI/ML catalog có thể được tải xuống, được triển khai và được thực hiện theo hai tùy chọn là triển khai dựa trên ảnh và triển khai dựa trên file. Trong cả hai trường hợp triển khai mô hình được thực hiện bằng cách sử dụng giao diên O1 và nút nơi mô hình thực hiện được gọi là máy chủ suy luận (Inference Host).

Trong triển khai dựa trên hình ảnh, mô hình AI/ML thực hiện như là một ảnh được côn-te-nơ hóa trong dạng một ứng dụng O-RAN (chẳng hạn: xApps và rApps) được tiển khai tại các nút O-RAN nơi mà nó được triển khai để thực hiện suy luận trực tuyến. Hiện thời các nút này chỉ giới hạn đến các RIC và triển khai AI tai các CU/DU vẫn đang được nghiên cứu. Triển khai dựa trên file xét trường hợp mô hình AI/ML được tải xuống như là một file đọc lập triển khai bên trong một môi trường suy luận (bên ngoài miền ứng dụng của O-RAN), chuyển tiếp đầu ra suy luận của mô hình đến một hay nhiều ứng dụng O-RAN. Thách thức chính iên quan đến triển khai AI/ML là thỏa mãn nhu cầu và (hoặc ý định) của nhà mạng. Thí dụ: nhà mạng muốn chọn các mô hình căt lát RAN dựa trên AI được đào tạo từ AI/ML catalog và triển khai chúng như là xApps sẽ được triển khai trong RIC gần thời gian thực.

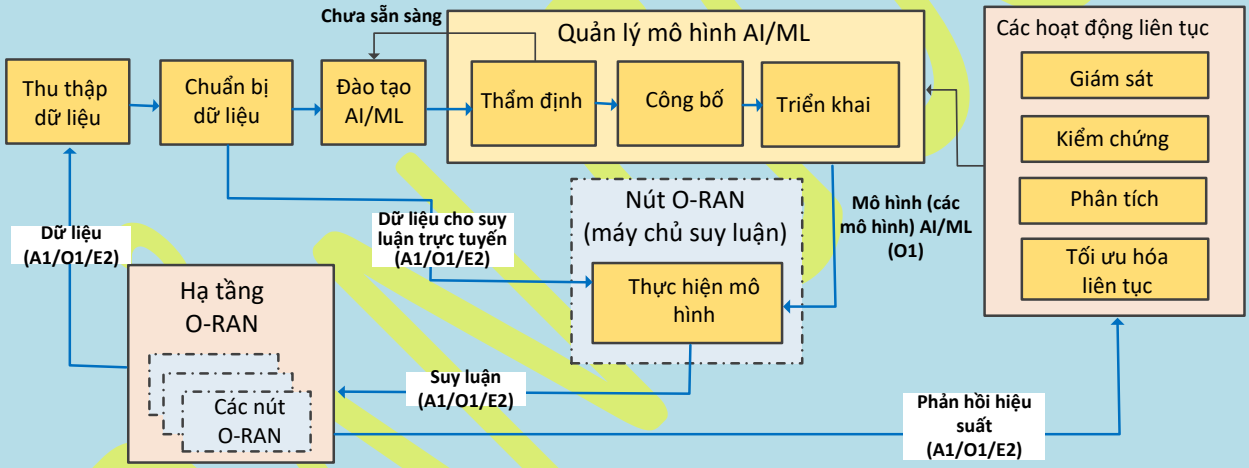
Triển khai và suy luận AI/ML. Sau khi các mô hình đã được triển khai trong máy chủ suy luận (Inference Host), chúng được cung cấp dữ liệu để thực hiện các nhiệm vụ suy luận trực tuyến đa dạng. Các nhiệm vụ này bao gồm các nhiệm vụ phân loại và dự đoán tại cả hai RIC (được phát trên các giao diện A1 và E2) và thực hiện quản lý cũng như các hành động điều khiển trên các giao diện O1 và E2). Tại bước này, điều cực kỳ quan trọng là triển khai các giải thuật suy luận phù hợp với thời gian biểu của của từng vòngđiều khiển. Thí dụ, tại RIC gần thời gian thực một xApps với thành phần AI/ML cần hoàn tất các tính toán lâu nhất là trong vài trăm mi li giây (có thể ít hơn) để đảo bảo vòng điều khiển đóng trong ít hơn 1s. Trong khi đo suy luận là nhiệm vụ ít phức tạp hơn đào tạo, thời gian thực hiện vòng cần được kiểm tra được lập hồ sơ để đảm bảo rằng nó đem đên hiệu suất thỏa mãn tại thang thời gian kỳ vọng (xem hình 12.2). Thí dụ: Sau khi xApps đã được trển khai trên RIC gần thời gian thực, điều khiển cắt lát RAN được thực hiện bằng cách thực hiện các hoạt động trong hình 12.9) trong đó: (i) xApp được cung cấp các KPM (chẳng hạn: các RRBvà trễ được yêu cầu, các đo đạc thông lượng) được thu thập trên giao diện E2; (ii) tính toán các hành động điều khiển sẽ dược sử dụng để hương dẫn DU và ấn định các PRB khả dụng cho các lát cắt khác nhau theo gần thời gian thực.

Các hoạt động liên tục. Một nét quan trọng của luồng công việc AI/ML là khả năng giám sát và phân tích trí tuệ được triển khai trên toàn bộ mạng để kiểm chứng rằng các đầu ra suy luận AI/ML là hiệu quả, chính xác và không ảnh hưởng tiêu cực hiệu suất mạng. Các hoạt động liên tục đảm bảo rằng các mô hình thực hiện trực tuyến xấu có thể được tinh chỉnh để cải thiện các chức năng của chúng, tránh các vấn đề liên quan đến không khả dụng dữ liệu và (hoặc) dịch vụ, và hỗ trợ cập nhật các mô hình AI/ML mà không làm gián đoạn dịch vụ. Thí dụ: trong trường hợp của chúng ta, nhà mạng có thể giam sát thường xuyên hiệu suất của xApp căt lát RAN và nếu bất kỳ các bất thường hay

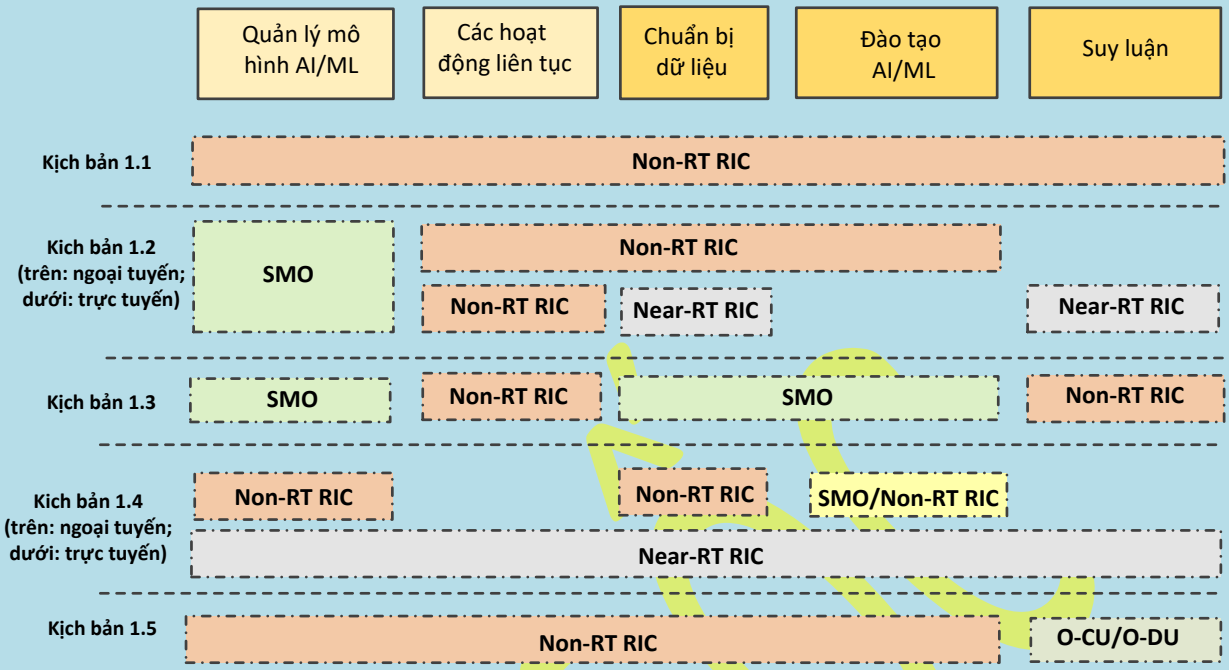
khiếm khuyến bị phát hiện, nó có thể quyết định đào tạo lại mô hình AI/ML được nhúng trong xApp theo dữ liệu mới được thu thập thông qua các giao diện O1 và E2.

12.6.1. Các kịch bản triển khai AI/ML

Một trong các tính năng chính của các mạng 5G là khả năng hỗ trợ rất nhiều dạng các trường hợp và các ứng dụng khác nhau. Xét tới tính đa dạng của hệ sinh thái 5G, rõ ràng rằng giải pháp một thiết kế phù hợp tất cả để triển khai và điều khiển trí tuệ mạng khó có khả năng tồn tại. Vì thế, liên minh O-RAN đã đặc tả năm kịch bản triển khai khác nhau để định nghĩa vị trí nơi các thành phần khác nhau của luồng công việc trong hình 12.15 được khởi đầu và thực hiện. Mặc dù các kịch bản triển khai này (mô tả trong hình 12.16) bao phủ một tập lớn các trường hợp sử dụng của thế giới thực, các triển khai thực tế có thể lệch khỏi các kịch bản này khi đáp ứng các yêu cầu của nhà mạng và ứng dụng đặc thù.



Hình 12.15. Luồng làm việc AI/ML trong O-RAN. Hạ tầng mạng cung cấp dữ liệu thông qua các giao diện O-RAN đến thu thập dữ liệu và chuẩn bị các khối logic chuẩn bị. Sau đó, các mô hình được đào tạo, được thẩm định, được triển khai và được thực hiện trên các nút O-RAN (các RIC). Các mô hình có thể được tinh chỉnh thêm thông qua giám sát và tối ưu liên tục dựa trên phản hồi hiệu năng từ RAN.



Ký hiệu:

Non-RT RIC (Non Real Time Radio Intellegence Controller): Bộ điều khiển trí tuệ RAN phi thời gian thực; near-RT RIC (near Real Time Radio Intellegence Controller): bộ điều khiển trí tuệ RAN gần thời gian thực.

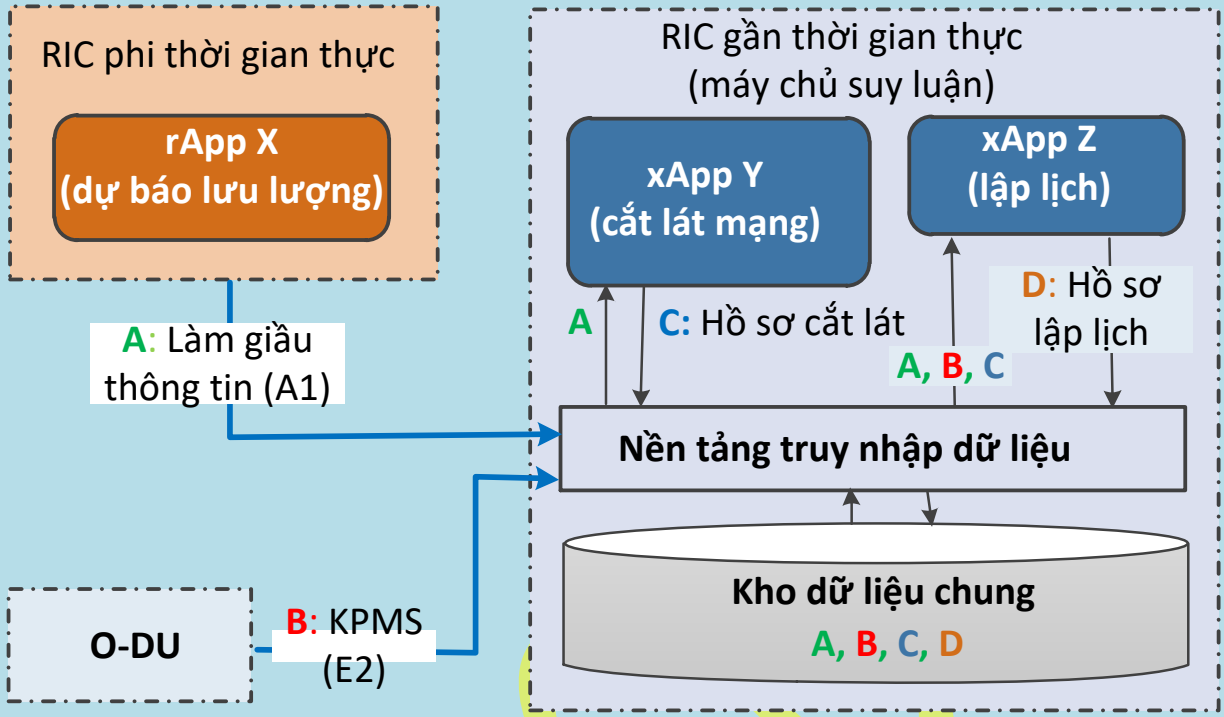
Hình 12.16.Các kịch bản triển khai O-RAN AI/ML. Các kịch bản khác nhau nhúng các thành phần khác nhau của luồng công tác trong các thành phần khác nhau của RAN, từ SMO/non-RT RIC đến near-RT RIC hay các nút RAN.

Như đã nói trước đây, Các đặc tả O-RAN được thiết kế đặc biệt cho phần RAN của mạng và các chức năng của nó. Tuy nhiên, cũng cần lưu ý rằng O-RAN và các RIC của nó có thể ảnh hưởng lên các quyết định suy luận liên quan đến hạ tầng mạng lõi và MEC (Multi-Access Edge Computing: điện toán biên đa kết nối). Tất nhiên, SMO có thể hành động như là một cổng giữa RAN và các thành phần mạng khác, vì nó có khả năng của các chức năng điều phối xuyên suốt toàn mạng. Theo cách này, thực hiện xApps và rApps bên trong môi trường O-RAN có thể được sử dụng để thu thập dữ liệu trên RAN (chẳng hạn: dự báo tải lưu lượng, dự báo di động, các thay đổi trạng thái mạng) có thể được sử dụng bởi SMO (chẳng hạn: ONAP ((Open Network Automation Platform) hay OSM (Open Service Management and Orchestration)) để nhận các thông báo quyết định về khởi đầu và chuyển dịch vụ MEC cũng như các chính sách cắt lát trong mạng lõi.

12.6.2. Thu thập các đầu vào cho suy luận trược tuyến

Dữ liệu cho suy luận trực tuyến có thể được thu thập từ nhiều người sản xuất dữ liệu (Data Producer) trên các giao diện O1, A1 và E2 được thiết kế để hỗ trợ các vòng điều khiển O-RA Nhoạt động trong các phạm vi thời gian khác nhau. Giao diện O1 cho phép các phần tử trong miền SMO/RIC phi thời gian thực thu thập dữ liễu từ mọi phần tử quản lý của O-RAN và thực hiện tối ưu hóa phi thời gian thực. Giao diện A1 có thể được sử dụng bởi RIC phi thời gian thực để gửi làm giầu thông tin từ miền SMO/RIC phi thời gian thực đến RIC gần thời gain thực và các ứng dụng của nó. Chẳng hạn, một rApp trong RIC phi thời gian thực có thể gửi làm giầu thông tin đến RIC gần thời gian thực về sự tiến hóa KPI được dự đoán sau vài giây tiếp theo. Cuối cùng, giao diện E2 cho phép RIC gần thời gian thực và các xApps của nó thu thập dữ liệu từ các nút E2 (chẳng hạn các KPI) cho điều khiển RAN gần thời gian thực. Cần lưu ý rằng, các đặc tả O-RAN cũng xét tới trường hợp trong đó dữx liệu có thể được tạo ra trong cùng máy chủ suy luận bơi các nhà sản xuất dữ liệu khác nhau. Đây là trường hợp các mô hình A/?ML bị ràng buộc theo chuỗi (Chianned AI/ML Models), trong đó một nhiệm vụ điều khiển bao gồm một số nhiệm vụ con, mỗi nhiệm vụ con bao gồm một mô hình khác nhau và yêu cầu nhiều dạng dưc liệu đầu vào và các kiểu.

Để điều chỉnh sản xuất và tiêu thụ dữ liệu (Data Consumption) giữa các ứng dụng được chủ trì bởi cùng một nút (chẳng hạn, RIC gần thời gian thực), các đặc tả O-RAN đặt ra cơ sở cho nền tảng truy nhập dữ liệu (Data Access Platfform) để điều chỉnh sản xuất, chia sẻ và truy nhập dữ liệu. Nền tảng này hoạt động như là một phần mềm trung gian giữa các ứng dụng và một kho dữ liệu chung trong đó dữ liệu được lưu trữ và chia sẻ. Để cung cấp một thí dụ mức cao cho vấn đề này, hình 12.17 cho thấy cách thức mà dữ liệu được tạo ra từ các nguồn khác nhau được các ứng dụng O-RA tiêu thụ khi thực hiện các nhiệm vụ không đồng nhất. Ta xét trường hợp một rApp dự báo tải lưu lượng (dữ liệu kiểu A). Hai xApps (xApp Y và xApp Z) sử dụng AI để điều khiển các chiến lược cắt lát và lập lịch. xApp Y tiêu thụ dữ liệu kiểu A nhận được từ nền tảng truy nhập dữ liệu và sản xuất ra một hồ sơ cắt lát (dữ liệu kểu C), trong khi đó xApp Z tiêu thụ dữ liệu các kiểu A, B, C đến từ nền tảng truy nhập dữ liệu (Với dữ liệu kiểu B bao gồm các KPI được gửi đi từ một DU trên giao diện E2) và chọn một hồ sơ lập lịch D được sử dụng bởi DU được điều khiển.



Hình 12.17. Thí dụ về các mô hình AI/ML bị ràng buộc theo chuỗi với đa dạng các kiểu đầu vào và các nhà sản xuất dữ liệu. Chữ A biểu thị dự báo lưu lượng được tạo bởi rApp X. Chữ B biểu thị các KPM từ RAN: xApp Y sử dụng dữ liệu A để tao ra hành động C (chẳng hạn, một hồ sơ cắt lát). xApp Z sử dụng dữ liệu A, B và điều khiển C làm đầu vào để tạo ra một hành động điều khiển mới (chẳng hạn: hồ sơ lập lịch).

12.6.3. Sử dụng AI/ML để thiết kế các hệ thống O-RAN

Một ứng dụng cụ thể cần nhắc đến là sử dụng chính cùng một hạ tầng do O-RAN cung cấp để thực hiện thiết kế và các hoạt động hiệu suất và hiệu quả hơn.

Mặc dù về mặt nguyên tắc ta có thể sử dụng AI/ML để cải thiện các hoạt động hay các chức năng được thực hiện trong kến trúc O-RAN, các ứng dụng thích hợp nhất bao gồm quản lý hạ tầng phần mềm , các giao diện mở và an ninh.

Chẳng hạn, một ứng dụng khá th ích hợp trong đó AI/ML đóng một vai trò quan trọng là đưa ra các quết định trí tuệ dựa trên điều chỉnh vòng đời của tất cả các thành phần phần mềm trong hạ tầng O-RAN. Hiển nhiên, AI/ML có thể được sử dụng để tự động hóa và tối ưu hóa khởi đầu, sắp đặt và quản lý các RIC để thích ứng động các điều kiện mạng và tải lưu lượng để giảm thiểu tiêu thụ năng lượng trong khi gảm gánh nặng của từng RIC.

Tương tự, AI/ML có thể được sử dụng để quyết định khởi đầu tối ưu của các CU và các DU.

Một ứng dụng thú vị khác là sử dụng AI/ML để hỗ trợ thiết kế các giao diện mở. AI/ML có thể được sử dụng để quyết định tốc độ thu dữ liệu tối ưu trên từng giao diện để thích ứng và nhận được các thay đổ hiện thời. Chẳng hạn, trong các môi trường thay đổi chậm, AI/ML có thể giảm tốc độ thu thập số liệu trên các giao diện này (chẳng hạn: giao diện E2 và O1), nhờ vậy giảm chi phí gián tiế (Overhead). Tương tự ta có thể tăng tốc độ thu dữ liệu trong trường hợp các điều kiện thay đổi nhanh đòi hỏi dữ liệu cập nhật thường xuyên để các RIC thực hiện các quyết định tối ưu và đúng lúc với trả giá tăng chi phí gián tiếp.

Cuối cùng an ninh là một trường hợp sử dụng có thể khác trong đó AI/ML có thể được sử dụng cho các cơ chế an ninh được tinh chỉnh để đảm để đảm bảo an ninh cho kiến trúc O-RAN. Chẳng hạn, các công cụ như DRL có thể được sử dụng để điều chỉnh thích ứng mức an ninh trong mạng bằng các điều khiển các tốc độ mật mã, phát hiện các kẻ xâm nhập, các dịch vụ và các chức năng di động để giảm khả năng chúng trở thành mục tiêu của các nút phản nghịch.

12.7. CÁC TRƯỜNG HỢP SỬ DỤNG VÀ NGHIÊN CỨU O-RAN

Các khả năng được RIC đưa ra như các giao diện mở và luồng cong việc AI/ML được xét trong các phần 3,4 và 5 làm cho nó có thể hỗ trợ các trường hợp sử dụng và các kịch bản tiên tiến cho điều khiển RAN và tự tối ưu hóa triển khai.

Những năm gần đây chứng kiên sự tăng nhanh các nghiên cứu được thúc đẩy bởi O-RAN về các trường hợp sử dụng và các ứng dụng, chẳng hạn về thiết kế các xApps và rApps hay về cấu hình tối ưu của mạng O-RAN. Phần này sẽ mô tả các lĩnh vực trong đó ứng dụng của mô hình O-RAN là hứa hẹn nhất và các kết quả nghiên cứu mới nhất cho thấy làm thế nào có thể sử dụng các giải pháp dựa trên O-RAN để tối ưu hóa RAN.

Liên minh O-RAN đã thu thập một danh mục bao quát gồm 9 trường hợp sử dụng mẫu cho các triển khai O-RAN và các tài liêu nghiên cứu cũng bàn luận ở mức cao một số trường hợp này. Ở mức cao, các kịch bản và các trường hợp sử dụng có thể được phân loại theo các cách khác nhau, chẳng hạn: bằng cách xem xet núm điều khiển (Control Knob) hay mục đích suy luận, hay miền được điều khiển hay được tối ưu hóa (chẳng hạn: một UE, một nút RAN hay toàn bộ mạng). Liên quan đến các núm kđiều khiên, ta nói về

các mục tiêu đặc thù của E2 trong phần 5.1. Một các tổng quát hơn, ta có thể nhận diện một số lĩnh vực quan tâm như sau.

Di động (Mobility). Các mạng RAN mở có thể ảnh hưởng lên quản lý di động hay hiệu suất của các người dùng di động bằng cách điều chỉnh chuyển giao, cân bằng tải, đa kết nối, chặn truy nhập và các thông số tạo búp trong RAN. Khác với một mạng 3GPP truyền thống, điều này có thể được thực hiện theo cách vòng kín bằng cách sử dụng hiểu biết trạng thái của nhiều trạm gốc và dự đoán di động của người dùng dựa trên thông tin từ RAN hoặc làm giầu thông tin từ bên ngoài. Chẳng hạn, một nghiên cứu đã trình bày một cơ chế dựa trên O-RAN cho dự đoán tải của nhiều trạm gốc trong một mạng tổ ong, với ứng dụng cho định tuyến động các thiết bị tự hành để tránh gây nghẽn. Một số tài liệu O-RAN cũng bao gồm chuyển giao dựa trên tình huống cho các kịch bản xe cộ, trong đó xApps khai thác thông tin làm giầu từ RIC phi thời gian thực và suy luận trên dữ liệu AI/ML RAN để quản lý các chuyển giao và các ứng dụng liên quan đến UAV, với cấu hình các thông số RAN dựa trên quỹ đạo kỳ vọng của UAV.

Ấn định tài nguyên. Điều khiển trong lĩnh vực này bao gồm cắt lát, lập lịch, cung cấp dịch vụ và các chức năng mạng. Liên quan đến điều khiển di động, các ưu điểm của RAM mở so với các mạng tổ ong truyền thống nằm ở khả năng thích ứng các bối cảnh tiến chiển độngvới các yêu cầu của người dùng (chẳng hạn đối với các lát cắt khác nhau) và với các sự kiện bên ngoài làm thay đổi trạng thái và cấu hình của RAN.

Vì thế, nghiên cứu điều khiển trí tuệ O-RAN đã tiếp nhận cắt lát mạng là một trong số các lĩnh vực hứa hẹn và quan tâm nhất cho tối ưu hóa dựa trên học máy. Bonati và các cộng sự phân tích các cách tiếp cần dựa trên dữ liệu trong O-RAN và cung cấp trình diễn đầu tiên về vòng điều khiển kín của một mạng tổ ong được mềm hóa được tạo ra trên một nền tảng thực nghiệm phạm vi lớn. Hiệu suất của RAN được tối ưu hóa thông qua các xApps điều khiển các chính sách lập lịch của các lát căt khác nhau trên các trạm gốc. Các lát cắt có các mục tiêu tối ưu hóa khác nhau, chẳng hạn: đói với lát cắt URLLC phần thưởng là cực tiểu hóa sự chiếm dụng bộ đệm đóng vai trò proxy (máy chủ trung gian) cho trễ đầu cuối đầu cuối. Một nghiên cứu khác đề xuất CoIO-RAN (Colosseum O-RAN: bộ giả lập mạng vô tuyến Colosseum sử dụng các thành phần O-RAN) , một đường ống cho thiết kế, đào tạo kiểm tra và đánh giá thực nghiệm của các vòng điều khiển dựa trên DRL (Deep Renforcement Learning). Các khả năng của ColO-RAN được phô diễn thông qua xApps điều khiển các chính sách lập lịch và ấn định cắt lát RAN và cho đào tạo các mô hình ML trực tuyến. Jonhson và các cộng sự đề xuất NexRAN, một xApp thực hiện điều khiển vòng kín cắt lắt các tài nguyên của các trạm gốc được mềm hóa và trình diễn nó trên nền tảng chương trình POWER U.S. Sarikaya và các cộng sự xem xét sắp đặt các lát cắt RAN trong 5G RAN mở nhiều tầng (Multi-Tier Open 5G

RAN) và chính thức hóa nó bằng toán học, chỉ ra các lợi ích của phân chia chức năng lnh hoạt so với các tùy chọn phân chia cố định. Niknam và các cộng sự phân tích các nguyên lý và các yêu cầu của các đặc tả O-RAN và đề xuất một sơ đồ thông minh cho quản lý các tài nguyên vô tuyến và nghẽn lưu lượng. Hiệu quả của giải pháp này đã được chứng minh trên dữ liệu của nhà mạng thực tế. Tương tự, Mnggarri đánh giá hiệu suất của quản lý tài nguyên vô tuyến dựa trên ML trong các mạng được O-RAN quản lý thông qua một xApp được triển khai trên RIC gần thời gian thực và dánh giá nó trong một thiết lập phòng thí nghiệm nhỏ. Một số tác giả cũng tập trung lên quá trình chọn ô, để chỉ ra cách thức O-RAN có thể hỗ trợ cải thiện ấn định các người dùng đến các ô đặc thù dựa trên các số đo thông lượng được dự báo thay vì các số đo đơn giản mức báo hiệu.

Lien và các cộng sự và Filali và các cộng sự xem xét tôi ưu hóa các tài nguyên vô tuyến dựa trên xApp cho các người dùng URLLC. Bài báo đầu tiên chi ra làm cách nào một tác nhân học tăng cường chạy trong RIC gần thời gian thực có thể điều khiển hiệu quả khởi tạo các phiên URLLC mới tốc độ bit được đảm bảo và lập cấu hình các thông số mức phiên để tăng xác suất hội nhập thành công các người dùng URLLC mới. Trong khi đó, bài báo của Filali và các cộng sự nghiên cứu một cơ chế cắt lát dựa trên O-RAN để điều khiển ấn định tài nguyên cho các người dùng URLLC và quản lý để đạt được các mục tiêu chất lượng dịch vụ.

Ituria và các cộng sự trình bày một khung cho phối hợp nhiều xApps và rApps thông hoa học nhóm nhiều tác nhân để có thể ấn định các tài nguyên thông qua hạ tầng AI/ML sử dụng quan hệ gần gui và phân cấp giữa nhiều xApps/rApps.

Có thể tìm thấy các thí dụ bổ sung trong các tài liệu được công bố, chẳng hạn ấn định tài nguyên cho các người dùng di động (bao gồm cả các UAV) bằng dự đoán di động trước và ấn định tài nguyên dựa trên QoS. Mục đích của ấn định tài nguyên dựa trên QoS là cung cấp động tập các tài nguyên mà các người dùng được lựa chọn yêu cầu để thỏa mãn QoS của họ, chẳng hạn thông qua cắt lát cho mục đích này và ấn định các PRB sau đó cho các lát cắt dựa trên QoS. Cuối cùng, hạ tầng O-RAN cũng có thể được sử dụng để dự đoán tình huống nghẽn và áp dụng khắc phục tương ứng bằng cách bổ sung tài nguyên.

Điều khiển dựa trên QoE/QoS. Tối ưu hóa các nguồn tài nguyên RAN để đáp ứng các yêu cầu chất lượng dịch vụ (QoS: Quality of Service) và chất lượng giải nghiệm (QoE: Quality of Experience) cũng mở rộng ra ngoài ấn định tài nguyên. Chẳng hạn, Bertizzolo và các cộng sự xem xét các ứng dụng phát video trực tuyến được hỗ trợ bởi Drone và đề xuất một hệ thống điều khiển cho O-RAN đẻ tối ưu hóa kết hợp định hướng phát và vị trí drone. Giải pháp này được đánh giá cả bằng thực nghiêm trong nền tảng thử nghiệm (Testbet) RAN ngoài nhà và thông qua các mô phỏng số.

Liên minh O-RAN sử dụng các trường hợp bao gồm một kịch bản tối ưu hóa QoE trong đó các đầu vào từ các hệ thống bên ngoài, bằng cách giám sát hiệu suất ứng dụng, và kết hợp dữ liệu đa chiều để nhận diện cấu hình RAN tố nhất cho tối ưu hóa QoE trên cơ sở người dùng. Ở đây, các giao diện O-RAN đóng vai trò duy nhất, vì chúng cho phép kết hợp đầu vào dữ liệu kết hợp (Fuse data) từ các nguồn hỗn tạp khác nhau (RAN, bên ngoài) theo một cách bình thường không thể có trong các triển khai RAN truyền thống.

Chia sẻ RAN (RAN Sharing). Trường hợp sử dụng này bao trùm các kịch bản khác nhau, từ chia sẻ phổ đến chia sẻ hạ tầng trên môt kiến trúc máy chủ tế bào thần kinh (Neural Host Architecture), nó được kỳ vọng tăng hiệu suất năng lượng, phổ và giảm chi phí đầu tư, vận hành. Kịch bản này kết hợp tính linh hoạt của mềm hóa và ảo hóa RAN với tính năng động và tính khả lập lại cấu hình của điều khiển vòng kín (bao gồm cả thông tin bên ngoài). Các đặc tả O-RAN bao gồm nhiều cơ chế cho chia sẻ phổ, bao gồm cát lát và điều khiển truy nhập phổ động tại DU/RU.

Một nghiên cứu kịch bản chia sẻ giữa một 5G RAN và chùm vệ tinh quỹ đạo thấp (cụ thể đường lên giữa tạm mặt đất và vệ tinh) được quản lý bởi RIC đã được thực hiện. Nghiên cứ unày bao gồm đầu vào một khung cảm biến phổ độc lập RAN và một cơ chế chia sẻ áp dụng trên một cơ sở UE bởi RIC. Một nghiên cứu khác cũng đề xuất một giải pháp chia sẻ phổ giữa các công nghệ được hỗ trợ bởi O-RAN, nhưng trong trường hợp này cảm biến được thực hiện trong chính DU và RU bằng cách phân tích học máy sâu các tính hiệu thu trên cơ sở I/Q. Sơ đồ chia sẻ được đề xuất phát hiện các người dùng Wi-Fi (hay các người dùng lạ chiếm phổ được quan tâm) và thích ứng cấu hình RAN để tránh nhiễu. Một công trình khác đề xuất một khung dựa trên kết hợp thông tin bối cảnh từ nhiều nguồn để tạo lập các bản đồ để các xAPPs sử dụng cho điều khiển truy nhập phổ động. Một bài báo khác đưa ra một quan điểm chia sẻ phổ được hỗ trợ bởi các bộ điều khiển O-RAN trong các ứng dụng 6G tương lai, trong đó một tần số sóng mang của liên kết tuyến sau được thay đổi bởi một bộ điều khiển tập trung dựa trên thông tin bên ngoài về các tổ chức có thể bị nhiễu từ các liên kết truyền thông.

Các các tiếp cận dược trên chuỗi khối (Blocchain) cũng được xem xét. Một khung chuỗi khối được nhúng vào hạ tầng O-RAN để hỗ trợ đảm bảo an ninh và các trao đổi tin tưởng các tài nguyên RAN (chẳng hạn: các DU, các RU, v.v…) giữa nhiều nhà mạng. Một nghiên cứu kết hợp hạ tầng tế bào thần kinh được O-RAN hỗ trợ và ấn định động dựa trên chuỗi khối như là một các thức thực tế để bắc cầu khoảng trống phân chia số (Digital Divide Gap) . Cuối cùng Cuối cùng, một nghiên cứu được thực hiện nhằm khám phá sử dụng các hợp đồng thông minh để kích hoạt và thôi kích hoạt kết hợp sóng mang trên các nhà mạng khác nhau trên một hạ tầng O-RAN chia sẻ.

MIMO kích thước lớn (Massive MIMO). Công nghệ này là một yếu tố hỗ trợ quan trọng của các mạng 5G. Thông qua một kiến trúc O-RAN, có thể nhúng điều khiển động và khả năng thích ứng vào cấu hình bảng mã MIMO (hay nhóm các búp sóng) hay xử lý chọn búp và làm cho trải nghiệm di động trở nên tin cậy và mạnh mẽ hơn.

Từ góc nhìn xử lý tín hiệu, tồn tại một khối lượng lớn nghiên cứu hưởng lợi từ C-RAN và các CU, các DU tập trung và ảo hóa. Một số tác giả nghiên cứu làm sao có hể cần chia sẻ thông tin trạng thái khả dụng tại RU hay DU trên hai nút. Các tác giả xem xét đặc biệt các khả năng được cung cấp bởi O-RAN FH. Trong bối cảnh này, các sơ đồ nén khác nhau cho giao diện O-RAN FH được thảo luận, đồng thờicũng xét cả các khả năng nhiều luồng điển hình của các thiết lập MIMO. Cuối cùng một số tác giả phân tích các tùy chọn tạo búp khác nhau trên cơ sở phân chia lớp vật lý 7.2x của O-RAN.

Khi nói đến các RIC được O-RAN đưa vào, điều khiển và tối ưu nói chung liên quan đến các thông số búp sóng và đến bảng mã (Codeboook) hay nhóm các búp trong DU và RU. Chẳng hạn, tài liệu kỹ thuật O-RAN về các trường hợp sử dụng nghiên cứu hai giải pháp dựa trên dữ liệu. Giải pháp thứ nhất thích ứng nhóm các búp dựa trên đo từ xa được thu thập tại RIC phi thời gian thực, chẳng hạn: sự hoạt động của người dùng và các báo cáo, các báo cáo đo đạc, các tọa độ GPS và lập lại cấu hình thông qua các giao diện O1 và O-RAN FH. Giải pháp thứ hai là một tối ưu hóa cấu hình di động trên RIC gần thời gian thực, chẳng hạn: các độ lệch búp sẽ quyết định người dùng cần thay đổi búp hay không. Một kịch bản khác là nhóm các người dùng vào các nhóm MIMO đa người dùng hoặc đơn người dùng , điều này sẽ hưởng lợi từ tăng cường dung lượng hay phân tập. Điều này có thể được thực hiên thông qua hướng dẫn chính sách hay làm giầu thông tin từ RIC phi thời gain thực và với tđiều khiển thực tế được thực hiện bởi RC gần thời gian thực. Lập lại cấu hình tạo búp động dựa trên dữ liệu bởi các RIC còn ít được nghiên cứu trong các tài liệu về O-RAN.

An ninh. Ta sẽ đề cập an ninh trong phân 8.

Các ứng dụng mới. Các thế hệ tiếpsau của các mạng không dây cũng nhúng và mở rộng đến các trường hợp sử dụng khác. Một thí dụ là hỗ trợ co truyền thông UAV và xe cộ, ta đã xét nó như là phần của các trường hợp di động và ấn định tài nguyên.

Một thí dụ khác được đại diện bởi các kịch bản IoT đòi hỏi tin cậy cao và định thời, đồng bộ chính xác đạt được bằng cách sao chép dữ liệu (Data Duplication), đa kết nối, QoS dành riêng và các kỹ thuật nén gói. Khi xem xét khối lượng các thông số có thể được điều chỉnh bởi các O-RAN RIC, điều khiển vòng kín được các O-RAN RIC hỗ trợ có thể cung cấp các cấu hình đàn hồi để thích ứng tiến triển các điều kiện tại thực địa.

Rời khỏi các kịch bản truyền thông, các mạng 5G từ phát hành 16 còn hỗ trợ định vị thông qua các báo hiệu dành riêng trên các giao diện vô tuyến (cả LTE lẫn NR) và một chức năng quản lý vị trí trong mạng lõi. Các dịch vụ vị trí cũng sẽ được sử dụng, chẳng hạn trong các kịch bản trong nhà (chẳng hạn: các nhà máy, các trung tâm thương mại) để cung cấp các dịch vụ giá trị gia tăng hay để thực hiện phát quảng bá thông tin an toàn dựa trên vị trí. Tuy nhiên chuyển tiếp thông tin vị trí đến mạng lõi để phân tíchvà xử lý có thể bị ảnh hưởng bởi trễ và jitter khiến cho đáng giá chính xác và kịp thời trở nên khó khăn. Trong trường hợp này, các đặc tả O-RAN xem RIC gần thời gian thực với một xApp định vị dành riêng như là một lựa chọn cho chức năng quản lý vị trí lõi bằng cách triển khai RIC gần thời gian thực tại biên mạng.

Tối ưu hóa các triển khai O-RAN. Ngoài tối ưu hóa RAN thông qua O-RAN, một số bài báo cũng nghiên cứu cách thực tối ưu hóa triển khai bản thân các phần tử O-RAN (chẳng hạn: các RIC, các xApps và các rApps, các nút RAN). Việc này sử dụng các chức năng quan lý của SMO, cũng như quan điểm tập trung của nó và đo từ xa, thông kê từ RAN.

D'Oro và các cộng sự thiết kế một khung mạng không cần khai thác (Zero-Touch Network) để tối ưu hóa vệc sắp đặt trí tuệ mạng trong các mạng được quản lý bởi O-RAN và xây dựng nó ở phạm vi Colossseum sử dụng RIC mã nguồn mở và các phần tử RAN. Trong ngữ cảnh này, bào báo đưa ra khái niệm RLOps, hay các hoạt động học máy tăng cường (Reinforcement Learning Operations), nghĩa là một khung để quản lý vòng đời trí tuệ (cụ thể, học máy cưỡng bức) cho O-RAN. RLOps bao phủ toàn bộ luồng công việc đầu cuối đầu cuối của AI/ML cho RAN (phần 4)., từ thiết kế và quản lý đến các hoạt động (nghĩa là: phát triển , các cập nhật) và quản lý an toàn, an ninh trong xuốt vòng đời trí tuệ. Mặt khác, Huff và các cộng sự phát triển một thư viện, cụ thể là RFT, để làm cho xApps có khả chịu đựng lỗi trong khi vẫn giữ đực khả năng định cỡ cao. Điều này đạt được thông qua các kỹ thuật như: phân vùng trạng thái, sao chép một phần và định tuyến lại nhanh bằng nhận thức vai trò.

Các bài báo khác tập trung lên các thành phần được ảo hóa của các trạm gốc được chia nhỏ. Tamim và các cộng sự cực đại hóa khả dụng cua mạng bằng đề xuất các chiến lược phát triển cho các đơn vị O-RAN được ảo hóa trong đám mây mở (O-Cloud). Pamuklu và các cộng sự đề xuất một kỹ thuật phân chia chức năng cho các O-RAN xanh. Giải pháp được đề xuất, đượcxây dụng trên cơ sở DRL, được đánh giá trên một tập dữ liệu thực tế. Một số tác giả phát triển một sơ đồ phù hợp (Matching Scheme) giữa các DU và các RU bằng một bài toán đóng gói thùng 2D (2D Bin Packing Problem). Cuối cùng, các đặc tả O-RAN xem xét một trường hợp sử dụng tương tự bằng gộp nhóm dựa trên số liệu các

CU và các DU trên các tài nguyên được chia xẻ, đươc điều phối thông quan RCI phi thời gian thực.

12.8. AN NINH TRONG O-RAN

Không thể phủ nhận rằng việc đưa vào sử dụng các thành phần kiến trúc, các giao diện mở, chia nhỏ và tích hợp các logic tùy biến và có thể điều khiển dựa trên dữ liệu sẽ khiến các mạng tổ ong thế hệ sau hiệu quả và linh hoạt hơn. Mặt khác, tuy nhiên, cuộc cách mạng này đi kèm với các thác thức an ninh chưa từng có trước hết bắt nuồn từ thực tế là tính chất phân tán và chia nhỏ của hạ tầng O-RAN sẽ hiệu quả mở rộng bề măt tấn công cho những người dùng ác ý, gây ra những đe dọa nghiêm trọng đối với mạng và các người dùng của nó, Chẳng hạn, một số tác giả nhấn mạnh rằng hỗ trợ một phần hay hạn chế cho xác thực (Authentication) có thể dẫn đến các lỗ ổng trong các giao diện O-RAN. Đồng thời, các khả năng giám sát vô đối, trí tuệ và triển khai điện toán đám mây gốc quyết định các kiến trúc O-RAN thực sự bổ sung cái nhìn sâu sắc về trạng thái mạng và cung cấp các công cụ cần thiết để thực hiện các giải pháp tiên tiến để giám sát, phát hiện, ngăn chặn và chống lại các đe dọa. Liên quan đến vấn này, liên minh O-RAN đã thành lập một nhóm công tác riêng để phân tích và định nghĩa mô hình đe dọa (Threat Model) cho các mạng O-RAN, cũng như cho định nghĩa các biện pháp và các chính sách an ninh cho các thành phần O-RAN hướng đến một mô hình không bao giờ tin tưởng (Zero-Trust Model).

12.8.1. Các bên liên quan

Nhóm tập trung an ninh (SFG: Security Focus Group) (hiện nay được nâng lên thành một nhóm đây đủ: WG11) đã định nghĩa một danh sách các bên liên quan cần chủ động đảm bảo an ninh RAN. Điều này mở rộng thêm các bên được quan tâm bên ngoài các mạng 4G và 5G, chẳng hạn: các nhà cung cấp, các nhà mạng và các người tích hợp hệ thống. Các nhà mạng sẽ đảm nhận một vai trò chủ yếu đảm bảo an ninh hạ tầng, vì sự mở của nền tảng và việc sử dụng các thành phần của nhiều nhà cung cấp cho phép họ điều chỉnh sự xây dựng (và do đó an ninh) hạ tầng. Điều này cũng có nghĩa là các nhà mạng có thể truy nhập và kiểm tra mức độ an ninh của các thành phần mở được đưa vào mạng, điều thường không thể trong các kiến trúc đóng kín hoàn toàn dựa trên các nhà cung cấp. Tài liệu công bố cũng định dạng các nhà cung cấp các chức năng mạng và nền tảng ảo như là các bên liên quan mới (chẳng hạn: các người phát triển xAppx và rApps, các nhà cung cấp O-Cloud bên thứ ba), cùng với các hồ sơ quản trị cho quản lý các thành phần được

ảo hóa và được phân nhỏ. Ngoài ra người điều phối (chẳng hạn: thực thể quản lý SMO) cũng có một vai trò trong đảm bảo an ninh các hoạt động của mạng.

12.8.2. Bề mặt đe dọa được mở rộng

Theo các phân tích về mối đe dọa trong một số nghiên cứu, mềm hóa và phân nhỏ sẽ không tránh khỏi mở rộng bề mặt đe dọa của các hệ thống tổ ong. Cho đến nay, WG1 đã định nghĩa bảy thể loại đe dọa bao gồm 84 các mối đe dọa có thể:

- **Các mối đe dọa đối với hệ thống O-RAN**. KIến trúc O-RAN đưa vào sử dụng các phần tử kiến trúc và các giao diện mới (từ tuyến trước để quản lý và điều khiển các giao diện), chúng trở thành một phần của một bề mặt đe dọa. Các thành phần này có thể bị các tấn công khác nhau gây tổn hại (i) sự khả dụng của hạ tầng (chẳng hạn: truy nhập không được phép đến các phần tử RAN được chia nhỏ nhằm giảm cấp hiệu suất mạng, hay triển khai độc hại các xApps với ý đồ gây ra các xung đột với các xApps khác), (ii) toàn vẹn của dữ liệu và hạ tầng (chẳng hạn phần mềm bi xâm phạm tin tưởng một chuỗi các cấu hình giao diện sai) và (iii) bảo mật dữ liệu (chẳng hạn: thông qua các tấn công làm mất khả năng mật mã hóa trên giao diện vô tuyến hay tạo điều khiện cho tuy nhập thiếu kiểm soat dữ liệu người dùng từ xApps và rApps). Để là thí dụ, một kẻ tấn công có thể khai thác thiết kế không đảm bảo an ninh của các thành phần RAN (chẳng hạn: các giao diện, các chức năng, v.v…), sự phức tạp của chúng hay cấu hình sai của chúng để đạt được truy nhập không được cấp phép đến hệ thống. Hay, các cơ chế điều khiển cấp phép/truy nhập yếu cho phép kẻ tấn công thâm nhập các dịch vụ để lộ thông qua các máy chủ web.
  Các mối đe dọa liên quan đến dữ liệu bao gồm: (i) thông tin được vận chuyển trên các giao diện O-RAN cho quản lý, điều khiển và lập cấu hình cho RAN, (ii) dữ liệu được sử dụng cho đào tạo các mô hình ML, (iii) dữ liệu nhạy cảm về người dùng, chẳng hạn: các mã nhận dạng của họ, (iii) các khóa mật mã được triển khai trên các phần tử mạng. Bề mặt đe dọa cũng mở rộng đến các thành phần logic mới của kiến trúc, nghĩa là: các RIC, SMO và các khung phần mềm (chẳng hạn: các xApps, các rApps) thực hiện trên các API của RIC gần thời gian thực. Tại RIC gần thời gian thực và các xApps các tấn công bao gồm các xApps độc hại sử dụng các RIC API không được xác thực (Unauthenticated) và cho phép các kẻ tấn công yêu cầu dịch vụ (“ăn cắp các dịch vụ”) hay thông tin nhạy cảm ((“dò rỉ dữ liệu”), theo dõi các UE và thay đổi ưu tiên của họ, truy nhập các tài nguyên. Các tấn công cũng có thể làm tổn hại cách ly giữa các xApps hay gửi đến chúng các chính

sách giả gây giảm cấp hiệu suất của chúng. Liên quan đến RIC phi thời gian thực và các rApps, một kẻ tấn công có thể gây ra các từ chối dịch vụ hay giảm hiệu năng, theo dõi người dùng, hỏng và thay đổi dữ liệu. Cuối cùng, các tấn công lên giao diện phân chia 7.2x của tuyến trước mở cũng thường xuyên được nhắc đến như là một điểm yếu tềm năng.

Cho đến nay,giao diện 7.2x không được mật mã hóa trên mặt phẳng điều khiển, vì các yêu cấu định thời đầy thách thức do mật mã hóa đưa ra. Điều này đưa vào các tấn công của kẻ xen giữa Man-in-the-Midle Attack), trong đó kẻ tấn công đóng giả DU (hay RU), gây tổn thương dữ liệu người dùng hay cấu hình tại cả hai đầu cuối, hay đạt được truy nhập đến DU và xa hơn thông qua tấn công lên RU hay giao diện tuyến trước, hoặc làm tổn thương khả dụng/bảo mật của giao diện này. Một tấn công khác có thể được thực hiên chống lại mặt phẳng S, bằng một tác nhân độc hại (Malicious Actor) gây tổn thương hạ tầng đồng bộ do đó gây giảm cấp hiệu năng, chẳng hạn bằng cách giả mạo đồng hồ chủ , loại bỏ các gói PTP (Precision Time Protocol) hay thông qua một thực thể lừa đảo (Rogue Instance) muốn trở thành đồng hồ chủ lớn PTP (PTP Grand master).

- **Các mối đe dọa đối với O-Cloud.** O-Cloud cung cấp một môi trường ảo hóa chứa các phần tử RAN cũng như các thành phần O-RA.N Vì mục đích này, các de dọa được nhận diện cho O-Cloud liên quan đến các tấn công trong các môi trường được ảo hóa. Các tấn công có thể có bao gồm: (i) làm tổn thương các chức năng mạng ảo, hoặc là đang thực hiện hoặc là các bản ghi (Snapshots) hoặc là các hình ảnh (Images) trạng thái máy ảo (chẳng hạn làm dò rỉ các bí mật mật mã được nhúng), (ii) khai thác giao diện O2 giữa O-Cloud và SMO để đạt được truy nhập hoặc trốn thoát cách ly, (iii) lạm dụng các con-te-nơ hay các máy ảo cho các chức năng mạng để tấn công các thực thể khác trong mạng, (IV) đánh lừa hay làm tổn thương các dịch vụ nối mạng các dịch vụ hay phụ trợ nền tảng và (V) lạm dụng các máy ảo (các VM) hoặc các con-ten- nơ đặc quyền hoặc trốn thoat khỏi các môi trường cách ly, ăn cắp dữ liệu, thay đổi các tài nguyên được ấn dịnh cho các VM hoặc các con-ten-nơ, làm xáo trộn hình ảnh và làm lộ bí mật trong các môi trường được ảo hóa.
- **Các mối đe dọa đối với mã nguồn mở.** Mềm hóa các phần tử RAN và O-RAN mở ra các lỗ hổng do các cửa hậu trong mã O-RAN bởi: (i) các nhà phát triển được tin cậy có ý đồ làm tổn thương các phần tử phần mềm hay bởi (ii) các thư viện đầu nguồn (Upstream Library) không nằm dưới sự điều khiển của các nhà phát triển O-RAN.
- **Các mối đe dọa vật lý**.Phần cứng bổ sung được đưa vào để hỗ trợ phân chia gNB và hạ tầng O-RAN có thể bị tổn thương bởi các kẻ tấn công đạt dược truy nhập vật

lý vào hạ tầng. Các tấn công có thể trải dài từ các tấn công khả dụng nguồn điện đến lập lại cấu hình đi cáp hay bổ sung thêm các cửa hậu phần cứng để làm hỏng dữ liệu nhạy cảm.

- **Các mối đe dọa đối với các chức năng vô tuyến**. Tấn công lên cảU hay gioa diện tuyến trước mở giữa cá RU và các DU có thể dẫn đến giảm cấp hiệu suất trên giao diện vô tuyến, bở các tấn công thông thường liên quan đến làm nghẽn dữ liệu hoặc các báo hiệu đồng bộ hoặc các từ chối dịch vụ. NGoài ra, kết hợp các thiết bị từ các nhà cung cáp khác nhau cú thể giảm hiệu suất mạng trong trường hợp không phù hợp các cấu hình hay các chức năng được hỗ trợ.
- **Các mối đe dọa đối với ngăn xếp giao thức**. Các tấn công cũng có thể được thực hiện hướng đế các lớp của ngăn xếp giao thức. Các tấn công liên quan đến tiêm mã độc (Injection), kịch bản chéo trang (Cross-Site Scripting; chèn các kịch bản (mã độc) vào các trang web đáng tin cậy), từ chối dịch vụ, để lộ các mã nhận dạng của các đối tượng, không có thẩm quyền thích hợp và các mã thông báo (Web Token) thông qua các REST API, các khai thác JSON hay HTTP (JSON/HTTP Exploit: đoạn mã được thiết kế để tấn công lỗ hổng khi xử lý JSON hoặc HTTP) có thể được thực hiện.
- **Các mối đe dọa đối với các thành phần AI/ML**. Cuối cùng đặc tả O-RAN và các tài liệu công bố cũng mô tả một lớp đe dọa mới, nghĩa là các tân công đối với các mô hình AI/ML được sử cho suy luận và điều khiển trong các xApps và rApps. Một ví dụ thực tế của một tấn công như thế là một tấn công định vị, trong đó một kẻ tấn công khai thác truy nhập không được quản lý đến dữ đực liệu được lưu trữ trong SMO/RIC phi thời gian thựcđể tiêm mã độc làm thay đổi và sai lệch dữ liệu trong các tập dữ liệu được sử dụng cho đào tạo AI/ML ngoại tuyến. Một thí dụ khác là điều khiển không hạn chế đối nghịch trên một hay nhiều nút để tạo ra dữ liệu tổng hợp (Synthetic Data) theo thời gian thực cho các giải pháp AI/ML được vi chỉnh trực tuyến hay được thực hiện để suy luận trực tuyến. Các tấn công này cực kỳ liên quan, vì chúng có thể cho ra các dự đoán sai hay đưa ra các quyết định điều khiển sai dẫn đến giảm hiệu suất hay thậm chí xấu hơn: ngừng dịch vụ. Tương tự các tấn công có tiếp trực tiếp nhắm đến mô hình ML (chẳng hạn: thay đổi các trong số hay các cấu hình của mô hình).

12.9. KẾT LUẬN

Chương này trình bầy tổng quan toàn diện các đặc tả, các kiến trúc và các hoạt động của O-RAN. Đầu tiên, các khối kiến trúc cơ sở và các nguyên lý cơ bản thiết kế các mạng O-RAN được trình bày. Sau đó các thành phần của RT gần thời gian thực, RC phi thời gian thực, SMO và các giao diện O-RAN như D2, O1 A1, FH và O2 được mô tả.Các

phần tieeps sau tập trung lên các vấn đề mở rộng các thành phần, các giao diện trong kiến trúc O-RAN. Các phần này này trình bày các chi tiết về luồng làm việc AI/ML, các trường hợp sử dụng của ORAN và các nghiên cứu, về các thánh thức và tiềm năng an ninh.

Chương 13

CÁC MẠNG TRUY NHẬP VÔ TUYẾN DỰA TRÊN AI/ML

13.1. TỔNG QUAN

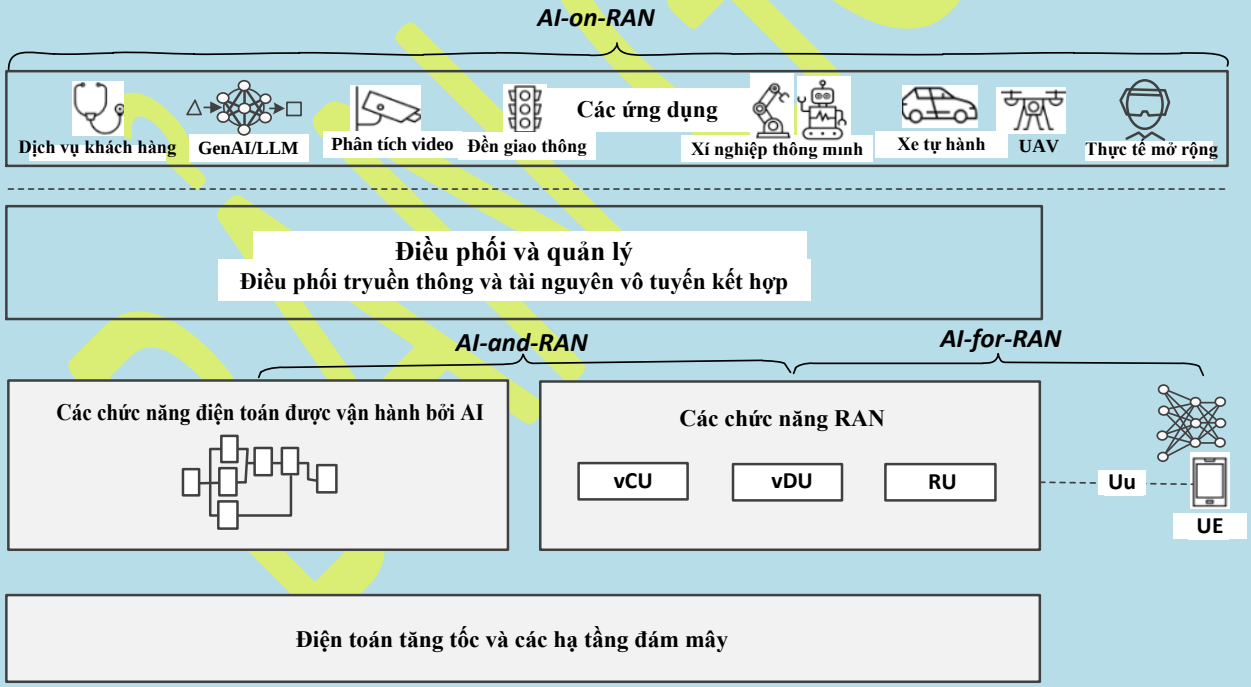
AI-RAN (Artificial Intelligence Radio Access Network: mạng truy nhập vô tuyến trí tuệ nhân tạo) là một công nghệ cho phép tích hợp hoàn toàn AI vào phần cứng và phần mềm để hỗ trợ các dịch vụ AI mới, các cơ hội kiếm tiền cùng với các lợi ích mang tính chuyển đổi trong việc sử dụng mạng, hiệu suất phổ tần và hiệu suất.

Cấu trúc cơ bản của AI-RAN được xây dựng dựa trên mục đích chung hoàn toàn thuần nhất, tăng tốc nền tảng tính toán mà không cần bất cứ phần cứng đặc thù nào của RAN, sao cho nó có thể chạy đồng thời khối lượng công việc tổ ong và AI với hiệu suất xác định cho từng khối lượng công việc. Nó chứa đựng các nguyên lý điện toán đám mây gốc (Cloud-Native) như định cỡ theo yêu cầu (On Demand Scaling), kiến trúc đa người thuê (Multi-Tenancy: nhiều người thuê, là một kiến trúc cho phép một phiên bản ứng dụng duy nhất phục vụ nhiều khách hàng với mỗi khách hàng có dữ liệu riêng) và công-ten-nơ hóa cả hai khối lượng công việc.

Phần mềm của AI-RAN được xây dựng dựa trên các nguyên tắc hoàn toàn được định nghĩa bằng phần mềm (Sofware-Defined) để cho phép con-ten-nơ hóa và tăng tốc các khối lượng công việc của AI và RAN nhằm đảm bảo các lợi ích của nền tảng tính toán được tăng tốc.

Với nền tảng phần cứng-phần mềm tăng tốc này, AI-RAN cho phép triển khai các khối lượng công việc của 5G/6G RAN và AI trên hạ tầng đám mây tăng tốc (Accelerated Cloud Infrastructure) phân tán và chia sẻ. Nó chuyển đổi hạ tầng RAN từ hạ tầng đơn mục đích vào đa mục đích.

Trong một hạ tầng tính toán-truyền thông thống nhất của mạng viễn thông, tích hợp các khối lượng công việc của AI và RAN được thể hiện trong ba lớp: mức giao thức/kiến trúc, mức dịch vụ và mức hạ tầng như được thể hiện trong hình 13.1. Trong khi các tổ chức tiêu chuẩn hóa toàn cầu như 3GPP và O-RAN căn bản tập trung lên hòa nhập AI và RAN mức giao thức/kiến trúc, thì các sáng kiến mới trong lĩnh vực này như liên minh AI-RAN (AI-RAN Alliance) và liên minh AI viễn thông toàn cầu (GTAA:Global Telecomm AI Alliance) lại đang khám phá các lợi nhuân tiềm năng trong việc liên kết AI và RAN để mở ra các cơ hội kinh doanh mới tại mức dịch vụ và hạ tầng. Các hoạt động đang diễn ra trong các tổ chức phát triển tiêu chuẩn khác nhau (SDO: Standard Development Organization) và các liên minh đang đang tiến đến khai thác sức mạng của AI trong việc chuyển đổi RAN. Tích hợp tính toán và truyền thông vào một hệ thống chung, các sáng kiến này tập trung lên ba lĩnh vực quan trọng: (i) cải thiện hiệu suất phổ và hoạt động tại mức giao thức/kiến trúc (AI for RAN: AI cho RAN), (ii) cho phép các ứng dụng AI mới được tổ chức bởi RAN (AI-on RAN: AI trên RAN) và (iii) hỗ trợ sử dụng tài sản cao hơn bằng cách chia sẻ các tài nguyên tính toán giữa các khối lượng công việc AI và RAN tại mức hạ tầng.



AI-on-RAN: AI trên RAN, AI-and-RAN: AI và RAN, AI-for-RAN: AI cho RAN, vCU (virtual Central Unit): đơn vị trung tâm ảo, vDU (virtual Distributed Unit) đơn vị phân tán ảo, GenAI (Generative AI): AI tạo sinh, LLM (Large Language Model): mô hình ngôn ngữ lớn.

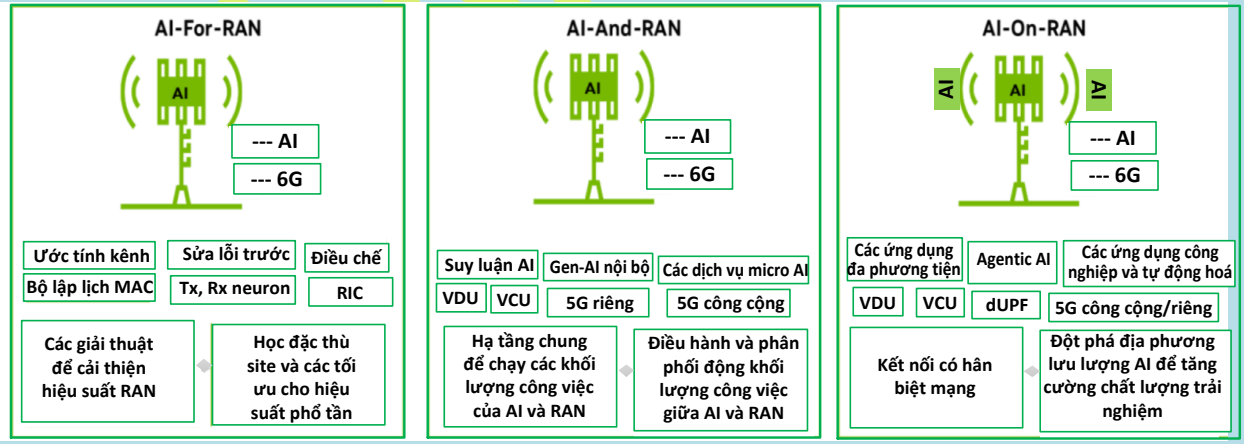
Hình 13.2. Tổng quan mức cao của AI-RAN

AI-RAN Alliance (Liên minh AI-RAN) vạch ra ba lĩnh vực đặc thù để tích hợp AI vào RAN:

- AI và RAN (AI and RAN), (còn được đề cập là AI với RAN): sử dụng một hạ tầng chung chia sẻ để chạy cả AI và RAN nhằm cực đại hóa sử dụng, giảm tổng chi phí quyền sở hữu (TCO: Total Cost of Onership) và tạo ra các cơ hội lợi nhuận mới bởi vận hành AI.
- AI cho RAN (AI for RAN): Nâng cao các khả năng của RAN thông qua nhúng các các giải thuật, các mô hình AI/ML và các mạng tế bào thần kinh vào lớp xử lý tín hiệu vô tuyến để cải thiện hiệu suất phổ tần, vùng phủ vô tuyến, dung lượng và hiệu suất.
- AI trên RAN (AI on RAN): cho phép các dịch vụ AI trên RAN tại biên mạng tăng hiệu năng khai thác và cung cấp các dịch vụ mới cho các người dùng di động. Đến lượt mình RAN từ một trung tâm chi phí trở thành một nguồn lợi nhuận.

AI-RAN mở rộng thêm RAN mở (Open-RAN) bằng cách sử dụng kiến trúc nền tảng mục đích chung hoàn toàn được định nghĩa bằng phần mềm cho phép các giao diện mở để đem đến tính linh hoạt, tương tác và hiệu quả chi phí cho RAN.

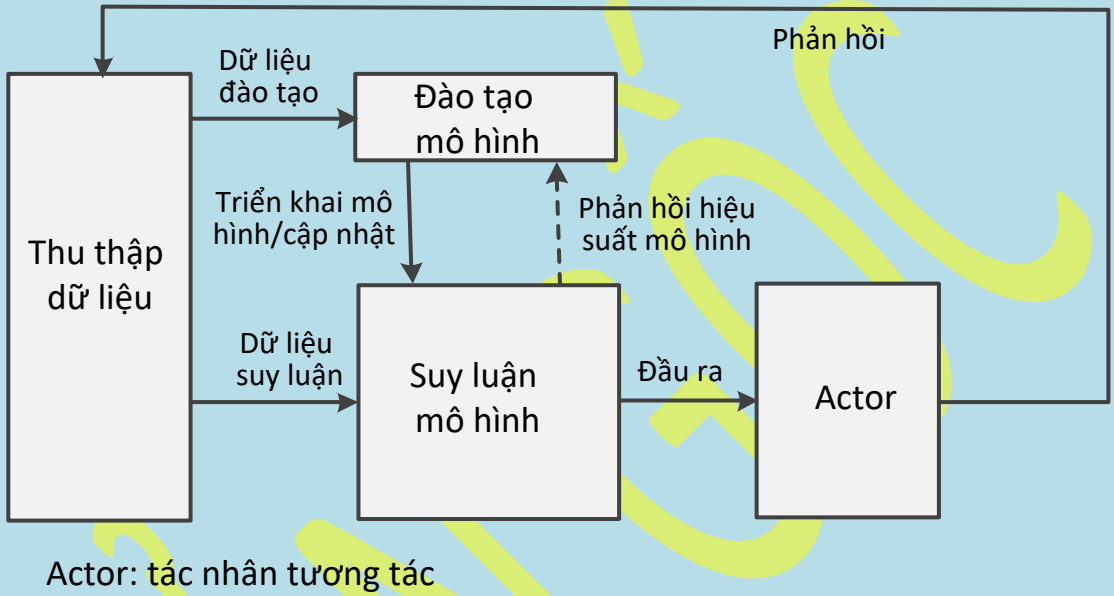
Hình 13.2 cho thấy các lĩnh vực ứng dụng của AI-RAN.



MAC (Medium Access Control); điều khiển truy nhập môi trường; RIC (RAN Intellegence Controller): bộ điều khiển trí tuệ RAN; VCU (Virtual Central Uni)t: đơn vị trung tâm ảo; VDU (Virtual Distributed Unit): đơn vị phân tán ảo; dUPF (Differentiated User Plane Function): chức năng mặt phẳng người dùng có phân biệt.

Hình 13.2. Các lĩnh vực ứng dụng của AI-RAN

Một khung cấu trúc cho trí tuệ RAN đã được mô tả trong 3GPP TR 37.817 , như thể hiện trong Hình 13.3 Chức năng Thu thập Dữ liệu cung cấp dữ liệu đầu vào cho các chức năng Huấn luyện Mô hình và Suy luận Mô hình. Ví dụ về dữ liệu đầu vào có thể bao gồm các phép đo từ UE hoặc các thực thể mạng khác nhau, phản hồi từ Actor, đầu ra từ mô hình AI/ML. Việc đạt được khả năng thu thập và quản lý dữ liệu hiệu quả và hợp lệ là chìa khóa để đảm bảo hiệu suất tốt của quá trình huấn luyện và suy luận AI/ML trong cả 5G và 6G.



Hình 13.3. Khung cấu trúc cho trí tuệ RAN

13.2. KIẾN TRÚC THAM CHIẾU

13.2.1. Các yêu cầu và các yếu tố của AI-RAN

Thiết kế AI-RAN đòi hỏi xem xét kỹ lưỡng cả hai chức năng tính toán và truyền thông. Mục đích này là để xây dựng một nền tảng RAN trí tuệ, hiệu suất và khả định cỡ cao cho phép kết hợp các khả năng tính toán tiên tiến với các chức năng truyền thông tối ưu. Dưới đây chúng ta sẽ nhận diện các yêu cầu và công nghệ quan trọng là các yếu tố cho thực hiện AI-RAN được minh họa trong hình N1.

Hạ tầng tính toán tăng tốc. Hạ tầng tính toán của RAN truyền thông thường sử dụng các vi mạch đặc thù ứng ụng được sản suất theo yêu cầu khách hàng (ASIC: Application Specific Integrated Circuit) cho xử băng gốc và đơn vị xử lý trung tâm cho xử lý lớp cao hơn. Hạ tầng như vậy không thể xử lý các nhiệm vụ phức tạp liên quan đến đa dạng các tường hợp sử dụng của AI-RAN. AI-RAN đòi hỏi các bộ tăng tốc phần cứng được định nghĩa bằng phần mềm đa mục đích, chẳng các GPU (Graphic Processing Unit: đơn vị xử

lý đồ họa), cho xử lý các khối lượng công vệc lớn không chỉ bao gồm tính toán mà cả đào tạo và suy luận mô hình AI. Hạ tầng cần được tối ưu hóa cho cả đào tạo mô hình và triển khai suy luận tại biên. Đào tạo ngoại tuyến có thể xẩy ra tại đám mây, còn suy luận cần thực hiện theo thời gian thực tai các nút biên với các khả năng tính toán tích hợp.

Ngoài ra, hạ tầng tính toán tăng tốc với xử lý song song quy mô lớn cũng cần thiết để xử lý khối lượng lớn số liệu được tạo ra trong AI-RAN và các dịch vụ AI. Khả năng này tăng tốc cái giải thuật AI cũng như xử lý tín hiệu băng gốc cho phép tính toán nhanh vơn đối với tạo búp của MIMO kích thước lớn (mMIMO: massive Multi-Input Multi-Output), lập lịch đa ô, cảm biến và truyền thông tích hợp (ISAC: Integrated Sensing and Communications) . Hạ tầng này cũng phải được trang bị các kết nối tốc độ cao trễ thấp để đảm bảo truyền thông nhanh và liền mạch trên tất cả các nút mạng.

Thiết kế điện toán đám mây gốc được, định nghĩa bằng phần mềm. Sự phát triển RAN thông thường từ thế hệ này sang thế hệ khác khởi động làm mới phần cứng hạ tầng RAN. Cách tiếp cận phát triển RAN này không phù hợp, đòi hỏi thiết kế đám mây điện tử gốc, được định nghĩa bằng phần mềm trong các mô hình Ai-RAN mới. Trong AI-RAN được định nghĩa bằng phần mềm, các phần tử chức năng (NF: Network Function) được trừu tượng hóa từ phần cứng cơ sở để cho phép linh hoạt, di động và khả lập trình đễ hơn. Đặc biệt, các NF trong AI-RAN, như là xử lý băng gốc, lập lịch và quản lý tài nguyên vô tuyến (RRM: Radio Resource Management), phải được thực hiện trong phần mềm để cho phép khởi tạo năng động các chức năng RAN như là các chức năng mạng đám mây điện tử gốc (CNF: Cloud-Native Network Function). Điều này cho phép các cập nhật, các lập lại cấu hình đễ ràng và khả năng đưa ra các tính năng và các dịch vụ mới (chẳng hạn: các ứng dụng AI biên mới nổi) mà không cần thay đổi phần cứng.

AI-RAN phải hỗ trợ định cỡ đàn hồi cả hai các tài nguyên tính toán và truyền thông, bằng cách thêm vào hoặc bớt đi các CNF theo yêu cầu. Thiết kế đám mây gốc là cực kỳ quan trọng để tạo điều kiện cho khả định cỡ, linh hoạt và hiệu suất.Các CNF có thể được phân nhỏ thành các vi dịch vụ (Microservice). Trong đó mỗi kiểu dịch vụ có thể xử lý một chức năng đặc thù và phù hợp với thay đổi khối lượng công việc. AI-RAN đám mây gốc sử dụng các con-ten-nơ (chẳng hạn: DOCKER, Kubernetes) để đóng gói và triển khai các CNF như là các vi dịch vụ để cho phép sử dụng hiệu quả tài nguyên trong kiến trúc đa người thuê (Multi-Tenancy), chẳng hạn: chạy nhều RAN và nhiều dịch vụ AI trên một hạ tầng chia sẻ). Tích hợp liên tục và triển khai liên tục (CI/CD: Continious Integration/ Continious Deployment) RAN và các dịch vụ AI sẽ tạo điều kiện triển khai nhanh các tính năng mới và nâng cấp với gián đoạn dịch vụ tối thiểu.

Điều phối kết hợp các tài nguyên truyền thông và tính toán. Bản chất đa người thuê cố hữu của AI-RAN cần có điều phối kết hợp các tài nguyên truyền thông và tính toán dẫn đến phải có một hạ tầng điều phối tông nhất. Điều này hỗ trợ ấn định theo yêu cầu thời gian thực các tài nguyên dựa trên lưu lượng mạng hiện thời, yêu cầu người dùng và các thỏa thuận mức dịch vụ (SLA: Service Level Agreement). Bộ điều phối phải quản lý cả hai tài nguyên tính toán (chẳng hạn: GPU, CPU, bộ nhớ) và truyền thông (chẳng hạn: băng thông, thời gian) theo các yêu cầu các các khối lượng công việc khác nhau cụ thể. Nó cần quyết định định nơi nào một dịch vụ hay khối lượng công việc cụ thể cần thực hiện. Chẳng hạn, bộ điều phối có thể ấn định tài các nguyên biên cho các khối lượng công việc nhạy cảm trễ, còn các nhiệm vụ định định tuyến nặng tính toán, phi thời gian thực cho các đám mây trung tâm. Các mô hình AI phải được tích hợp vào nền tảng điều phối để đưa ra các quyết định thông minh liên quan đến phân phối tối ưu các tài nguyên tính toán và truyền thông cho các nhiệm vụ như tối ưu hóa công suất và quản lý tải.

Tính đàn hồi là vô cùng quan trọng cho xử lý các thay đổi các điều kiện mạng và các yêu cầu tính toán. Bộ điều phối phải định cỡ các tài nguyên tăng hoặc giảm để xử lý hiệu quả lưu lượng thay đổi và các yêu cầu tính toán, trong cả giờ cao điểm và các giai đoạn yêu cầu thấp. Cần khai thác các khả năng của phần cứng cơ sở để đạt được định cỡ động. Chẳng hạn kỹ thuật GPU nhiều thực thể (MIG: Multi-Instance GPU) có thể được sử dụng phân đoạn một GPU vào một số thực thể được phân chia cứng và được cách ly bộ nhớ/lỗi, cho phép bộ điều phối có thể hỗ trợ xử lý đồng thời các khối lượng công việc của RAN và các ứng dụng AI biên trên các tài nguyen GPU chung.

Hỗ trợ AI gốc (Native AI). Mô hình AI RAN mới nổi phải hỗ trợ AI gốc bằng cách nhúng trực tiếp các khả năng AI vào hạ tầng RAN để cho phép đưa ra quyết định thông minh thời gian thực, tự động hóa và tối ưu hóa trên cả các lớp mạng và các lớp điện toán. Từ viễn cảnh thiết kế RAN, AI phải được tích hợp vào tất cả các lớp mạng ngay từ khi bắt đầu và trải dài tư từ xử lý lớp vật lý đến các phần tử mạng lớp cao hơn, như: quản lý tài nguyên vô tuyến (RRM), quản lý di động, cân bằng tải và lập lịch. Các mô hình AI phải có khả năng điều phối các quyết định tên các lớp khác nhau của mạng để đạt được tối ưu hóa mạng toàn diện, đầu cuối đầu cuối, trong đó quyết định tai một lớp có xét đến ảnh hưởng của các lớp khác. Để các tối ưu hóa và các quyết định dựa trên AI chính xác, hệ thống AI-RAN phải có các ống dữ liệu bền vững để thu tập và xử lý dữ liệu từ các phần tử, các nút và các lớp khác nhau của mạng.

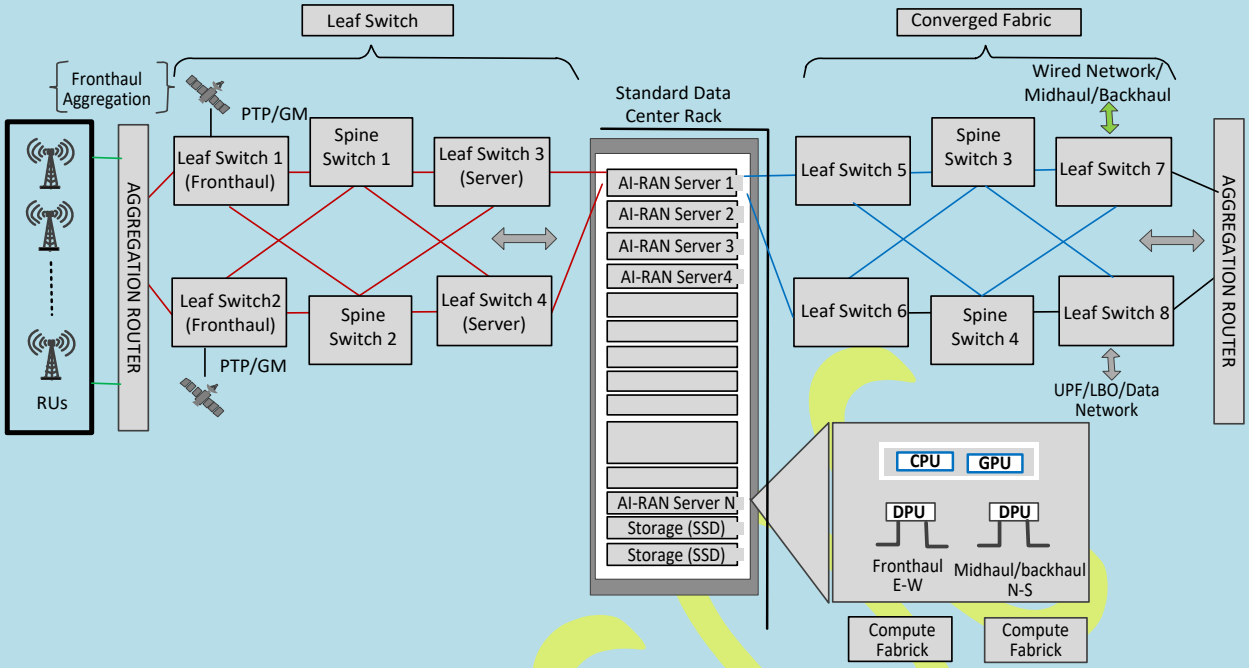
Hỗ trợ AI gốc trong AI-RAN cần phải sử dụng song sinh số mạng hay bản sao số mạng (Network Digital Twin) , là một bản sao ảo của mạng vật lý cho thu tập dữ liệu, mô phỏng, và tối ưu hóa. NDT cung cấp một hộp cát (Sandbox: một môi trường nhỏ biệt lập) để đào tạo các mô hình mà không gây rủi ro gián đoạn các hoạt động trực tiếp của mạng

AI-RAN. Sau khi được đào tạo, các mô hình AI có thể được thẩm định (Validated) trong mi trường NDT để kiểm tra hành vi của chúng trong các điều kiện khác nhau. Điều này đảm bảo các mô hình hoạt động như được kỳ vọng và không gây ra các vấn đề ngoài ý muốn như thời gian chết (Downtime) khi được triển khai trong mạng AI-RAN trực tiếp.

Kết hợp tất cả các đặc tính và các công nghệ quan trong này với nhay sẽ cho phép thành lập kiến trúc AI-RAN, mà dựa trên nó các nhà mạng viễn thông có thể xây dựng các triển khai riêng, tùy chỉnh riêng của họ được tăng cường trên hạ tầng RAN của họ. TRong phần tiếp theo ta sẽ xét chi tiết một phác thảo kiến trúc AI-RAN kết hợp các yếu tố công nghệ cơ bản.

13.2.2. Thiết kế kiến trúc tham chiếu

Kiến trúc tham chiếu của AI-RAN có thể được xây dựng trên các nguyên tắc cơ bản: hiệu suất cao, khả định cỡ và mo-đun trong hội nhập tính toán AI và RAN. Để hướng dẫn triển khai, một bản vẽ thiết kế bao gồm giá máy trung tâm dữ liệu tiêu chuẩn với tất cả các AI-RAN server (với các CPU, các GPU, các đơn vị xử lý dữ liệu (DPU), các SSD (Solid-State Drive: ổ cứng thể rắn) và các chuyển mạch ethernet dựa trên cấu trúc mạng (Networking Fabric: hệ thống kết nối các phần tử mạng) như được mô tả trong hình 13.4. Sơ đồ này cung cấp cho các nhà mạng viễn thông một kiến trúc tham chiếu để triển khai trung tâm dữ liệu tăng tốc và được định nghĩa bằng phần mềm thế hệ sau cho AI-RAN để giải quyết các nhu cầu tính toán tính toán của các khối lượng công việc AI và RAN đồng thời. Tính toán tăng tốc đám mây điện toán gốc (Cloud Native) là lõi của kiến trúc tham chiếu này, cho phép triển khai các hệ thống AI-RAN nhanh với các mức độ định cỡ và các các yêu cầu tính toán thay đổi theo lưu lượng RAN và khối lượng công việc AI đến từ các trung tâm dữ liệu biên của các nhà mạng viễn thông. Thiết kế triển khai AI-RAN từ đầu đến cuối này bao gồm các phần tử quan trọng như các đơn vị vô tuyến (RUs), mạng tuyến trước (Fronthaul Network), các đơn vị phân tán (DUs) và trong một số trường hợp các đơn vị trung tâm (CUs) và cả các mạng lõi (CNs) chạy trên các AI-RAN server, kết nối toàn bộ đến internet. Lưu ý, để đơn giản, hình 13.4 mô tả cấu hình topo mạng tuyến trước kết nối nhiều RU đến một AI-RAN server (nghĩa là ánh xạ nhiều đến một), trong khi đó trong các triển khai thực tế các kết nối gữa các RU đến các AI-RAN server sẽ là nhiều đến nhiều.



Fronthaul Aggregation: kết hợp tuyến trước, Leaf Switch: chuyển mạch lá, Spine Switch: chuyển mạch sương sống, Aggregation Router: router kết hợp, PTP GM (Precision Time Protocol GrandMaster): đồng hồ chủ lớn giao thức thời gian chính xác, Standard Data Center Rack: giá máy trung tâm dữ liệu tiêu chuẩn, AI-RAN Server: máy chủ AI-RAN, Storage: bộ lưu trữ, SSD (Solid-State Drive) : ổ cứng thể rắn, CPU (Central Processing Unit): đơn vị xử lý trung tâm, GPU (Graphics Processing Unit): bộ xử lý đồ họa, DPU (Data Processing Unit): đơn vị xử lý dữ liệu, Fronthaul E-W: tuyến trước Đông-Tây, Midhaul/Backhaul N-S (tuyến giữa, tuyến sau Bắc-Nam), Converged Fabric: mạng hội tụ, Compute Fabric: mảng tính toán, UPF/LBO/Data Network (User Plane Function/Local Breakout/Data Network: chức năng mặt phẳng người dùng/đột phá tại chỗ/ mạng dữ liệu, Wired Network/Midhaul/Backhaul: mạng hữu tuyến/tuyến giữa/tuyến sau.

Hình 13.4. Sơ đồ kiến trúc tham chiếu AI-RAN với giá máy server và cấu hình topo mạng

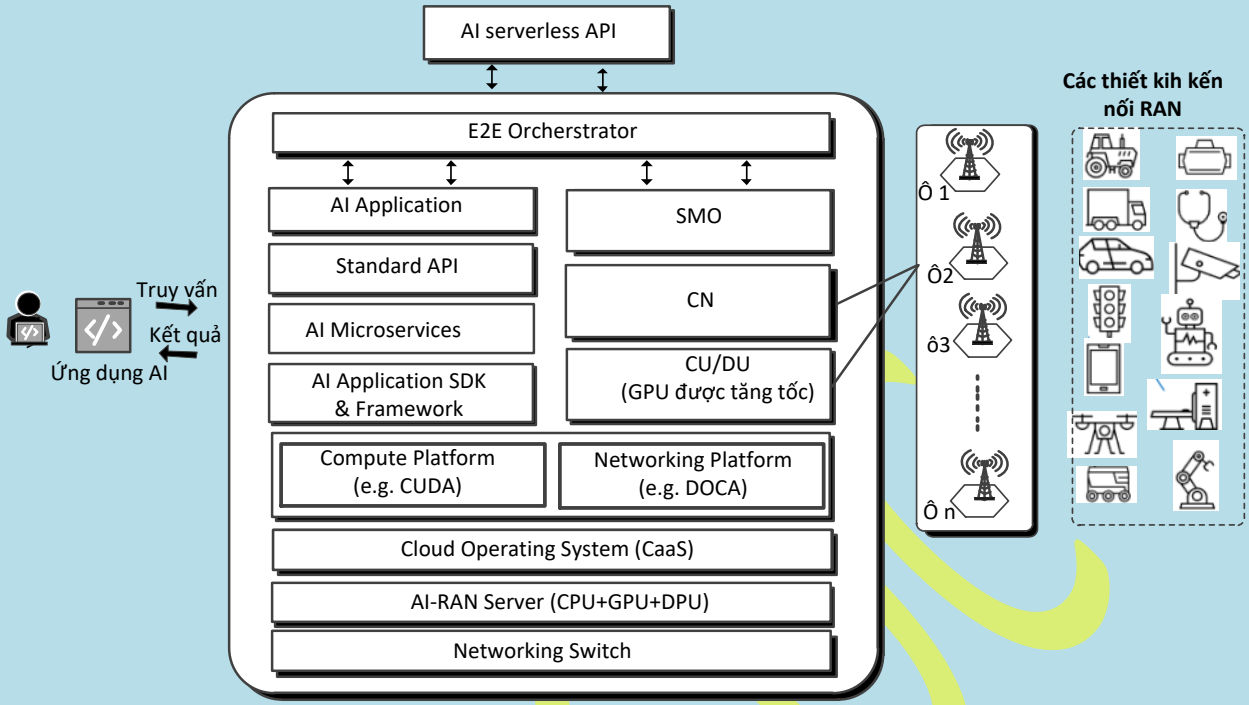
Để luồng lưu lượng AI và RAN liền mạch thông qua cùng một hạ tầng, mạng kết nối (Networking Fabrici) được chia thành hai phần: mạng tính toán (Compute Fabric) giữa các RU và các AI-RAN server và mạng hội tụi hội tụ (Converged Fabric) giữa các AI-RAN và internet. Mạngtính toán phân phối khối lượng công việc của RAN trông qua tuyến trước (Fronthaul) trên các AI-RAN server, các luồng lưu lượng đông- tây (est-west (E-W)). Mạng hội tụ mang các khối lượng công việc của RAN và AI đến và từ các AI-RAN server thông qua tiyến giữa/tuyến sau (Fronhaul/Backhaul) bắc-nam (north-eath (N-E)) cho lưu lượng không dây, cũng như cung cấp kết nối cho các mạng hữu tuyến cho lưu

lượng AI không bắt nguồn từ mạng không dây. Trong hình 13.4, cả hai mạng này đều được trình bày bởi cấu hình topo thiểu cây hai sương sống (two -Spine), bốn lá (Leaf) có thể mở rộng thêm trong các triển khai thực tế. Trong mạng tính toán, các kết nối tuyến trước đến từ các RU được kết hợp trong một site ô hoặc router kết hợp vận tải (Transport Aggregation Router) và được kết nối đến các AI-server thông qua một mạng kết nối "sương sống-lá" (Spine-Leaf). Kiến trúc lớp hai chuyển mạch này thường được sử dụng trong cấu hình topo kết nối mạng của trung tâm dữ liệu cho mục đích khả định cỡ, dự phòng, hiệu suất và đơn giản hóa quản lý mạng.

Trong một hệ thống sương sống-lá điển hình, các chuyển mạch lá (Leaf Switch) kết nối trực tiếp đến các điểm cuối của biên mạng (chẳng hạn: các server và các thiết bị biên khác), kết hợp lưu lượng từ chúng trước khi gửi đến lớp sương sống, trong khi đó các chuyển mạch sương sống (Spine Switch) hình thành lõi của mạng kết nối, định tuyến giữa các chuyển mạch lá. Mạng tính toán bao gồm hai loại chuyển mạch, cụ thể lad: cặp chuyển mạch lá tuyến trước phục vụ các điểm truy nhập cho các RU (chẳng hạn: các chuyển mạch lá 1-2 trong hình 13.4) và cặp chuyển mạch lá AI-RAN (chẳng hạn: hai chuyển mạch lá 3-4 trong hình 13.4) kết nối lớp sương sống đến các server AI-RAN biên. Từng chuyển mạch lá phân phối định thời thông qua đồng hồ chủ lớn giao thức thời gian chính xác (PTPGM: Precision Time Protocol Grand/ Master), nguồn đồng bộ định thời sơ cấp trong mạng tính toán của mạng. Sử dụng giao thức PTP, các chuyển mạch lá phân phối thông tin định thời chính xác cho các RU được kết nối đến mạng tuyến trước cũng như đến các DU trong các AI-server thông qua cấu hình topo đồng bộ của cấu hình phân chia lớp thấp hơn 3 (LLS-C3: Lower Layer Split Configuration 3) theo đặc tả O-RAN. Cấu hình mạng được tạo bởi các chuyển mạch lá tuyến trước/server liên kết nối thông qua lớp sương sống tạo nên một kiến trúc mạng khả định cỡ và dự phòng cao.

Trong khi từng AI-RAN server kết nối đến mạng tính toán tại đầu trước (Frontend), nghĩa là mạng tuyến trước, thì đầu sau của nó kết nối đến các cặp lá của mạng hội tụ (chẳng hạn: các chuyển mạch lá 5-8 trong hình 13.4) được liên kết nối thông qua các chuyển mạch sương sống (chẳng hạn các chuyển mạch sương sống trong hình 13.3). Mạng hội tụ kết nối các AI-RAN server đến tuyến giữa, tuyến sau hay internet phụ thuộc và các AI-RAN server làm chủ chỉ DU, DU và CU kết hợp hay DU, CU và CN kết hợp. Chẳng hạn, Các AI-RAN server chỉ làm chủ DU, DU có thể kết nối qua tuyến giữa đến CU, trong khi đó các AI-RAN server làm chủ đồng thời DU và CU có thể kết nối đến UPF LBO (User Plane Function Local Breakout: đột phá tại chỗ chức năng mặt phẳng người dùng) hay đến UPF trong CN (mạng lõi). Mặt khác, Một DU+CU+CN tập trung chạy trên các AI-server có thể kết nối đến internet (qua giao diện N6) thông qua mạng hội tụ như minh họa trong hình 13.4.

Phóng to các AI-RAN server trong hình 13.5, chúng ta khám phá ngăn xếp phần mềm trên các server này để hỗ trợ kiến trúc nhiều người thuê của AI và RAN trên cùng một nền tảng. Hình 13.5 minh họa các thành phần khác nhau của ngăn xếp phần mềm. Nó được thiết kế để trở thành đám mây điện tử gốc, hay bẩm sinh đám mây (Cloud Native), với hệ điều hành đám mây cấp độ thương mại (chẳng hạn Kubeernetes) cung cấp đám mây như là một dịch vụ (CaaS: Cloud-as-a-Service) cho mục đích điều phối tài nguyên động và quản lý hạ tầng. Hệ thống điều hành đám mây làm chủ các nền tảng tính toán và các mô hình giao diện lập trình ứng dụng (API) như CUDA (Compute United Device Architecture: kiến trúc thết bị tính toán hợp nhất) cũng như các nền tảng kết kết nối và các mô hình API như DOCA (Datacenter Infrastucture on-a-Chip: Kiến trúc cơ sở hạ tầng trung tâm dữ liệu trên chip) để chạy hiệu quả các ứng dụng RAN và AI khác nhau được hỗ trợ bởi tính toán tăng tốc. Đối với ngăn xếp RAN, DU, CU và CN được điều phối bởi thực thể SMO (Service Management and Orchestration: quản lý và điều phối dịch vụ) để hỗ trợ khối lượng lớn các ô và các ứng dụng RAN, trong khi đó đối với ngăn xếp AI, các thành phần phần mềm khác nhau làm việc cùng nhau dưới tác nhân của cụm API (API Cluster Agent) giám sát và quản lý khối lượng công việc của AI server trong các cụm Kubernetes. Đối với ngăn xếp AI, các khối cơ sở nền tảng bao gồm các SDK (Sofware Development Kit: bộ công cụ phát triển phần mềm) cho ứng dụng AI và các khung (chẳng hạn khung các mô-dun tế bào thần kinh NVIDIA (NeMo: NVIDIA neural Modules)), các vi dịch vụ AI (chẳng hạn: các vi dịch vụ suy luận NVIDIA (NIM: NVIDIA Inference Microservice)) và các AI API tiêu chuẩn để kết nối các thành phần này với các ứng dụng AI khác nhau (chẳng hạn: văn bản, giọng nói, video, hình ảnh) trên nền tảng này bẩm sinh hoặc thông qua các API phi máy chủ AI (AI Serverles). Một bộ điều phối bao trùm đầu cuối đầu cuối (E2E) làm việc đồng thời RAN SMO và tác nhân của cụm AI để theo dõi sử dụng tài nguyên và điều phối khối lượng công việc RAN và các yêu cầu suy luận AI trên cùng phần cứng chia sẻ hỗ trợ kiến trúc đa người thuê trong khi vẫn duy trì chất lượng dịch vụ (QoS) và chất lượng trải nghiệm dịch vụ (QeS) cho RAN.



AI Serverless API (AI Serverless Application Programming Interrface): giao diện lập trình ứng dụng phi AI server, E2E Orchestrator (End-to-End Orchestrator): bộ điều phố từ dầu đến cuối, SMO (Service Management and Orchestrationquản): quản lý dịch vụ và điều phối, AI Application: ứng dụng AI, Standard API (Standard Application Programming Interface): giao diện lập trình ứng dụng tiêu chuẩn, AI Microservice: vi dịch vụ AI, AI Application SDK (AI Application Sofware Development Skit): bộ phát triển phần mềm ứng dụng AI, CN (Core Network): mạng lõi, CU/DU (Central Unit/distributed Unit): đơn vị trung tâm/đơn vị phân tán, Compute Platform: nền tang tính toán, CUDA (Compute United Device Architecture): kiến trúc thết bị tính toán hợp nhất, Networking Platform: nền tảng nối mạng, DOCA (Datacenter Infrastucture on-a-Chip): Kiến trúc hạ tầng trung tâm dữ liệu trên chip, Cloud Operating System: hệ điều hành đám mây, CaaS (Cloud-as-a-Service): Đám mây như là một dịch vụ, AI-RAN Server: máy chủ AI-RAN, CPU (Central Processing Unit): đơn vị xử lý trung tâm, GPU (Graphics Processing Unit): đơn vị xử lý đồ họa, DPU (Data Processing Unit) đơn vị xử lý dữ liệu, Networking Switch: chuyển mạch nối mạng.

Hình 13.5. Ngăn xếp phần mềm AI-RAN nhiều người thuê

13.2.3. Các kiểu chuyển mạch và các Card giao diện mạng (NIC)

AI-RAN khi được triển khai trong trung tâm dữ liệu nhiều người thuê là máy chủ (Host) đồng thời cho C-RAN và các dịch vụ AI, nó sử dụng công nghệ dòng X ccủa NVIDEA (NVIDEA Spectrum-X technology) để hỗ trợ mạng nối mạng (Networking Fabric) hiệu suất cao.

Nền tảng nối mạng dòng X (Specrum-X Networking Platform) có các chuyển mạch Spectrum-4 và BF3 SuperNIC (Bluefield3 Super Network Interface Card: Card siêu giao diện mạng trường sắc xanh 3) cho phép tăng tốc hiệu suất mạng tạo sinh AI 1,5 lần trên các mạng Ethernet. Nó được tối ưu hóa cho tính toán AI và có nối mạng khác với tính toán đám mây. Bảng 13.1 cho thấy các điểm khác nhau nổi bật giữa yêu cầu ethernet truyền thống dựa trên các đám mây và nối mạng ethernet tính toán AI.

Bảng 13.1. Các điểm khác nhau nổi bật giữa yêu cầu ethernet truyền thống dựa trên các đám mây và nối mạng tính toán AI

Ethernet truyền thống dựa trên các đám mây	**Nối mạng Ethernet tính toán AI**
Các ứng dụng được ghép lỏng	Xử lý phân tán ghép chặt
Các luồng TCP băng thông và sử dụng thấp	Các luồng RoCE* băng thông và sử dụng cao
Khả năng chịu đừng jitter cao	Khả năng chịu đựng Jitter thấp
Lưu lượng không đồng nhất, đa đường dẫn thông kê	Dung lượng mạng mang tính bùng nổ, các luồng voi**

* RoCE (Remote Direct Memory Access over Converged Ethernet): Truy nhập bộ nhớ trực tiếp từ xa trên ethernet hội tụ.

** Elephan Flow: là các luồng dữ liệu cực lớn liên tục được thiết lập bởi giao thức TCP (hoặc các giao thức khác).

Ngoài ra, chuyển mạch Specrum cùng với MVIDIA BF3 thực hiện các chức năng sau:

1. Mạng tuyến trước với giao thức thời gian chính xác (PTP: Precision Time Protocol) đê kết nối RU đến DU do đám mây làm chủ
2. Các khả năng tuyến trước được định nghĩa bằng phần mềm để điều phối đàn hồi các khả năng của RAN và AI trên cùng hạ tầng.
3. Ánh xạ lại địa chỉ MAC (Medium Access Control: điều khiển truy nhập môi trường) để đảm bảo chuyển đổi DU liền mạch.
4. Đa kết nối (Multi-Homing) mạng ethernet riêng ảo (EVPN: Ethernet Virtual Private Network) và dự phòng tích cực-tích cực cho lưu lượng AI và RAN với sự hỗ trợ của cầu kết nối Linux tăng tốc (Accelerated Linux Bridge) được tăng cường bởi NIC FW (Network Interface Card Firmware: phần mềm nhúng của card giao tiếp mạng) hỗ trợ dự phòng PTP.
5. Chuyển tải an ninh (Security Offload) và tạo chuỗi chức năng dịch vụ.

AI-RAN mở rộng cho cả triển khai C-RAN (Celtralized RAN: RAN tập trung) và D-RAN (Distributed RAN: RAN phân tán). AI-RAN khi triển khai như là một D-RAN có

thể không sử dụng các khả năng của Spectrum-X như đã trình bày ở trên. Đối với D-RAN, NVIDIA cung cấp hai giải pháp nối mạng:

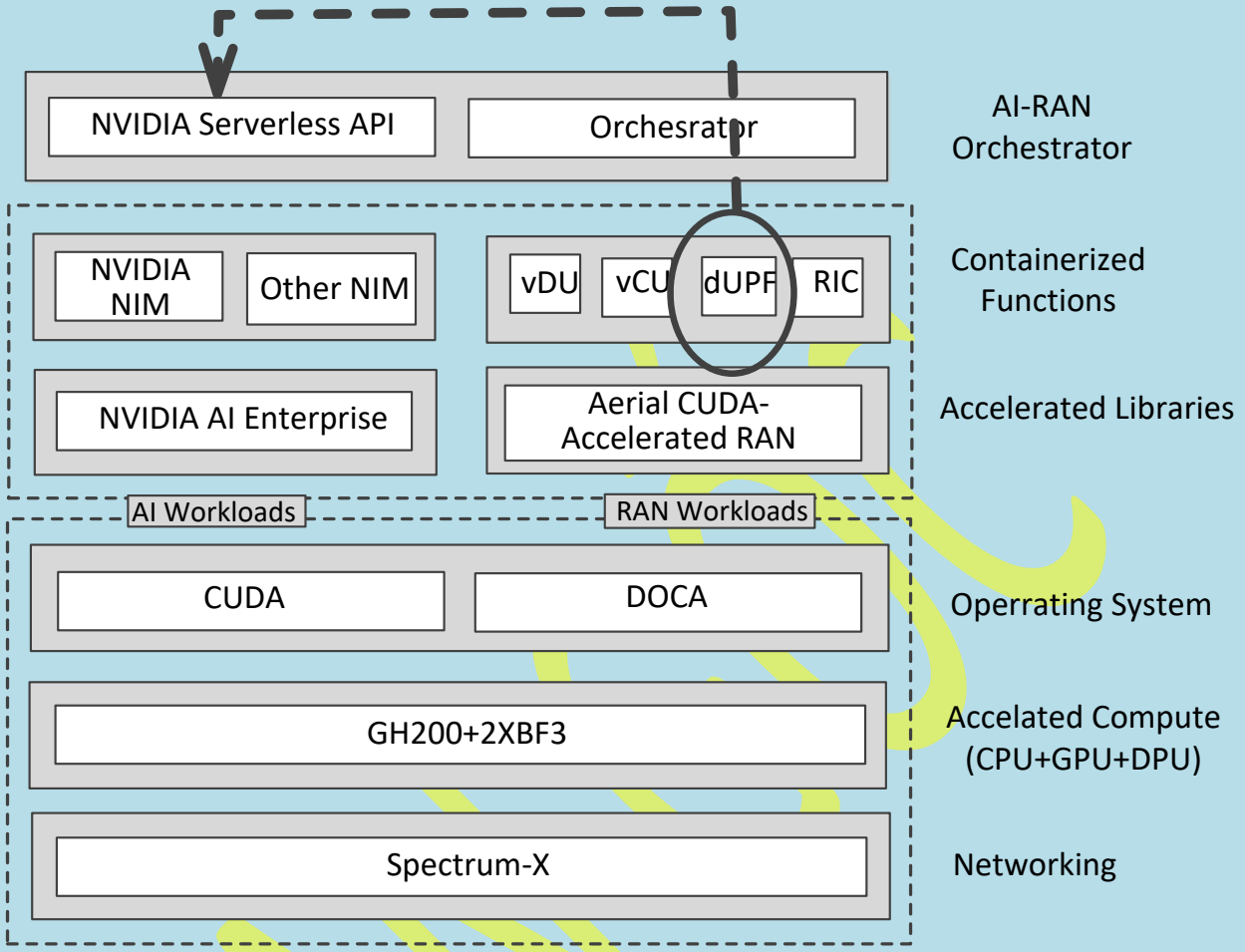
- BF3 DPU (BlueField 3 Data Processing Unit: đơn vị xử lý dữ liệu trường sắc xanh 3)
- ConnectX7 NIC (NIC kết nối X7) dựa trên các xem xét định cỡ và hiệu suất của một triển khai.

13.2.4. Giao diện tuyến trước được định nghĩa bằng phần mềm.

Card giao diện mạng (NIC: Network Interface Card) được sử dụng để triển khai AI-RAN phải có khả năng hỗ trợ kết nối mạng AI hiệu suất cao, cũng như hỗ trợ định thời RAN và các cấu hình RAN khác nhau. Giao diện đa dụng được định nghĩa bằng phần mềm cho phép quản lý mạng và nâng cấp phần mềm dễ ràng hơn, vì thế hỗ trợ tất cả ba trường hợp sử dụng như đã được xét trong phần 13.1.

AI ngày càng được tích hợp và được sử dụng nhiều hơn trrên các ngàng dọc (Industry Verticals). Các ứng dung phát triển rất nhanh đến AI tác nhân (Agentic AI)có các khả năng ra quyết định, thích ứng và tự mình thực hiện hành động. Nó được thiết kế để làm việc với sự giám sát của con người và có thể thực hiện các nhiệm vụ phức tạp. Các ứng dụng như các xe tự hành, các robot công nghiệp, phiên dịch ngôn ngữ thời gian thực, camera XR, tất cả đều yêu cầu xử lý tất định và trễ thấp để đem đến trải nghiện chất lượng cao.

Dự kiến AI-RAN cũng xử lý lưu lượng AI được phát trên các mạng không dây bằng cách cung cấp không gian và tài nguyên gần nhau cho 5G RAN và ngăn xếp AI trong các trung tâm dữ liệu của nhà mạng viễn thông: các vị trí kết hợp (Aggregation Sites), các tổng đài chuyển mạch di động (MSO: Mobile Switching Office) và các văn phòng trung tâm (Central Office). Kiến trúc mới này sử dụng các chức năng mặt phẳng người dùng phân tán 5G (5G dUPF: 5G Distributed User Plane Function) để kết nối hiệu quả lưu lượng AI đến một ngăn xếp suy luận AI bằng các API phi server của NVIDIA (NVIDIA Serverless API) và các NIM (NVIDIA Inferencing Microservice: vi dịch vụ suy ulận của NVIDIA) (hình 13.6).



NVIDIA Serverless API: API phi server của NVIDIA, Orchestrator: bộ điều phối, NVIDIA NIM (NVIDIA Inferencing Microservice): vi dịch vụ suy luận của NVIDIA, vDU (virtual Distributed Unit) : đơn vị phân tán ảo, vCU (virtual Central Unit): đơn vị trung tâm ảo, dUPF (Distributed User Plane Function): chức năng mặt phẳng người dùng phân tán, NVIDIA AI Enterprise: doanh nghiệp AI NVIDIA, Aeral CUDA (Aeral Compute United Device Architecture): kiến trúc thiết bị tính toán hợp nhất không trung, AI Workload: khối lượng công việc AI, RAN Workload: khối lượng công việc RAN, GH 200 Server: máy chủ siêu máy tính của NVIDIA được thiết kết cho tải công việc AI quy mô lớn và tính toán hiệu năng cao, BF3 (BlueField 3): trường sắc xanh 3 là thế hệ thứ ba của bộ xử lý dữ liệu DPU, 2x: hai bộ, Spectrum-X: các chuyển mạch Spectrum-X.

AI-RAN Orchestrator: bộ điều phối AI-RAN, Containerized Functions: các chức năng được con-ten-nơ hóa, Accelerated Libraries: các thư viện tăng tốc, Operating System: hệ điều khành, Accelerated Compute: tính toán tăng tốc, CPU (Central Processing Unit):

đơn vị xử lý trung tâm, GPU (Graphics Processing Unit): đơn vị xử lý đồ họa, DPU (Data Processing Unit): đơn vị xử lý dữ liệu, Networking: nối mạng.

Hình 13.6. AI-RAN hưởng lợi từ các chức năng được chia nhỏ

AI-RAN vì thế mở ra cơ hội cho các nhà mạng viễn thông chuyển đổi các trung tâm dữ liệu phân tán, các MSO, các CO và các POP (Point of Presence: điểm hiển diện), vào các trung tâm dữ liệu AI-RAN và cung cấp giá trị duy nhất bằng cách kết hợp các dịch vụ AI và RAN để đạt được trễ thấp và trải nghiệm tất định cho các ứng dụng mới nổi thấm đẫm AI mà kiến trúc AI tập trung hiên nay khó hỗ trợ.

Điều này chỉ có thể được thực hiện trên một hạ tầng được thiết kế kỹ lưỡng với các thành phần sau: các server DGX GH200, các chuyển mạch Spectrum và các BF3 DPU với các khả năng quan trọng của chúng như đã được mô tả.

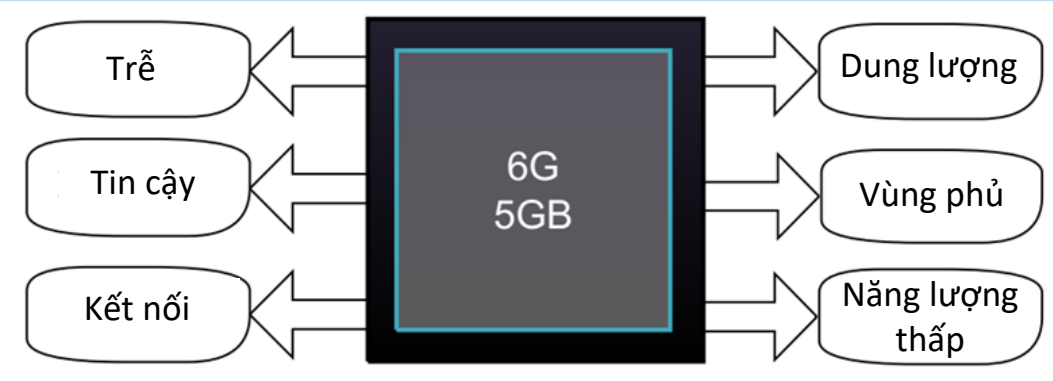
13.3. AI-RAN TRONG MẠNG TRUYỀN THÔNG KHÔNG DÂY DI ĐỘNG 6G

6G là công nghệ truyền thông thế hệ sau được xây dựng trên các mạng 5G và các trải nghiệm của người dùng hiện có. Thông qua sự tích hợp cua AI và ML, mạng lõi 5G được nâng cấp vào một mạng 6G thông minh. Mạng truy nhập vô tuyến trí tuệ nhân tạo 6G (6G AI-RAN) được mong đợi sẽ cung cấp các tính năng tiên tiến như trễ giảm, băng thông, tốc độ dữ liệu và vùng phủ được cải thiện. Ngoài ra, AI-RAN được kỳ vọng hỗ trợ các trường hợp sử dụng phức tạp như kết nối mức độ cao nhất, thuyền thông đa người dùng và truy nhập phổ động.

AI-RAN sẽ hỗ trợ hiệu quả giảm chi phí và cho phép đưa ra quyết định thông minh hơn cho các mạng 6G. AI-RAN là phiên bản nâng cấp của các RAN truyền thống được sử dụng để kết nối các thiết bị không dây đến mặt phẳng điều khiển và mạng lõi. Chẳng hạn, các hệ thống AI-RAN có thể dự đoán các mẫu lưu lượng mạng, thay đổi dung lượng mạng và giảm thiểu yêu cầu cung cấp quá mức. Ngoài ra AI-RAN có thể tối ưu hóa ấn định tài nguyên mạng theo thời gian thực, đảm bảo rằng các tài nguyên khả dụng được phân phát hiệu quả để thỏa mãn các yêu cầu của người dùng. Điều này có thể hỗ trợ giảm chi phí liên quan đến cân bằng tải. Ngoài ra, AI-RAN có thể hỗ trợ tăng khả năng phục hồi và khả dụng mạng bằng cách chủ động phát hiện và chẩn đoán các khiếm khuyết và các lỗ hổng an ninh của mạng. Các cách tiếp cận DP (Dynamic Programming: lập trình động) có thể quét các khối lượng lớn dữ liệu mạng theo thời gian thực để phát hiện các mẫu và các sự bất thường để có thể gợi ý các vấn đề có thể xảy ra. Nó có thể duy trì sự khả dụng mạng 6G và giảm thiểu thời gian ngừng hoạt động thậm chí khi đối diện sự

tăng đột biến lưu lượng. Ngoài ra AI-RAN có thể thúc đẩy tự động hóa và điều phối thông minh nhờ đó đạt đựa việc đưa ra quyết định thông minh hơn.

Hình 13.7 cho thấy các tính năng quan trọng của 6G AI-RAN.



5GB (5G Beyond): sau 5 G

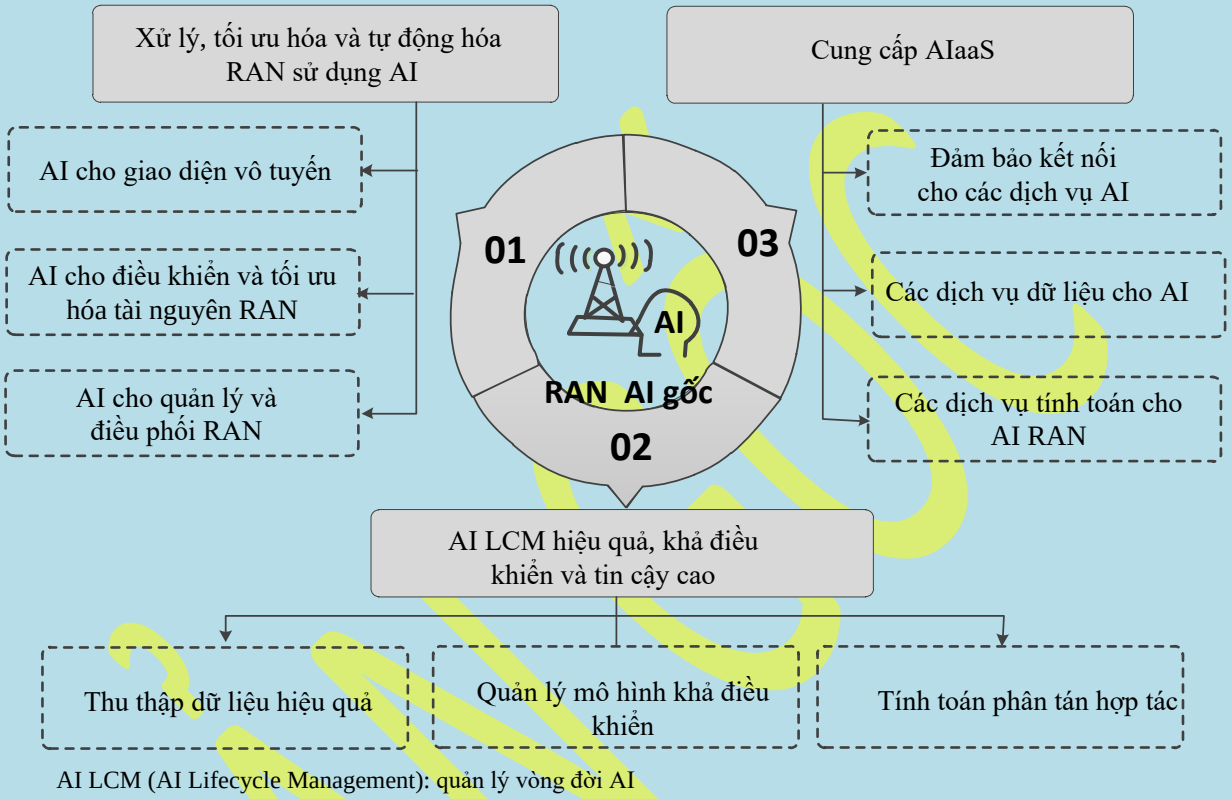
Hình 13.10. Các tính năng quan trọng của 6G AI-RAN

13.4. CÁC KHÁI NIỆM VÀ CÁC KHẢ NĂNG THEN CHỐT CỦA RAN AI GỐC

Kinh nghiệm nhận được từ tích hợp AI/ML vào 5G và 5G-A nhấn mạnh cả về tiềm năng cũng như các hạn chế của các cách tiếp cận hiện nay. Trong khi đã đạt được các tiến bộ nổi bật, bản chất phân khúc và đặc thù trường hợp của các công việc hiện thời để lộ việc thiếu một phiên bản kiến trúc dài hạn và thống nhất cho hòa nhập sâu của AI và RAN. Điều này đòi hỏi một chuyển dịch mô hình căn bản trong thiết kế 6G, tại đây AI không chỉ là cộng thêm, mà một khả năng gốc được nhúng khắp RAN . Một khái niệm được diễn đạt mạch lạc của RAN gốc AI là cốt yếu đến thống nhất các nỗ lực tiêu chuẩn hóa, dẫn dăt cách mạng kiến trúcvà cung cấp hướng dẫn rõ ràng, thống nhất cho phát triển sản phẩm tương lai. Điều này phản ảnh việc nền tảng tính toán đám mây góc (CNCF: Cloud Native Computing Foundation) đóng vai trò then chốt trong việc định hình khái niệm tính toán gốc AI, đặt nền mong cho cách mạng đám mây.

Như minh họa trong hình 13.11, việc thực hiện tầm nhìn của RAN AI gốc đòi hỏi ba khả năng quan trọng. Thứ nhất là xử lý, tối ưu hóa và tự động hóa RAN được vận hành bởi AI, cho phép điều khiển thông minh trên giao diện vô tuyến. Thứ hai, quản lý vòng đời (LCM: Lifecycle Management) AI hiệu quả, khả điều khiển và tin cậy, bao gồm thu thập

dữ liệu, quản lý mô hình và tính toán hợp tác. Thứ ba là cung cấp AI như là một dịch vụ (AIaaS: AI as a Service), cho phép chuyển đổi RAN vào một nền tảng thống nhất cho truyền thông và các ứng dụng AI.



Hình 13.11. Ba khả năng quan trọng của RAN AI gốc.

13.4.1. Xử lý, tối ưu hóa và tự động hóa được vận hành bởi AI

Với sự tăng không ngừng mức độ phức tạp của các mạng không dây hiện đại, các phương pháp tối ưu hóa dựa trên quy tắc hay mô hình truyền thống đang đối mặt với sự tăng không ngừng các thách thức. Ngược lại, các công nghệ AI chứng tỏ các ưu điểm độc nhất vô nhị trong việc xử lý xử lý sự phức tap, định cỡ, và sự không chắc chắn, thiếu rõ ràng thông tin. Trong ngữ cảnh xử lý, tối ưu hóa và tự động hóa, AI cung cấp các khả năng quan trọng: khả năng học các kịch bản, nơi mà lập mô hình tường minh không khả thi, khả năng đưa ra các quyết định thời gian thực trong các môi trường quy mô lớn, đa chiều và năng lực thực hiện dự đoán chính xác và thích ứng chủ động dựa trên các mẫu không gian thời gian cho tối ưu hóa chủ động.

Tổng thể, các khả năng này đặt AI vào vị trí như là một yếu tố mạnh mẽ trên lớp vật lý (L1), lớp liên kết (L2) và lớp mạng (L3). Bảng 13.2 cho thấy các trường hợp sử dụng của AI cho giao diện vô tuyến (L1) và AI cho điều khiển tà tối ưu hóa tài nguyên (L2/L3) từ góc nhìn hệ thống.

Bảng 13.2. Các trường hợp sử dụng cho xử lý, tối ưu hóa và tự động hóa RAN được vận hành bởi AI

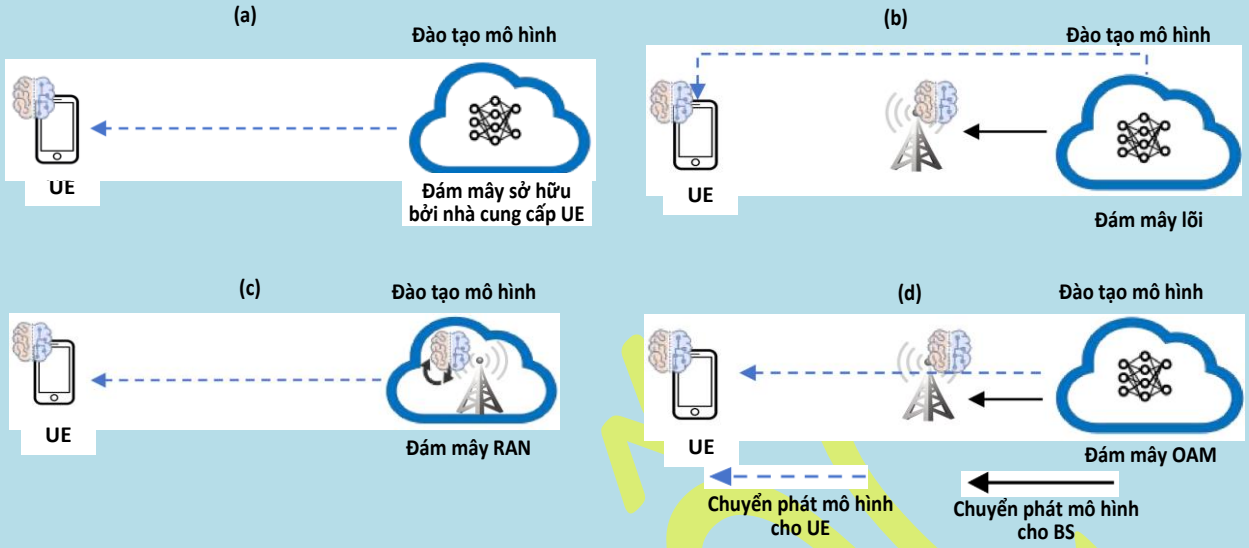
Các kịch bản	Trường hợp sử dụng	Các thách thức chính cho các giải pháp dựa trên không AI	Vị trí của suy luận AI	Các phương pháp AI tiềm năng & quy mô thông số	Trạng thái nghiên cứu
AI cho điều khiển và tối ưu hóa tài nguyên RAN	Cân bằng tải	Tối ưu hóa thụ động giảm cấp hiệu năng; đa chiều, phức tạp và thường gặp các vấn đề tối ưu hóa không lồi; khó tìm được giải pháp tối ưu thời gian thực	Phía BS	DNN (1M) LSTM (10M) DRL (<10M)	5G-A 3GPP Release 18, O-RAN
	Quản lý di động		Phía UE, phía BS	Rừng cây Tăng cường gradient LSTM (<1M)	5G-A 3GPP Release 18/19, O-RAN
	Tiết kiệm năng lượng mạng		Phía BS	DNN (1M) LSTM (10M) DRL (<10M)	5G-A 3GPP Release 18, thử nghiệm hiện trường trong mạng thương mại 5G-A, O-RAN
	Cắt lát mạng			LSTM (10M) GRU (<10M) Transformer[1] (10M-1B) Dubling Duel DQN[2] (1M)	5G-A 3GPP Release 19, O-RAN
	Tối ưu hóa dung lượng và vùng phủ			DRL(<10M) GNN (10M) LSTM (<10M) Transformer (10-1 B)	
	Ấn định tài nguyên và lập biểu đa người dùng			DNN (1M) DRL (10M)	Trường hợp sử dụng ứng viên 6G , O-RAN
	Tối ưu hóa QoS/QoE và đảm bảo thỏa thuận mức dịch vụ (SLA)			k-means [3](<1M) DNN (10M) CNN (1M) DRL (10M)	Thử nghiệm hiện trường trong mạng thương mại 5G-A, trường hợp sử dụng ứng viên 6G,

BẢN GỐC

					O-RAN
AI cho giao diện vô tuyến	Định vị	Chính xác bị giới hạn	Phía UE, phía BS Phía LMF[5]	DNN (<30M) CNN (<30M)	5G-A 3GPP Release 18/19
	Quản lý búp sóng	Chính xác bị giới hạn trong kịch bản tốc độ trung bình và cao	Phía UE, phía BS	CNN (<10M) LSTM (<10M) Transformers (<10M)	
	Nén CSI	Khó lập mô hình/lập công thức về mặt lý thuyết	Mô hình hai phía	CNN (<10M) LSTM (<10M) Transformers (<10M)	
	Dự đoán CSI	Chính xác bị giới hạn	Phía UE, phía BS	MLP-Mixer (<10M) 2D-FCN (1M) CNN (<1M)	
	Máy phát/ thu dựa trên AI	Khó lập mô hình/lập công thức về mặt lý thuyết		ESN (10M) Transformers (<10M)	Trường hợp sử dụng ứng cử 6G
	Bù méo phi tuyến tính khuếch đại công suất	Khó lập mô hình/lập công thức về mặt lý thuyết		ESN (<10M)	
	Dự đoán nhiễu và xử lý	Chính xác bị hạn chế và lỗi thời		LSTM (<10M) Transformers (<10M)	
	Giảm chi phí gián tiếp RS[4] (gồm cả hoa tiêu được chồng lên)	Chi phí gián tiếp báo hiệu tham chiếu cao	phía BS	CNN (<10M) LSTM (<10M) Transmers (<10M)	

Transformer[1] : bộ chuyển đổi, Double Dueling DQN[2] (Double Dueling Deep Q-Network: mạng Q sâu đối đầu nhân đôi), k-means[3]: thuật toán phân cụm k-mean,: tính hiệu tham chiếu, LMF[5] (Location Management Function): chức năng quản lý vị trí.

Các mô hình được áp dụng cho các trường hợp khác nhau trước hết phải trải qua đào tạo, kiểm tra và thẩm định để đảm bảo hiệu suất xử lý, tối ưu hóa và tự động hóa RAN dựa trên AI trong các hệ thống thực tế. Hình 13.12 cho thấy bốn phương pháp điển hình cho cách thức có thể đào tạo, chuyển phát và triển khai mô hình AI.



Hình 13.13. Bốn phương pháp điển hình cho cách thức có thể đào tạo, phân phát và triển khai mô hình AI.

Một kiểu các kênh mang vô tuyến (RB: Radio Bearer) có thể được định nghĩa để hỗ trợ AI/ML liên quan đến chuyển phát dữ liệu trên giao diện vô tuyến như được thể hiện trên hình 13.13. Kênh mang vô tuyến mới này chỉ được quản lý bởi RAN, cho phép RAN thiết lập và lập cấu hình động cho nó theo yêu cầu của các trường hợp sử dụng trong 6G RAN.

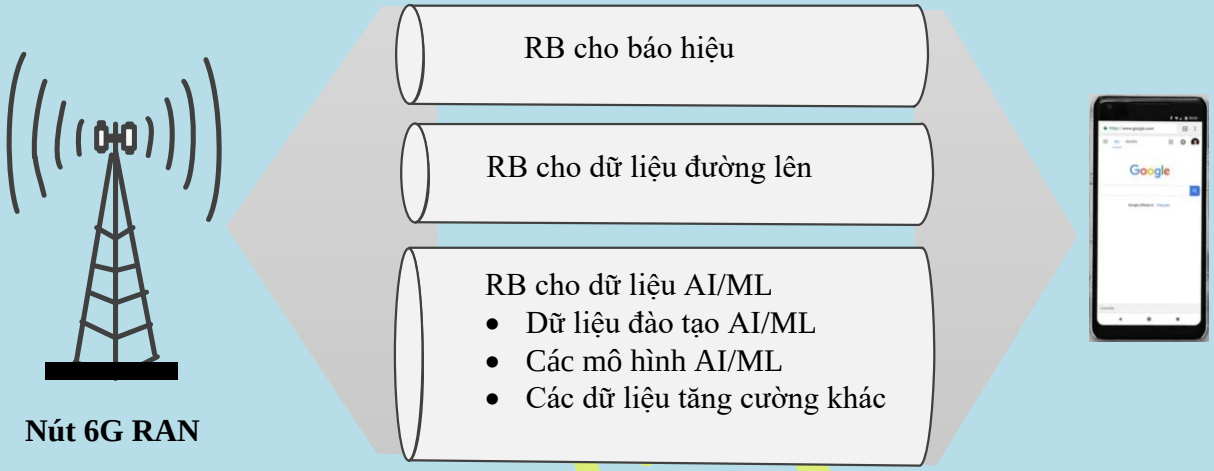
Các cân nhắc tiêu chuẩn hóa. Tiêu chuẩn hóa 5G đã mở đường cho các ứng dụng AI trong các mô-dun RAN đơn lẻ, đặt nền móng cho tich hợp AI rộng hơn. Trong 6G, được kỳ vọng sẽ phát triển vượt qua mô-đun đơn lẻ tiến đến các trường hợp sử dụng phức tạp và giá trị cao, như là tối ưu hóa kết hợp trên nhiều mô-đun và hợp tác AI liên lớp, nhằm khac thác hoàn toàn AI cho các trải nghiệm siêu tin cậy và hiệu suất cao của người dùng. Hứa hẹn rằng các trường hợp sử dụng có thể bao gồm máy thu phát, bù méo phi tuyến tính của bô khuếch đại công suất (PA) và giảm chi phí gián tiếp tín hiệu tham chiếu dựa trên AI…. Để hỗ trợ tối ưu hóa các trường hợp sử dụng này, rất cần có một khung đánh giá hiệu năng có tính hệ thống. Cần vượt qua độ chính xác hay các số đo thông lượng cơ sở. Thay vào đó, cần kết hợp các tiêu chí đa chiều, bao gồm hiệu suất năng lượng, khả năng tổng quát hóa mô hình và chi phí tổng thể của hệ thống, bao gồm các chi phí đặc thù AI như tính toán, thu thập dữ liệ và lưu trữ. Trong khi các mô hình đặc thù nằm ngoài pham vi của tiêu chuẩn hóa, các yếu tố then chốt như các cơ chế thu thập dữ liêu, các thủ tục quản lý mô hình và các khung tính toán AI phân tán cần được tiêu chuẩn hóa cho các triển khai RAN gốc AI khả định cỡ và khả tương tác.

13.4.2. Thu thập dữ liệu hiệu quả

Thiết kế giao thức 5G hiện thời được định hướng cho các mục đích truyền thông, nhưng ít cho các nu cầu thu thập dữ liệu AI/ML. Nhận xét tổng tổng thể các cách tiếp cận thu thập dữ liệu hiện thời đã được tiến hành trong 3GPP RAN. Trên giao diện vô tuyến, mặt phẳng diều khiển phải chịu sự không khả định cỡ kích thước tải tin cực đại. Trong khi mặt phẳng người dùng có ít các hạn chế hơn trong khối lượng dữ liệu, thì các luồng dữ liệu lại bị kết cuối bởi UPF (chức năng mặt phẳng người dùng) trong mạng lõi và có thể bị đánh cước làm cho nó kém thuận tiện hơn cho các nhiệm vụ AI/ML trong RAN. Ngoài ra, khi một khối lượng lớn dữ liệu được thu thập trong không trung vô tuyến, tồn tại rủi do làm kiệt quệ liên kết vô tuyến và cản trở các dịch vụ kết nối trong RAN. Ngoài ra, nhiều khía cạnh bao gồm quyền riêng tư và an ninh dữ liệu cũng đặt ra các thách thức cho thu thập dữ liệu. Trong phát hành 19, các đường dẫn khác nhau cho thu thập dữ liệu từ UE (server thu thập dữ liệu trong OTT trong lõi hoặc trong OAM) đang được bàn luận trong 3GPP theo khung 5G hiện có. Tuy nhiên, khi nó phát trển đến 6G, các thiết kế mới sẽ được xem xét cho hỗ trợ thu thập dữ liệu gốc trong RAN.

13.4.2.1. Kênh mang vô tuyến AI được dành riêng (Dedicated Radio Bearer).

Một kiểu mới của các kênh mang vô tuyến (Radio Bearer) có thể được định nghĩa để hỗ trự phân phát dữ liệu liên quan đến AI/ML trên giao diện vô tuyến như được thể hiện trên hình 13.13. Kênh mang vô tuyến này chỉ được quản lý bởi RAN, quản lý này cho phép RAN thiết lập và lập cấu hình nó một cách năng động theo yêu cầu của các trường hợp sử dụng AI/ML trong 6G RAN. Về phần tải tin, kênh mang vôt tuyến có thể hỗ trợ nhiều kiểu dữ liệu liên quan đến AI/ML bao gồm (nhưng không phải tất cả) dữ liệu đào tạo AI/ML, các mô hình AI/ML và các kiểu dữ liệu tăng cường khác có thể hỗ trợ các luồng công việc AI/ML (chẳng hạn: các số đo hiệu suất của các mô hình AI/ML). Kỳ vọng rằng các kênh mang vô tuyến mới sẽ đảm bảo cả khả năng linh hoạt lẫn hiệu suất trong các luồng công việc AI/ML trong 6G RAN. Ngoài kênh mang vô tuyến mới, một tunnel dữ liệu mới có thể được đặc tả cho các giao diện mạng để liên kết nối các nút 6G RAN, để cho phép treo đổi dữ liệu liên quan đến AI/ML. Chẳng hạn, một nút 6G RAN có thể thu thập dữ liệu từ các nút lân cận hoặc từ các UE được phục vụ của chúng cho đào tạo mô hình tập chung và sau đó nút này có thể phân phát các mô hình đào tạo đến các nút lân cận hay đến các UE theo yêu cầu.



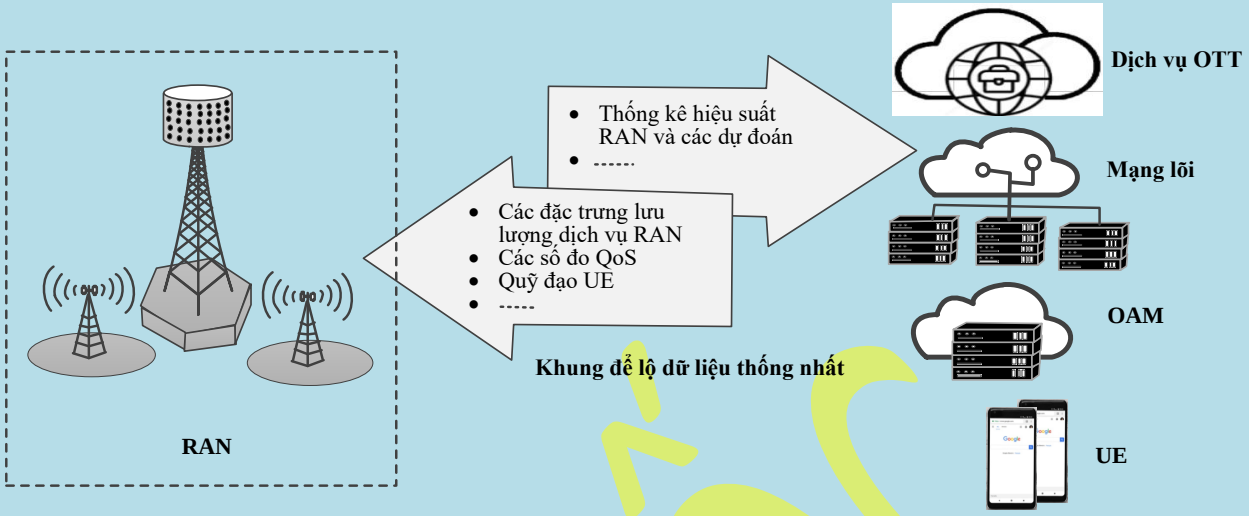
RB (Radio Bearer): kênh mang vô tuyến.

Hình 13.13. Một kiểu kênh mang vô tuyến AI dành riêng mới có thể được định nghĩa để hỗ trợ AI/ML để chuyển phát dự liệu trên giao diện vô tuyến.

13.4.2.2. Thu thập dữ liệu liên miền (Cross-Domain Data Collection)

RAN AI gốc của 6G được kỳ vọng cho phép AI/ML liên miền (Cross Domain AI/ML) để đảm bảo dịch vụ từ đầu đến cuối, dẫn đến thu thập dữ liệu liên miền (Cross Domain). Một trường hợp sử dụng là tối ưu hóa RAN nhận thức dịch vụ cho các dịch vụ AI . 6G RAN có thể tối ưu hóa hành vi của nó bằng cách thu tập các đặc tính thời gian thực từ các server ứng dụng. Các số đo QoE (Quality of Experience: chất lượng cảm nhận) cho dịch vụ AI/ML cũng có thể hữu ích cho một cách tiếp cân chuyên biệt, chẳng hạn: TTFT (Time to First Token: thời gian phản hồi token đầu tiên) trong suy luận mô hình ngôn ngữ lớn (LLM: Large Language Model). Dựa trên thông tin lớp dịch vụ , RAN gốc AI có thể áp dụng tối ưu hóa đặc thù dịch vụ như lập bản đò tập PDU (Protocol Data Unit: đơn vị dữ liệu giao thức) và là lập cấu hình DBR (Dedicated Radio Bearer: kêhn mang vô tuyến dành riêng). Trong khi đó, các số đo hiệu suất RAN và các dự đoán thời gian thực cũng có thể được thu thập bởi mạng lõi để điều chỉnh cấu hình QoS và hành vi UPF (User Plane Function: chức năng mặt phẳng người dùng) một cách năng động hơn. Với các nguồn dữ liệu đa dạng và các bộ thu thập chúng, 6G RAN có thể cung cấp hiệu quả hơn một khung toàn diện cho thu thập dữ liệu liên miền vượt trội để lộ khả năng của 5G.

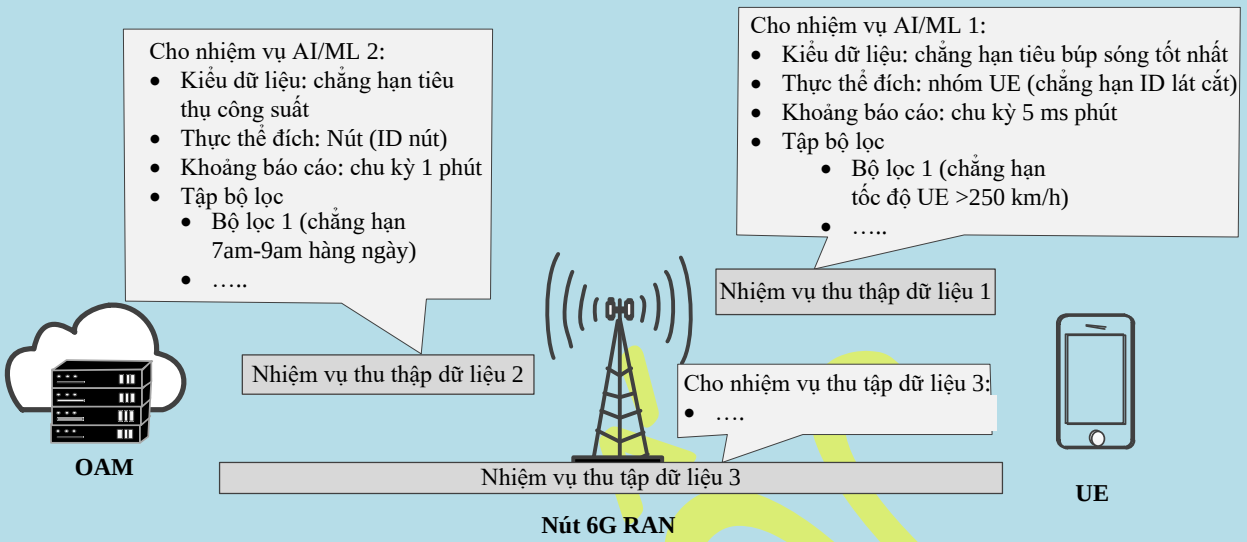
Hình 13.14 minh họa thu thập dữ liệu liên miền được mường tượng trong 6G.



Hình 13.14. Minh họa thu thập dữ liệu liên miền trong 6G.

13.4.2.3.Cơ chế thu thập dữ liệu RAN khả tùy chỉnh dựa trên nhiệm vụ

Giảm thiểu kiểm tra trên xe MDT (MDI:Minimization of Drive Test) là một giải sử dụng cho phép sử dụng dữ liệu thu thập từ các thiết bị di động để đo lường chất lượng mạng di động thay vì phải kiểm tra chuyên dụng trên xe để giảm thiểu chi phí. Vượt trội hơn MDT, các đo đạc hiệu suất (PM: Performance Meassurements) và các báo cáo đo đạc (Meassurement Reports), một cơ chế thu thập dữ liệu mới là yếu tố thiết yếu để thực hiện các yêu cầu thu thập dữ liệu đa dạng trong 6G RAN. Cơ chế mới nhấn mạnh thời gian thực, khả năng chi tiết và linh hoạt. So với các cơ chế định hướng theo OAM làm việc với khoảng thời gian 15 phút, cơ chế mới được điều khiển bởi 6G RAN có thể hỗ trợ trễ 10 ms hay thấp hơn cho các nhiệm vụ suy luận AI/ML thời gian thực. Nó cũng có thể thu thập dữ liệu RAN trên các UE đặc thù để cho phép các nhiệm vụ tối ưu hóa AI/ML chi tiết sâu. Ngoài ra, cơ chế mới này được đặc tả để giảm thu thập dữ liệu không cần thiết, chẳng hạn đào tạo một mô hình cho các UE tốc độ cao, việc thu thập thông tin UE từ tất cả các người dùng trong ô sẽ lãng phí tài nguyên vô tuyến. Cho mục đích này, cơ chế có thể hỗ trợ một tập các bộ lọc khả tùy chỉnh để đảm bảo chỉ thu thập các dữ liệu RAN cần thiết. Bên cạnh kiểu dữ liệu, một bộ lọc có thể đặc tả kiểu dịch vụ liên kết (chẳng hạn, thông lượng với 5 QI đặc thù), vùng quan tâm, dải thời gian (chẳng hạn ban ngày) hay trạng thái UE (chẳng hạn không thu thập dữ liệu từ các UE pin thấp). Các bộ lọc này có thể đặc tả một tiêu chuẩn bao gồm các ký hiệu toán học (chẳng hạn lơn hơn “>”) và các giá trị ngưỡng (chẳng hạn “khi tốc độ dữ liệu > 250 km/h”). Biểu thức logic này được đánh giá tại nguồn dữ liệu mà với nó chỉ một tập con dữ liệu thỏa mãn tiêu chuẩn được phân phát đến bộ thu thập dữ liệu. Hình 13.15 minh họa các thí dụ điển hình của cơ chế thu thập dữ liệu RAN khả tùy thỉnh dựa trên nhiệm vụ.



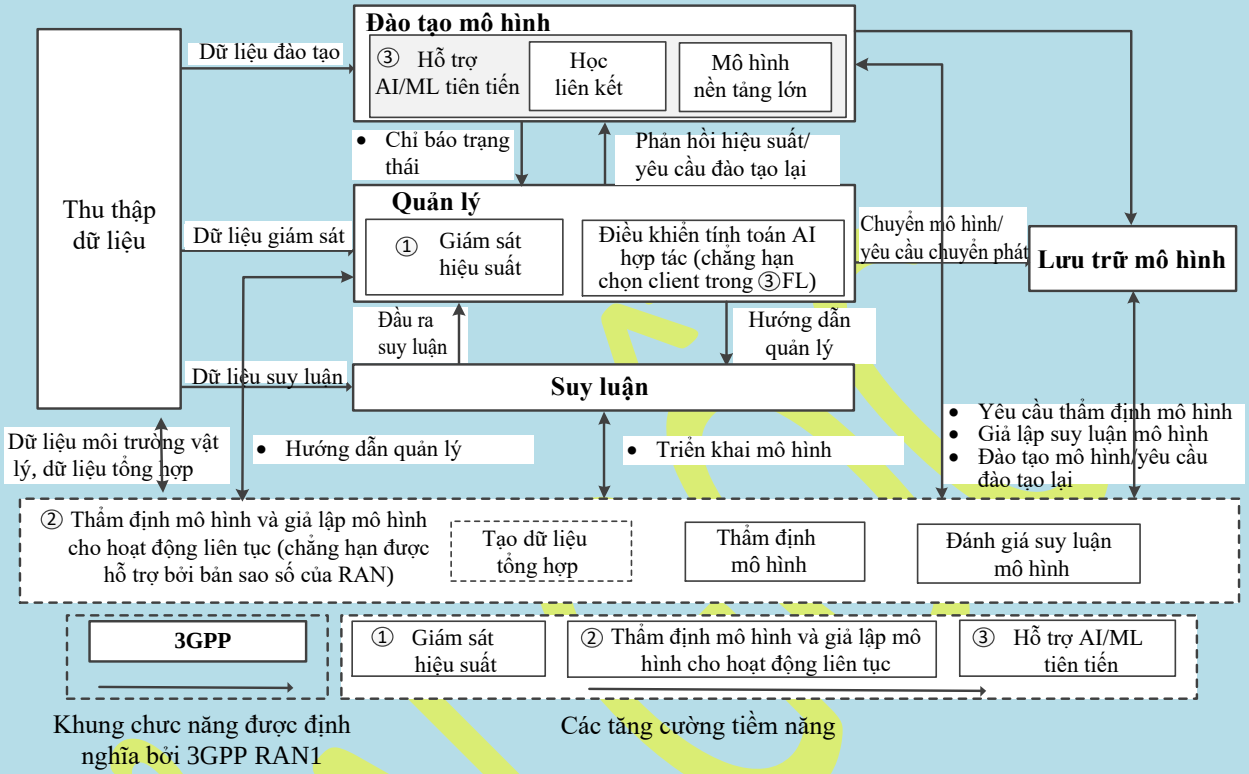
Hình 13.15. Cơ chế thu thập dữ liệu RAN khả tùy thỉnh dựa trên nhiệm vụ.

Các cấn nhắc tiêu chuẩn hóa. Ảnh hưởng của thu thập dữ liệu lên tiêu chuẩn hóa 6G RAN gồm hai khía cạnh. Một khia cạnh là các giao thức chung tên các giao diện liên quan bao gồm giao diện vô tuyến (chẳng hạn Uu), các giao diện RAN (chẳng hạn các giao diện Xn/F1/E1 trong 5G) và các giao diện với OAM và mạng lõi. Thiết kế các giao diện này sẽ không chỉ đảm bảo hiệu quả cho các nhiệm vụ thu thập dữ liệu đa dạng, mà còn khả năng quan sát và khả năng điều khiển của các nhà mạng di động. Khía cạnh khác là các đặc tả chi tiết của các kiểu dữ liệu và các định dang lên quan đến AI/ML(bao gồm dữ liệu đầu vào/đầu ra cho suy luận mô hình, dữ liệu đào tạo, v.v…), và các cấu hình liên kết cho các nhiệm vụ thu thập dữ liệu, Các đặc tả này sẽ được xây dựng một cách nhất quán trên đỉnh các giao thức chung nói trên và dựa trên các kịch bản đặc thù và các trường hợp sử dụng. Ngoài thu thập dữ liệu, các khía cạnh như xử lý trước dữ liệu, lưu trữ và truy xuất dữ liệu cần được chú ý hơn trong các tiêu chuẩn, điều này có thể yêu cầu một khung quản lý dữ liệu toàn diện hơn.

13.4.3. Quản lý mô hình AI/ML khả điều khiển và tin cậy cao hơn

Vì AI trở thành thiết yếu của điều khiển và tối ưu hóa 6G RAN, nên quản lý mô hình AI bền vững và khả điều khiển là điều tối quan trọng do bản chất xác suất của AI gây phức tạp cho khả năng dự đoán trong các môi trường năng động. Trong khi 3GPP đặt nền móng cho quản lý AI/ML trong 5G, bao phủ thu thập dữ liệu, đào tạo, suy luận và giám sát, thì các khung hiện thời không đạt cho triển khai thương mại , thiếu sự phát hiện sai lệch hoạt động, thẩm định trước khi triển khai và học mô hình liên tục để thích ứng các điều khiển thay đổi. Ngoài ra, các kỹ thuật mới nổi như học liên kết và các mô hình nền tảng lớn cung cấp cải thiện khả năng chính xác và thích ứng. Các tiến bộ cho quản lý mô

hình được kỳ vọng để để tăng cường quản lý cho 6G RAN AI gốc. Hình 13.16 minh họa khung quản lý mô hình được tăng cường trong 6G xây dựng trên khung 5G.



Hình 13.16. Quản lý mô hình được tăng cường cho 6G RAN.

Giám sát hiệu suất mô hình tăng cường. Sau khi được triển khai trong RAN, các mô hình AI được giám sát và điều chỉnh liên tục để đảm bảo sự ổn định và tin cậy của chúng. Trong khi 3GPP đã khởi xướng các nghiên cứu sơ bộ về giám sát hiệu suất của mô hình AI trong 5G/5G-A, tập trung lên sự chính xác của suy luận, các KPI và phân phối dữ liệu. RAN gốc AI đòi hỏi một khung ba chiều gồm hiệu suất mô hình AI, hiệu suất mạng và sử dụng tài nguyên. Giảm cấp chất lượng thầm lặng từ trôi dữ liệu (Data Drift), các kịch bản không lường trước, hay các thay đổi lưu lượng cần có giám sát từ đầu đến cuối để tích hợp các số đo mức độ chính xác của mô hình cùng với các KPI (chẳng hạn: trễ, thông lượng). Các chỉ số tài nguyên (trễ suy luận, tiêu thụ bộ nhớ và hiệu suất năng lượng) cũng rất quan trọng cho tính khả thi vận hành và đánh giá chi phí, tạo nên nền tảng cho các khung giám sát được tiêu chuẩn hóa. Các chỉ số đại diện được minh họa trong bảng 13.3 cung cấp nền tảng cho các khung giám sát khả định cỡ. Các số đo này kỳ vọng sẽ được thu thập từ đa dạng các nút suy luận AI và các giao diện.

Bảng 13.3. Các thí dụ về các số đo giám sát hiệu suất.

Chiều	Các số đo thí dụ
Hiệu năng mô hình	Độ chính xác, độ chuẩn xác, Độ thu hồi (Recall), MAE ((Mean Abssolute Error: sai số tuyệt đối trung bình), MSE (Mean Squared Error: sai số bình phương trung bình), RMSE (Root Mean Squared Error: sai số bình phương trung bình gốc), v.v…
Hiệu suất mạng	Tiết kiệm năng lượng mạng: tiêu thụ/hiệu suất năng lượng, tỷ lệ mất gói, tốc độ dữ liệu của người dùng, v.v… Quản lý di động: thất bại chuyển giao, lỗi liên kết vô tuyến, trễ, tỷ lệ mất hói, v.v… Nén CSI: BLER (Block Error Rate: tỷ lệ lỗi khối), thông lượng người dùng trung bình (UPT: Units Per Transacstion: số lượng đơn vị trên một giao dịch), 5%-UPT, v.v…
Hiệu suất tài nguyên	Tải tính toán, chẳng hạn sự sử dụng CPU/GPU Chi phí lưu trữ gián tiếp, chẳng hạn: lưu trữ bộ nhớ/độ chiếm bộ nhớ đệm Trạng thái sử dụng năng lượng của UE và RAN

Hoạt động liên tục của mô hình. Hoạt đông liên tục của mô hình là thiết yếu để đảm bảo hành vi AI bền vững và thích ứng trong RAN gôc AI, đặc biệt trong các điều kiện mạng và sự di động của người dùng liên tục thay đổi. Vì các UE chuyển động xuyên qua các ô, suy luận nhất quán phải được duy trì thông quan việc đồng bộ ngữ cảnh (chẳng hạn: ID, phiên bản mô hình) xuyen qua các gNB , được hỗ trợ bởi các giao diện Xn hay OAM-gNB chuẩn hóa. Để thích ứng với các môi trường thay đổi, mạng phải tiếp nhận đào tạo lại được khởi động bởi hiệu suất, nơi mà giảm cấp mô hình bị phát hiện bởi gNB hay UE khởi xướng thu thập dữ liệu, đào tạo lại, thẩm định và giả lập suy luận trước khi triển khai lại, hay đào tạo lại định kỳ với cập nhật chủ động các mô hình tại các khoảng thời gian đều đặn sử dụng các tập dữ liệu được làm mới. Cả hai cách tiếp cận đều đòi hỏi kiểm soát phiên bản mô hình kỹ lưỡng và khả năng theo dõi tập dữ liệu để đảm bảo sự nhất quán giữa đào tạo và suy luận. Để tăng cường tính bền vững hơn nữa, các bản sao số (Digital Tween) có thể cung cấp dữ liệu hợp lực và hỗ trợ thẩm định bằng cách mô phỏng các kịch bản thực tế để cho phéo kiểm tra trước khi triển khai, tọo dữ liệu tổng hợp và tối ưu hóa vòng kín. Khung tích hợp này hỗ trợ triển khai AI an toàn hơn, tin cậy hơn và phát triển mô hình bền vững trong các mạng 6G tương lai.

Hỗ trợ công nghệ AI/ML tiên tiến. Học liên kết (FL) tăng cường tổng quát hóa mô hình AI trong 6G RAN bằng cách cho phép đào tạo hợp tác bảo toàn quyền riêng tư xuyên xuốt các gNB phân tán. Mỗi nút đào tạo một mô hình nội bộ trên dữ liệu riêng của mình, trong khi đó cập nhật được tổng hợp, hoặc thông qua cộng tác đồng cấp hay các nút RAN

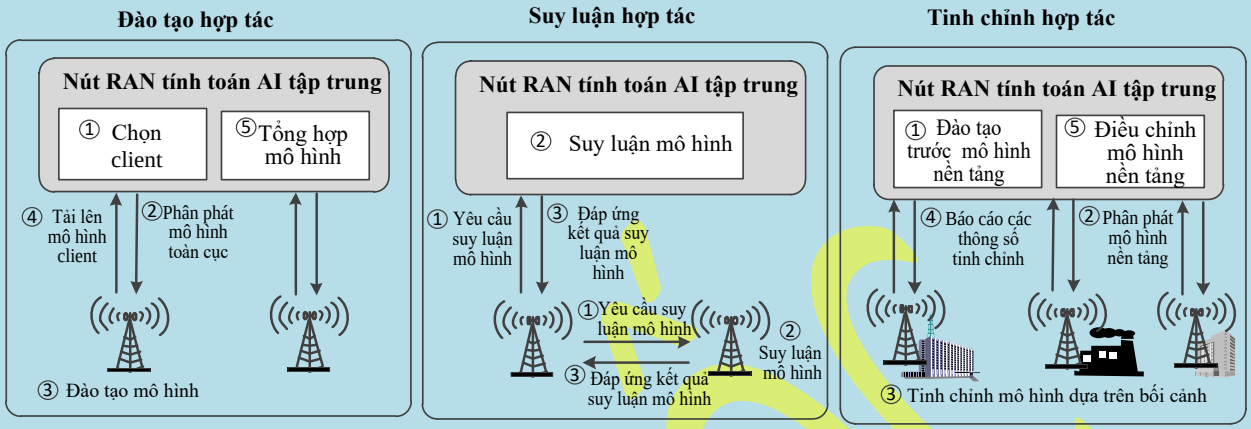
tập trung, để tạo ra một mô hình toàn cục phản ảnh được đa dạng các điều kiện mạng, như là các tải lưu lượng và các mẫu di động. Việc triển khai hiệu quả FL trong RAN đòi hỏi hỗ trợ kiến trúc cho chọn lựa thích ứng đối tượng tham gia để tối ưu hóa các nút với dữ liệu giá trị cao, các cơ chế để xử lý sự đa tạp thiết bị và dữ liệu cho việc hội tụ ổn định và kịp thời và tổng hợp cập nhật hiệu quả. Các cân nhắc này là thiết yếu để thực hiện các luồng công việc của FL khả định cỡ và bền vững trong các môi trường RAN gốc AI. Các mô hình nền tảng lớn cung cấp một cách tiếp cận thống nhất để lập mô hình năng đông vô tuyến phức tạp trong 6G RAN, cho phép các nhiệm vụ như dự đoán kênh và chọn lựa búp sóng với tổng quát hóa mạnh trên khắp các môi trường đa dạng. Để cho phép nó, thích ứng phân tán hiệu quả, nén mô hình cho suy luận trễ thấp và giá sát nhận thức nhiệm vụ cần được xem xét để đảm bảo hiệu suất trong các thiết lập hạn chế tài nguyên.

Các cân nhắc tiêu chuẩn hóa. Để đảm bảo sự ổn định và tin cậy trong các mô hình AI trong 6G RAN, điều cốt yếu là nghĩa định nghĩa tập hợp các số đo giám sát hiệu suất đa chiều qua các giao diện như Uu, Xn và các giao diện liên kết với quản lý mạng. Dựa trên hiệu suất được giám sát, cần có các thủ tục và các giao diện được chuẩn hóa để đảm bảo trao đổi các hướng dẫn quản lý mô hình, như khởi động đào tạo lại mô hình, cập nhật các phiên bản mô hình hay quay lại phiên bản không phải AI, giữa các UE, các nút RAN và các hệ thống OAM. Khi xây dựng trên các mô hình này, quản lý vòng đời mô hình từ đầu đến cuối phải được hỗ trợ thông qua các luồng công việc khả tương tác và được chuẩn hóa, bao gồm đào tạo lại, thẩm định và triển khai. Ngoài ra các cơ chế cho theo rõi ID mô hình và tập dữ liệu, kiểm soát phiên bản và quản lý siêu dữ liệu vòng đời phải được đặc tả để hành vi AI nhất quán và tin cậy trên các triển khai của nhiều nhà cung cấp. Để kiến trúc đảm bảo tương lai, một khung khả mở rộng cũng cần chứa đựng được các công nghệ mới nổi, như học liên kết (FL) và hỗ trợ tích hợp liền mạch chúng và hệ sinh thái 6G RAN.

13.4.4. Tính toán AI phân tán hợp tác (Distributed Collaborative AI Computing)

Tích hợp AI vào 6G RAN hứa hẹn cách mạng hóa tối ưu hóa mạng và chuyển phát dịch vụ. Tuy nhiên việc triển khai rộng rãi bị hạn chế bởi các tài nguyên tính toán, quỹ năng lượng bị giới hạn và dữ liệu bị phân khúc, cục bộ trong các nút cá thể. Các nhân tố này cản trở đào tạo và thực hiện các mô hình AI bền vững, nhấn mạnh sự cần thiết một mô hình tính toán phân tán, hợp tác trên khắp RAN. Để khắc phục các thánh thức này, các nút RAN tính toán AI tập trung các các chức năng hợp tác có thể được đưa ra để cho phép tập hợp tài nguyên AI và điều phối các khối lượng công việc trên nhiều UE và gNB. Các cơ chế AI hợp tác hiệu quả phải được phát triển để tối ưu hóa sử dụng tài nguyên và thống nhất trí tuệ trên khắp các phần tử mạng đa tạp, bao gồm các gNB, các UE và các

nút AI tập trung. Ba chế độ hợp tác cơ bản được mường tượng: đào tạo, suy luân và tinh chỉnh hợp tác được thể hiện trong hình 13.17.



Hình 13.17. Ba kiểu tính toán AI hợp tác trong RAN gốc AI.

Đào tạo hợp tác. Các nút gNB hay một nút RAN tính toán AI dành riêng đào tạo các mô hình AI toàn cục hay được cá nhân hóa với sử dụng các tập dữ liệu phân tán, mà điển hình là thông qua các mô hình học liên kết và phân chia. Cách tiếp cận này giảm bớt tính thưa tbớt và đa tạp của dữ liệu tại các nút cá thể trong khi vẫn bảo toàn quyền riêng tư dữ liệu và giảm chi phí truyền thông gian tiếp. Chẳng hạn, trong một kịch bản học liên kết cho dự đoán tính di động của người dùng, mỗi gNB thu thập dữ liệu được cục bộ hóa (chẳng hạn nhật ký chuyển giao, các theo dõi cường độ tín hiệu) và đào tạo mô hình cục bộ tại chỗ, Nút RAN tính toán AI tập trung tổng hợp các cập nhật này để luyện ra một mô hình di động toán cục, sau đó phân phát lại cho các gNB cho các vòng đào tạo tiếp theo. Vòng đào tạo lặp này đảm bảo cải thiện tổng quát hóa mô hình trên toàn bộ mạng.

Suy luận hợp tác. Các khối lượng suy luận có thể linh hoạt được phân chia giữa các gNB và các nút tính toán AI để tối ưu hóa hiệu suất trong các điều kiện hạn chế tài nguyên. Trong khi các gNB và UE có năng lực xử lý các nhiệm vụ suy luận nhẹ ký cục bộ, các gNB bị giới hạn tài nguyên có thể trực tiếp yêu cầu suy luận từ nút tập trung. Sử dụng dữ liệu được tổng hợp hay các suy luận tức thì từ nhiều vị trí, nút tính toán AI tập trung áp dụng các giải thuật tiên tiến hơn để tạo ra các chiến lược tối ưu hóa toàn cục, chẳng hạn giảm bới nhiễu và ấn định tài nguyên năng động, sau đó phân phát chúng cho các gNB để thực hiện hợp tác. Mô hình hợp tác này tăng cường tính thích ứng, hiệu suất và trí tuệ toàn bộ mạng trong thời gian thực.

Đào tạo trước và tinh chỉnh. Các mô hình đào tạo trước và được tinh chỉnh tại các nút biên sự dụng dữ liệu được cục bộ hóa, cho phép thích ứng với các điều kiện đặc thù trong khi đóng góp các cập nhật cho mô hình chia sẻ toán cục. Chẳng hạn một nút trung tâm

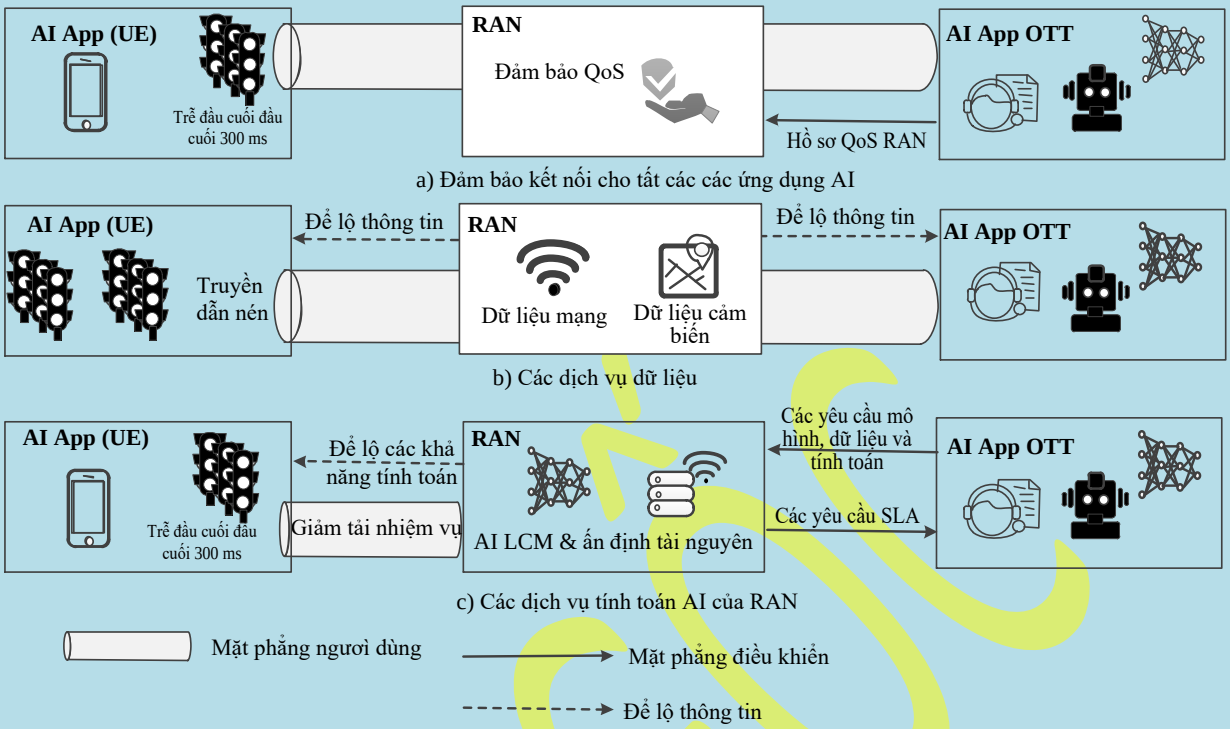
phân phát một mô hình phân loại lưu lượng tổng quát, sau đó các gNB điều chỉnh nó dựa trên các mẫu lưu lượng cục bộ để tăng cường lập lịch và thi hành QoS. Các UE cũng thích ứng các mô hình QoE với phản hồi được cá nhân hóa, cho phép các tối ưu hóa thời gian thực như điều chỉnh tốc độ bit video. Các cập nhật được tinh chỉnh được gửi ngược lại một các có chọn lọc để cải thiện mô hình toàn cục.

Để hỗ trợ các mô hình phân tán này, các cơ chế như gộp tài nguyên và lập lịch động hiệu quả là vấn đề then chốt. Các gNB và các UE có thể đăng ký các khả năng tính toán với các nút RAN tập trung, mà các nút tạo nên các hồ tính toán logic dựa trên địa lý, kiểu phần cứng hay các nhu cầu tốc độ. Các hồ này trừu tượng hóa tính pha tạp và hỗ trợ triển khai các khối lượng công việc của AI. Lập lịch thông minh năng động ấn định các các tài nguyên như các CPU/GPU dựa trên sự ưu tiên của nhiệm vụ, QoS và hiệu suất năng lượng để đảm bảo sử dụng tối ưu trên toàn RAN.

Các cân nhắc chuẩn hóa. Việc cho phép tính toán AI hợp tác trong RAN đòi hỏi các định nghĩa mang tính kiến trúc cho các chức năng trí tuệ phân tán trên các gNB và các nút tính toán AI. Vấn đề này bao gồm các thủ tục và các giao diện được tiêu chuẩn hóa để hỗ trợ đào tạo, suy luận và thích ứng mô hình hợp tác. Một khung quản lý và điều phối dành riêng cũng rất quan trong, bao gồm các chức năng như phát hiện tài nguyên giám sát sử dụng, và quản lý cấu trúc liên kết mạng (Topology). Để đảm bảo hợp tác hiệu quả, các giao diện giữa các thống quản lý và hạ tầng tính toán của RAN phải cho phép báo cáo tài nguyên tính toán thời gian thực, định cỡ và di chuyển khối lượng công việc AI, và tích hợp liền mạch với các hệ thống quản lý RAN hiện có trên các nút của mạng đa tạp.

13.4.5. Cung cấp AIaaS

AIaaS được công nhận rộng rãi như là một tính năng chính của các hệ thống 6G tương lai. Với mô hình này, các nhà mạng kỳ vọng phát triển từ các nhà cung cấp kết nối đơn thuần đến các nhà cung cấp tính toán AI biên và cuối cùng đến các nhà cung cấp dịch vụ AI biên. Đặc biệt, RAN gốc AI đóng một vai trò then chốt trong việc tạo điều kiện cho cuộc cách mạng này, cho phép ba mô hình phân phát dịch vụ được thiết kế để đáp ứng nhu cầu của các ứng dụng AI, như được mô tả trong hình 13.15.



Hình 13.15. Ba chế độ dịch vụ định hướng theo ứng dụng AI: a) đảm bảo kết nối cho các ứng dụng AI, b) các dịch vụ dữ liệu, c) các dịch vụ tính toán AI của RAN

- Đảm bảo kết nối cho các dịch vụ AI đề cập đến khả năng của AI cung cấp truyền dẫn tin cậy và thích ứng được thiết kế cho các yêu cầu độc nhất của các khối lượng công việc đòi hỏi sự phát triển vượt quá các năng lực mạng truyền thống. Ba hướng then chốt là: (i) tăng cường khung QoS để xử lý các luồng dữ liệu phức tạp, không rõ ràng, mhư các đầu vào đa chế độ và các cập nhật mô hình, (ii) hỗ trợ các mẫu lưu lượng mới như các luồng bên trong lưu lượng tong trí tuệ hiện thân (Embodied Intellegence) và lưu lượng đường lên cao. bùng phát (Uplink- heavy, Burstly Traffic) trong các trợ lý cá nhân, (iii) đảm bảo trễ tất định – chẳng hạn dưới 200 ms cho các trợ lý và dưới 100 ms cho AI hiện thân- có tiềm năng thông qua quyền ưu tiên có chọn lọc của lưu lượng mạng nền tảng.
- Các dịch vụ dữ liệu cho phép các ứng dụng AI thích ứng thời gian thực cho trải nghiệm người dùng bằng cách để lộ các số đo bên mạng nội bộ (chẳng hạn: tải lưu lượng, chất lượng liên kết) và dữ liệu cảm biến (chẳng hạn mẫu di động, vị trí). Chẳng hạn, các ứng dụng trợ lý riêng có thể điều chỉnh các tốc độ nén video cho truyền dẫn đường lên để giảm trễ, trong khi đó các hệ thống robot có thể điều chỉnh các tần suất cảm biến và cập nhật để tránh mất gói. Để hỗ trợ điều này, RAN cũng phải cung cấp các khả năng như dự đoán, tăng cường và làm sạch dữ liệu để đảm bảo để lộ dữ liệu chính xác và kịp thời.

- Các dịch vụ tính toán AI của RAN hỗ trợ các ứng dụng AI biên bên ngoài bằng cách để lộ các tài nguyên tính toán như các CPU, các GPU và các NPU (Neural Processing Unit: đơn vị xử lý nơron) trong RAN, trong khi cũng đáp ứng yêu cầu nội bộ trí tuệ RAN. Yêu cầu sử dụng kép này đòi hỏi lập lịch hiệu quả, xử lý song song và hợp tác chắt chẽ giữa truyền thông và tính toán để đảm bảo hiệu suất và trải nghiệm người dùng. Chẳng hạn một số tác giả đề xuất giải thuật tối ưu hóa truyền thông thích ứng kênh vô tuyến-tính toán kết hợp để chọn lọc năng động phần mô hình và các điểm thoát sớm, trong khi đó ấn định các tài nguyên đường lên và tính toán dựa trên các trạng thái kênh thời gian thực.

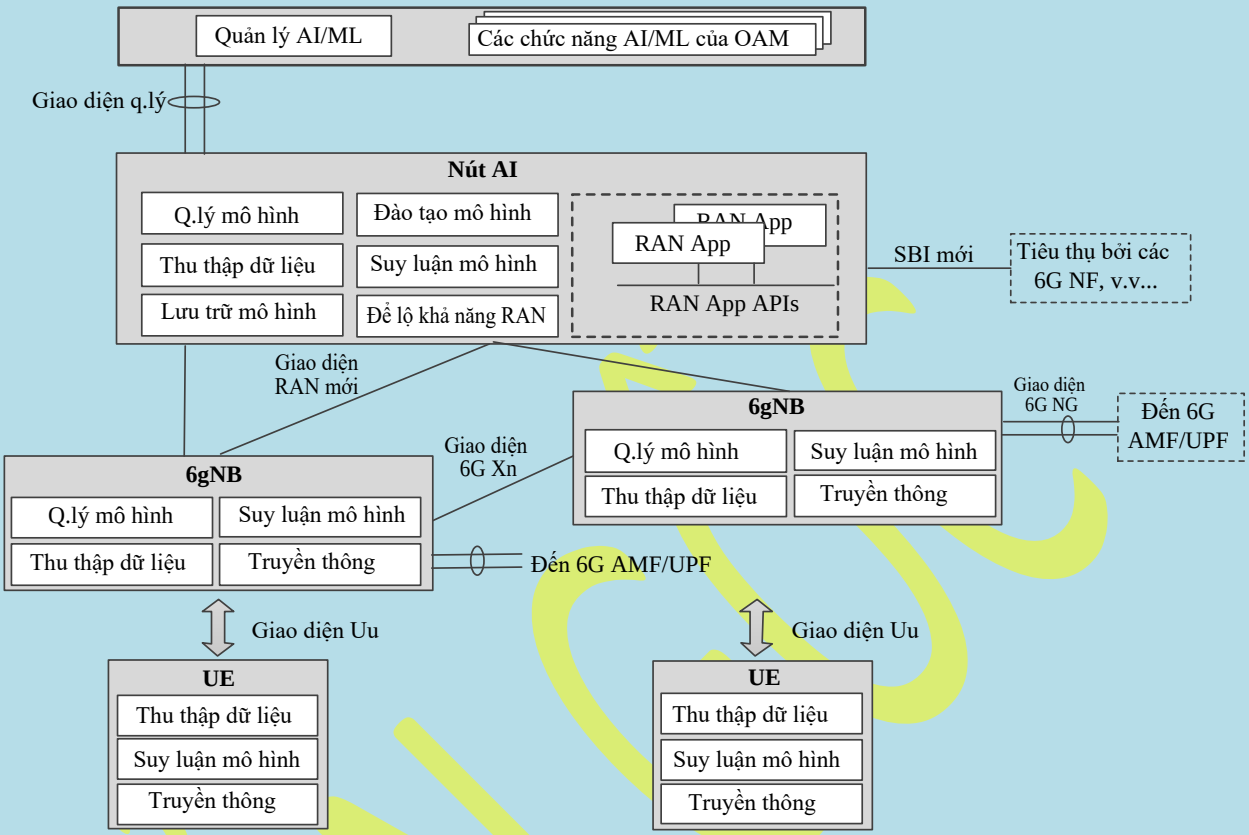
Cân nhắc tiêu chuẩn hóa. Để hoàn toàn giải phóng tiềm năng thương mạicủa các khả năng kết nối, các tài sản dữ liệu và hạ tầng của RAN, điều quan trọng bậc nhất là tiêu chuẩn hóa cơ chế đảm bảo QoS cho AI, để lộ dữ liệu RAN và chương trình khung cho các dịch vụ tính toán AI trễ thấp. Việc cung cấp RAN AIaaS đòi hỏi một khung API để tạo điều kiện truy nhập liền mạch đến các tài nguyên của RAN. Thông qua các API được chuẩn hóa, các nhà mạng có thể giảm các chi phí triển khai, tăng cường hiệu suất vận hành và phân phát các chức năng AI kết cấu mô-dun như là các dịch vụ theo yêu cầu. Ngoài ra các API mở thúc đẩy phát triển một hệ sinh thái hợp tác bởi sự tích hợp các khả năng AI ba bên, vì thế tăng tốc tiến hóa đến các mạng trí tuệ và linh hoạt.

13.5. CÁC THIẾT KẾ THEN CHỐT CHO RAN AI GỐC GIAI ĐOẠN ĐẦU

Tiêu chuẩn hóa thực sự cho RAN góc AI sẽ bao gồm các trường hợp sử dụng, kến trúc cad các khối cơ sở. Toàn bộ quá trình cho chuẩn hóa RAN gốc AI sẽ tuân theo cách tiếp cận dựa trên trường hợp sử dụng, bắt đầu từ nhận dạng cẩn thận các trường hợp sử dụng trong số các trường hợp trng phân 13.4.1.

13.5.1. Kiến trúc tham khảo cho 6G RAN gốc AI

Hình 13.16 cho thấy một kiến trúc được đề xuất cho 6G RAN gốc AI.



RAN App (RAN Application): ứng dụng của RAN, RAN App API (RAN Application Application Programming Interrface: Giao diện lập trình ứng dụng của ứng dụng RAN, SBI (Service Based Interface): giao diện dựa trên dịch vụ, 6G NG (6G Next Generation) thế hệ sau 6G, 6G AMF/UPF (6G Access Management Function/ User Plane Function: chức năng quản lý truy nhập/ chức năng mặt phẳng người dùng của 6G, 6gNB: trạm gốc 6G, UE (User Equipment): thiết bị người dùng, q.lý: quản lý, Giao diện Uu: giao diện vô tuyến., Giao diện Xn: giao diện giữa các 6gNB.

Hình 13.16. Một kiến trúc được đề xuất cho 6G RAN gốc AI.

Phần tử then chốt của kiiến trúc này là một nút định hướng theo AI/ML được đặt tên là "Nút AI". Nút AI và 6G BS ("6gNB"), có số lượng từ một đến nhiều phần tử được liên kết nối với nhau bởi một giao diện RAN mới, cùng nhau tạo nên 6G RAN. Các khía cạnh quản lý phi thời gian thực được hỗ trợ bởi RAN OAM trong đám mây trung tâm. Đám mây này không chỉ có khả năng của riêng mình (chẳng hạn: phân tích dữ liệu quản lý (MDA: Management Data Analystics) cho 6G) mà còn cho phép các nhà mạng di động giám sát và hợp tác tổng thể đối với các chức năng AI/ML được phân tán trong các nút RAN.

Nút AI được trang bị các tài nguyên tính toán AI/ML đủ đáp ứng được nhu cầu (chẳng hạn: các tài nguyên trên một nền tảng biên) và thường được triển khai như một nút được tập trung cục bộ. Nó có các chưc năng sau:

- **Thu thập dữ liệu**. Có thể thu thập dữ liệu từ RAN, UE và các miền khác cho các nhiệm vụ AI/ML bao gồm đào tạo mô hình, suy luận và giám sát hiệu suất. Cũng có thể lưu trữ dữ liệu RAN cho đào tạo AI/ML tương lai theo yêu cầu.
- **Quản lý mô hình.** Có thể thực hiện quản lý và điều khiển nội bộ cho các mô hình AI/ML sử dụng cho các trường hợp sử dụng L3. Chẳng hạn: triển khai mô hình, kích hoạt/hủy hoạt động, chuyển mạch giữa các mô hình đa ứng dụng.Tuy nhiên, có tiềm năng FL server phối hợp các 6NB phân tán nếu các 6GNB được trang bị các khả năng đào tạo AI/ML tương ứng.
- **Kho mô hình:** Có thể lưu trữ và danh mục các mô hình AI/ML cho RAN và UE, điều này có thể được triển khai cho L1,L2,L3. Cũng có thể phục như là một kho mô hình AI/ML cho các ứng dụng dịch vụ AI.
- **Đào tạo AI/ML.** Hỗ trợ đào tạo mô hình cho RAN và UE, bất kể các trường hợp sử dụng L1, L2 hay L3. Tài nguyên tính toán AI/ML được gộp cũng cho phép nút AI hỗ trợ các mô hình đào tạo quy mô lớn cho RAN 6G. Ngoài ra, hiệu quả của mô hình đào tạo được tăng cường bởi các khả năng bản sao số trong nút AI.
- **Suy luận AI.** Có thể thực hiện suy luận mô hình cho các trường hợp sử dụng L3 trong RAN. Các mô hình quy mô lớn cho RAN có thể được triển khai cho các trường hợp sử dụng phức tạp. Ngoài ra, có thể thực hiện các nhiệm vụ suy luận AI/ML được giảm tải từ các 6GNB trong các kịch bản suy luận hợp tác.
- **Để lộ khả năng của RAN.** Có thể để lộ các khả năng tính toán dự phòng (AI hoặc không AI) và khả năng lập trình cho các khách hàng được phép. Dịch vụ tính toán có thể được tiêu thụ bởi các ứng dụng dịch vụ OTT được triển khai nội bộ (không được thể hiện trên hình 13.16). Dữ liệu RAN có thể được tiêu thụ bởi mạng lõi thông qua một giao diện dựa trên dịch vụ (SBI: Service Based Interface) hay bởi OAM thông qua các giao diện quản lý. Khả năng lập trình AI/ML có thể được để lộ thông qua các giao diện mở API đến các ứng dụng RAN được chứa tại nút AI, điều này có thể tùy chỉnh các mô hình AI/ML và luồng công việc liên kết cho các trường hợp sử dụng đặc thù.

6GNB là một nút RAN phân tán cho truy nhập vô tuyến của UE. Ngoài các chức năng truyền thông cơ sở (chẳng hạn lập lịch năng động), nó còn quản lý AI/ML LCM (LifeCycle Management: quản lý vòng đời) và suy luận AI/ML cho các trường hợp sử dụng, cũng như các chức năng thu thập dữ liệu. Sự phân chia chức năng AI/ML đối với nút AI/ML như vậy đảm bảo các hoạt động thời gian thực trong L1/L2. Thiết kế này cũng

giảm thiểu yêu cầu tính toán trên các 6GNB, vì hầu hết các nhiệm vụ đào tạo mô hình AL/ML đòi hỏi tính toán lớn được tập trung tại nút AI.

Một số UE cũng cho các khả năng AI/ML cho RAN. Tuy nhiên do hạn chế về pin và kích thước của nó, các 6G UE này có lẽ chỉ hỗ trợ suy luận AI/ML và thu thập dữ liệu có liên quan, ngoài trừ các chức năng truyền thông.

Kiến trúc tổng thể đặt nền móng cho môt RAN gốc AI trong 6G có các tính năng chính sau:

- **Sử dụng hiệu qua tài nguyên tính toán**. Bằng cách giảm tải các nhiệm vụ đào tạo AI thỉnh thoảng mới xẩy ra từ các 6GNB và các UE đến các nút AI tập trung, có thể đạt được ghép kênh thống kê. Ngoài ra, nút AI có thể cung cấp dịch vụ suy luận AI/ML cho 5G/6G BS không có các khả năng tính toán, điều này đặc biệt hấp dẫn cho triển khai giá thành thấp trong các nước phát triển. Ngoài ra tài nguyên tính toán AI có thể được chia sẻ với các ứng dụng dịch vụ AI cục bộ và vì thể tăng cường hơn nữa sử dụng tài nguyên.
- **Hợp tác AI/ML phân cấp**.Thiết kế đượcđề xuất này ấn địnhh các chức năng AI/ML tho một cách tiếp cận có hệ thông, dựa trên bản chất đa dạng các nhiệm vụ AI/ML trong RAN và các đòi hởi kịp thời của chúng.Các cộng tác AI/ML liên miền với OAM hay mạng lõi cũng được hỗ trợ. Thiết kế này cũng tạo điều kiện cho khung học tiên tiến như học liên kết, chẳng hạn: giữa BS và các UE hay giữa OAM và các nút AI.
- **Luồng công việc AI/ML khả lập trình**. Các ứng dụng RAN nhẹ ký cho phép khả năng lập trình cao cho RAN gốc AI, cho phép các nhà mạng tùy chỉnh và tối ưu hóa các chức năng AI/ML dựa trên các trường hợp dịch vụ và các kịch bản triển khai đặc thù.Mô hình này đảm bảo 6G RAN thích ứng với đa dạng các yêu cầu (chẳng hạn, các chiều đứng) một cách linh hoạt (chẳng hạn, bằng cách chi nhỏ nhiệm vụ) so với quá trình R&D truyền thống định hướng theo thiết bị.

13.5.2. Các khối cơ sở cho RAN gốc AI giai đoạn đầu

Gựa trên các xem xét tổng quát và kiến trúc RAN được đề xuất, mội số mục cần được tiêu chuẩn hóa với các thực hiện chi tiết (còn được biết với cái tên là “giai đoạn 3” trong 3GPP hay nhiều tổ chức tiêu chẩn khác được mô tả dưới đây và được tổng kết trong bảng 13.4:

- **Thu thập dữ liệu**. Thiết kế gốc cho thu thập dữ liệu là bước đầu tiên hương đến RAN gốc AI trong 6G. Các cơ chế thu thập dữ liệu mới phải được đưa vào trước tiên cho giao diện vô tuyến và sau đó trên các giao diện lĩnh vực chéo (Cross

Domain). Các cơ chế này bao gồm chủ yếu là các giao thức và các thủ tục để lập cấu hình các nhiệm vụ thu thập dữ liệu và cách thức chuyển phát các gói giữa các thực thể với các cân nhắc về trạng thái, tính di động của UE và khả năng phối hợp trên các dịch vụ kết nối cơ sở. Các cơ chế thu thập dữ liệu chi tiết có thể tùy nhỉnh được cho đa dạng các nhiệm vụ thu thập dữ liệu phải liên tục được cải thiện trong vòng đời của 6G RAN.

- **Quản lý mô hình**. Được công nhận rộng rãi rằng quản lý mô hình trong mạng 6G sẽ tiến hóa đến một luồng công việc AI/ML trong mạng và liên tục và vì vậy, các thủ tục quản lý mô hình xem xét giai đoạn đào tạo mô hình và thẩm định trước trước khi triển khai và cập nhật trong vòng đời của nó có thể không cần thiết trong các tiêu chuẩn giai đoạn đầu. Các thủ tục này có thể bao gồm khám phá khả năng, lập cấu hình nhiệm vụ, kích hoạt/thôi kích hoat và báo cáo trạng thái, v.v… Trong khi đó, một tập mang tính hệ thống hơn của các thiết kế giám sát hiệu suất là rất quan trọng trong giai đoạn đầu của 6G, bao gồm các đặc tả các kiểu số đo hiệu năng có thể áp dụng cho nhiều mô hình hợp tác giữa các nút/các lớp cũng như các mô hình cá thể. Để tạo thuận lợi cho phát triển RAN với các công nghệ AI tiên tiến nhất, các công nghệ giai đoan đầu phải cung cấp sự ổn định và và khả năng linh hoạt cho các kỹ thuật tiên tiến như FL và các mô hình nền tảng lớn.
- **Tính toán AI hợp tác**. Hỗ trợ của giao thức cơ sở cho hợp tác giữa nút AI và 6gNB phải sẵn sàng trong giai đoạn đầu để tận dụng tối đa kiến trúc RAN được đề xuất liên quan đến sử dụng tài nguyên tính toán AI. Sự hỗ trợ cơ sở này phải liên quan đến các tương tác giữa nút AI và 6gNB cho đào tạo mô hình hợp tác và suy luận. Quản lý và điều phối các tài nguyên tính toán AI cũng rất quan trọng để cho phép tái sử dụng hạ tầng và đảm bảo khả năng tương thích tiến (Forward Compability). Điều này đòi hỏi nhận thức (Awareness), ảo hóa, giám sát và lập lịch được tối hưu hóa của hạ tầng AI của RAN. Trong giai đoạn sau, nhiều mẫu hợp tác hơn, chẳng hạn: đào tạo trước và tinh chỉnh chỉnh hợp tác, có thể được đưa và vì nhiều khả năng tính toán AI hơn được trang bị trong các 6gNB.
- **Đảm bảo dịch vụ AI**. Điều thiết yếu đối với 6G RAN gốc AI là phân phát kết nối tuyệt vời cho các dịch vụ AI di động mới nổi và vì thể các cơ chế cho đảm bảo QoS sẽ được sẵn sàng trong những ngày đầu. Để đáp ứng các yêu cầu của dịch vụ mới này, các cơ chế mới có thể liên quan đến thu nhận được các đặc tính lưu lượng dịch vụ và thích ứng năng động thời gian thực trong RAN, cùng với các tăng cường khung QoS để cân nhắc các KPI dịch vụ mới vượt ra ngoài các KPI truyền thống, chẳng hạn thông lượng, trễ và mất gói. Khi 6G RAN phát triển với các dịch vụ tính toán mới vượt ra ngoài kết nối, cần xem xét tối ưu hóa các tài nguyên vô tuyến và tính toán để đảm bảo trải nghiệm người dùng.

- **Dịch vụ dữ liệu RAN**. Trước hết một khung để lộ dịch vụ thống nhất là thiết yếu trong những ngày đầu cho RAN để để lộ các dịch vụ dữ liệu của nó đến các khách hàng bên ngoài. Khung phải cho phép phát hiện dịch vụ, kiểm sóat truy nhập và giám sát, tính cước, v.v… Thiết kế chi tiết cho khung dịch vụ này có thể dễ ràng hơn khi khởi đầu từ CAPIF (Common API Framework: khung API chung) trong 3GPP, nhưng cũng cần xử lý các đặc trưng trong RAN như triển khai phân tán và tính ẩn danh của UE.Trên dỉnh khung để lộ dịch vụ, một số các dịch vụ của dữ liệu RAN có thể xẩy ra trong giai đoạn đầu và sau đó là các khả năng như tăng cường dữ liệu tại giai đoạn muộn hơn.
- **Dịch vụ tính toán RAN AI**. Cùng một khung để lộ dịch vụ phải được sử dụng lại để để lộ các dịch vụ tính toán AI. Với khung để lộ dịch vụ thống nhất này, kỳ vọng răng các dịch vụ tính toán AI có thể sẵn sàng trên 6G giai đoạn đầu cho cả các nhà cung cấp dịch vụ OTT và các ứng dụng RAN. Khi các dịch vụ này được tiêu thụ bởi nhà cung cấp dịch vu OTT, hỗ trợ giảm tải lưu lượng dịch vụ là vấn đề rất quan trọng trong các tiêu chuẩn. Khi các dịch vụ này được tiêu thu bởi RAN, các API chuẩn hóa phải dược để lộ để cho phép các ứng dụng RAN ảnh hưởng lên lên hành vi của RAN. Khi 6G RAN phát triển, có nhể nhiều hơn các dịch vụ tính toán AI được xem xét, chẳng hạn: sử dụng dữ liệu RAN (chẳng hạn dữ liệu cảm biến) để tối ưu hóa/chuyên biệt hóa các mô hình AI cho các dịch vụ OTT.

Bảng 13.5. Các khối cơ sở cho RAN trong 6G giai đoạn đầu và các tăng cường hơn cho phát triển.

Các khả năng then chốt	Ảnh hưởng tiêu chuẩn hóa	Các khối cơ sở cho 6G ngày đầu	Các tăng cường hơn nữa cho phát triển
Xử lý, tối ưu hóa và tự động hóa dựa trên AI	AI cho các trường hợp xử dụng	- Các trường hợp sử dụng được cho phép bởi AI; - Các công dụng mới của 6G (chẳng hạn: giảm chi phí gián tiếp RS dựa trên, xem xét độ phức tạp và các độ lợi hiệu xuất	Các trường hợp sử dụng liên miền
	Thu thập dữ liệu	- Kênh mang vô tuuyếnAI dành riêng trên giao diện vô tuyến - Thu thập dữ liệu liên miền	Thu thập dữ liệu RAN tùy chỉnh dựa trên nhiệm vụ

khả điều khiển và tin cậy * LCM (Life Cycle Management): quản lý vòng đời	Quản lý mô hình	Luồng công việc AI/ML liên tục và trong mạng, cho phép tương thích tương lai - Đào tao/đào tạo lại mô hình, suy luận; - Thẩm định mô hình, triển khai và cập nhật mô hình; - Giám sát hiệu suất mô hình; - Phát hiện khả năng, lập cấu hình nhiệm vụ, kích hoạt/hủy hoat động mô hình và báo cáo trạng thái	Hỗ trợ công nghệ AI/ML tiên tiến
	Tính toán AI khả điều khiển	- Đào tạo hợp tác và suy luận hợp tác (cho sử dụng hiệu quả tài nguyên và cải thiện hiếu suất); - Quản lý và điều phối tài nguyên tính toán AI (cho phép hạ tầng RAN chia sẻ, tái sử dụng và tương thích tương lai)	Đào tạo hợp tác trước và tinh chỉnh
Cung cấp AIaaS	Đảm bảo dịch vụ AI	- Đảm bảo QoS kết nối	Đảm bảo cho các dịch dụ truyền thông và tính toán tích hợp
	Dịch vụ dữ liệu RAN	- Khung để lộ dịch vụ RAN; - Các dịch vụ để lộ dữ liệu RAN	Tăng cường dữ liệu RAN
	Dịch vụ tính toán AI của RAN	- Khung để lộ dịch vụ RAN (giống như trên); - Các dịch vụ tính toán AI của RAN	Các dịch vụ tính toán AI tiên tiến

13.5.3. CỘNG TÁC GIỮA 3GPP, ITU VÀ CÁC TỔ CHỨC PHÁT TRIỂN TIÊU CHUẨN KHÁC

Một phân chia rõ ràng công việc giưã 3GPP, ITU và các SDO (Standard Development Organization) tổ chức phát triển tiêu chuẩn khác là rất quan trọng cho hợp tác đa SDO đển hoàn toàn đạt được tàm nhìn RAN gốc AI.

Là một tổ chức liên kết giữa các chính phủ và công nghiệp viễn thông, ITU đóng một vai trò tổ chức có tầm nhìn xa trong việc thiết lập các mục tiêu và mạng thế hệ sau. Ngoài ra, nó phải đảm nhận vai trò lãnh đạo trong việc vạch ra một khung bao trùm thống nhất toàn cầu và các công nghệ cho các mạng gốc AI. Một hoạt động liên quan là nhóm tập trung lên gốc AI cho các mạng viễn thông (FG-ANN: Focus Group- AI-Native for Telecommunication Networks) được thành lập năm 2024 với mục tiêu đặc tả các công nghệ, nhận dạng các trường hợp sử dụng AI/ML mới, các ảnh hưởng của tiêu chuaant hóa và định hình lại kiến trúc mạng thế hệ sau đối với các thiết kế AI/ML gốc.

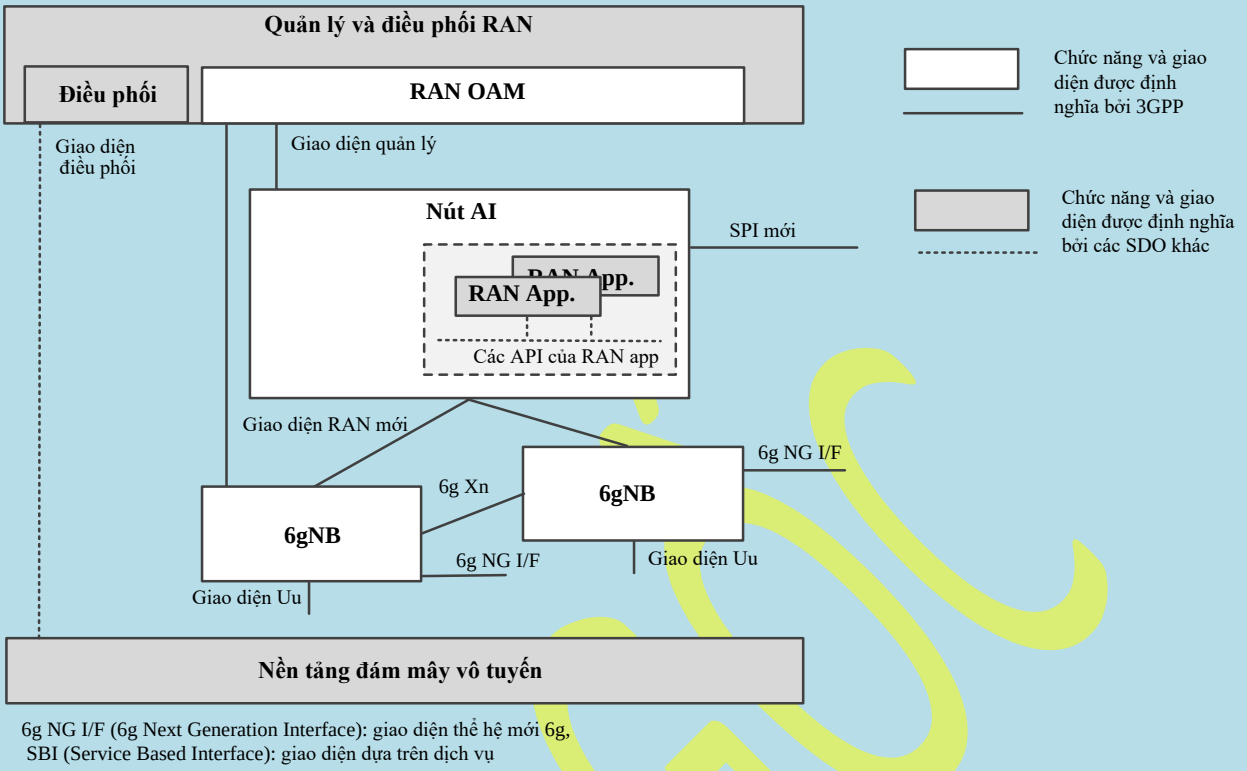
3GPP, là SDO quan trọng nhất cho công nghiệp viễn thông, phải chịu trách nhiệm cho các giao thức kiến trúc 6G từ đầu đến cuối và các giao diện đảm bảo khả tương tác, khả năng định cỡ và tương thích ngược trong các mạng thương mại. Nó cộng tác với các bên liên quan, các cơ quan quy định và giới học thuât để xử lý các nhu cầu mới nổi của 6G

bao gồm tích hợp với AI/ML, từ ITU và các nguồn khác.Cụ thể, 3GPP sẽ đảm bảo tương tác đa nhà cung cấp trong RAN giữa các trạm gốc, với mạng lõi cũng như các UE.

Vì NFV (Network Function Virtualization: ảo hóa phần tử mạng) và mô hình đám mây gốc bắt đầu thay đổi cách thức thức hiện các thiết bị RAN, các SDO khác bao gồm ETSI và liên minh O-RAN có thể đóng một vai trò thiết yếu trong hạ tầng đám mây RAN,mà hạ tầng này điều khiển các chức năng RAN và đóng vai trò quan trọng cho tương thích tương lai liên quan đến phần cứng. Ngoài ra các SDO không phải 3GPP cũng bổ sung kiến trúc RAN với các đổi mới đột phá với giả định rằng các tăng cường này phù hợp với các tiêu chuẩn của 3GPP và rằng công việc được để mở và không phân biệt đối xử.

Dưới khung bao trùm được thiết lập bởi ITU, các tùy chọn khác nhau cho công việc giữa 3GPP và các những SDO khác có thể được xem xét cho tiêu chuẩn hóa RAN AI gốc của 6G. Một tùy chọn được khuyến nghị được minh họa trong hình 13.17. Trong tùy chọn này, 3GPP chịu trách nhiệm cho các nút 6G RAN (6GNB và nút AI) và các khía cạnh quản lý cũng như mạng lõi và các UE. Các giao diện liên kết với các chức năng này cũng nằm trong phạm vi của 3GPP. Ngoại trừ kiến trúc phân cấp liên kết với nút AI, một tập AI/ML LCM nhất quán sẽ dược đặc tả trong 3GPP, sẽ ảnh hưởng lên tất các giao diện giao và các giao thức của chúng, cụ thể là giao diện mới giữa nút AI và các 6gNB. Bổ sung cho 3GPP RAN, các ứng dụng RAN được quản lý bởi nút AI và các giao diện API liên kết có thể hưởng lợi từ sự am hiểu chuyên sâu của liên minh O-RAN và có thể tiếp tục mở rộng ra ngoài 3GPP.

BAN CƠ



Hình 13.17. Phân chia công việc giữa 3GPP và O-RAN, EtSI, v.v…

13.6. CÁC KIẾN TRÚC O-RAN AI GỐC (AI-NATIVE O-RAN) CHO 6G

13.6.1. Tổng quan

Sự phát triển nhanh chóng của các mạng di động tiến tới các kiến trúc thế hệ sáu (6G) đòi hỏi tăng cường khả năng thích ứng, tự động hóa và hiệu suất trong các điều kiện đa dang và không thể dự đoán. Mạng truy nhập vô tuyến mở (O-RAN) đã nổi trội như là một mô hình chuyển đổi để đáp ứng các nhu cầu này. Bằng cách đón nhận các kiến trúc phân nhỏ, các giao diện được chuẩn hóa và các vòng điều khiển được vận hành bởi AI, O-RAN cho phép triển khai các ứng dụng điều khiển mô-đun (xApps, rApps) trên các bộ điều khiển vô tuyến thông minh (RIC). Liên quan đến phạm vi của xApps và rApps mở rộng trong các RIC, một trong các chức năng quan trong nhất và phát triển tích cực là ấn định tài nguyên, đang trải qua một sự chuyển dịch vào các hệ thống RAN mở vận hành theo yêu cầu điều khiển thời gian thực, thông minh và hiểu biết bối cảnh được hỗ trợ bởi 6G. Các mô hình lý thuyết trò chơi hợp tác đã chứng minh tiềm năng cho tăng cường chất lượng cung cấp dịch vụ thời gian thực, trong khi các thực hiện giao diện vô tuyến mở

(OpenAirInterface) đã phục vụ như là các dàn thử nghiệm thực tế cho cho 5G và mẫu thử nghiệm O-RAN. Những hiểu biết xâu sắc từ sự phát triển 5G sau phát hành 16 của 3GPP cho thấy các chiến lược thết kế cho các kiến trúc 5G khả định cỡ, với 5G NR đặt nền móng cho truyền thông siêu tin cậy trễ thấp.

Trong bối cảnh này, các bề mặt thông minh khả lập cấu hình (RIS: Reconfigurable Intellegence Surface) nổi lên như là các yếu tố then chốt với chuyển giao nhận thức vật cản (Brockage-Drine Handover) và tạo búp động (Dynamic Beaforming) cho phép cải thiện hiệu suất trong các một trường sóng mi-li-met dầy đặc. Các khung như ATHENA minh họa cách thức mà học máy tăng cường sâu (DRL: Deep Reinforcement Learning), cùng với lý luận biểu tượng (Symbolic Resonning) tăng cường chất lượng lập lịch tài nguyên vô tuyến trong các RAM ảo. Các kỹ thuật DRL đa tác nhân (Multi-Agent DLR) được áp dụng cho thiết kế bảng mã RIS và điều khiển búp sóng thích ứng cho phép nhận thức mô trường chi tiết. Sự kết hợp ngày càng tăng giữa RIS và và tối ưu hóa dựa trên học máy đã là khởi đầu cho nhiều đề xuất áp dụng DRL để tăng cường chất lượng tạo búp, điều khiển dịch pha và cực đại hóa tổng tốc độ (Sum-Rate maximization) trong các hệ thống được RIS hỗ trợ. Các công trình này mở rộng các mục tiêu thiết kế và các khung học máy, bao gồm cả tạo búp lai trong các mạng RIS đa chặng THz, các phương pháp diễn viên-nhà phê bình (Actor-Critic Method) cho điều khiển điều khiển đường xuống MISO (Multi-Input Single-Output: nhiều đầu và một đầu ra), tối ưu hóa được thúc đẩy bởi mục đích (PRO-Driven Optimization) cho các triển khai RIS xếp chồng, DRL phân cấp cho thích ứng pha rời rạc, học sâu cho điều chỉnh RIS và học máy tăng cường dựa trên kiến thức (Knowledge-Guided Reinforcement Learning) cho điều khiển điều khiển RIS thực tế- tất cả củng cố hứa hẹn về thích ứng thông minh, nhận thức bối cảnh trong các hệ thống không dây tương lai.

Ngoài ra quản lý dựa trên ý định (Intent-Based), được cung cấp nền tảng bởi các mô hình ngôn ngữ lớn (LLM: Large Language Model), đang định nghĩa lại tự động hóa và chuyển đổi chính sách trong các mạng thế đệ sau. Điều này phù hợp với các tầm nhìn kiến trúc rộng hơn của hạ tầng gốc AI trong lộ trình 6G. Chuyển đổi được đánh dấu bởi phát hành 19 của 3GPP củng cố lộ trình từ 5G-Advanced (5G-tiên tiến) đến các thực hiện sớm của 6G, cùng với các sáng kiến như AutoRAN, thúc đẩy tự động hóa không chạm (Zero-Tuch) trên vòng đời của RAN mở.

Điều khiển hợp tác RIS được vận hành bởi DRL hỗ trợ cả các chế độ ngủ hiệu suất năng lượng (Energy-Efficient Sleeep Mode) và lái búp sóng thời gian thực, trong khi đó các bộ chuyển đổi đa chế độ cho cảm biến và truyền thông tích hợp (ISAC: Integrated Sensing and Coomunication) cho phép chọn búp sóng linh hoạt. Các khung bản sao số (Digital Twin) hay còn gọi là song sinh số mới nổi cho RIS nâng cao hơn nữa khả năng cho ấn

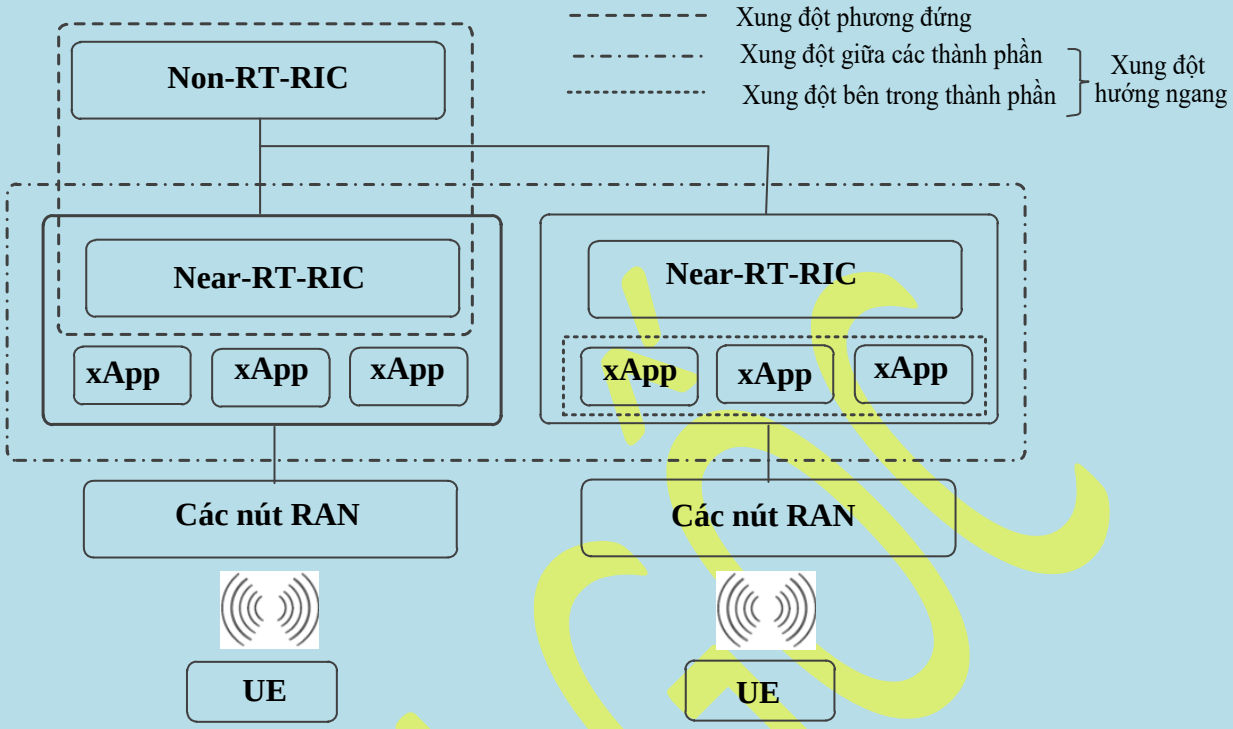
định tài nguyên bến vững, nhận thức bối cảnh và dự đón được được trong các hệ thống 6G gốc AI.

Tuy nhiên, điều phối AI gốc (AI Native Orchestration) trong O-RAN hiện nay phải đối mặt các thách thức đan xen. Thứ nhất, các hành động xung đột giữa các xApps và RApps được phát triển độc lập có thể dẫn đến giảm cấp hiệu suấtvà sư ổn định mạng. Thứ hai, thiếu ngữ cảnh trễ thấp được đồng bộ trên các tác nhân gây trở ngại đưa ra quyết định có điều phối. Thứ ba, một khoảng trống ngữ nghĩa vẫn còn giữa ý định nhà mạng mức cao và các chính sách mức thấp mà các bộ điều khiển mạng yêu cầu. Các thách thức này lại bị làm trầm trọng hơn bởi việc đưa vào phần cứng thời gian thực như các RIS, chúng đòi hỏi các thích ứng nhận thức môi trường trễ thấp vượt quá khuôn khổ các trừu tượng hóa truyền thống. Sử dụng RIS ngày càng tăng trong các hệ thống 6G đưa vào một lớp phức tạp mới, đặc biệt trong các triển khai dày đặc và đa nhà mạng . Alexxandropoulos và các cộng sự cung cấp một phân tích toàn diện về các kịch bản triển triển khai và tích hợp chúng vào các kiến trúc hợp chuẩn 3GPP và O-RAN. Họ đưa ra hai khái niệm nền tảng : *băng thông suy luận* (BoI: Bandwith of Inference) và *vùng suy luận* (AoI: Area of Inference), đặc trưng cho các miền phổ và thời gian mà các tấm panel RIS ảnh hưởng truyền lan tín hiệu. Các hiểu biết xâu sắc này tiết lộ rằng các thay đổi truyền sóng gây ra bởi RIS không đồng dạng cũng không tĩnh tại, và rằng điều khiển RIS hiệu quả phải nhận thức được bối cảnh và phải được tối ưu hóa động. Vì thế, các khung điều phối có nhận thức về RIS phải có khả năng giám sát có thể giám sát các hiệu ứng RIS cục bộ, thích ứng các chính sách theo gần thời gian thực và phối hợp với các chức năng RAN truyền thống thông qua các giao diện chuẩn hoá. Điều này dẫn đến một thiết kế xApps điều khiển RIS mô-đun hóa có khả năng thích ứng dựa trên học thời gian thực.

Dưới đây chúng ta sẽ xét các thách thức khi nghiên cứu thiết kê kến trúc O-RAN gốc AI.

13.6.2. Các xung đột quyết định trong O-RAN

Trong O-RAN xẩy ra các xung đột của quyết định điều khiển. Các xung đột duyết định trong các hệ thống O-RAN cho thể phân loại rộng thanh các loại chiều dọc và chiều ngang, như thấy trên hình 13.17. Nhận dạng và xử lý đúng các xung đột này là chìa khóa để đạt được điều phối mạng tin cậy và hiệu quả. Các xung đột được phân chia thành ba thể loại: (i) các xung độ chiều đứng: xuất hiện giữa các Non-RT và Near-RT-RIC,(ii) các xung đột giữa các phần tử: giữa các thực thể của Near-RT-RIC và (iii) các xung đột bên trong phần tử: bên trong một Near-RT-RIC. Các xung đột hường ngang đề cập đến va chạm giã các xAppx hoặc bên trong một Near-RT-RIC hoặc giữa các thực thể Near-RT-RIC. Các khác biệt này rất quan trọng để việc hiểu được cách thức mà sự không thống nhất này truyền lan trên các lớp RAN.



Non-RT-RIC: RIC phi thời gian thực; Near-RT-RIC: Ric gần thời gian thực.

Hình 13.17. Các kịch bản xung đột trong kiến trúc O-RAN.

Các xung đột chiều đứng. Các xung đột chiều đứng xảy ra giữa các mức phân cấp khác nhau của kiến trúc O-RAN. Một thí dụ điển hình là xung đột giữa một Non-RT-RIC và một Near-RT-RIC, trong đó cả hai thực thể có thể phát đi các lệnh điều khiển chồng lấn hoặc đối nghịch do sự khác nhau về thang thời gian hay các mục tiêu tối ưu. Sự không thống nhất trong các lện này có thể dẫn đến các cấu hình không nhất quán trê các lớp kiến trúc tại các nút E2, cần một cơ chế hài hóa trên các lớp kiến trúc.

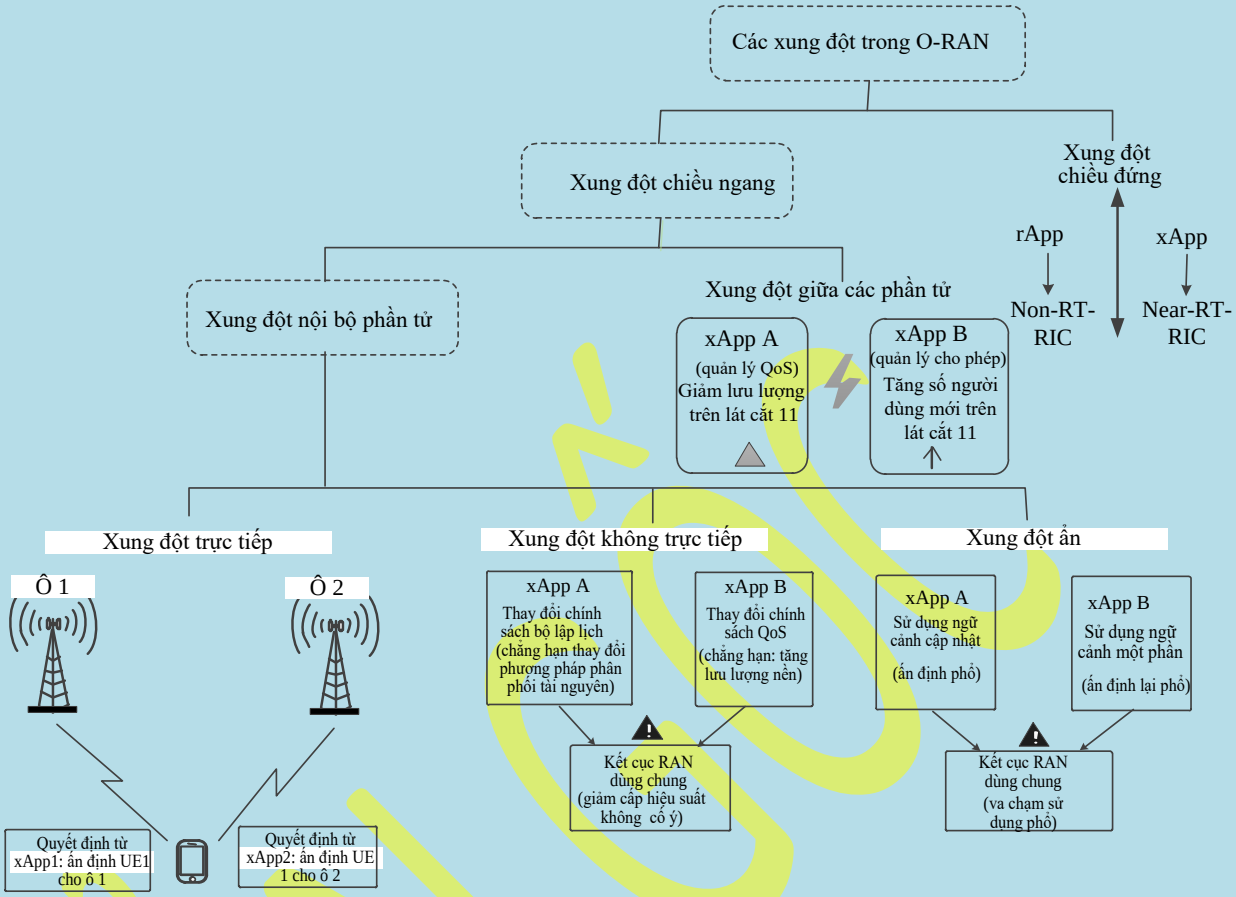
Các xung đột hướng ngang. Các xung đột hướng ngang xảy ra giữa các thành phần hoạt động trên cùng một mức phân cấp và được chia thành:

- **Các xung đột nội bộ thành phần**. Các xung đột này xảy ra bên trong một thành phần giữa các xApp trong cùng một Near-RT-RIC.
- **Các xung đột giữa các thành phần**. Các xung đột này nổi lên giữa các thành phần giống nhau, chẳng hạn các Near-RT-RIC cạnh nhau, đặc biệt khi quản lý các ô chồng lấn nhau.

Các xung đột nội bộ thành phần. Các xung đột nội bộ thành phần đặc biệt nguy hiểm do việc thực hiện đồng thời nhiều xApp với đa dạng các mục tiêu tối ưu hóa. Các xung đột này được chia thành các thể loại sau:

- **Các xung đột trực tiếp. Xaỷ ra giưa các xApp phát tường minh các lênh đối nghịch, chẳng hạn: ấn định các giá trị khác nhau cho cùng một thông số. Các xung độ này dễ rang được phát hiện và xử lý thông qua các cơ chế phân giải trước hành động, nhưng vẫn dẫn đến các thỏa hiệp dưới mức ưu nếu không được cân bằng tối ưu.**
- **Cac xung đột không trực tiếp. Xảy ra khi một thay đổi cb[ỉ một xApp ảnh hưởng không cố ý miền hiệu suất của một xApp khác, chẳng hạn điều chỉnh các mức công suất dẫn đến gián tiếp giảm thông lượng được nhắm tới bởi một xApp khác. Phát hiện đòi hỏi lập mô hình các các mối quan hệ phụ thuộc lẫn nhau tiên tiến.**
- **Các xung đột ẩn. Các xung đột này bắt nguồn từ các tương tác không rõ ràng, trong đó các hành động rời rạc ảnh hưởng lên các KPI dùng chung như trễ hay QoS. Việc nhận dạng các xung đột này đòi hỏi phân tích dữ liệu công phu và các kỹ thuật suy luận nhân quả để phát hiện các tương tác ẩn này.**

Hình 13.17 tổng kết các phân loại này, hình này minh họa trực giác bản chất phân cấp của các xung đột điều khiển trong O-RAN bao gồm các nhánh con hướng ngang, như trực tiếp, không trực tiếp, ẩn tàng và sự không thống nhất chiều đứng giữa các lớp.



Hình 13.17. Phân loại các kiểu xung đột trong các hệ thống O-RAN. *Chiều ngang* (trong cùng một lớp RIC) và *chiều đứng* (giữa Non-RT-RIC và Near-RT-RIC). Các xung đột chiều ngang lại được phân loại tiếp thành: (i) các xung đội nội bộ phần tử, xảy ra trong cùng một thực thể RIC, (ii) các xung đột giữa các phần tử, xuất hiện giữa các xApps khác nhau với các ý đồ trái ngược nhau và (iii) các biểu hiện thời gian thực chi tiết bao gồm các xung đột trực tiếp (chẳng hạn điều khiển đồng thời trên cùng một UE), các xung đột không trực tiếp (chẳng hạn xung đột các thay đổi đối với các thông số liên quan), và các xung đột ẩn (chẳng hạn các quyết định dựa trên ngữ cảnh lỗi thời và một phần). Hình vẽ nhấn mạnh bản chất phân lớp phức tạp trong các môi trường O-RAN bị phân nhỏ.

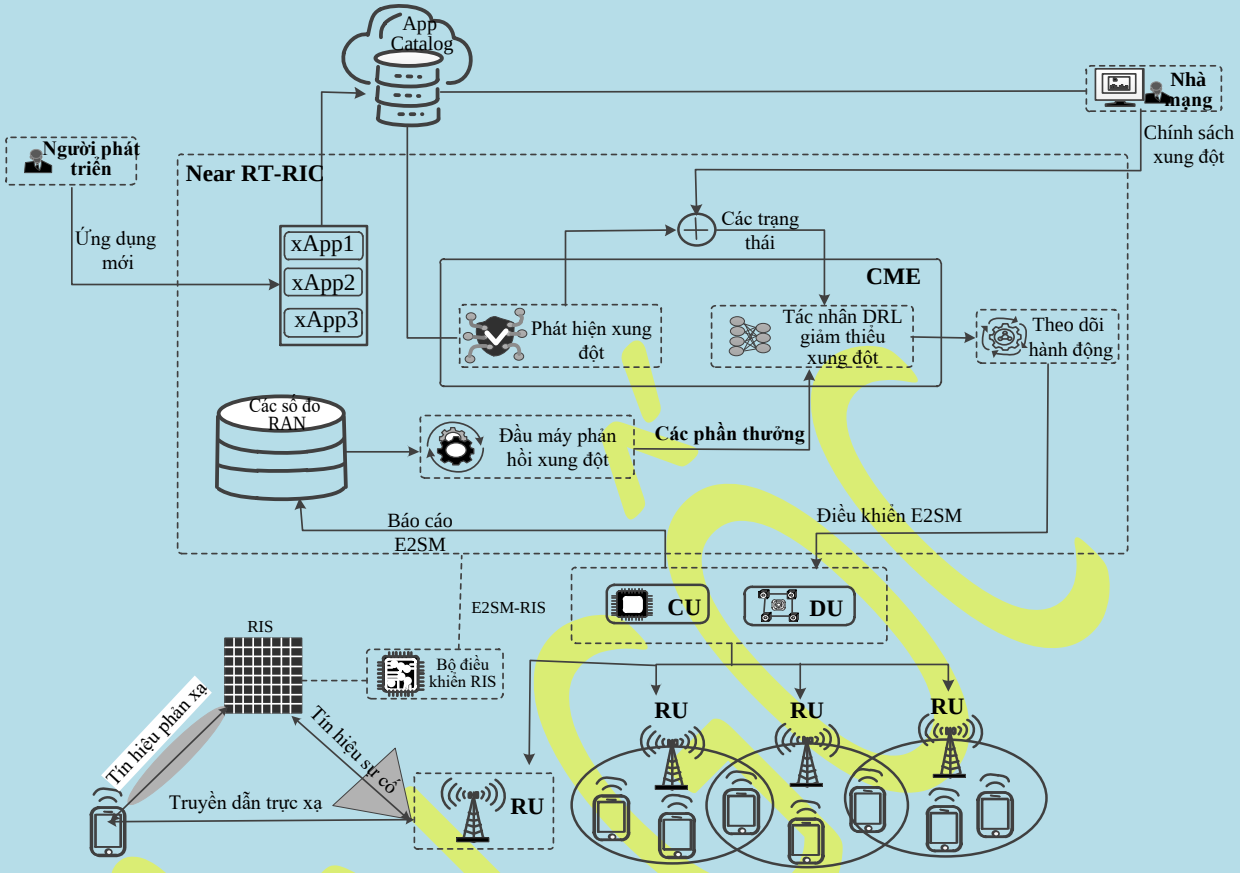
Để xử lý các thách thức này, một số tác giả đã đưa ra một đầu máy quản lý xung đột (CME: Conflict Management Engine) tích hợp một tác nhân học máy tăng cường sâu (DRL) trực tiếp vào mô-đun giảm thiểu xung đột (CM: Miligation Modul) của RIC. Giải pháp này cho phép tự học các chiến lược giảm thiểu tối ưu thông qua tương tác liên tục

với môi trường RAN mà không cần dựa trên các loại xung đột được định nghĩa trước hay các cơ chế hợp tác tĩnh giữa các xApp. CME này thích ứng năng động cac chính sách của nó dựa trên các trạng thái hệ thống có cấu trúc, bằng các kết hợp không chỉ các siêu dữ liệu xung đột và các đặc trưng của xAPP mà còn cả các chính sách cua nhà mạng và các chỉ số hiệu suất thời gian thực của RAN.

13.6.3. Kiến trúc RIC gần thời gian thực (Near RT-RIC) với đầu máy quản lý xung đột CME

Hình 13.18 minh họa kiến trúc RIC gần thời gian thực (Near RT-RIC) với đầu máy quản lý xung đột (CME: Conflict Management Engine) trong RIC gần thời gian thực. CME sử dụng một tác nhân học máy tăng cường sâu (DRL Deep Reinforcement Learning: học máy tăng cường sâu) dể học các chiến lược giảm thiểu xung đột thông qua tương tác liên tục với môi trường RAN. Nó xây dựng các vectơ trạng thái có cấu trúc tích hợp siêu dữ liệu xung đột, các đặc tính của xAPP, các chính sách của nhà mạng và các RAN KPI thời gian thực. Khi phát hiện một xung đột, DRL chọn các hành động như làm trễ, tối ưu hóa lại hay kết thúc các xApp và thực hiện chúng thông qua bộ theo dõi hành động (Action Tracker). Đầu máy phản hồi xung đột (Conflict Feeback Engine) tính toán các tín hiệu phần thưởng (Reward) dựa trên sự tuân thủ QoS và chính sách để cho phép thích ứng chinh sách theo thời gian.Hệ thống vòng kín cho phép giải quyết xung đột thích ứng theo thời gian , nhận thức ngữ cảng và tự chủ.

BẢN GỐC



E2SM (E2ServiceModel): giao diện mô hình dịch vụ E2, Near-RT-RIC: RIC gần thời gian thực.

Hình 13.18. Minh họa kiến trúc RIC gần thời gian thực (Near RT-RIC) với đầu máy quản lý xung đột (CME)

CME bao gồm tác nhân DRL, bộ theo dõi hành động và đầu máy phản hồi xung đột được tích hợp trong Nearr-RT-RIC. Tại từng bước quyết định t, tác nhân này quan sát môi trường thông qua một vecto trạng thái có cấu trúc, về quy tắc được định nghĩa như là một cấu trúc dữ liệu sau:

$$\mathbf{s}_t = \phi_t, \delta_t, \mathbf{q}_t, \mathbf{k}_t$$

Trong đó:

ϕ_t ký hiệu cho cặp số nhận dạng của các xApp xung đột cùng với các vai trò điều khiển và các mức ưu tiên của chúng.

δ_t ghi lại kiểu xung đột và một điểm số mức độ nghiêm trọng được tính toán.

$\mathbf{q}_t$ mã hóa các ràng buộc chính sách như các mức bảo vệ xApp hay các ngưỡng trễ được phép

$\mathbf{k}_t$ chứa các số đo hiệu suất thực tế thời guan thực, bao gồm sử dụng PRB, trễ trung bình và thông lượng thu thập được từ đo lường từ xa thông qua giao diện E2.

Dựa trên trạng thái quan trắc $\mathbf{s}_t$, tác nhân DRL chọn một hành động $a_t \in \mathcal{A}$ từ một không gian hành động rời rạc:

$a_t \in \{\text{Trễ}_i, \text{Kết thúc}_j, \text{Thiết lập lại thứ tự ưu (I,j)}, \text{ghi đè}_i, \text{không hành động}\}$

Mỗi hành động tương ứng một hoạt động giảm thiểu xung đột đặc thù, được thực hiện thông qua bộ thực hiện hành động (Action Maker) giao diện với mặt phẳng điều khiển của RIC để thực thi các quyết định trên các thực thể xApp.

Tiếp sau thực hiện hành động, CPE (Conflict Feedback Engine: đầu máy phản hồi xung đột) tính toán một r_t vô hướng dựa trên thành công của phân giải xung đột, bảo bảo trì QoS và sự tuân thủ các chính sách của nhà mạng. Tác nhân cập nhật chính sác của nó $\pi\ a_t | \mathbf{s}_t$ sử dụng phản hồi này, học dần dần để tổng quát hóa trên các kịch bản xung độ đa dạng và đang dần dần tiến hóa.

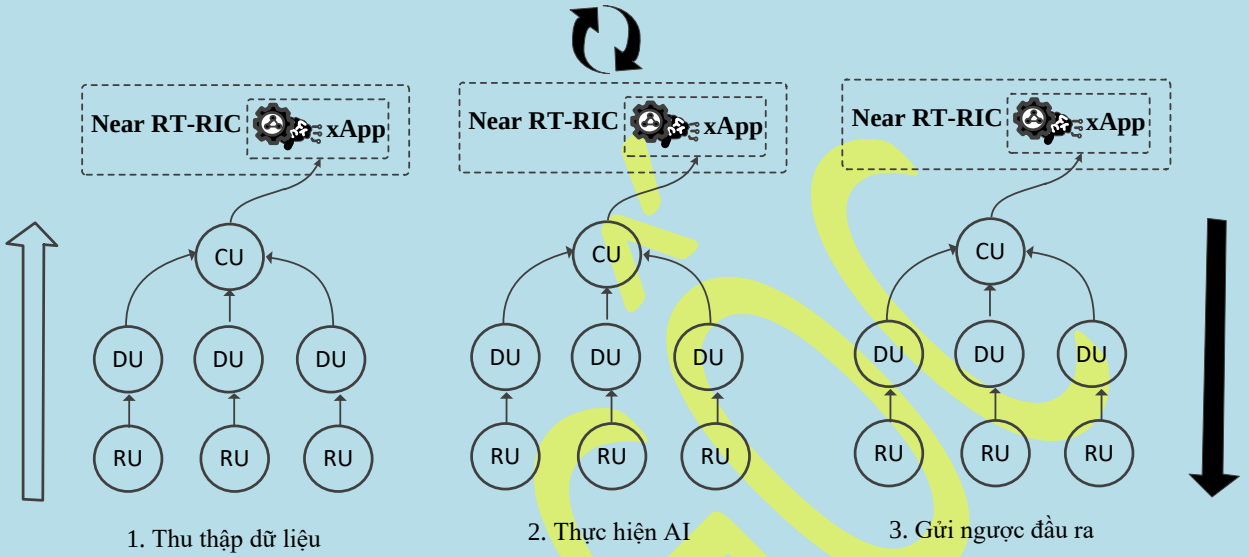
Bằng cách tích hợp DRL vào quá trình giảm thiểu xung đột, khung CME cho phép RIC thích ứng tự động đến các biến chuyển xung đột thời gian chạy, tăng cường năng lực thích nghi điều phối và thúc đẩy các hệ thống O-RAN tự tối ưu và hoàn toàn thông minh.

13.6.4. Không đồng chỉnh thời gian và không nhất quan ngữ cảnh

Hứa hẹn về điều khiển thông minh trong các kiến trúc O-RAN phụ thuộc rất nhiều vào khả năng điều khiển của các tác nhân (chẳng hạn các xApp) để quan xát chính xác trang thái mạng và đáp ứng kịp thời. Tuy nhiên trong thực tế, bản chất phân tán của kiên truc tạo ra một thách thức chủ yếu: không đồng chỉnh gia tăng giữa định thời thu thập dữ liệu đo lường từ xa, quyết định tính toán và thực hiện hành động.

Vòng điều khiển thực hiện trong RIC gần thời gian thực tiến hành theo ba bước (hình 13.19): (i) thu thập đo lường từ xa từ các nút RAN phân tán, (ii) thực hiện suy luận hay logic tối ưu hóa và (iii) phát đi các quyết định ngược về RAN. Trong hai đoạn thứ nhất, các báo cáo đo đạc (chẳng hạn: RSRP (Reference Signal Received Power: công suất thu

tín thêu tham chiếu), sự sử dụng PRB (Physical Resource Block: khối tài nguyên vật lý) và trạng thái UE đi qua các giao diện vật lý và logic từ RU đến DU và CU trước khi đạt được RIC thông qua giao diện E2. Truyền lan này bị các trễ phức tạp và dẫn đến tổng hợp ngữ cảnh có thể bị sai lệch theo thời gian hoặc phân mảnh theo không gian.



Hình 13.19. Minh họa ống thời gian và thách thức trễ tong điều khiển của xApp dựa trên AI. Qúa trình bao gồm ba giai đoạn: (i) Thu thập dữ liệu: các kết quả đo đạc thời gian thực được thu thập từ các RU và các DU và được truyền đến CU và RIC gần thời gian thực, bị trễ đường lên (mũi tên mầu trắng); (ii) Thực hiện AI: xApp xử lý đo lường từ xa thông qua DRL và các giải thuật khác (mũi tên mầu đen), dẫn đến trễ và ứng dụng điều khiển AI: các lệnh điều khiển kết quả được gửi ngược đường xuống qua CU, các DU và cán RU (mũi tiên mầu đen) bổ sung thêm trễ đường xuống. Tại thời điểm các quyết định đạt đến các nút vô tuyến, trạng thái mạng có thể đã thay đổi gây nên các hành động lỗi thời hoặc không tối ưu.

Tại thời điểm bắt đầu giai đoạn hai, các xApp phải hoạt động trên dữ liệu có thể không còn phản ảnh các tính trạng mạng hiện thời. Kết quả là các quyết định được đưa ra dựa trên một các bản sao lưu RAN lỗi thời và chỉ một phần làm giảm sự hợp lệ và thích hợp của chúng. Tại giai đoạn ba, khi các quyết định này được truyền ngược thông qua phân cấp CU/DU/RU cho thực thi, trang thái mạng có thể lại thay đổi làm cho thậm chí các quyết định chính xác không hiệu quả hay hay phản tức dụng.

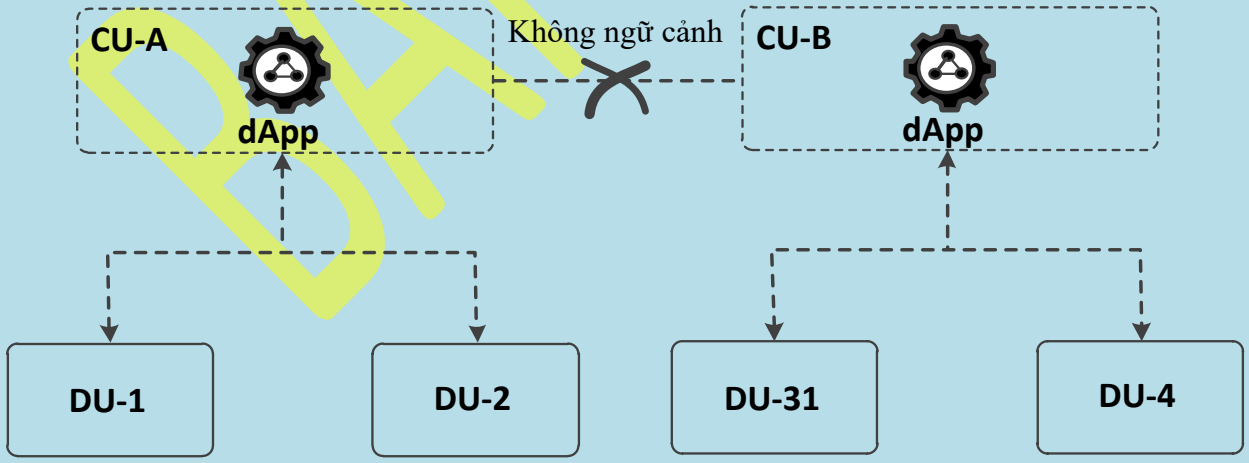
Sự thiếu nhất quán và đồng bộ ngữ cảnh này hạn chế nghiêm trọng tiềm năng điều khiển dựa trên AI trong O-RAN. Nó không chỉ giảm sự hiệu quả của các xAPP cá lẻ mà còn dẫn đến hành vi không nhất quán trên nhiều ứng dụng và các vòng điều khiển hoạt động

trên thông tin trạng thái phân kỳ. Cuối cùng, điều này dẫn đến sự không ổn định của hiệu suất, các đảm bảo QoS bị giảm và hiệu suất mạng bị giảm.

Để xử lý các hạn chế thời điểm và sự phân mảnh nói ở trên, các nghiên cứu đã đưa ra một số đổi mới kiến trúc nhằm cải thiện sự đáp ứng và tính mô-dun hóa trong các hệ thống O-RAN. Một trong số đó là khái niệm các ứng dụng phân tán (dAppS: Distributed Application) nhằm dịch chuyển đưa ra quyết định đến gần mạng vật lý hơn bằng cách trển khai logic điều khiển nhẹ ký trực tiếp trên các CU và các DU.

Bằng việc hoạt động ngay tại chính nút băng gốc, dApps đạt được truy nhập trực tiếp các đo đạc vô tuyến thời gian thực như RSRP, CQI và phản hồi HARQ. Trạng thái gần gũi kiến trúc cho phép chúng tránh được trễ và chi phí gian tiếp của giao diện E2 và RIC gần thời gian thực, cho phép các hành động trễ siêu thấp (điển hình trong gải dưới mi-li-giây) cho các nhiệm vụ nhạy cảm thời gian như thích ứng liên kết, tạo búp và điều khiển động công suất. Về điểm này, dApps là một cơ chế mạnh mẽ cho xử lý các thách thức thời gian thực được mô tả trong hình 13.20.

Tuy nhiên, bất chấp tốc độ và tính địa phương của chúng, dApp về mặt nguyên lý bị giới hạn bởi chính kiến trúc mà kiến trúc này làm cho chúng nhanh hơn: chúng được thiết kế để hoạt động cách ly. dApps được thiết kế trên CU-A, chẳng hạn, chỉ có khả năng nhìn đến các kênh UE và các chính sách lập lịch của cụm nội bộ của chính nó (chẳng hạn: DU-1 và DU-2) nhưng không ttuy nhập được đến dữ liệu hay các quyết định thời gian thực xẩy ra trên các nút lân cân chẳng hạn CU-B (đang quản lý DU- 3 và DU- 4), (hình 13.19).



Hình 13.20. Minh họa việc thiếu chia sẻ ngữ cảnh giữa các dApps trên các CU. Trong thiết lập này, các thực thể dApp tại CU-A và CU-B hoạt động độc lập, với bộ nhớ và đồng bộ không được chia sẻ (thể hiện bởi dấu gạch chéo). Kết quả là các

quyết định được đưa ra bởi từng dApp, chẳng hạn lập lịch hay ấn định tài nguyên trên các DU, chỉ dựa trên một phần thông tin hay thông tin cục bộ. Sự phân mảnh này có thể dẫn đến xung đột điều khiển, các hành vi không nhất quán hayu hiệu suất RAN không tối ưu do các hành động không được phối hợp trên các miền điều khiển.

Như được chỉ ra trên hình 13.20, sự thiếu luồng ngữ cảnh chiều ngang có nghĩa là không có cơ chế cho các dApp trên các CU khác nhau chia sẻ trạng thái RAN hay phối hợp các quyết định. Điều này trở thành vấn đề trong các kịch bản đòi hỏi trí tuệ liên nút:

- **Quản lý di động (Mobility Management)**. Khi một UE chuyển động từ DU-1 (dưới CU-A) đến DU-3 (dưới CU-B), dApp trên CU-A không dự đoán được tải hay các điều kiện tín hiệu của DU-3 dẫn đến khởi động chuyển giao muộn hay không tối ưu.
- **Cân bằng tải (Load Balancing)**. Vì không chia sẽ đo lường từ xa, nên các CU không thể hợp tác phân phối vại lưu lượng, thậm chí nếu các ô lân cận bị quán tải hau không được sử dụng.
- **Điều khiển nhận thức lát căt (Slice-Aware Control)**. Khi một lát cát trải rộng trên nhiều miền CU, dApps không thể thực thi toàn cục QoS hay các ràng buộc trễ không có ngữ cảnh từ các miền đồng cấp.

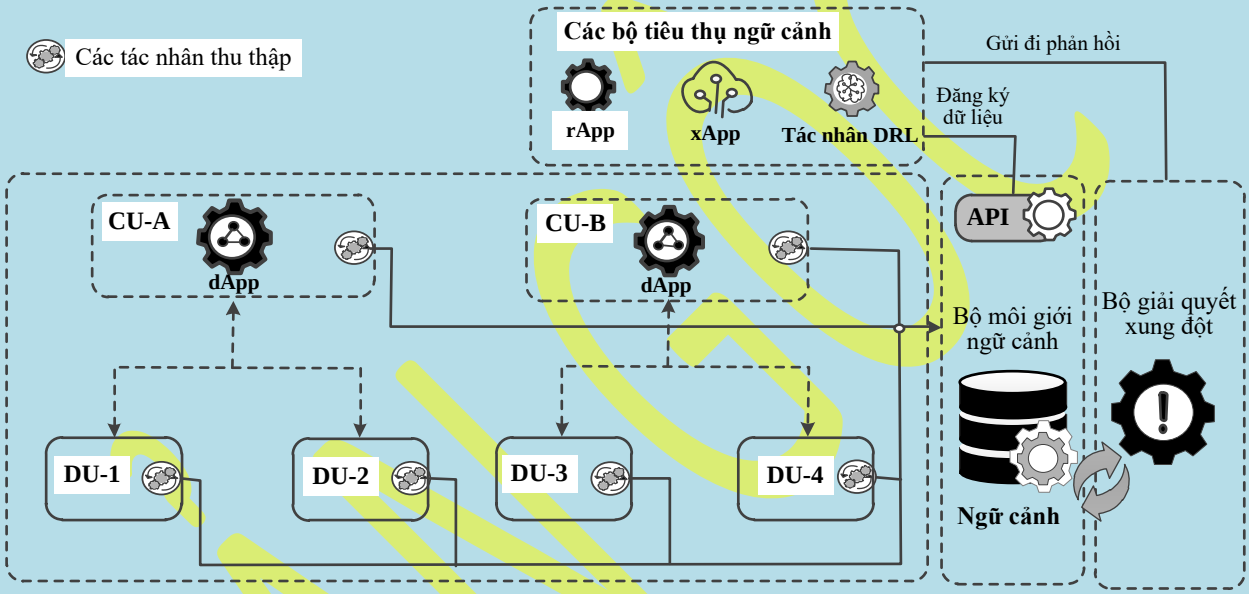
Điều này minh họa một giới hạn cốt lõi: trong khi các dApp thành công trong việc chuyển phát nhanh, điều khiển được địa phương hóa, thì chúng về bản chất lại thất bại trong việc xử lý mọi nhiệm vụ đòi hỏi sự nhận thức ngữ cảnh toàn cục hay đồng bộ giữa các nút. Thiết kế kiến trúc này hy sinh sự hợp tác cho tốc độ, để lại một điểm mù nghiêm trọng của mặt phẳng điều khiển. Giải pháp cho vấn đề này là đưa ra một lớp chia sẻ ngữ cảnh (CSL: Context Sharing Layer). CSL cho phép khắc phục các hạn chế tầm nhìn bị phân mảnh và điều khiển được đồng bộ trong O-RAN. Đây là một kiến trúc phần mềm trung gian được thiết kế để để thống nhất đo lường từ xa và thông tin trạng thái trên các tác nhân điều khiển RAN phân tán. CSL phục vụ như là một lớp đồng bộ được tập trung về mặt logic và còn có thể định cỡ về mặt vật lý, dù hoạt động cục bộ (các dApp) hay toàn cục (các xApp, các rApp), có thể tiếp cận quang cảnh nhất quán và thời gian thực của môi trường RAN.

Tại lõi của nó, CSL tách rời quá trình đưa ra quyết định điều khển khỏi thu thập và phấn phối ngữ cảnh. Thay vì đòi hỏi từng tác nhân thăm dò độc lập hay tổng thợp dữ liệu từ nhiều nguồn, CSL liên tục thu thập, thẩm định, đánh dấu thời gian và để lộ ngữ cảnh RAN thông qua một giao diện dựa trên đăng ký thuê bao. Hạ tầng ngữ cảnh chia sẻ đảm bảo tất cả các tác nhân điều khiển hoạt động trên các đầu vào được đồng bộ để tránh sự

không nhất quán, trễ và thiếu đồng bộ của quyết định do các kiến trúc RAN phân tán gây ra.

13.6.5. Kiến trúc lớp chia sẻ ngữ cảnh (CSL)

Hình 13.21 minh họa một kiến trúc cho phép tích hợp bốn thành phần quan trọng vào một kiến trúc hệ thống cấu trúc mô-đun và khả tương tác được thiết kế cho nhất quán điều khiển RAN thời gian thực.



Hình 13.21. Kiến trúc lớp chia sẻ ngữ cảnh (CSL) để đạt được điều khiển nhất quán ngữ cảnh trong O-RAN. CSL cho phép các tác nhân điều khiển phân tán (gồm các dApp, các xApp và các rApp) hoạt đông trên một tầm nhìn nhất quán và thời gian thực của RAN bằng cách tách rời thu thập ngữ cảnh khỏi quá trình đưa ra quyết định. Các tác nhân thu thập nhẹ ký được triển khai trên các mút CU và DU thu thập dữ liệu đo lường từ xa và dữ liệu điều khiển (chẳng hạn: RSRP, các trạng thái nhớ đệm, các khởi đầu lập lịch) và gửi chúng đến một bộ môi giới ngữ cảnh tập trung. Bộ môi giới này thực hiện đồng chỉnh thời gian, loại bỏ trùng lặp, đánh chỉ số và để lộ ngữ cảnh có cấu trúc đến nhiều tác nhân có đăng ký thông qua các giao diện AI được chuẩn hóa. Mô-dun giải quyết xung đột giám sát các quyết định chồng lấn và các vi phạm chính sách, đảm bảo phối hợp trên các tác nhân. CSL hỗ trợ đồng bộ khả định cỡ và cho phép điều khiển tiên tiên trên cơ sở ML bằng cách duy trì các không gian quan sát nhất quán thông qua RAN.

1) **Các tác nhân thu thập ngữ cảnh (Context Collector Agent)**. Đây là các mô-đun phần mềm nhẹ ký được triển khai tại các vị trí chiến lược của RAN,bao gồm các DU, các CU và các điểm cuối của RAN phi thời gian thực. Vai trò của chúng là liên tục chiết xuất đo lường từ xa và các sự kiện liên quan, như:
    - Các số đo lớp vật lý như: RSRP (Reference Signal Received Power: công suất thu tín hiệu tham chiểu), RSRQ (Reference Signal Received Quality: chất lượng thu tín hiệu tham chiếu) , SINR (Signal to Interference and Noise Ratio: tỷ số tín hiệu trên nhiễu cộng tạp âm)
    - Các trạng thái MAC/RLC (sử dụng
    - Các đầu ra hay các khởi động của dApp địa phương

Mối số đo được ghi chú với một dấu thời gian chính xác, số nhận dạng nút và điểm số tin cậy (nếu áp dụng), sau đó được chuyển đến bộ môi giới ngữ cảnh.

2) **Bộ môi giới ngữ cảnh (Context Brocker)**. Bộ môi giới phục vụ như là trái tim của CSE, hoạt động như là một kiến trúc dữ liệu tạm thời, nó nhập, đánh chỉ số và lưu trữ ngữ cảnh thu thập trong một định dạng được tối ưu hóa cho cả truyền luồng thời gian thực lẫn truy vấn quá khứ. Bộ môi giới đảm bảo:
    - Đồng chỉnh thời gian dữ liệu đo xa trên tất cả các nút
    - Nhất quán dữ liệu thông qua khử trùng lặp và phát hiện xung đột
    - Truy nhập theo yêu cầu và dựa trên sự kiện bởi các tác nhân ngoài thông qua API được chuẩn hóa (REST/gRTC)
3) **Các bộ tiêu thụ ngữ cảnh (Context Consumer)**. Các bộ này bao gồm các dApps, xApps và rApps đăng ký cho các luồng ngữ cảnh thông qua API công cộng của CSC. Các bộ tiêu thu yêu cầu
    - Dữ liệu thô đo lường từ xa thời gian thực (chẳng hạn: RSRP hiện thời cho UE_123)
    - Các dữ liệu thống kê được tổng hợp (chẳng hạn sử dụng PRB trung bình trong ô-A)
    - Xu thế đa nút (chẳng hạn bản đồ nhiệt nhiễu)

    Thông qua việc hoạt động trên dữ liệu được thẩm định và được đồng chỉnh thời gian, các bộ tiêu thụ có thể thực hiện các tối ưu hóa, chẳng hạn các quyết định chuyển giao, cân bằng tải hay cắt lát với ngữ cảnh nhất quán.
4) **Bộ giải quyết xung đột (Conflict Resolver)**. Để cải thiện hơn nữa sự nhất quán điều khiển, CSL tùy chọn có thể có một mô-đun loogic phối hợp để phát hiện và giải quyết các xung đột giữa các quyết định tạo ra bởi nhiều tác nhân điều khiển, cho dù địa phương (dApps) hay tập trung (xApps, rApps). Nó giám sát:

- Các miền điều khiển chồng lấn (chẳng hạn: UE_X bị quản lý đồng thời bởi một dApp và một xApp).
- Các va chạm thời gian (chẳng hạn các cập nhật gần như đồng thời ho một thông số lập lịch).
- Các quy tắc thích ứng dựa trên chính sách (chẳng hạn: ưu tiên QoS hơn so với hiệu suất năng lượng).

Khi phát hiện xung đột, bộ giải quyết có thể khởi xướng một logic trọng tài, thông báo cá tác nhân bị ảnh hưởng hay trì hoãn thực hiện cho đến khi đồng chỉnh đạt được.

Khi được triển khai, CSL hoạt động song song với các luồng công tác điều khiển truyền thống. Chẳng hạn khi chất lượng tín hiệu của một UE giảm sút, một dApp địa phương có thể khởi động một báo cáo cho bộ thu thập ngữ cảnh. Sự kiện này được gửi đến bộ môi giới, được đồng chỉnh với tải và các trạng thái di động hiện thời.Một xApp được đăng ký đến loại các sự kiện này sẽ đánh giá quyết định chuyển giao bằng việc sử dụng các đầu vào thời gian thực và nhất quán, Nếu một tác nhân khác đồng thời đề xuất ân định lại lát cắt, bộ giải quyết xung đội kiểm tra sự tương thích trước khi cả hai hành động được áp dung.

13.6.6. Vấn đề nhận thức môi trường

Mặc dù để mở kiến trúc và khả lập trình được đưa vào bởi mô hình O-RAN, một số phần tử quan trọng của môi trường vô tuyến vẫn nằm ngoài phạm vi điều khiên của RIC phi thời gian thực. Một phần tử như thế là bản thân môi trường truyền sóng, một nhân tố chính ảnh hưởng cường độ tín hiệu, nhiễu và độ tin cậy của đường truyền. Trong khi O-RAN cho phép quản lý linh hoạt các chức năng của BS thông qua xApp chạy trên RIC gần thời gian thực, thì hiện thời nó lại thiếu cơ chế cho thích ứng trực tiếp hay ảnh lưởng lên môi trường không dây thời gian thực.

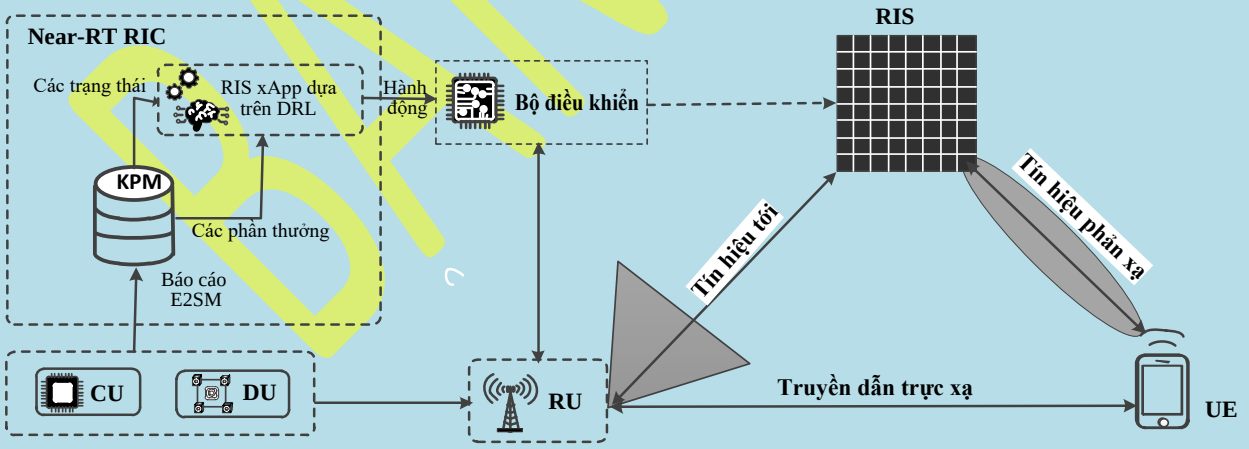
RIS (Reconfigurable Intellegence Surface: bề mặt thông minh khả lập trình) cho thấy một cơ hội đầy hứa hẹn để xử lý hạn chế này bằng việc định hình thụ động các phản xạ tín hiệu để cải thiện phủ sóng, giảm nhiễu và tối ưu hóa hiệu suất phổ tần. Tuy nhiên các hệ thống RIS thông thường được lập cấu hình thông qua các kịch bản thủ công hay các quy tắch tĩnh, không kết nối thời gian thực với các điều kiện mạng thực tế. Sự thiếu kết nối này hạn chế sự đáp ứng và sự hữu dụng thổng thể của chúngtrong các kịch bản RAN năng động.

Ngoài ra, các kiến trúc điều khiển hiện hời không cung cấp cơ chế dựa trên phản hồi để phối hợp hành vi RIS dựa trên các số đo hiệu suất thực tế, như trễ, thông lượng vhay chất lượng tín hiệu. Kết quả là RIS vẫn là một tài sản chưa được tận dụng hết, không có khả năng tham gia vào tối ưu hóa vòng kín cùng với các phần tử RAN khác.

Để khắc phục các hạn chế của điều khiển tĩnh và thích ứng môi trường không được phối hợp trong các hệ thống O-RAN, một đề xuất cho phép tích hợp RIS và khung điều khiển của RIC gần thời gian thực sử dụng học máy tăng cường sâu (DRL: Deep Reiforcement Learning) dựa trên xApp.

Hình 13.22 minh họa kiến trúc điều khiển vòng kín cho RAN được hỗ trợ bởi RIS sử dụng xAPP dựa trên DRL. RIC gần thời gian thực điều hành một tác nhân DRL, nó nhận các số đo hiệusuất của RAN thời gian thực (chẳng hạn thông lượng, trễ, SINR) thông qua mô hình dịch vụ E2SM-KPM từ mô-đun KPM. Dựa trên các quan sát trạng thái này, tác nhân chọn các cấu hình RIS tối ưu và gửi các hành động điều khiển đến một bộ điều khiển RIS dành riêng để lập trình phần cứng RIS. RIS chọn tín hiệu tới để lái hướng hay tăng cường đường truyền vô tuyến, bù phụ vào truyến dẫn thẳng từ RU. Hệ thống này hỗ trợ tuyền hai đường truyền và lập lại cấu hình thích ứng để cải thiện chất lượng liên kết và QoS của người dùng trong các kịch bản không trực xạ (NLOS: Non-Line-Of-Sight).

Kiến trúc này kết nối gBN và RIS thông qua một truyến điều khiển được quản lý bởi RIC gần thời gian thực, trong đó một xAPP tùy chỉnh chạy trên một tác nhân DRL chịu trách nhiệm cho lập cấu hình động RIS để đáp ứng các điều kiện vô tuyến thay đổi.



Hình 13.22. Minh họa kiến trúc điều khiển vòng kín cho RAN được hỗ trợ bởi RIS sử dụng xAPP dựa trên DRL dưới dự điều hành của RIC gần thời gian.

Quá trình bắt đầu bằng thu thập các số đo mạng thông qua giao diện E2 sử dụng mô hình dịch vụ K2SM-KPM được chuẩn hóa. Các số đo này, bao gồm thông tin như thông lượng, trễ và chất lượng liên kết, liên tục được báo cáo cho mô-đun KPM trong RIC. Từ đây, hai luồng dữ liệu quan trọng được chiết suất và được chuyển phát đến DRLxApp:

- **Các trạng thái (Stats)**. Các trạng thái này trình bày các điều kiện được quan sát bởi tác nhận DRL. Một trạng thái có thể gồm các giá trị như cường độ tín hiệu đo được tại UE, trễ đường xuống trung bình hay sơ đồ điều chế và mã hóa (MCS: Modulation and Coding Scheme) hiện đang được gNB sử dụng.
- **Các phần thưởng (Rewwards)**. Các phần thường này được tính toán bằng cách đánh giá thay đổi hiệu suất mạng trước và sau một lập cấu hình RIS được áp dụng. Hàm phần thưởng được thiết kế để ép buộc hành vi mong muốn, chẳng hạn giảm trễ hay cải thiện thông lượng bằng cách ấn định các giá trị dương cho các hành động thành công và các giá trị âm cho các hành động dẫn đến giảm cấp hiệu năng.

Dựa trên trạng thái hiện thời và phần thưởng, tác nhân DRL chọn ra một hành động phù hợp với một cấu hình RIS được định nghĩa trước. Cấu hình này có thể, chẳng hạn đặc tả các dịch pha áp dụng cho từng phần tử của RIS để chọn ra một búp sóng phản xạ theo một phương nhất định. Sau khi hành động đã được chọn, nó được phát đến một bộ điều khiển RIS thông qua một giao diện truyền thông (chẳng hạn: REST API hay kết nối socket). Bộ điều khiển diễn giải lệnh này và áp dụng lập cấu hình trực tiếp cho phần cứng của RIS.

Sự thích ứng động môi trường không dây này cho phép RIS phục vụ như là một bộ phản xạ thông minh tăng cường liên kết đường xuống giữa RU và UE. Chẳng hạn, khi có sự chặn lại tín hiệu (chẳng hạn một bức tường giữa RU và UE)), RIS có thể được lập lại cấu hình để phản xạ tính hiệu vòng qua vật chắn để duy trì một kết nối mạnh và tin cậy. Trong một kịch bản khác, RIS có thể chuyển hướng tín hiệu đến một UE đang chuyển động để bù trừ sự giảm cấp do chuyển động gây ra.

Hệ thống cũng kết hợp cả các hai tín hiệu truyền thẳng và và phản xạ. RU phát các tín hiệu và chúng có thể đạt đến UE qua một đường truyền trực xạ (LOS: Line of Sight) hay gián tiếp được phản xạ bởi RIS. Phân tập hai đường truyền này đặc biệt quan trong trong các điều kiện không trực xạ (NLOS : Non Line Of Sight).

Cần nhấn mạnh rằng kiến trúc nói trên duy trì một sự phân chia rõ ràng giữa logic điều khiển (trong DRL, xApp) và thực hiện phần cứng (trong bộ điều khiển RIS). Sự mô đun hóa nàf đảm bảo rằng các nâng cấp phần cứng RIS hay các thực hiện đặc thù nhà cung

cấp không ảnh hưởng lên quá trình học và đưa ra quyết định của xApp, nó vẫn duy trì trạng thái được ảo hóa tại mức chính sách.

Tổng thể, giải pháp nói trên cho phép RIS trở thành một phần tử tham gia tích cực trong vòng điều khiển của RAN, cho phép điều khiển truyền sóng thông minh, thích ứng và nhận thức môi trường, Điều này không chỉ cải thiện QoS và độ tin cậy cho người sử dụng đầu cuối mà còn thể hiện tính khả thi của vệc tích hợp lập lại cấu hình lớp vật lý trong kiến trúc dựa trên phần mềm của O-RAN.

13.6.6. Vấn đề phiên dịch ý định ngữ nghĩa (Sematic Intent Transation) và dồng chỉnh AI-con người (Human-AI Alignment)

O-RAN thể thiện một sự phát triển đáng kể trong hạ tầng không dây, thúc dẩy sự để mở, phân nhỏ và khả lập trình trên các phần tử truy nhập vô tuyến. Bằng cách tách rời phần cứng khỏi phần mềm và đưa ra các giao diện được chuẩn hóa, O-RAN cho phép phát triển các chức năng điều khiển thông minh thông qua các RIC. Tuy nhiên, vệc tăng khả năng mô-đun và khả năng linh hoạt điều khiển cũng làm cho mức độ phức tạp hệ thống cao hơn, đặc biệt trong diễn dải ý định, phối hợp chính sách, chẩn đoán sự cố và quản lý tài nguyên thích ứng.

Những tiến bộ gần đây trong các mô hình ngôn ngữ lớn (LLM: Large language Model) đã chứng tỏ các khả năng tuyệt vời trong việc lý luận ngôn ngữ tự nhiên (Nature Language Reasoning), tạo mã nguồn và điều phối nhiệm vụ dựa trên mục đích. Các khả năng này đặt LLM vào vị trí như là các công cụ hứa hẹn để hỗ trợ tự động hóa và tương tác nhà mạng trong các miền nối mạng được định nghĩa bằng phần mềm. Trong ngữ cảnh O-RAN, các LLM có thể đóng vai trò trụ cột trong diễn giải mục đích của nhà mạng, tạo loogic điều khiển (chẳng hạn các xAppx và giải thích các hành vi về khía cạnh khả năng đọc được của con người.

Các tiến bộ gần đây trong lĩnh vực viễn thông cho thấy mô quan tâm tích hợp các công nghệ AI , đặc biệt các mô hình ngôn ngữ lớn (LLM), để hỗ trợ quản lý mạng thông minh không ngừng tăng. Các LLM ngày càng được coi là các tác nhân tự chủ có khả năng lập kế hoạch nhiệm vụ, quan sát các trạng thái hệ thống và thực hiện các hành động nhận thức ngữ cảnh. Trong thực tế chúng đã được áp dụng cho đa dạng các trường hợp sử dụng liên quan đến viễn thông, bao gồm tái cấu trúc mã nguồn tự động và thiết kế mạng, tạo cấu hình từ ý định của nhà mạng và tối ưu hóa búp sóng dự đoán. Ngoài ra các LLM đã được tiếp nhận như là các trợ lý tương tác để tăng tốc triển khai và tìm sự cố mạng. Các triển khai này nhấn mạnh tiềm năng không ngừng tăng của LLM như là đa dạng các đầu máy lý luận (Reasoning Engine) trong các môi trường viễn thông tương lai.

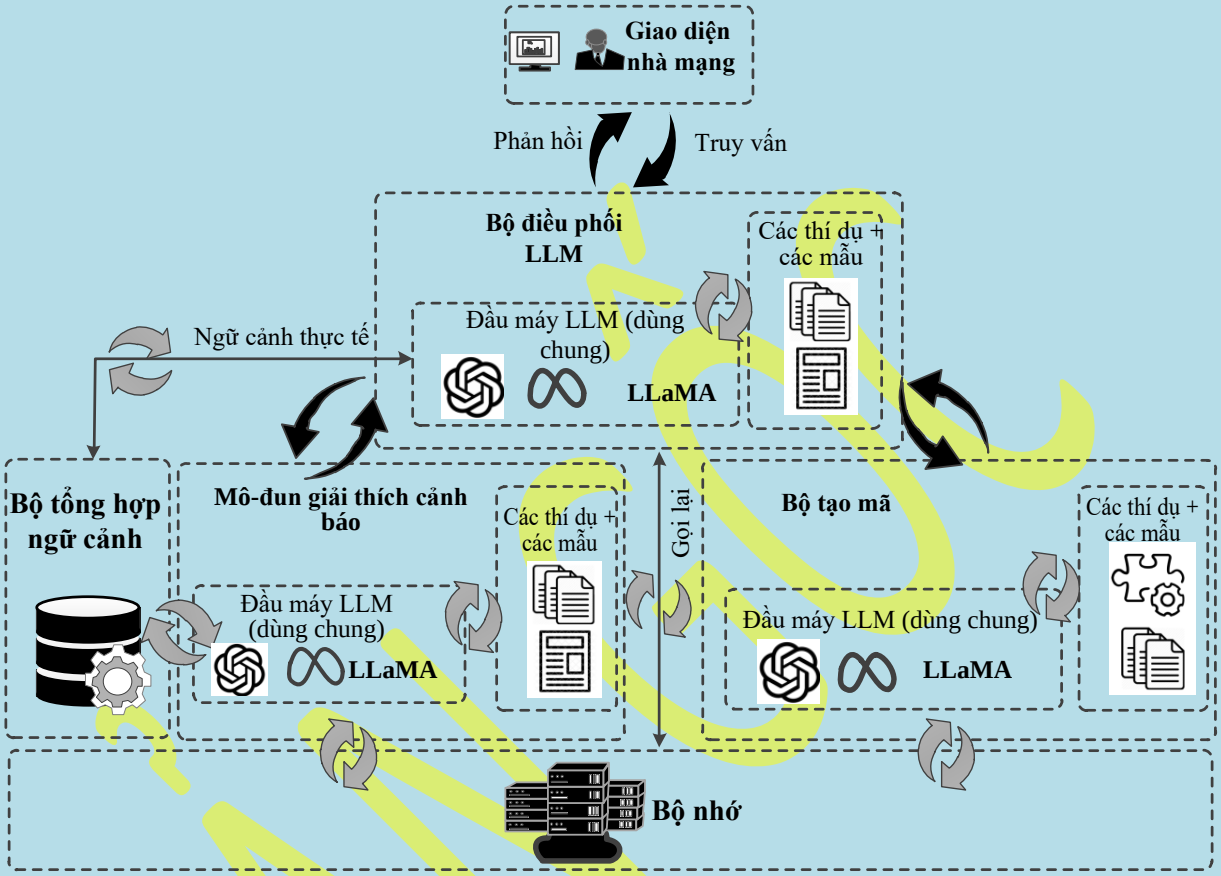
Cho dù hứa hẹn như vậy, việc tích hợp các LLM trong O-RAN vẫn chưa phát triển rộng. Hầu hết các hệ thống hiện thời dựa trên logic theo qua tắc hay các chứ năng được mã hóa tĩnh dẫn đến hạn chế khả năng thích ứng và sự thân thiện nhà mạng. Vẫn còn nhu cầu rõ ràng cho các khung LLM mang tính mô-đun và có khả năng tái ử dụng được thiết kế cho các ràng buộc kiến trúc và các mục đích hoạt động của các RAN hiện đại.

Dưới đây ta sẽ xét các vấn đề liên quan đến sử dụng các LLM cho O-RAN.

1) **LLM cho điều khiển và quản lý tài nguyên O-RAN.** Nghiên cứu gần đây đã chứng tỏ giá trị của việc mở rộng các cơ chế điều khiển dựa trên học máy tăng cường với ngữ cảnh ngữ nghĩa được cung cấp bởi các LLM. Chẳng hạn, một số tác giả đã đề xuất một khung DRL đa tác nhân được tăng cường bởi LLM cho cắt lát mạng năng động. Bằng cách phiên dịch các số đo mạng thô vào các trạng thái ngữ nghĩa mức cao, LLM cải thiên hiệu quả đưa ra quyết định của các tác nhân DRL. Một đề xuất khác tập trung lên các mô-đun lý luận dựa trên lời nhắc (Prompt Based Reasoning) có khả năng hội chẩn sự cố va tổng hợp xApp.
2) **LLM cho an ninh O-RAN**. Trong miền an ninh, một số tác giả đưa ra hệ thống phát hiện sự xâm nhập dựa trên LLM. Cách tiếp cận của họ tích hợp nhiều xApp để phát hiện hành vi bất thường và thi hành các chính sách cắt lát an ninh. Trong khi hệ thống này tập trung lên các mối đe dọa an ninh, thì một số tác giả khác đưa ra kiến trúc an ninh cho phép tổng quát hóa chiến lược suy luận theo mô-đun cho các trường hợp sử dụng điều khiển và điều phối rộng hơn.

Các phương pháp truyền thống dựa trên quy tắc gặp khó khăn lớn để chuyển đổi ý định nhà mạng mức cao vào các chính sách mạng khả thực hiện để hội chẩn và giải thích các tình trạng sự cố phức tạp hoặc tổng hợp logic điều khiển khả triển khai như các xApp và các rApp. Vấn đề này tạo ra một khoảng trông nghiêm trọng trong việc cho phép điều khiển thông minh và thích ứng trong các hệ thống O-RAN.

Để giải quyết các thách thức này, một số tác giả đề xuất kiến trúc suy luận dựa trên LLM theo mo-đun với việc đưa vào một đầu máy LLM (LLM Engine) dung chung tại lõi với ba tác nhân phối hợp: Bộ điều phối LLM (LLM Orchestrator), Mô-đun giải thích cảnh báo (Alert Explanation Module) và Bộ tạo mã (Code Generator). Các mô-đun này phối ghợp để diễn giải đầu và nhà mạng gọi ra ngữ cảnh liên quan và gửi lại các giải thích giầu ngữ nghĩa và logic khả thực hiện, như được minh họa trong hình 13.23.



LLaMA (Large Language Model Meta AI): mô hình ngôn ngữ lớn do Meta phát triển.

Hình 13.23. Kiến trúc lý luận cho điều khiển RAN dựa trên ý định và thích ứng. Hệ thống có một đầu máy LLM (chẳng hạn LLaMA) với ba mô-đun lý luận: bộ điều phối LLM (LLM Orchestrator), mô-đun giải thích cảnh báo (Alert Explanation Module) và bộ tạo mã (Code Generator). Các mô-đun này hợp tác để diễn giải các ý định của nhà mạng, gọi ra ngữ cảnh thực tế thời gian thực từ bộ tổng hợp ngữ cảnh (Context Aggregator) và tạo ra các giải thích hay logic khả triển khai (chẳng hạn các xApp/rApp). Mô-đun nhớ (Memory) đảm bảo tính liên tục của phiên và tinh chỉnh lời nhắc (Prompt). Thiết kế theo mô-đun và nhẹ ký này lấp đây khoảng trống giữa các đầu vào ngôn ngữ tự nhiên và các chính sách thực hiện được trên máy và đưa ra quyết định tự động trong các môi trường O-RAN.

Kiến trúc bao gồm một số phần tử lõi, môi phần tử đóng một vai trò khác nhau trong đường ống (Pipeline) lý luận (Reasoning)và điều ph (Orchestration):

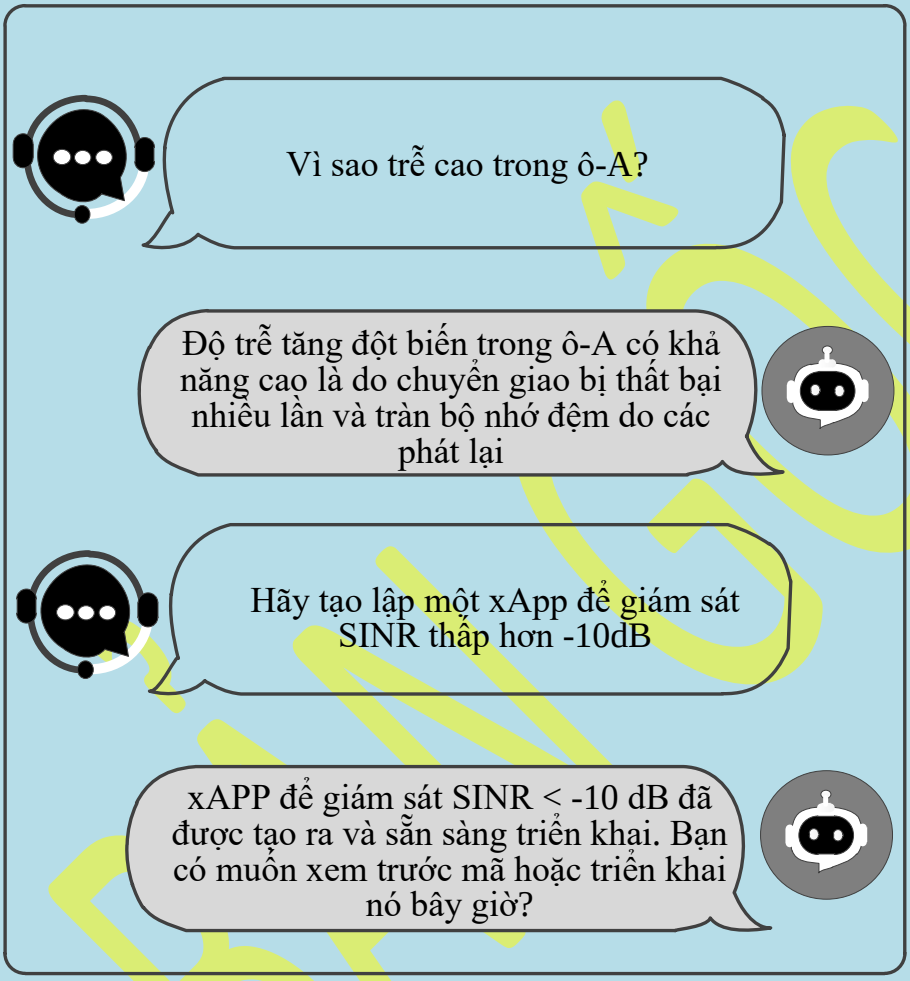
- **Bộ tổng hợp ngữ cảnh (Context Aggregator).** Chịu trách nhiệm thu tập dữ liệu đo lường từ xa thời gian thực và dữ liệu hoạt động. Nó giao diện với Near-RT-RIC trhông qua các giao diện E2 tiêu chuẩn, bao gồm E2SM-KPM cho các số đo hiệu suất (chẳng hạn SIRN, sự sử dụng PRB, trễ) và E2SM-RC cho dữ liệu cấu hình mặt phẳng điều khiển. Nó cũng có thể truy nhập các nhật ký (Logs) và các cảnh báo từ các công cụ có khả năng quan sát mức SMO. Dữ liệu này được cấu trúc vảo các ngữ nghĩa để các mô-đun lý luận có thể truy vấn theo yêu cầu.
- **Gọi lại bộ nhớ (Memory Recal).** Duy trì một lịch sử tương tác nhẹ ký và các dấu vết sự kiện nổi bật. Điều này cho phép sự liên tục của ngữ cảnh trên các phiên, hỗ trợ tránh các truy vấn thừa và hỗ trợ tinh chỉnh lặp các ý định của nhà mạng.
- **Đầu máy LLM (LLM Engine).** Cung cấp tất cả lý luận cực cao. Mô hình ngôn ngữ này (chẳng hạn LLaMA hay Mixtral) xử lý các lời nhắc (Promps) có cấu trúc và gửi trả lại các giải thích ngôn ngữ tự nhiên, các mẫu mã (Code Templates) hay logic chính sách (Po;icy Logic). Nó được dùng chung trên các tác nhân lý luận để đảm bảo tính nhất quán và hiệu quả.
- **Bộ điều phối LLM (LLM Orchestrator).** Phục vụ như là một điểm khởi đầu cho các tương tác của nhà mạng. Nó phân loại ý định đầu vào, giao phó yêu cầu đến mô-đun thích hợp và quản lý luồng đối thoại.
- **Mô-đun giải thích cảnh báo**. Xử lý các truy vấn hội chẩn. Nó thu thập các KPI liên quan từ bộ tổng hợp ngứ cảnh, xây dựng lời nhắc ngữ nghĩa (Sematic Prompt) và truy vấn LLM cho giải thích con người có thể đọc đươc về các bất thường quan sát được.
- **Bộ tạo mã (Code Generator)**. Chịu trách nhiệm cho tổng hợp các xApp hay các rApp khi nhà mạng phát đi các lệnh tạo lập logic (Logic-Creation). Nó phát đi lời nhắc, các tiện ích mở rộng (Plugins) và các mẫu để tạo ra logic điều khiển khả triển khai.

Cùng nhau, các phần tử này tạo thành một đường ống (Pipeline) lý luận cấu trúc mô-đun dựa trên ý định để chuyển đổi đầu vào ngôn ngữ tự nhiên vào logic điều khiển chuẩn đoán và khả triển khai, như được thể hiện trong hình 13.23, kiến trúc này phối hợp nạp dữ liệu, bộ nhớ, lý luận ngữ nghĩa và chuyển phát đầu ra để cho phép điều khiển RAN tự động, có thể giải thích đươc và thích ứng.

Hình 13.24 cho thấy một tương tác mẫu giữa nhà mạng và hệ thống. Khi nhà mạng đưa ra một mục tiêu như “Vì sao trễ cao trong ô A”, Bộ điều phối phân loại điều này như là một yêu cầu chẩn đoán và giao phó nhiệm vụ này cho mô-đun giải thích cảnh báo. Phần tử này giao diện với bộ tổng hợp ngữ cảnh để thu thập các KPI thời gian thực (chẳng hạn: tràn bộ nhớ đệm RLC, các thất bại chuyển giao) và xây dựng một lời nhắc ngữ nghĩa cho đầu máy LLM. Khi này LLM tạo ra một trả lời như “ Độ trễ tăng đột biến trong ô A có khả năng cao là do các chuyển giao bị thất bại nhiều lần và tràn bộ nhớ đệm do các phát lại”.

Nếu thay vào đó nhà mạng yêu cầu “Hãy tạo lập một xApp để giám sát SINR<-10dB”, bộ điều phối nhận dạng điều này như là một nhiệm vụ tổng hợp và định tuyến đầu vào này đến bộ tạo mã. Mô-đun này kết hợp các mẫu lời nhắc, các thư viện tiện ích mở rộng (Plugin) và các quy tắc ngữ nghĩa để tạo ra mã (Code) và các bộ mô tả triển khai (Deployment Descriptors) khả tương

thích với các mô hình dịch vụ E2SM-KPM. Chatbot (Trợ lý ảo dựa trên AI) trả lời bằng một bản tin súc tích, có thể hành động như: “xAPP đã được tạo ra và sẵn sàng cho triển khai. Ban có muốn xem trước mã hay triển khai nó bây giờ”.



Hình 13.24. Luồng tương tác nhà mạng-chatbot minh họa được cho phép bởi kiến trúc dựa trên LLM trong hình 13.23. Nhà mạng đầu tiên đưa ra truy vấn chẩn đoán, truy vấn này được trả lời bằng cách sử dụng ngữ cảnh thời gian thực được truy xuất bởi mô-đun giải thích cảnh báo. Tiếp theo, một yêu cầu tổng hợp chính sách được đưa ra để tạo lập một xApp cho giám sát SINR. Bộ tao mã trả lời bằng một logic điều khiển khả triển khai, đóng lại vòng ý định-giải thích (Intent-Explanation Loop) thông qua ngôn ngữ tự nhiên.

13.7. TỔNG KẾT

AI-RAN (Artificial Intelligence Radio Access Network: mạng truy nhập vô tuyến trí tuệ nhân tạo) là một công nghệ cho phép tích hợp hoàn toàn AI vào phần cứng và phần mềm để hỗ trợ các dịch vụ AI mới.

Cấu trúc cơ bản của AI-RAN được xây dựng dựa trên mục đích chung hoàn toàn thuần nhất, tăng tốc nền tảng tính toán mà không cần bất cứ phần cứng đặc thù nào của RAN, sao cho nó có thể chạy đồng thời khối lượng công việc tổ ong và AI với hiệu suất xác định cho từng khối lượng công việc. Nó chứa đựng các nguyên lý điện toán đám mây gốc (Cloud-Native) như định cỡ theo yêu cầu (On Demand Scaling), kiến trúc đa người thuê (Multi-Tenancy: nhiều người thuê, là một kiến trúc cho phép một phiên bản ứng dụng duy nhất phục vụ nhiều khách hàng với mỗi khách hàng có dữ liệu riêng) và công-ten-nơ hóa cả hai khối lượng công việc.

Phần mềm của AI-RAN được xây dựng dựa trên các nguyên tắc hoàn toàn được định nghĩa bằng phần mềm (Sofware-Defined) để cho phép con-ten-nơ hóa và tăng tốc các khối lượng công việc của AI và RAN nhằm đảm bảo các lợi ích của nền tảng tính toán được tăng tốc.

Kiến trúc tham chiếu của AI-RAN có thể được xây dựng trên các nguyên tắc cơ bản: hiệu suất cao, khả định cỡ và mo-đun trong hội nhập tính toán AI và RAN.

Mạng truy nhập vô tuyến trí tuệ nhân tạo 6G (6G AI-RAN) được mong đợi sẽ cung cấp các tính năng tiên tiến như trễ giảm, băng thông, tốc độ dữ liệu và vùng phủ được cải thiện. Ngoài ra, AI-RAN được kỳ vọng hỗ trợ các trường hợp sử dụng phức tạp như kết nối mức độ cao nhất, thuyền thông đa người dùng và truy nhập phổ động.

AI-RAN sẽ hỗ trợ hiệu quả giảm chi phí và cho phép đưa ra quyết định thông minh hơn cho các mạng 6G.

RAN AI gốc đòi hỏi ba khả năng quan trọng. Thứ nhất là xử lý, tối ưu hóa và tự động hóa RAN được vận hành bởi AI, cho phép điều khiển thông minh trên giao diện vộ tuyến. Thứ hai, quản lý vòng đời (LCM: Lifecycle Management) AI hiệu quả, khả điều khiển và tin cậy, bao gồm thu thập dữ liệu, quản lý mô hình và tính toán hợp tác. Thứ ba là cung cấp AI như là một dịch vụ (AIaaS: AI as a Service), cho phép chuyển đổi RAN vào một nền tảng thống nhất cho truyền thông và các ứng dụng AI.

Bằng cách đón nhận các kiến trúc phân nhỏ, các giao diện được chuẩn hóa và các vòng điều khiển được vận hành bởi AI, O-RAN cho phép triển khai các ứng dụng điều khiển mô-đun (xApps, rApps) trên các bộ điều khiển vô tuyến thông minh (RIC). Liên quan đến phạm vi của xApps và rApps mở rộng trong các RIC, một trong các chức năng quan

trong nhất và phát triển tích cực là ấn định tài nguyên, đang trải qua một sự chuyển dịch vào các hệ thống RAN mở vận hành theo yêu cầu điều khiển thời gian thực, thông minh và hiểu biết bối cảnh được hỗ trợ bởi 6G. Ngoài ra quản lý dựa trên ý định (Intent-Based), được cung cấp nền tảng bởi các mô hình ngôn ngữ lớn (LLM: Large Language Model), đang định nghĩa lại tự động hóa và chuyển đổi chính sách trong các mạng thế đệ sau. Điều này phù hợp với các tầm nhìn kiến trúc rộng hơn của hạ tầng gốc AI trong lộ trình 6G.

Chương này đã xét:

- Kiến trúc tham chiếu AI-RAN với giá máy server và cấu hình topo mạng.
- Ngăn xếp phần mềm AI-RAN nhiều người thuê.
- RAN gốc AI
- Các phương pháp điển hình cho đào tạo, phân phát và triển khai mô hình AI trong 6G
- Các kiên trúc O-RAN gốc AI hương đến thích ứng thời gian thực, giiar quyết xung độ và quản lý tài nguyên hiệu quả
- Khung điều phối AI gốc cho O-RAN được thiết kế để xử lý sự phát triển các yêu cầu của các mạng không dây 6G. Kiến trúc O-RAN được đặt ra trong chương này xử lý các thách thức then chốt về khả năng thích ứng thời gian thực, đồng chỉnh ngữ nghĩa, điều khiển nhạy cảm trễ và phối hợp đa tác nhân. Bằng cách tích hợp các phần tử, như SME, CSL, xApp điều khiển RIS dựa trên DRL và các mô-đun lý luận dựa trên LLM trong phân cấp RIC, chưng đã cho thấy cách thức điều khiển thông minh, mô-đun hóa và khả định cỡ. Hệ thống được xét cho phép thúc đẩy cộng tác cả theo chiều ngang lần tho chiều dọc, hỗ trợ tự động hóa dựa trên ý định và cho phép tối ưu hóa thời gian thực thông qua các vòng quyết định nhận thức môi trường. Khung minh họa tiềm năng kết hợp các tác nhân học có cấu trúc và lý luận LLM để lấp đầy khoảng cách giữa ý định của con người, tạo lập chính sách AI và tự động hóa thời gian thực.

THUẬT NGỮ VÀ KÝ HIỆU

3GPP	Third Generation Parnership Project	Dự án đối tác thế hệ ba
6G AMF/UPF	6G Access Management Function/ User Plane Function	Chức năng quản lý truy nhập/ chức năng mặt phẳng người dùng của 6G
6G NG	6G Next Generation	Thế hệ sau 6G
6gNB	Nút B 6G	Trạm gốc 6G
	Action Tracker	Bộ theo dõi hành động
	Actor-Critic Method	Phương pháp diễn viên-nhà phê bình
AI	Artificial Intelligence	Trí tuệ nhân tạo
AI4NET	AI for NET	AI cho mạng
AIaaS	AI as a Service	AI như là một dịch vụ
	Accelerated Cloud Infrastructure	Hạ tầng đám mây tăng tốc
	Accelerated Libraries	Các thư viện tăng tốc
	Accelerated Linux Bridge	Cầu kết nối Linux tăng tốc
	AI Native Orchestration	Điều phối AI gốc
AI/ML	Artificial Intellegence/ Machine Learning	Trí tuệ nhận tao/ học máy
AI/ML Model LCM	AI/ML Model Life Cycle Managment	Qản lý vòng đời mô hình AI/ML
	AI/ML Function Identification	Nhận dạng chức năng AI/ML
	AI Microservice	Vi dịch vụ AI
	AI-Native O-RAN	O-RAN gốc AI
AI-RAN	Artificial Intellegence Radio Access Network	Mạng truy nhập vô tuyến trí tuệ nhân tạo
	AI-RAN Orchestrator	Bộ điều phối AI-RAN
AI Serverless API	AI Serverless Application Programming Interface	Giao diện lập trình ứng dụng phi AI server
	Anti- Fraud Billing System	Hệ thống phát hiện gian lận thanh toán
	API Cluster Agent	Tác nhân của cụm API
	AI-RAN Alliance	Liên minh AI-RAN
AoI	Area of Inference	Vùng suy luận
API	Application Programming Interface	Giao diện lập trình ứng dụng
	Artificial Universal Intellegence	Trí tuệ nhân tạo toàn năng

BẢN NHÁP

AR	Augmented Reality	Thực tế tăng cường
ASIC	Application Specific Integrated Circuit	Vi mạch đặc thù ứng dụng
	Backhaul	Tuyến sau
BF3 DPU	BlueField 3 Data Processing Unit	Đơn vị xử lý dữ liệu trường sắc xanh
BF3 SuperNIC	Bluefield3 Super Network Interface Card	Card siêu giao diện mạng trường sắc xanh 3
BLER	Block Error Rate	Tỷ lệ lỗi khối
BoI	Bandwith of Inference	Băng thông suy luận
	Brockage-Drine Handover	Chuyển giao nhận thức vật cản
CaaS	Cloud-as-a-Service	Đám mây như là một dịch vụ
CI/CD	Continious Integration/ Continious Deployment	Tích hợp liên tục/ triển khai liên tuc
	Cloud Operating System	Hệ điều hành đám mây
	Cloud Native	Đám mây điện tử gốc
CAPIF	Common API Framework	:Khung API chung
CME	Conflict Management Engine	Đầu máy quản lý xung đột
	Closed-loop Conrol	Điều khiển vòng kín
CNF	Cloud-Native Network Function	Chức măng đám mây điện tử gốc
	Corpus	Kho ngữ liệu
C-plane	Control Plane	Mặt phẳng điều khiển
	Containerized Functions	Các chức năng được con-ten-nơ hóa
	Context Collector Agent	Tác nhân thu thập ngữ cảnh
CQI	Channel Quality Index	Chỉ số chất lượng kênh
CSL	Context Sharing Layer	Lớp chia sẻ ngữ cảnh
C-RAN	Celtralized RAN	RAN tập trung
	Cross Domain AI/ML	AM/ML liên miền
CP	Cyclic Prefix	Tiền tố chu trình
CPU	Celtral Processing Unit	Đơn vị xử lý trung tâm
CQI	Channel Qualiy Indicator	Chỉ số chất lượng kênh
CU	Celtral Unit	Đơn vị trung tâm
CUDA	Compute United Device Architecture	Kiến trúc thết bị tính toán hợp nhất
	Data Domain	Miền dữ liệu
CSI	Channel State Information	Thông tin trạng thái kênh
	Differential Privacy	Quyền riêng tư vi sai

BẢN GỐC

	Distributed Collaborative AI Computing	Tính toán AI phân tán hợp tác
	Digital Divide Gap	khoảng trống phân chia số
DOCA	Datacenter Infrastucture on-a-Chip	Kiến trúc cơ sở hạ tầng trung tâm dữ liệu trên chip
DRB	Dedicated Radio Bearer	Kênh mang vô tuyến dành riêng
D-RAN	Distributed RAN	RAN phân tán
DRL	Deep Renforcement Learning	Học tăng cường sâu
DP	Dynamic Programming	Lập trình động
DPU	Digital Processing Unit	Đơn vị xử lý dữ liệu
DT	Digital Twin	Bản sao số
	DT Entity Management	Quản lý thực thể DT
DU	Distributed Unit	Đơn vị phân tán
Double Dueling DQN	Double Dueling Deep Q-Network	Mạng Q sâu đối đầu nhân đôi
eMBB	Enhanced Mobile Broadband	Băng rộng di động tăng cường
eNA	enabler for Network Automation	Công nghệ cho phép tự động hóa mạng
ETSI	Europeen Telecommunications Standars Institute	Viện tiêu chuẩn viễn thông Châu Âu
FCAPS	Fault, Configuration, Accounting, Performance and Security	Sự cố, cấu hình, thanh toán, hiệu suất và an ninh
FL	Fedeerated Learning	Học liên kết
	Fronthaul	Tuyến trước
	Fronthaul Network	Mạng tuyến trước
	Function Based LCM	LCM dựa trên chức năng
	Garbled Circuits	Kỹ thuật các mạch điện lộn xộn
	Go beyond Data intellegence	Trí tuệ vượt xa hơn dữ liệu
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
GTAA:	Global Telecomm AI Alliance	Liên minh AI viễn thông toàn cầu
HARQ	Hybrid Automatic Repeat Request	Yêu cầu phát lại tự động lai ghép
	Hormomorphic	Mật mã hóa hóa đồng

BẢN NỘP

	Encryption	hình
	Human-AI Alignment	Đồng chỉnh AI-con người
IFFT	Inverse Fast Fourier Transformation	Biến đổi Fourier nhanh đảo
	Immersive Communication	Truyền thông nhập vai
INC	In-Network Computing	Tính toán trong mạng
	Intent-based	Dựa trên ý định
ISAC	Integrated Sensing and Communications	Cảm biến và truyền thông tích hợp
	Knowledge-Guided Reinforcement Learning	Học máy tăng cường dựa trên kiến thức
KPI	Key Performance Indicator	Chỉ số đánh giá hiệu suất chính
KPM	Key Perfomance Measurement	Đo đạc chỉ số hiệu suất chính
LAM	Large Scale AI Model	Mô hình AI quy mô lớn
	Leaf Switch	Chuyển mạch lá
LMF	Location Management Function	Chức năng quản lý vị trí.
LCM	Life Cycle Management	Quản lý vòng đời
	Linux Foudation	Quỹ Linux
LLM	Large Language Model	Mô hình ngôn ngữ lớn
MAC	Medium Access Control	Điều khiển truy nhập môi trường)
	Management Domain	Miền quản lý
MAE	Mean Abssolute Error	Sai số tuyệt đối trung bình
MARL	Multi-Agent Reinforcement Learning	Học tăng cường đa tác nhân
MCS	Modulation and Coding Scheme	Sơ đồ điều chế và mã hóa
MDT	Minimization of Drive Test	Giảm thiểu kiểm tra trên xe
	Meassurement Reports	Các báo cáo đo đạc
	Midhaul	Tuyến giữa
MIG	Multi-Instance GPU	GPU nhiều thực thể
mMTC	massive Machine-Type Communications	Truyền thông kiểu máy số đông
	Model Domain	Miền mô hình
	Model Switching Technique	Kỹ thuật chuyển đổi mô
MR	Mixied Reality	Thực tế pha trộn

BẢN GỐC

MSE	Mean Squared Error	Sai số bình phương trung bình
M-plane		Mặt phẳng M
mMIMO	massive Multi-Input Multi-Output	Anten nhiều đầu vào nhiều đầu ra kích thước lớn
	Native Intellegence	Trí tuệ gốc
NET4AI	NET for AI	Mạng cho AI
	Network Application Layer	Lớp ứng dụng mạng
	Network Digital Tween Layer	Lớp bản sao số của mạng
NWDAF	Network Data Analytics Function	Chức năng phân tích dữ liệu mạng
	Native AI	AI gốc
Near-RT-RIC	Near-Real Time-Radio Intellegence Controller	Vộ điều khiển vô tuyến thông minh gần thời gian thực
	Network Digital Twin	Bản sao số của mạng
NF	Network Function	Chức năng mạng
NeMo	NVIDIA neural Module	Mô đun tế bào thần kinh của MVIDEA
NIC	Network Interface Card	Card giao diện mạng
NIC FW	Network Interface Card Firmware	Phần mềm nhúng của card giao tiếp mạng
	ConnectX7 NIC	Card giao tiếp mạng ConnectX7
NIM	NVIDIA Inference Microservice	Vi dịch vụ suy luận NVIDIA
Non-RT-RIC	Non-Real Time-Radio Intellegence Controller	Bộ điều khiển vô tuyến thông minh phi thời gian thực
NWDAF	Network Data Analytíc Function	Chức năng phân tích dữ liệu mạng
	NVIDEA Spectrum-X technology	Công nghệ dòng X ccủa NVIDEA
	Operating System	Hệ điều hành
OAM	Operation and Maintenance	Khai thác và bảo dưỡng
	Oblivious Transfer	Kỹ thuật chuyển giao vô thức
O-CU-UP	Open-Celtral Unit-User Plan	Mặt phẳng người dùng đơn vị trung tâm mở
O-DU-UP	Open-Distributed Unit-	Mặt phẳng người dùng

	User Plan	đơn vị phân tán mở
ONAP	Open Network Automation Platform	Nền tảng tự đông hóa của mạng mở
O-RAN	Open Radio Access Network	Mạng truy nhập vô tuyến mở
O-RAN FH	O-RAN Fronthaul	Tuyến trước O-RAN
OTT	Over the Top	Ứng dụng và nền tảng cung cấp nội dung
QI	Quality Indicator	Chỉ số chất lượng
QoAIS	Quality of AI Service	Chất lượng dịch vụ AI
QoS/QoE	Qualitiy of Service/ Quality of Experience	Chất lượng dịch vụ/ chất lượng trải nghiệm
PDCP	Packet Data Convergence Protocol	Giao thức hội tụ dữ liệu gói
	Physical Network Layer	Lớp mạng vật lý
PDU	Protocol Data Unit	Đơn vị dữ liệu giao thức
PM	Performance Meassurements	Các đo đạc hiệu suất
Publish-Sunscribe		Công bố-đăng ký
	Recall	Độ thu hồi
	Symbolic Resonning	Lý luận biểu tượng
RIS	Reconfigurable Intellegence Surface	Bề mặt thông minh khả lập cấu hình
RLC	Radio Link Control	Điều khiển liên kết vô tuyến
RMSE	Root Mean Squared Error	Sai số bình phương trung bình gốc
R-NIB	RAN Network Information Base	Cơ sở thông tin mạng RAN
RoCE	Remote Direct Memory Access over Converged Ethernet	Truy nhập bộ nhớ trực tiếp từ xa trên ethernet hội tụ
RRC	Radio Resource Control	Điều khiển tài nguyên vô tuyến
RRM	Radio Resource Management	Quản lý tài nguyên vô tuyến
RS	Reference Signal	Tín hiệu tham chiếu
RSRP	Reference Signal Received Power	Công suất thu tín thêu tham chiếu
RSRQ	Reference Signal Received Quality	Chất lượng thu tín hiệu tham chiếu
PRB	Physical Resource Block	Khối tài nguyên vật lý
PTP GM:	Precision Time Protocol	Đồng hồ lớn/chủ giao thức thời gian chính xác

BẢN GỐC

	Grand/ Master	
RU	Radio Unit	Đơn vị vô tuyến
SBA	Service Based Archtecture	Kiến trúc dựa trên dịch vụ
SDAP	Service Data Adaptation Protocol	Giao thức thích ứng dữ liệu dịch vụ
	Sharing Secre	Chia sẻ bí mật
SDO	Standard Development Organization	Tổ chức phát triển tiêu chuẩn
SMPC	Sercure Multi-Party Computation	Tính toán nhiều bên an ninh
SDK	Sofware Development Kit	bộ công cụ phát triển phần mềm
SI	Semantic Information	Thông tin ngữ nghĩa
	Sematic Intent Transation	Phiên dịch ý định ngữ nghĩa
SINR	Signal to Interference and Noise Ratio	Tỷ số tín hiệu trên nhiễu cộng tạp âm
SLA	Service Level Agreement	Thỏa thuận mức dịch vụ
	Slice-Aware Control	Điều khiển nhận thức lát cắt
SM	Service Model	Mô hình dịch vụ
SMO	Service Management and Orchestration	quản lý và điều phối dịch vụ
SL	Split learning	Học học phân chia
	Spine Switch	Chuyển mạch sương sống
S-plane	Synchronization Plane	Mặt phẳng đồng bộ
	Specrum-X Networking Platform	Nền tảng nối mạng dòng X
	Spectrum-4	Chuyển mạch Spectrum-4
SSD	Solid-State Drive	Ổ cứng thể rắn
TCO	Total Cost of Onership	Tổng chi phí quyền sở hữu
TCP	Transmission Control Protocol	Giao thức điều khiển truyền dẫn
	Transfer Learning	Học chuyển giao
TTFT	Time to First Token	Thời gian phản hồi token đầu tiên
	Virtual Machine Hypervisor	Phần mềm giám sát máy ảo
UE-NIB	UE Network Information Base	Cơ sở thông tin mạng UE
	Uniffied Repossitory	Kho dữ liệu thống nhất
U-plane	User Plane	Mặt phẳng người dùng

BẢN GỐC

UPF	User Plane Function	Chức năng mặt phẳng người dùng
UPF LBO	User Plane Function Local Breakout	Đột phá tại chỗ chức năng mặt phẳng người dùng
UPT	Units Per Transacstion	Số lượng đơn vị trên một giao dịch
URLLC	Ultra reliable and Low lattency Communication	Truyền thông siêu tin cậy và trễ thấp
vDU	virtual Distributed Unit	Đơn vị phân tán ảo
vCU	virtual Central Unit	Đơn vị trung tâm ảo
VR	Virtual Reality	Thực tế ảo
XR	Extended reality	THực tế mở rộng bao gồm: VR,AR,MR
	Zero-Tuch Automation	Tự ffoongj hóa không chạm

TÀI LIỆU THAM KHẢO

1. Nguyễn Phạm Anh Dũng, "5G và lộ trình phát triển lên 6G". Nhà xuất bản Thông tin và Truyền thông, 12/2022
2. Nguyễn Phạm Anh Dũng, **"**IoT (Internet vạn vật): *Kiến trúc IoT, IoT công ghiệp và công nghiệp 4.0, IoT tổ ong"*. Nhà xuất bản Thông tin và Truyền thông, 12/2022
3. Nguyễn Phạm Anh Dũng, **"**Truyền thông vệ tinh và truyền thông 3D trong các hệ thống 5G, 6G**".** Nhà xuất bản Thông tin và Truyền thông, 2/2024
4. Nguyễn Phạm Anh Dũng "Tích hợp RIS vào các mạng truyền thông không dây 5G và 6G". Sách điện tử, 4/2025
5. Vivienne Sze and others "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". arXiv:1703.09039v2 [cs.CV] 13 Aug 2017
6. J. Woodhouse "Big, big, big data: higher and higher resolution video surveillance". Technology.ihs.com, January 2016.
7. Jasneeet Kaur and Others,"Machine Learning Techniques for 5G and Beyond". IEEE Access, 2021
8. Yiping Kang and others "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge". ACM. ISBN, 2017
9. Chen Sun and others "On the Combination of AI and Wireless Technologies: 3GPP Standardization Progress". arXiv:2407.10984v1 [cs.NI] 17 Jun 2024
10. Omar Nassef and others "A survey: Distributed Machine Learning for 5G and beyond". Computer Networks 207 (2022) 108820

11. Maria Vakalopoulou and others "Deep learning: basics and convolutional neural networks (CNN)". HAL. 2023
12. Tsung-Yi Lin and others "Feature Pyramid Networks for Object Detection". arXiv:1612.03144v2 [cs.CV] 19 Apr 2017
13. 3GPP TS 23.288 "Architecture enhancements for 5G system (5GS) to support network data analytics services" V17.8.0, March 2023
14. 3GPP TR 37.817 "Study on enhancement for data collection for NR and EN-DC" V17.0.0, April 2022
15. 3GPP TS 28.533 "Management and orchestration; Architecture framework" V17.2.0, March 2022
16. 3GPP TR 22.874 "Study on traffic characteristics and performance requirements for AI/ML model transfer" V18.2.0., December 2021
17. 3GPP TS 22.261 "Service requirements for the 5G system" V18.6.0, March 2022
18. 3GPP TR 23.700-80, "Study on 5G system support for AI/MLbased services" V18.0.0, December 2022
19. 3GPP TR 33.898 "Study on security and privacy of AI/ML-based services and applications in 5G" V0.6.0., May 2023
20. 3GPP TR 26.927 "Study on artificial intelligence and machine learning in 5G media services" V0.3.1, February 2023
21. 3GPP TS 28.105 "Artificial intelligence/machine learning (AI/ML) management" V17.3.0., March 2023
22. 3GPP TR 28.908, "Study on artificial intelligence/machine learning (AI/ML) management," V0.2.1., March 2023
23. RP-213602 "Artificial intelligence (AI)/machine learning (ML) for NG-RAN," 3GPP TSG RAN Meeting #94e, Dec. 2021
24. 3GPP TR 38.843 "Study on artificial intelligence (AI)/machine learning (ML) for NR air interface" V0.0.0., June 2022
25. Huaijiang Zhu, Manali Sharma, Kai Pfeiffer, Marco Mezzavilla, Jia Shen, Sundeep Rangan, and Ludovic Righetti, "Enabling Remote Whole-body Control with 5G Edge Computing", to appear, in Proc. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. Available at: https://arxiv.org/pdf/2008.08243.pdf
26. B. Taylor, V. S.Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive deep learning model selection on embedded systems," in Proc. ACM LCTES, 2018, pp. 31–43.
27. Technical specification group radio access networks; Study on artificial intelligence (AI)/machine learning (ML) for NR air interface (Release 18), Std. 3GPP TR 38.843 V0.0.0, 2022.
28. Technical specification group radio access networks; Study on artificial intelligence (AI)/machine learning (ML) for NR air interface (Release 18), Std. 3GPP TR 38.843 V0.0.0, 2022.
29. 3GPP TR 38.843, "Study on artificial intelligence (AI)/machine learning (ML) for NR air interface," V0.1.0., June 2023

30. Cheng-Xiang Wang and Others, “Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges”. ArXive 2001

31. Abdelfatteh Haidine and Others, “Artificial Intelligence and Machine Learning in 5G and beyond: A Survey and Perspectives”. IntechOpen, 2021

32. Yang H, Alphones A, Xiong Z, et al. Artificial-intelligence-enabled intelligent 6G networks. IEEE Netw, 2020

33. Jianjun Wu and Others, “Toward Native Artificial Intelligence in 6G Networks: System Design, Architectures, and Paradigms”. ArXive, 2021

34. Alexandros-Apostolos A. Boulogeorgos, “AI and ML – Enablers for Beyond 5G Networks”. 5G PPP Technology Board, 2021

35. Jakob Hoydis and Others,“Toward a 6G AI-Native Air Interface”. ArXive, 2021

36. Abdulrahman Yarali, “From 5G to 6G : technologies, architecture, AI, and security”. John Wiley & Sons, 2023

37. Dr. Irene Vilà Muñoz, “Reinforcement Learning for 5G and beyond radio access networks: from design to implementation”. EuCNC & 6G Summit. 2023

38. Michele Polese, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges”. U.S. National Science Foundatio, 2023

39. Xingqin Lin , “An Overview of the 3GPP Study on Artificial Intelligence for 5G New Radio”. NVIDIA, 2023

40. Xingqin Lin , “Artificial Intelligence in 3GPP 5G-Advanced: A Survey”. NVIDIA, 2024

41. Naveed Ali Khan and Others, “AI-RAN in 6G Networks State-of-the-Art and Challenges”. Supported by 6G-RIC and funded by the German Federal Ministry of Education and Research, 2023

42. Lopamudra Kundu and Others , “AI-RAN: Transforming RAN with AI-driven Computing Infrastructure”, NVIDIA, 2025

43. Robin Chataut and Others, “6G Networks and the AI Revolution-Exploring Technologies, Applications, and Emerging Challenges”. Sensors 2024

44. Swatilakshmi Boppana and Others, “A Machine Learning Approach in Communication 5G, 6G Network”. Journal of Theoretical and Applied Information Technology, May 2024

45. Abdulraqeb Alhammadi and Others, “Artificial Intelligence in 6G Wireless Networks: Opportunities, Applications, and Challenges”. International Journal of Intelligent Systems, volume 2024.

46. Qimei CUI and Others, “Overview of AI and communication for 6G network: fundamentals, challenges, and future research opportunities”. Science China Information Sciences, February 2025

47. Nouri Omheni and Others,“Artificial Intelligence for 5G and 6G Networks: A Taxonomy-Based Survey of Applications, Trends, and Challenges". MDPI, 2025

48. NVIDIA, “AI-RAN: Artificial Intelligence-Radi Access Networks”. 2025

49. Sifeddine Salmi and Others, “AI-Native O-RAN Architectures for 6G: Towards Real-Time Adaptation, Conflict Resolution, and Efficient Resource Management”. 2025

50. Nan Li and Others, “Towards AI-Native RAN: An Operator’s Perspective of 6G Day 1 Standardization”. ArXiv, 6/2025

51. Pradnya Taksande and Others, “AI/ML in 3GPP 5G Advanced- Services and Architecture”. ArXiv, 12/2025

BẢN GỐC