TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN ANH ĐẮC : 19133020
NGUYỄN THANH TÂN KỶ : 19133031

Đề Tài:

TÌM HIỂU VỀ FEDERATED LEARNING VÀ ỨNG DỤNG CỦA NÓ

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

ThS. QUÁCH ĐÌNH HOÀNG

TP. HCM, ngày tháng năm 2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN ANH ĐẮC : 19133020
NGUYỄN THANH TÂN KỶ : 19133031

Đề Tài:

TÌM HIỂU VỀ THUẬT TOÁN FEDERATED LEARNING VÀ ỨNG DỤNG CỦA NÓ

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

ThS. QUÁCH ĐÌNH HOÀNG

TP. HCM, ngày Tháng năm 2022

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên 1: Nguyễn Anh Đắc MSSV: 19133020

Họ và tên sinh viên 2: Nguyễn Thanh Tân Kỷ MSSV: 19133031

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu về thuật toán Federated Learning và ứng dụng của nó.

Họ và tên giáo viên hướng dẫn: ThS.Quách Đình Hoàng

NHẬN XÉT:

1. Về nội dung và đề tài khối lượng thực hiện:

.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp.Hồ Chí Minh, ngày...tháng...năm 2022

Giáo viên hướng dẫn

Ký & ghi rõ họ tên

LỜI CẢM ƠN

Trong quá trình nghiên cứu đề tài nghiên cứu, các giảng viên đã luôn hỗ trợ, hướng dẫn sinh viên một cách hết sức là nhiệt tình và chu đáo, với tất cả sự kính trọng, chúng tôi xin được bày tỏ lòng biết ơn sâu sắc đến Thầy, Cô vì đã luôn theo dõi và hướng dẫn trong suốt thời gian thực hiện đề tài.

Đầu tiên, chúng tôi xin gửi lời cảm ơn sâu sắc nhất đến Ban giám hiệu trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo điều kiện, môi trường học tập, cơ sở vật chất chất lượng và hiệu quả để chúng tôi có thể phát huy một cách tốt nhất việc nghiên cứu đề tài.

Đồng thời, chúng tôi xin gửi lời cảm ơn đến Ban chủ nhiệm khoa Công nghệ Thông tin và các Thầy, Cô khoa Công nghệ Thông tin - Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo ra một môi trường học tập và làm việc chuyên nghiệp, nhiệt tình trong phương pháp giảng dạy để chúng tôi có thể thực hiện tốt đề tài nói riêng và sinh viên trong khoa Công nghệ Thông tin nói chung trong quá trình học tập và làm việc tại trường.

Đặc biệt, chúng tôi xin gửi lời cảm ơn chân thành nhất đến Thầy Quách Đình Hoàng – Giáo viên hướng dẫn tiểu luận chuyên ngành – Khoa Công nghệ Thông tin – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, đã hướng dẫn, quan tâm, góp ý và luôn đồng hành cùng chúng tôi trong những giai đoạn khó khăn nhất của việc nghiên cứu đề tài.

Tuy nhiên vì thời gian hoàn thành đề tài ngắn, nên đề tài khó lòng tránh khỏi những sai sót và hạn chế nhất định. Kính mong nhận được những phản hồi, đóng góp ý kiến và chỉ bảo thêm từ Quý Thầy Cô, để chúng tôi có thể đạt được những kiến thức hữu ích, nâng cao trình độ để phục vụ cho sự nghiệp sau này.

Xin chân thành cảm ơn!

KẾ HOẠCH THỰC HIỆN

Tuần	Thời gian	Nội dung công việc	Ghi chú
Tuần 2-3	28/8 - 11/9	Lựa chọn và xác định đề tài tiểu luận chuyên ngành	
Tuần 4-5	12/9 - 25/9	Tìm hiểu sơ lược về đề tài	
Tuần 6	26/9 - 2/10	Tìm hiểu sơ lược về mô hình của federated learning và so sánh với các loại mô hình học máy khác	
Tuần 7	3/10 - 9/10	Tìm hiểu về các loại và kiến trúc của federated learning	
Tuần 8	10/10 - 16/10	Tìm hiểu về hoạt động và đặc điểm	
Tuần 9-10	17/10 - 30/10	Tìm hiểu về thuật toán FedAvg	
Tuần 11	31/10 - 6/11	Tìm hiểu lĩnh tình hình phát triển và ứng dụng của federated learning	
Tuần 12	7/11 - 13/11	Tìm hiểu cách triển khai mô hình federated learning	
Tuần 13	14/11 - 20/11	Chọn tập dữ liệu và tìm hiểu thuật toán CNN	
Tuần 14-15	21/11 - 4/12	Triển khai mô hình	
Tuần 16	5/12 - 11/12	Đánh giá mô hình và viết app nhận dạng chữ viết tay.	
Tuần 17-18	12/12 - 25/12	Viết báo cáo và powerpoint	

MỤC LỤC

DANH SÁCH HÌNH ẢNH

CHƯƠNG 1: MỞ ĐẦU

1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Lĩnh vực trí tuệ nhân tạo đang phát triển nhanh chóng. Chỉ mới 8 năm kể từ khi kỷ nguyên hiện đại của học sâu bắt đầu tại cuộc thi ImageNet năm 2012. Sự tiến bộ trong lĩnh vực này kể từ đó thật ngoạn mục và không ngừng phát triển.

Không chỉ đã phát triển với tốc độ nhanh, mà chúng còn đang tăng tốc từng ngày. 5 năm tới, lĩnh vực AI sẽ còn trông rất khác so với hiện nay. Các phương pháp hiện được coi là tiên tiến sẽ trở nên lỗi thời; các phương pháp ngày nay mới ra đời hoặc đang ở ngoài rìa sẽ là xu hướng chủ đạo.

Một trong những thách thức bao trùm của kỷ nguyên kỹ thuật số là quyền riêng tư của dữ liệu. Vì dữ liệu là mạch máu của trí tuệ nhân tạo hiện đại, các vấn đề về quyền riêng tư của dữ liệu đóng một vai trò quan trọng (và thường là giới hạn) trong quỹ đạo của AI. Trí tuệ nhân tạo bảo vệ quyền riêng tư – các phương pháp cho phép các mô hình AI học hỏi từ bộ dữ liệu mà không ảnh hưởng đến quyền riêng tư của chúng – do đó nó ngày càng trở thành mục tiêu theo đuổi quan trọng. Có lẽ cách tiếp cận hứa hẹn nhất để bảo vệ quyền riêng tư của AI là giải pháp học liên kết hay Federated Learning.

Cách tiếp cận tiêu chuẩn để xây dựng mô hình học máy ngày nay là tập hợp tất cả dữ liệu đào tạo ở một nơi, thường là trên đám mây, và sau đó đào tạo mô hình trên dữ liệu. Nhưng cách tiếp cận này không khả thi đối với phần lớn dữ liệu trên thế giới, vì lý do riêng tư và bảo mật cho dữ liệu của người dùng.

Được thúc đẩy bởi các mối quan tâm về quyền riêng tư và tầm nhìn của học sâu, bốn năm qua đã chứng kiến sự thay đổi mô hình trong cơ chế ứng dụng của học máy (Machine Learning - ML). Một mô hình mới nổi, được gọi là **Học Liên Kết (Federated Learning - FL)**, đang vượt lên trên cả hệ thống tập trung (centralized systems) và phân tích tại chỗ (on-site analysis), trở thành một thiết kế kiểu mới cho việc triển khai ML. Đây là một cách tiếp cận phi tập trung bảo vệ quyền riêng tư, giữ dữ liệu thô trên các thiết bị và liên quan đến việc đào tạo ML trên các thiết bị cục bộ. Sau đó, một liên kết giữa các mô hình đã được đào tạo sẽ được chia sẻ và tổng hợp lại tại máy chủ trung tâm, tiếp đó sẽ trả lại kết quả cuối cùng cho các người tham gia. Chúng tôi bắt đầu bằng cách kiểm tra và so sánh các kiến trúc triển khai dựa trên ML khác nhau, tiếp theo là điều tra sâu và rộng về FL. Trong bối cảnh này, chúng tôi xây dựng các phân loại toàn diện bao gồm các khía cạnh thách thức, đóng

góp và xu hướng khác nhau trong tài liệu, bao gồm các mô hình và thiết kế hệ thống cốt lõi, các lĩnh vực ứng dụng, quyền riêng tư và bảo mật cũng như quản lý tài nguyên. Hơn nữa, chúng tôi thảo luận về những thách thức quan trọng và mở ra các hướng nghiên cứu hướng tới các hệ thống FL mạnh mẽ hơn.

1.2. MỤC TIÊU VÀ NHIỆM VỤ NGHIÊN CỨU

Mục tiêu của đề tài là tập trung nghiên cứu cơ sở lý thuyết của bài toán học liên kết (Federated Learning), các ứng dụng liên quan đến thuật toán và từ đó khai thác chiều sâu của bài toán cũng như những thuật toán được sử dụng trong bài toán đó. Trong đề tài này, chúng tôi muốn xây dựng một mô hình phân tích để có thể phân tích và nhận dạng được các chữ số viết tay. Để đạt được những điều đó, điều đầu tiên chúng tôi cần tìm hiểu một số vấn đề sau:

- Tìm hiểu cơ sở lý thuyết của bài toán Federated Learning.
- Tìm hiểu các thuật toán liên quan đến bài toán như Gradient Descent, Stochastic Gradient Descent (SGD), Federated Averaging (FedAVG) để xử lý, tính toán và sau đó sẽ phân tích thuật toán để chọn phương pháp phù hợp để sử dụng cho bài toán.
- Ứng dụng bài toán vào một tập dữ liệu cụ thể để trực quan hóa bài toán.
- Đánh giá và giải thích kết quả đạt được.

1.3. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

Bài toán phân tích và nhận dạng chữ số viết tay cách tiếp cận phổ biến nhất là sử dụng mô hình học sâu. Với các thuật toán học máy có giảm sát như Decision Trees, Logistic Regression, CNN,.. Ở hướng tiếp cận các chữ số viết tay, thì mỗi người là có một kiểu chữ kiểu viết nên muốn có thể có sự chính xác tốt nhất vẫn phải tùy thuộc vào bộ dữ liệu đang có, có đủ tốt hay không, vì thế cần bổ sung dữ liệu nhiều hơn để tăng tỷ lệ và cải thiện mô hình tốt hơn.

1.4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC

Nhóm chúng tôi mong muốn sau khi thực hiện quá trình nghiên cứu nhiều công trình cũng như các ứng dụng từ các tác giả đi trước, nhóm có thể học hỏi và đúc kết thành một bài báo cáo khai sâu về nội dung lý thuyết của federated learning và triển khai vào bài toán nhận dạng chữ số viết tay, các cách phân tích, xử lý cho bài toán.

Về phần ứng dụng, để trực quan hóa bài toán nhóm sẽ xây dựng một mô hình đơn giản để trực quan hóa kết quả sau khi phân tích từ tập dữ liệu nhằm có cái nhìn

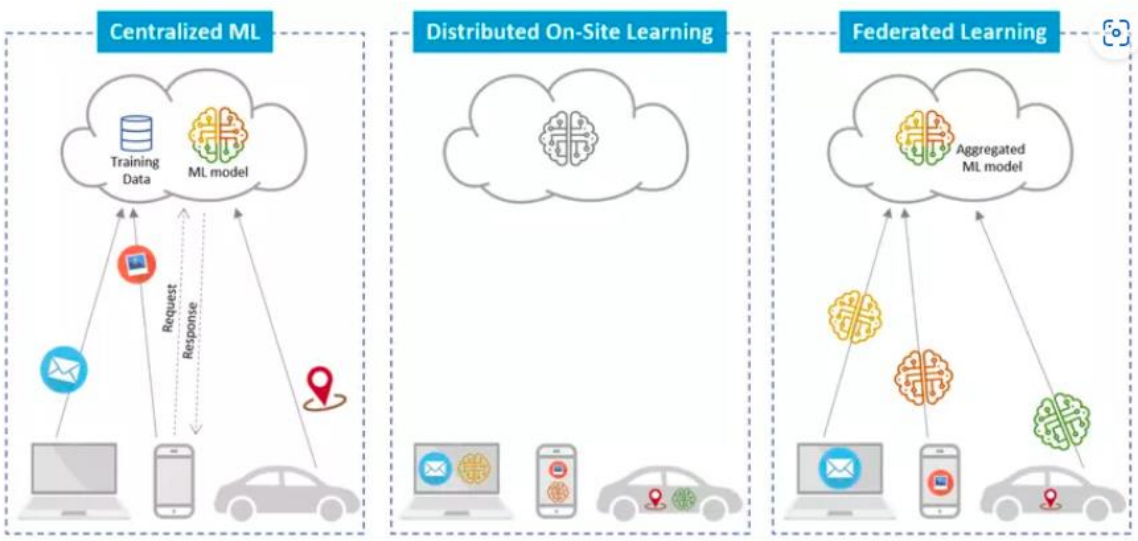
cụ thể hơn cũng như thấy được sự hữu ích khi áp dụng federated learning vào thực tế.

CHƯƠNG 2: NỘI DUNG

2.1. TỔNG QUAN VỀ FEDERATED LEARNING

2.1.1. MÔ HÌNH

2.1.1.1. SO SÁNH VỚI NHỮNG MÔ HÌNH TRUYỀN THỐNG



Hình 1: So sánh mô hình [1]

2.1.1.1.1. MÔ HÌNH CENTRALIZED ML (BÊN TRÁI)

Machine Learning nói chung và Deep Learning nói riêng đang tìm đường đi vào cuộc sống hàng ngày của chúng ta khi chúng ta ngày càng bị cuốn hút bởi việc ra quyết định của AI. Các ứng dụng DL rất đa dạng, từ đơn giản như Netflix đang theo chân Google và Facebook để cải thiện dịch vụ của mình, đến phức tạp như ô tô tự lái, chăm sóc sức khỏe thông minh, phát hiện gian lận, dự báo động đất và nhiều ứng dụng khác. Đằng sau sự thành công của Deep Learning là lượng dữ liệu khổng lồ được tạo bởi các thiết bị di động và IoT. Trong các phương pháp đào tạo, thông thường là liên tục truyền dữ liệu được tạo vào đám mây, nơi dữ liệu được phân tích tốt hơn, nhiều tính năng hơn được trích xuất và các mô hình được đào tạo tốt hơn trên các máy chủ hiệu suất cao. Phương pháp này được minh họa trong kịch bản ML tập trung ở phía bên trái của *Hình 1*. Amazon Web Services, Google Cloud và Microsoft Azure nằm trong số các nhà cung cấp dịch vụ ML có sẵn, nơi các mô hình có thể được triển khai và sử dụng trên quy mô lớn. Khi có nhiều tương tác với các dịch vụ khả dụng trong đám mây, nhiều dữ liệu huấn luyện hơn sẽ được thu thập và

do đó, nhiều ứng dụng dựa trên ML thông minh hơn sẽ được tạo ra. Tuy nhiên, tính riêng tư của dữ liệu có sẵn được sử dụng để đào tạo và cho sự thành công vượt bậc của DL đang trở thành mối quan tâm ngày càng tăng đối với người dùng. Dữ liệu đó có thể rất riêng tư và thuộc bất kỳ loại nào, chẳng hạn như Thông tin nhận dạng cá nhân (ví dụ: bằng lái xe, thông tin hộ chiếu, v.v.), Dữ liệu thanh toán (ví dụ: tài khoản ngân hàng, số thẻ tín dụng, v.v.), Thông tin sức khỏe được bảo vệ (ví dụ: hồ sơ chẩn đoán và y tế , v.v.), Dữ liệu bí mật (ví dụ: tài liệu tài chính, v.v.) và các dữ liệu khác. Khi dữ liệu này được chia sẻ với đám mây, rất có thể quyền riêng tư của người dùng sẽ bị xâm phạm do các cuộc tấn công nghe lén. Các vấn đề khác phát sinh trong cách tiếp cận dựa trên đám mây/tập trung: (1) Độ trễ, vì dữ liệu có thể được truyền đi hàng trăm, thậm chí hàng ngàn dặm để đến đám mây và (2) Chi phí truyền dữ liệu, vì việc di chuyển dữ liệu qua mạng vào và ra khỏi điện toán đám mây là không miễn phí. Để khắc phục những vấn đề như vậy, ML tại chỗ đã được nâng cao, trong đó một số tác vụ ML được chuyển đến các thiết bị có tài nguyên mạnh mẽ. [1]

2.1.1.1.2. MÔ HÌNH DISTRIBUTED ON-SITE LEARNING

Với rủi ro ngày càng tăng của việc di chuyển dữ liệu sang một thực thể tập trung, nhu cầu về trí thông minh theo thời gian thực thúc đẩy ML phân tán tại chỗ, nơi đào tạo, dự đoán và suy luận dựa trên dữ liệu phát trực tiếp. Thay vì gửi yêu cầu cùng với dữ liệu riêng tư từ người dùng lên đám mây, ML tại chỗ sẽ tương tác với máy chủ để phân phối mô hình ML không chung hoặc được đào tạo trước cho các thiết bị, như được minh họa trong phần giữa của *Hình 1*. Sau khi triển khai mô hình, mỗi thiết bị có thể cá nhân hóa thiết bị bằng cách huấn luyện sử dụng dữ liệu cục bộ của thiết bị, có thể thực hiện một số dự đoán cho dữ liệu của thiết bị để dự đoán kết quả hoặc có thể chạy tính toán suy luận để suy ra một số mẫu thử nghiệm và tìm hiểu về quy trình tạo dữ liệu. Trong những hệ thống như vậy, lợi thế riêng tư là chắc chắn, vì dữ liệu không rời khỏi máy chủ của nó. Trí thông minh trên thiết bị đã được áp dụng trong nhiều ứng dụng như Phát hiện ung thư da, Ứng dụng y tế, Lớp học thông minh, Dịch vụ hỗ trợ mạng thần kinh, v.v. các thiết bị giới hạn các mô hình cục bộ được tạo đối với từng trải nghiệm người dùng mà không có bất kỳ lợi ích nào từ dữ liệu ngang hàng. Vì mục đích này, FL đã được nâng cao, trong đó tính toán của người dùng được liên kết trong khi vẫn bảo vệ quyền riêng tư. [1]

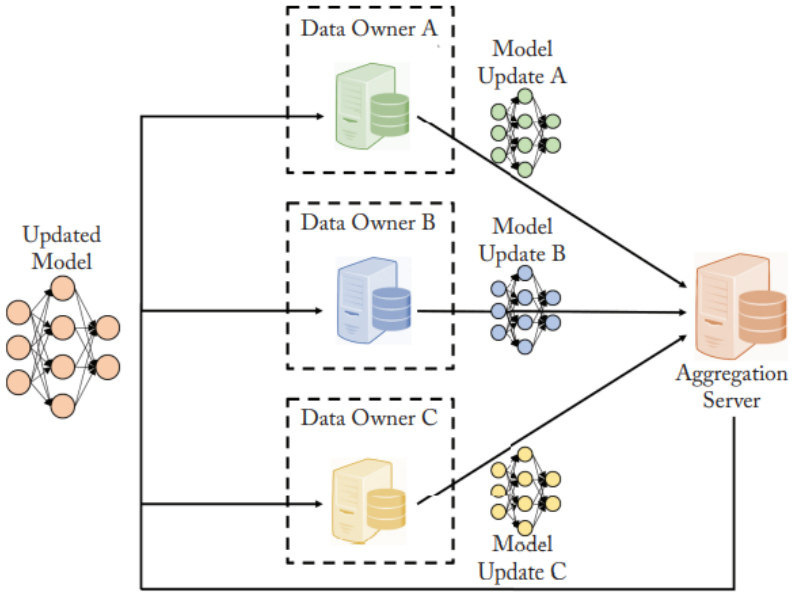
2.1.1.1.3. MÔ HÌNH FEDERATED LEARNING

Các nhà nghiên cứu của Google đã tạo ra FL vào năm 2016 và kể từ đó, nó đã càn quét thế giới bằng cách trải qua sự phát triển mạnh mẽ trong cả giới học thuật

và ngành công nghiệp. Ngoài ML trên thiết bị, FL được phát triển để chuyển nhiệm vụ đào tạo sang chính thiết bị, đồng thời liên kết các mô hình cục bộ và học tập. Mục tiêu chính của nó là xây dựng một khuôn khổ hướng tới ML bảo vệ quyền riêng tư. Phía bên phải của *Hình 1* cho thấy kiến trúc FL so với các phương pháp hiện có khác. Giữa việc gửi dữ liệu cục bộ chưa được bảo mật đến máy chủ và hưởng lợi từ các ứng dụng ML (centralized ML), thực hiện các tác vụ ML trên thiết bị mà không hưởng lợi từ dữ liệu của máy ngang hàng và loại trừ quyền truy cập trực tiếp vào dữ liệu thô cũng như liên kết các mô hình ML đào tạo cục bộ (distributed on-site learning), thì kiến trúc FL có nhiều khả năng được người dùng lựa chọn hơn . Do đó, FL bảo vệ quyền riêng tư của dữ liệu và giảm chi phí giao tiếp dữ liệu bằng cách giữ dữ liệu thô trên thiết bị, hưởng lợi từ dữ liệu của các máy ngang hàng và tổng hợp các bản cập nhật mô hình được tính toán cục bộ. [1]

2.1.2. Ý TƯỞNG THUẬT TOÁN FEDERATED LEARNING

Tìm kiếm các giải pháp để xây dựng mô hình Machine Learning không phụ thuộc vào việc thu thập tất cả dữ liệu vào một bộ lưu trữ tập trung nơi quá trình đào tạo mô hình có thể diễn ra. Một ý tưởng là đào tạo một mô hình tại nơi có nguồn dữ liệu, sau đó để các máy khách (client) truyền các mô hình tương ứng của chúng để đạt được sự đồng thuận cho một mô hình toàn cục. Để đảm bảo quyền riêng tư và bảo mật dữ liệu của người dùng, quy trình liên lạc được thiết kế cẩn thận để không máy khách nào có thể đoán được dữ liệu của bất kỳ máy khách nào khác. Đồng thời, mô hình toàn cục được xây dựng như thể các nguồn dữ liệu được kết hợp. Đây là ý tưởng đằng sau “Federated machine learning” hay gọi tắt là “Federated Learning”.



Hình 2: Mô hình FL [2]

Trong mô hình của Federated Learning sẽ bao gồm các client và máy server, mỗi client sẽ sở hữu dữ liệu riêng (Data Owner), server sẽ có vai trò tổng hợp các mô hình từ các client sau khi đã được đào tạo với các dữ liệu ở client. Điều này đảm bảo dữ liệu của các bên tham gia sẽ không bị rò rỉ. [2]

2.1.3. ĐỊNH NGHĨA VỀ FEDERATED LEARNING

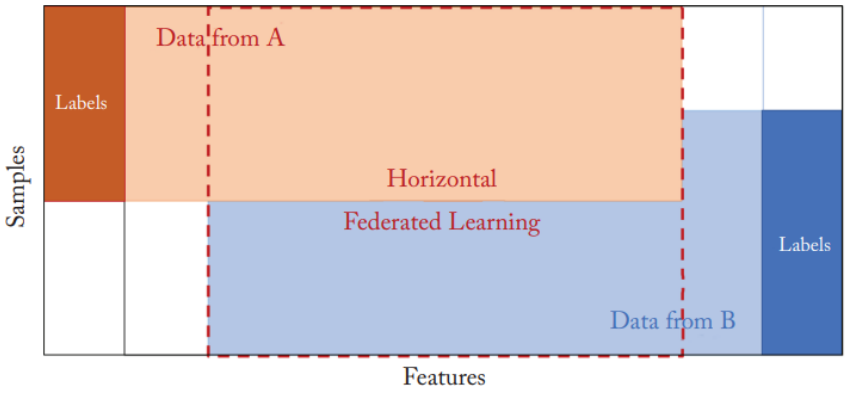
Federated Learning (Học liên kết): là một kỹ thuật máy học (machine learning), đào tạo một thuật toán trên nhiều thiết bị hoặc máy chủ phi tập trung (decentralized) đang giữ các tập dữ liệu cục bộ, mà không trao đổi các dữ liệu đó. Kỹ thuật này trái ngược với kỹ thuật máy học tập trung truyền thống, với các mẫu dữ liệu được tải lên chung một máy chủ. Federated Learning cho phép xây dựng mô hình máy học phổ biến, mạnh mẽ mà không cần chia sẻ dữ liệu, nhờ đó giải quyết được các vấn đề quan trọng như bảo mật, quyền truy cập và truy cập dữ liệu không đồng nhất. [3]

Có hai quá trình trong học tập liên kết: đào tạo mô hình và suy luận mô hình. Trong quá trình đào tạo mô hình, thông tin về mô hình có thể được trao đổi giữa các bên chứ không phải dữ liệu. Trao đổi không tiết lộ bất kỳ phần dữ liệu riêng tư nào được bảo vệ của mỗi người tham gia. Mô hình được đào tạo có thể nằm ở một bên hoặc được chia sẻ giữa nhiều bên. [2]

2.1.4. PHÂN LOẠI FEDERATED LEARNING

2.1.4.1. HORIZONTAL FEDERATED LEARNING

Horizontal federated learning (HFL) đề cập đến trường hợp những người tham gia học tập liên kết chia sẻ các thuộc tính (feature) dữ liệu chồng chéo, nghĩa là các thuộc tính dữ liệu được căn chỉnh giữa những người tham gia, nhưng chúng khác nhau về mẫu dữ liệu. Do đó, chúng ta cũng gọi HFL là phương pháp học liên kết được phân vùng theo mẫu.



Hình 3: Phân vùng HFL [2]

Ví dụ, khi hai bên là hai ngân hàng phục vụ hai thị trường khu vực khác nhau, họ có thể chỉ chia sẻ một số ít người dùng nhưng dữ liệu của họ có thể có không gian thuộc tính (feature) rất giống nhau do mô hình kinh doanh tương tự. Nghĩa là, với sự trùng lặp hạn chế về người dùng nhưng chồng chéo lớn về thuộc tính dữ liệu, hai ngân hàng có thể cộng tác trong việc xây dựng các mô hình ML thông qua học tập liên kết ngang (HFL). [2]

2.1.4.1.1. KIẾN TRÚC CỦA HFL

2.1.4.1.1.1. KIẾN TRÚC CLIENT-SERVER

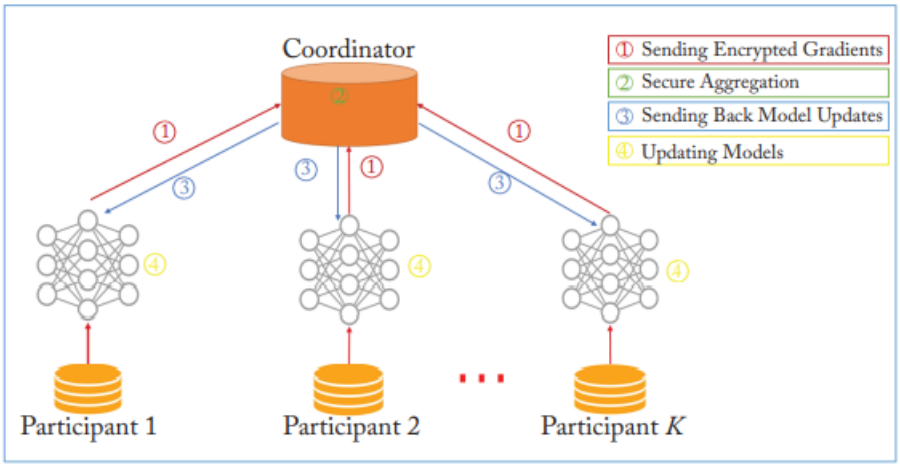
Còn được gọi là kiến trúc master-worker. Trong hệ thống này, K người tham gia (còn được gọi là khách hàng hoặc người dùng hoặc các bên) có cùng cấu trúc dữ liệu hợp tác đào tạo mô hình máy học (ML) với sự trợ giúp của máy chủ server (còn được gọi là máy chủ tham số hoặc máy chủ tổng hợp hoặc điều phối viên). Gồm các bước:

Bước 1: Những người tham gia tính toán cục bộ trọng số hoặc độ dốc đào tạo, che dấu lựa chọn độ dốc bằng kỹ thuật mã hóa hoặc chia sẻ bí mật và gửi kết quả che giấu đến máy chủ.

Bước 2: Máy chủ thực hiện tổng hợp bảo mật, ví dụ: thông qua lấy trung bình trọng số.

Bước 3: Máy chủ gửi lại kết quả tổng hợp cho người tham gia.

Bước 4: Những người tham gia cập nhật các mô hình tương ứng của họ với các gradient được giải mã.



Hình 4: Kiến trúc HFL [2]

Các lần lặp qua các bước ở trên tiếp tục cho đến khi hàm mất mát hội tụ hoặc cho đến khi đạt đến số lần lặp tối đa cho phép hoặc cho đến khi đạt đến thời gian đào tạo tối đa cho phép.

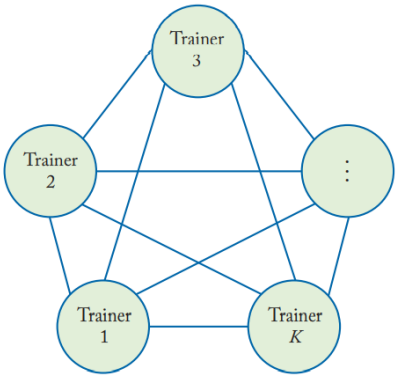
Lưu ý rằng trong các bước trên, người tham gia gửi các gradient đến máy chủ, máy chủ này sẽ tổng hợp các gradient nhận được. Chúng tôi gọi phương pháp này là lấy trung bình độ dốc (gradient averaging). Gradient averaging cũng được biết đến là synchronous stochastic gradient descent hoặc federated SGD (FedSGD). Ngoài ra, thay vì độ dốc, những người tham gia có thể chia sẻ trọng số mô hình. Nghĩa là, những người tham gia tính toán trọng số mô hình cục bộ và gửi chúng đến máy chủ. Máy chủ tổng hợp các trọng số mô hình cục bộ nhận được và gửi kết quả tổng hợp lại cho những người tham gia. Chúng tôi gọi phương pháp này là mô hình lấy trung bình (model averaging). Lưu ý rằng cả trung bình độ dốc và trung bình mô hình đều được gọi là trung bình liên kết (FedAvg).

Kiến trúc trên có thể ngăn chặn rò rỉ dữ liệu đối với máy chủ bán trung thực, nếu việc tổng hợp độ dốc được thực hiện bằng tính toán an toàn của nhiều bên. Tuy nhiên, nó có thể dễ bị tấn công bởi một người tham gia đào tạo độc hại trong quá trình học tập hợp tác. [2]

2.1.4.1.1.2. KIẾN TRÚC PEER-TO-PEER

Một hệ thống HFL cũng có thể sử dụng kiến trúc ngang hàng. Trong kiến trúc này không có máy chủ trung tâm hoặc điều phối viên. Trong các tình huống như vậy, những người tham gia K của hệ thống HFL còn được gọi là người đào tạo hoặc người

đào tạo phân tán hoặc worker. Mỗi người đào tạo chịu trách nhiệm đào tạo cùng một mô hình ML chỉ sử dụng dữ liệu cục bộ của nó. Hơn nữa, các người đào tạo cần các kênh an toàn để chuyển các trọng số mô hình cho nhau. Để đảm bảo liên lạc an toàn giữa hai người đào tạo bất kỳ, các biện pháp bảo mật, chẳng hạn như các lược đồ mã hóa dựa trên khóa công khai (key public), có thể được áp dụng.



Hình 5: Kiến trúc Peer – to - peer [2]

Do không có máy chủ trung tâm nên người huấn luyện phải thống nhất trước thứ tự gửi và nhận trọng số mô hình. Có hai cách chủ yếu để làm điều này:

Cyclic transfer (Chuyển giao theo chu kỳ): Trong chế độ truyền theo chu kỳ, các người đào tạo được tổ chức thành một chuỗi. Người đào tạo đầu tiên (tức là phần trên cùng của chuỗi) gửi các trọng số mô hình hiện tại của nó tới người đào tạo sau của nó. Một người đào tạo nhận các trọng số mô hình từ người đào tạo trước nó của nó và nó cập nhật các trọng số mô hình nhận được bằng cách sử dụng các lô nhỏ dữ liệu huấn luyện từ tập dữ liệu của chính nó. Sau đó, nó sẽ gửi các trọng số mô hình được cập nhật tới trình huấn luyện xuôi dòng của nó.Ví dụ: người đào tạo thứ 1 tới người đào tạo thứ 2,...người đào tạo thứ k-1 tới người đào tạo thứ k, người đào tạo thứ k tới người đào tạo thứ 1. Quy trình này được lặp lại cho đến khi các trọng số của mô hình hội tụ hoặc cho đến khi đạt đến số lần lặp lại tối đa hoặc cho đến khi đạt đến thời gian huấn luyện tối đa cho phép.

Random transfer: người đào tạo thứ k chọn 1 số i từ tập {1,...,L} ngẫu nhiên với xác suất bằng nhau và gửi trọng số của nó trới cho người đào tạo thứ i. Người đào tạo thứ i nhận trọng số và cập nhật trọng số bằng cách sử dụng các lô nhỏ dữ liệu huấn luyện từ tập dữ liệu của chính nó. Sau đó người đào tạo thứ i sẽ chọn một số j từ tập {1,....,L}/{i} ngẫu nhiên và gửi trọng số của nó trới cho người đào tạo thứ j. Quy trình này diễn ra đồng thời giữa K người đào tạo cho đến khi các người đào tạo đồng ý rằng các trọng số mô hình đã hội tụ hoặc cho đến khi đạt đến thời gian huấn luyện tối đa cho phép. Phương pháp này còn được gọi là Gossip Learning

So với kiến trúc Client – Server, ưu điểm rõ ràng của kiến trúc ngang hàng là khả năng loại bỏ máy chủ trung tâm, nên nó loại bỏ khả năng rò rỉ thông tin đến máy chủ. Tuy nhiên, có một số nhược điểm. Chẳng hạn, trong chế độ truyền theo chu kỳ, do không có máy chủ trung tâm, các tham số trọng lượng được cập nhật nối tiếp thay vì theo đợt song song, điều này sẽ mất nhiều thời gian hơn để đào tạo một mô hình. [2]

2.1.4.1.2. ĐÁNH GIÁ MÔ HÌNH TOÀN CỤC

Trong HFL, việc đào tạo và đánh giá mô hình được thực hiện một cách phân tán ở mỗi người tham gia và không thể truy cập bộ dữ liệu của những người tham gia. Do đó, mỗi người tham gia có thể dễ dàng kiểm tra hiệu suất của chế độ bằng cách sử dụng tập dữ liệu thử nghiệm cục bộ của mình để có được hiệu suất của mô hình cục bộ, nhưng phải mất nhiều nỗ lực hơn để có được hiệu suất của mô hình toàn cầu trên tất cả những người tham gia. Ở đây, hiệu suất mô hình cục bộ có nghĩa là hiệu suất của mô hình HFL được kiểm tra trên bộ dữ liệu thử nghiệm cục bộ của một người tham gia và hiệu suất mô hình toàn cầu đề cập đến hiệu suất của mô hình HFL được đánh giá trên bộ dữ liệu thử nghiệm của tất cả những người tham gia. Hiệu suất của mô hình có thể được biểu thị bằng accuracy, precision, recall, and AUC,...

Đối với kiến trúc Client – Server, những người tham gia và điều phối viên có thể hợp tác để có được hiệu suất mô hình toàn cục. Trong quá trình đào tạo mô hình và sau khi hoàn thành đào tạo mô hình trong HFL, chúng ta có thể có được hiệu suất mô hình toàn cục theo các bước sau:

Bước 1: người tham gia thứ k thực hiện đánh giá hiệu suất của mô hình với tập dữ liệu cục bộ (local data test).

Bước 2: người tham gia thứ k gửi kết quả đánh giá mô hình tới người điều phối.

Bước 3: Sau khi thu thập kết quả đánh giá mô hình của k người tham gia, người điều phối có thể tính toán hiệu suất mô hình toàn cục.

Bước 4: Người điều phối gửi lại kết quả đánh giá mô hình toàn cục cho tất cả người tham gia.

Đối với kiến trúc ngang hàng, do không có điều phối viên trung tâm nên sẽ phức tạp hơn để đạt được hiệu suất mô hình toàn cục. Một cách khả thi là chọn một trong các người đào tạo làm điều phối viên tạm thời. Sau đó, chúng ta có thể làm theo quy trình trên được đề xuất cho kiến trúc máy khách-máy chủ để có được hiệu suất mô hình toàn cục cho kiến trúc ngang hàng. Phương pháp này được khuyến nghị để đánh giá mô hình HFL cuối cùng sau khi hoàn thành khóa đào tạo. Tuy nhiên,

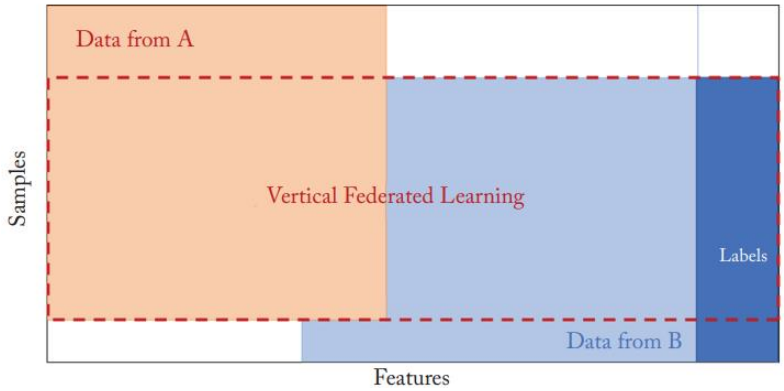
nếu chúng tôi áp dụng phương pháp này trong quá trình đào tạo, nó sẽ gây quá tải cho điều phối viên tạm thời, điều này có thể không được chấp nhận nếu giảng viên là thiết bị di động hoặc IoT có nguồn lực hạn chế (ví dụ: pin). Một cách khả thi để khắc phục vấn đề này là yêu cầu các người đào tạo thay phiên nhau làm điều phối viên tạm thời. [2]

2.1.4.2. VERTICAL FEDERATED LEARNING

Học liên kết ngang (HFL) có thể áp dụng cho các tình huống trong đó bộ dữ liệu của những người tham gia chia sẻ cùng một không gian đặc trưng nhưng khác nhau về không gian mẫu. Do đó, HFL thuận tiện khi được áp dụng để xây dựng các ứng dụng được hỗ trợ bởi một lượng lớn thiết bị di động. Trong những trường hợp đó, các mục tiêu được liên kết là những người tiêu dùng cá nhân của các ứng dụng, có thể được coi là mô hình B2C (business-to-consumer). Tuy nhiên, trong nhiều tình huống thực tế, những người tham gia học tập liên kết là các tổ chức đã thu thập các tính năng dữ liệu khác nhau cho cùng một nhóm người để theo đuổi các mục tiêu kinh doanh khác nhau. Các tổ chức này thường có động lực hợp tác mạnh mẽ để nâng cao hiệu quả kinh doanh, có thể coi đây là mô hình B2B (business-to- business). [2]

2.1.4.2.1. ĐỊNH NGHĨA CỦA VFL

Vertical federated learning (VFL) khác với HFL, VFL áp dụng cho tình huống trong đó những người tham gia học tập liên kết chia sẻ các mẫu dữ liệu chồng chéo, nghĩa là các mẫu dữ liệu được căn chỉnh giữa những người tham gia, nhưng chúng khác nhau về thuộc tính (feature) dữ liệu. Nó giống với tình huống dữ liệu được phân vùng theo chiều dọc bên trong dạng xem dạng bảng. Do đó, chúng tôi cũng đặt tên VFL là học tập liên kết được phân vùng theo thuộc tính (feature).



Hình 6: Phân vùng VFL [2]

Các bộ dữ liệu được duy trì bởi các tổ chức khác nhau có các mục tiêu kinh doanh khác nhau thường có các không gian tính năng khác nhau, trong khi các tổ chức đó có thể chia sẻ một nhóm lớn người dùng chung. Điều này được minh họa trong *Hình 6*. Với VFL, còn được gọi là học tập liên kết được phân vùng theo tính năng, chúng ta có thể tận dụng các không gian tính năng không đồng nhất của các bộ dữ liệu phân tán được duy trì bởi các bộ dữ liệu đó các tổ chức để xây dựng các mô hình máy học (ML) tốt hơn mà không cần trao đổi và phơi bày các dữ liệu riêng tư. Trong khuôn khổ học tập liên kết, danh tính và trạng thái của mỗi người tham gia các bên là như nhau, và liên đoàn giúp mọi người thiết lập một chiến lược "khối thịnh vượng chung", đó là lý do tại sao điều này được gọi là "học tập liên kết." Đối với một hệ thống VFL như vậy, chúng tôi có:

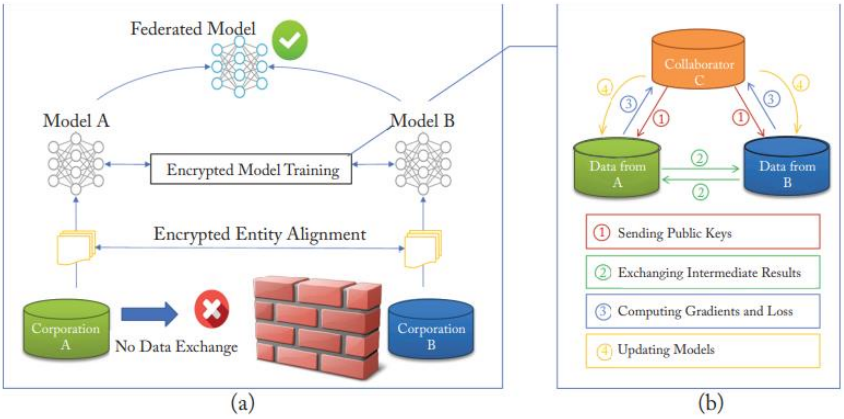
$$X_i \neq X_j,\ Y_i \neq Y_j, I_i = I_j\ \forall D_i, D_j, i \neq j$$

Trong đó X và Y lần lượt biểu thị không gian đặc trưng và không gian nhãn. I là ID mẫu không gian và ma trận D đại diện cho dữ liệu do các bên khác nhau nắm giữ. Mục tiêu cho tất cả các bên là cộng tác xây dựng mô hình ML dùng chung bằng cách khai thác tất cả các tính năng đã thu thập bởi các bên tham gia.

Trong cài đặt VFL, có một số giả định cơ bản để đạt được bảo mật và quyền riêng tư trước khi phục vụ. Đầu tiên, người ta cho rằng những người tham gia là những người trung thực nhưng tò mò. Điều này có nghĩa là rằng những người tham gia cố gắng suy luận càng nhiều càng tốt từ thông tin nhận được từ những người tham gia khác, mặc dù họ tuân thủ giao thức mà không làm phiền nó theo bất kỳ cách nào. Từ họ cũng có ý định xây dựng một mô hình chính xác hơn, họ không thông đồng với nhau. Thứ hai, người ta cho rằng quá trình truyền thông tin là an toàn và đủ tin cậy để bảo vệ chống lại các cuộc tấn công. Người ta còn giả định rằng giao tiếp là không mất dữ liệu mà không can thiệp vào các kết quả trung gian. Một bên thứ ba bán trung thực (STP – Semi-honest third party) cũng có thể tham gia cùng những người tham gia để hỗ trợ hai bên ngang nhau. STP độc lập với cả hai bên. STP thu thập các kết quả trung gian để tính toán độ dốc và độ mất mát, đồng thời phân phối kết quả cho mỗi bên. Thông tin mà STP nhận được từ những người tham gia được mã hóa hoặc làm xáo trộn. Dữ liệu thô của những người tham gia không được tiếp xúc với nhau và mỗi người tham gia chỉ nhận được các tham số mô hình liên quan đến các tính năng riêng của nó. [2]

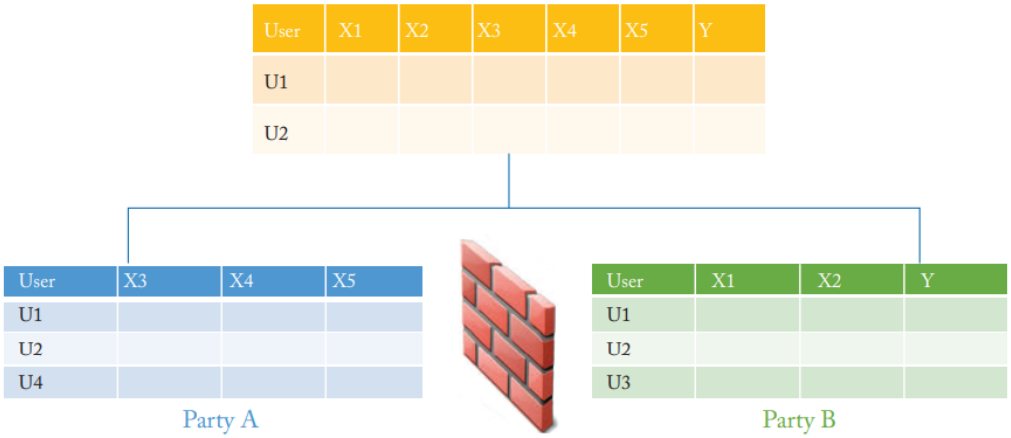
2.1.4.2.2. KIẾN TRÚC CỦA VFL

Để dễ xây dựng, chúng tôi sử dụng một ví dụ để mô tả kiến trúc của VFL. Giả sử rằng Công ty A và B muốn cùng đào tạo một mô hình ML. Mỗi người trong số họ có dữ liệu riêng của họ. Ngoài ra, B còn có dữ liệu đã được gán nhãn mà mô hình cần để thực hiện nhiệm vụ dự đoán. Vì lý do bảo mật dữ liệu và quyền riêng tư của người dùng, A và B không thể trao đổi dữ liệu trực tiếp. Để đảm bảo bí mật dữ liệu trong quá trình đào tạo, có thể có sự tham gia của cộng tác viên bên thứ ba C. Ở đây, chúng tôi giả định rằng C trung thực và không thông đồng với A hoặc B, nhưng A và B là trung thực nhưng tò mò. Bên thứ ba đáng tin cậy C là một giả định hợp pháp vì vai trò của C có thể được thực hiện bởi các cơ quan có thẩm quyền như chính phủ hoặc được thay thế bằng các nút tính toán an toàn như Intel Software Guard Extensions (SGX). Quá trình đào tạo của một hệ thống VFL thường bao gồm hai phần. Đầu tiên, nó thiết lập sự liên kết giữa các thực thể chia sẻ cùng ID của hai bên. Sau đó, quy trình đào tạo được mã hóa (hoặc bảo vệ quyền riêng tư) được tiến hành trên các đối tượng được liên kết đó.



Hình 7: Mô hình VFL [2]

Phần 1: Liên kết thực thể được mã hóa. Do nhóm người dùng của hai công ty A và B không giống nhau nên hệ thống sử dụng kỹ thuật căn chỉnh ID người dùng dựa trên mã hóa để xác nhận những người dùng chung được chia sẻ bởi cả hai bên mà A và B không tiết lộ dữ liệu thô tương ứng của họ. Trong quá trình căn chỉnh thực thể, hệ thống sẽ không hiển thị những người dùng thuộc về một trong hai công ty, như trong hình 7. (a) ở trên.



Hình 8: Minh họa căn chỉnh thực thể được mã hóa [2]

Phần 2: Đào tạo mô hình mã hóa. Sau khi xác định các thực thể chung, chúng tôi có thể sử dụng dữ liệu của các thực thể chung này để đào tạo một mô hình ML chung. Quá trình đào tạo có thể được chia thành bốn bước sau (như minh họa ở hình 7. (b):

Bước 1: C tạo các cặp mã hóa và gửi khóa chung cho A và B.

Bước 2: A và B mã hóa và trao đổi kết quả trung gian để tính toán độ dốc (gradient) và hàm mất mát

Bước 3: A và B tính toán độ dốc được mã hóa và thêm lớp bảo vệ bổ sung tương ứng. B cũng tính toán hàm mất mát và mã hóa nó. A và B gửi kết quả được mã hóa tới C

Bước 4: C giải mã độ dốc và độ mất mát rồi gửi kết quả lại cho A và B. A và B mở khóa lớp bảo vệ độ dốc (gradient) và cập nhật các tham số mô hình tương ứng. [2]

2.1.4.3. FEDERATED TRANSFER LEARNING

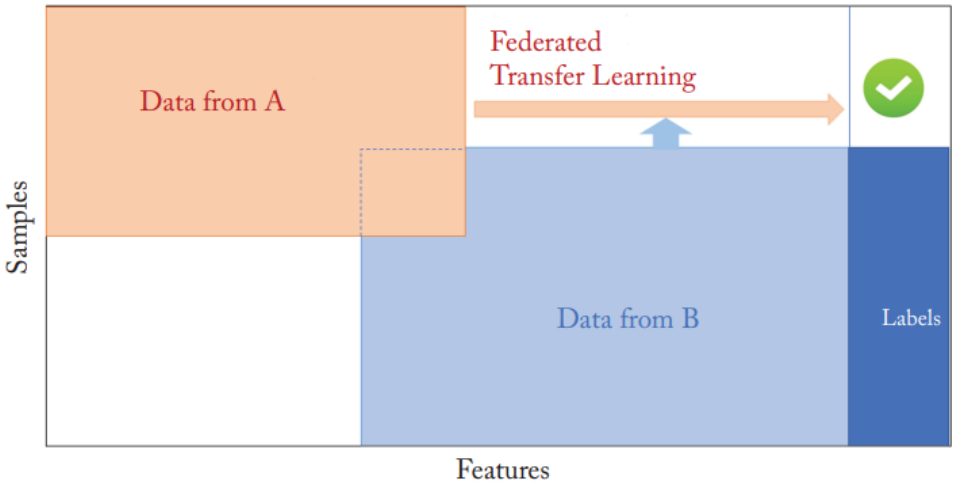
HFL yêu cầu tất cả các bên tham gia chia sẻ cùng một không gian đối tượng trong khi VFL yêu cầu các bên chia sẻ cùng một không gian mẫu. Tuy nhiên, trên thực tế, chúng tôi thường gặp phải tình huống không có đủ các thuộc tính hoặc mẫu được chia sẻ giữa các bên tham gia. Trong những trường hợp đó, người ta vẫn có thể xây dựng mô hình học liên kết kết hợp với học chuyển giao để chuyển giao kiến thức giữa các bên nhằm đạt được hiệu quả tốt hơn. Chúng tôi gọi sự kết hợp giữa học tập liên kết và học tập chuyển giao là Học tập chuyển giao liên kết (FTL). [2]

2.1.4.3.1. ĐỊNH NGHĨA CỦA FTL

Học chuyển giao là một kỹ thuật học tập để cung cấp các giải pháp chuyển giao kiến thức giữa các miền. Trong nhiều ứng dụng, chúng ta chỉ có một lượng nhỏ

dữ liệu được gắn nhãn hoặc khả năng giám sát yếu nên không thể xây dựng các mô hình ML một cách đáng tin cậy. Trong những tình huống như vậy, chúng ta vẫn có thể xây dựng các mô hình ML hiệu suất cao bằng cách tận dụng và điều chỉnh các mô hình từ các nhiệm vụ hoặc miền tương tự.

Federated transfer learning (FTL) có thể áp dụng cho trường hợp không có chồng chéo trong các mẫu dữ liệu cũng như trong các tính năng. Nhằm mục đích xây dựng mô hình hiệu quả cho miền mục tiêu trong khi tận dụng lợi thế kiến thức từ các miền (nguồn) khác.



Hình 9: Phân vùng FTL [2]

Từ hình trên, xem xét hai bên A và B, trong đó chỉ có một sự trùng lặp nhỏ trong không gian thuộc tính (features) và không gian mẫu giữa A và B, một mô hình đã học trên B được chuyển sang A bằng cách tận dụng các thuộc tính và dữ liệu trùng lặp nhỏ giữa 2 bên. Khả năng sử dụng dữ liệu được truyền trên dữ liệu không trùng lặp trong A làm cho FTL khác với VFL. [2]

Từ góc độ kỹ thuật, FTL khác với học chuyển giao truyền thống chủ yếu theo hai cách sau:

- FTL xây dựng các mô hình dựa trên dữ liệu được phân phối giữa nhiều bên và dữ liệu mong muốn của mỗi bên không thể được tập hợp lại với nhau hoặc hiển thị cho các bên khác. Học chuyển giao truyền thống không có ràng buộc như vậy.
- FTL yêu cầu bảo vệ quyền riêng tư của người dùng và bảo vệ dữ liệu (và mô hình), đây không phải là mối quan tâm đáng kể trong học tập chuyển giao truyền thống

2.1.4.3.2. PHÂN LOẠI FTL

Bản chất của học chuyển giao là tìm ra sự bất biến giữa miền nguồn giàu tài nguyên và miền đích khan hiếm tài nguyên, và khai thác sự bất biến đó để chuyển tri thức từ miền nguồn sang miền đích. Dựa trên các phương pháp được sử dụng để tiến hành học chuyển đổi, Pan và Yang [2010] chia học chuyển giao truyền thống thành ba loại chủ yếu: chuyển giao dựa trên mẫu, chuyển giao dựa trên thuộc tính và chuyển giao dựa trên mô hình. FTL mở rộng quá trình học chuyển đổi truyền thống sang mô hình máy học phân tán (DML) bảo vệ quyền riêng tư. Ba loại kỹ thuật học chuyển tiếp này có thể được áp dụng cho HFL và VFL tương ứng. [2]

2.1.4.3.2.1. CHUYỂN GIAO DỰA TRÊN MẪU

Đối với HFL, dữ liệu của các bên tham gia thường được lấy từ các bản phân phối khác nhau, điều này có thể dẫn đến hiệu suất kém của các mô hình ML được đào tạo trên những dữ liệu đó. Các bên tham gia có thể chọn một cách có chọn lọc hoặc cân nhắc lại các mẫu dữ liệu đào tạo để giảm bớt sự khác biệt về phân phối sao cho chức năng tổn thất khách quan có thể được giảm thiểu một cách tối ưu.

Đối với VFL, các bên tham gia có thể có những mục tiêu kinh doanh khá khác nhau. Do đó, các mẫu được căn chỉnh và một số tính năng của chúng có thể có tác động tiêu cực đến quá trình học chuyển giao được phân loại liên kết, được gọi là chuyển giao tiêu cực. Trong trường hợp này, các bên tham gia có thể chọn một cách có chọn lọc các tính năng và mẫu để tránh chuyển nhượng tiêu cực. [2]

2.1.4.3.2.2. CHUYỂN GIAO DỰA TRÊN THUỘC TÍNH

Các bên tham gia cùng nhau tìm hiểu một không gian biểu diễn tính năng chung, trong đó sự khác biệt về phân phối và ngữ nghĩa giữa các biểu diễn tính năng được chuyển đổi từ dữ liệu thô có thể được giải tỏa và nhờ đó kiến thức có thể được chuyển giao giữa các miền khác nhau.

Đối với HFL, không gian biểu diễn tính năng chung có thể được học thông qua việc giảm thiểu sự khác biệt trung bình tối đa (MMD) giữa các mẫu của các bên tham gia.

Trong khi đối với VFL, không gian biểu diễn tính năng chung có thể được học thông qua việc giảm thiểu khoảng cách giữa các biểu diễn của các mẫu được căn chỉnh thuộc về các bên khác nhau. [2]

2.1.4.3.2.3. CHUYỂN GIAO DỰA TRÊN MÔ HÌNH

HFL là một loại FTL dựa trên mô hình vì trong quá trình đào tạo, một mô hình toàn cầu được chia sẻ đang được học dựa trên dữ liệu của tất cả các bên và mô hình

toàn cầu được chia sẻ đó được dùng làm mô hình được đào tạo trước để mỗi bên hoàn thiện trong mỗi vòng giao tiếp.

Đối với VFL, các mô hình dự đoán có thể được học từ các mẫu được căn chỉnh để suy ra các thuộc tính và nhãn bị thiếu. Sau đó, các mẫu đào tạo mở rộng có thể được sử dụng để đào tạo một mô hình chia sẻ chính xác hơn. [2]

2.2. ĐẶC ĐIỂM CHÍNH

2.2.1. HỌC LẶP ĐI LẶP LẠI (ITERATIVE LEARNING)

Để đảm bảo thực hiện tốt nhiệm vụ cuối cùng của mô hình học máy trung tâm, học liên kết dựa trên một quy trình lặp đi lặp lại được chia thành một tập hợp nguyên tử của các tương tác giữa máy khách và máy chủ được gọi là vòng học liên kết. Mỗi vòng của quá trình này bao gồm việc truyền trạng thái mô hình toàn cầu hiện tại tới các nút tham gia, đào tạo mô hình cục bộ trên các nút cục bộ này để tạo ra một tập hợp các bản cập nhật mô hình tiềm năng tại mỗi nút, sau đó tổng hợp và xử lý các bản cập nhật cục bộ này thành một bản cập nhật toàn cục duy nhất và áp dụng nó vào mô hình toàn cục.

Trong phương pháp dưới đây, một máy chủ trung tâm được sử dụng để tổng hợp, trong khi các nút cục bộ thực hiện đào tạo cục bộ tùy thuộc vào lệnh của máy chủ trung tâm. Tuy nhiên, các chiến lược khác dẫn đến kết quả tương tự mà không có máy chủ trung tâm, theo cách tiếp cận ngang hàng là phương pháp Peer – to – Peer.

Giả sử một vòng liên kết được tạo bởi một lần lặp lại quá trình học tập, quy trình học tập có thể được tóm tắt như sau:

- **Khởi tạo:** theo đầu vào của máy chủ, một mô hình học máy (ví dụ: hồi quy tuyến tính, mạng nơ ron, tăng cường) được chọn để đào tạo trên các nút cục bộ và khởi tạo. Sau đó, các nút được kích hoạt và chờ máy chủ trung tâm đưa ra các nhiệm vụ tính toán.
- **Chọn client:** một phần nhỏ các nút cục bộ được chọn để bắt đầu đào tạo về dữ liệu cục bộ. Các nút đã chọn có được mô hình thống kê hiện tại trong khi các nút khác chờ vòng liên kết tiếp theo.
- **Cấu hình cài đặt:** máy chủ trung tâm ra lệnh cho các nút đã chọn phải trải qua quá trình đào tạo mô hình trên dữ liệu cục bộ của chúng theo cách được chỉ định trước (ví dụ: đối với một số cập nhật theo lô nhỏ của gradient descent)**.**

- **Báo cáo:** mỗi nút được chọn gửi mô hình cục bộ của nó đến máy chủ để tổng hợp. Máy chủ trung tâm tổng hợp các mô hình đã nhận và gửi lại các bản cập nhật mô hình cho các nút. Nó cũng xử lý các lỗi cho các nút bị ngắt kết nối hoặc các bản cập nhật mô hình bị mất. Vòng liên kết tiếp theo được bắt đầu quay trở lại giai đoạn lựa chọn khách hàng.

Thủ tục được xem xét trước khi giả định cập nhật mô hình được đồng bộ hóa. Các phát triển học tập liên kết gần đây đã giới thiệu các kỹ thuật mới để giải quyết sự không đồng bộ trong quá trình đào tạo hoặc đào tạo với các mô hình thay đổi một cách chủ động. So với các phương pháp tiếp cận đồng bộ trong đó các mô hình cục bộ được trao đổi sau khi các tính toán đã được thực hiện cho tất cả các lớp của mạng nơron, các phương pháp không đồng bộ tận dụng các thuộc tính của mạng nơron để trao đổi các cập nhật mô hình ngay khi các tính toán của một lớp nhất định có sẵn. Các kỹ thuật này cũng thường được gọi là học tập phân tách và chúng có thể được áp dụng cả ở thời gian đào tạo và suy luận bất kể cài đặt học tập liên kết tập trung hay phi tập trung. [4]

2.2.2. NON – IID DATA

Trong hầu hết các trường hợp, giả định về các mẫu độc lập và được phân phối giống nhau trên các nút cục bộ không phù hợp với các thiết lập học liên kết. Theo cài đặt này, hiệu suất của quá trình đào tạo có thể thay đổi đáng kể theo tính không cân bằng của các mẫu dữ liệu cục bộ cũng như phân phối xác suất cụ thể của các ví dụ đào tạo (tức là các tính năng và nhãn) được lưu trữ tại các nút cục bộ.

Mô tả của dữ liệu non - iid dựa trên phân tích xác suất chung giữa các đối tượng địa lý và nhãn cho mỗi nút. Điều này cho phép tách từng đóng góp theo phân phối cụ thể có sẵn tại các nút cục bộ. Các danh mục chính cho dữ liệu non - iid có thể được tóm tắt như sau:

- **Covariate shift:** các nút cục bộ có thể lưu trữ các ví dụ có phân phối thống kê khác nhau so với các nút khác. Một ví dụ xảy ra trong bộ dữ liệu xử lý ngôn ngữ tự nhiên nơi mọi người thường viết các chữ số / chữ cái giống nhau với độ rộng hoặc độ nghiêng nét khác nhau
- **Prior probability shift:** các nút cục bộ có thể lưu trữ các nhãn có phân phối thống kê khác nhau so với các nút khác. Điều này có thể xảy ra nếu tập dữ liệu được phân vùng theo khu vực và / hoặc nhân khẩu học. Ví dụ: tập dữ liệu có chứa hình ảnh của các loài động vật khác nhau đáng kể giữa các quốc gia.

- **Concept drift (*same label, different features):*** các nút cục bộ có thể chia sẻ các nhãn giống nhau nhưng một số trong số chúng tương ứng với các tính năng khác nhau tại các nút cục bộ khác nhau. Ví dụ: hình ảnh mô tả một đối tượng cụ thể có thể thay đổi tùy theo điều kiện thời tiết mà chúng được chụp.
- **Concept shift (*same features, different labels*):** các nút cục bộ có thể chia sẻ các tính năng giống nhau nhưng một số trong số chúng tương ứng với các nhãn khác nhau tại các nút cục bộ khác nhau. Ví dụ, trong xử lý ngôn ngữ tự nhiên, phân tích tình cảm có thể mang lại những cảm nhận khác nhau ngay cả khi cùng một văn bản được quan sát.
- **Unbalancedness:** lượng dữ liệu có sẵn tại các nút cục bộ có thể thay đổi đáng kể về kích thước.

Việc mất độ chính xác do dữ liệu Non-iid có thể bị giới hạn thông qua việc sử dụng các phương tiện chuẩn hóa dữ liệu phức tạp hơn, thay vì chuẩn hóa hàng loạt. [4]

2.2.3. THUẬT TOÁN FEDERATED LEARNING

2.2.3.1. GRADIENT DESCENT

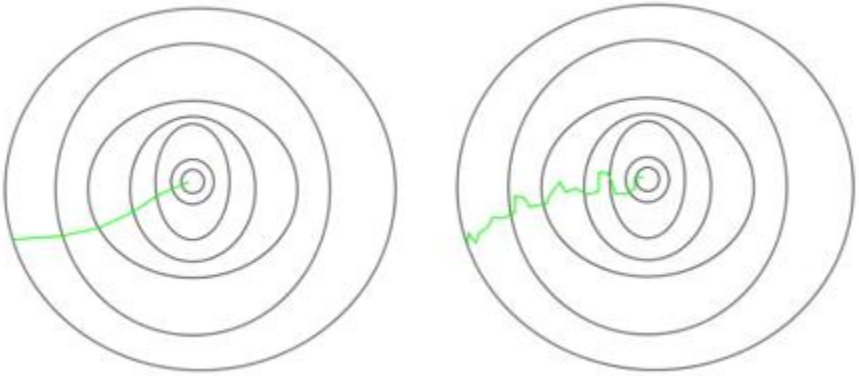
Gradient Descent là một kỹ thuật tối ưu hóa phổ biến trong ML và học sâu và nó có thể được sử dụng với hầu hết các thuật toán học tập. Gradient là độ dốc của một hàm. Nó đo lường mức độ thay đổi của một biến để đáp ứng với những thay đổi của một biến khác. Về mặt toán học, Gradient Descent là một hàm lồi có đầu ra là đạo hàm riêng của một tập các tham số đầu vào của nó. Gradient càng lớn thì độ dốc càng lớn .Trong mỗi lần lặp i, gradient ∇ của hàm mất mát F của mô hình M được đặc trưng bởi các tham số (trọng số) Wi được tính bằng cách tối thiểu hóa F trên tập con Si của các mẫu huấn luyện trong tập dữ liệu. Sau đó, các thông số mô hình được cập nhật theo hướng ngược lại của các giá trị gradient. Tốc độ học tập η (eta) chỉ định kích thước bước của bản cập nhật.

$$W_i + 1 = W_i - \eta \nabla \mathrm{F}\ (W_i,\ S_i)$$

Có các biến thể khác nhau của độ dốc gradient tùy thuộc vào cách sử dụng các mẫu của tập dữ liệu đào tạo để cập nhật các thông số mô hình. Trong full (batch) gradient descent (FGD), tất cả các mẫu đều được sử dụng để tính toán các gradient; stochastic gradient descent (SGD) tính toán độ dốc bằng cách sử dụng một mẫu được chọn ngẫu nhiên duy nhất của tập dữ liệu đào tạo; mini-batch gradient descent (MBGD) tối ưu hóa chức năng mất mát trên một lô mẫu nhỏ ngẫu nhiên. [5]

2.2.3.2. STOCHASTIC GRADIENT DESCENT (SGD)

Trong Stochastic Gradient Descent, một mẫu được chọn ngẫu nhiên thay vì toàn bộ tập dữ liệu cho mỗi lần lặp, vì chỉ có một mẫu từ tập dữ liệu được chọn ngẫu nhiên cho mỗi lần lặp, nên con đường mà thuật toán thực hiện để đến cực tiểu thường biến động hơn so với thuật toán Gradient Descent.



Hình 10: Tính biến động của thuật GD và SGD [5]

Thuật toán FedSGD (Giảm độ dốc có liên kết) là chuyển vị trực tiếp của thuật toán này sang cài đặt có liên kết, nhưng bằng cách sử dụng một phân số ngẫu nhiên ***C*** của các nút và sử dụng tất cả dữ liệu trên nút này. Độ dốc được máy chủ tính trung bình tương ứng với số lượng mẫu đào tạo trên mỗi nút và được sử dụng để thực hiện bước giảm dần độ dốc. [5]

2.2.3.3. THUẬT TOÁN FEDERATED AVERAGING (FedAvg)

2.2.3.3.1. TỔNG QUAN

Federate Averaging (FedAvg) hay còn gọi là thuật toán tính trung bình liên kết. Ý tưởng chính đằng sau FedAvg là thực hiện một số lượng lớn các cập nhật cục bộ trong các máy khách và sau đó lấy trung bình có trọng số đơn giản cho các thông số mô hình cục bộ trên máy chủ. Thuật toán này kết hợp độ dốc ngẫu nhiên (SGD) trên mỗi client với một máy server thực hiện tính trung bình của mô hình. Đây là một cách tiếp cận hiệu quả về giao tiếp đối với việc học liên kết, nhằm mục đích đạt được mô hình toàn cầu chính xác với số lượng vòng giao tiếp hiệu quả giữa máy khách và máy chủ. FedAvg có thể giảm đáng kể số lượng vòng giao tiếp nếu dữ liệu được phân phối độc lập và giống hệt nhau (IID) trên các máy khách.

Mô hình toàn cầu do FedAvg đào tạo có thể không hội tụ đến mức tối ưu trong môi trường non-IID và liên quan đến việc đào tạo liên kết có thể không cung cấp hiệu suất tương đương như đối với cài đặt IID. Hơn nữa, FedAvg vẫn có thể yêu cầu

một số lượng lớn các vòng giao tiếp để đạt được hiệu suất mục tiêu trong các cấu hình Non-IID

Mô tả thuật toán:

Server thực thi:

Khởi tạo tham số W_0

Lặp mỗi vòng t = 1, 2, . . . **thực hiện:**

m ← max(C · K, 1)

S_t ← (nhóm khách ngẫu nhiên được chọn)

Lặp mỗi client k ∈ S_t **song song thực hiện:**

w_{t+1}^k ← **ClientUpdate**(k, w_t)

$w_{t+1} \leftarrow \sum_{k=1}^{k} \frac{n_k}{n} . w_{t+1}^k$

ClientUpdate(k, w): // Chạy trên client k

A ← (Chia P_k thành các lô nhỏ với kích thước B)

Lặp mỗi vòng cục bộ I từ 1 tới E **thực hiện:**

Lặp mỗi $a \in A$ **thực hiện:**

w ← w − η∇F (w; a)

Trả kết quả tham số w cho server

Trong đó: K: danh sách client, C: Số lượng client tham gia vào mỗi vòng đào tạo, w: các tham số, P_k Dữ liệu ở client, B: Kích thước minibatch cục bộ của client, E: số lượt đào tạo mà mỗi khách hàng thực hiện trên tập dữ liệu cục bộ của mình trên mỗi vòng, ∇F (w) : gradient hàm chi phí, η: tốc độ học

Các bước thực hiện:

Bước 1: Lúc đầu, một mô hình được khởi tạo ngẫu nhiên trên máy chủ trung tâm.

Bước 2: Đối với mỗi vòng t:

- Nhóm khách hàng được chọn ngẫu nhiên
- Mỗi máy khách thực hiện các bước giảm độ dốc cục bộ
- Máy chủ tổng hợp các tham số mô hình do khách hàng gửi.

Chúng tôi sử dụng thuật toán FedAvg được trình bày trong [6] (để tổng hợp các cập nhật của khách hàng sau mỗi vòng đào tạo tại chỗ trên thiết bị để tạo ra một mô hình toàn cầu mới. Tại vòng huấn luyện t, một mô hình toàn cục với các tham số w_t, được gửi đến K thiết bị được chọn từ tập hợp thiết bị. Mỗi thiết bị có một tập dữ liệu cục bộ P_k được chia thành các lô có kích thước B. Áp dụng giảm độ dốc ngẫu nhiên (SGD) trên các máy khách để tính toán các thông số mô hình mới w_{t+1}^k. Các

trọng số máy khách này sau đó được tính trung bình trên các thiết bị, trên máy chủ, để tính toán các thông số mô hình mới w_{t+1}.

2.2.3.3.2. FEDERATED OPTIMIZATION

Non-independent identical distributions (Non-IID): Các bản phân phối giống hệt nhau không độc lập của bộ dữ liệu. Để tối ưu hóa phân tán trong một trung tâm dữ liệu, có thể đảm bảo rằng các nút điện toán khác nhau có bộ dữ liệu IID để tất cả các bản cập nhật cục bộ trông rất giống nhau. Trong tối ưu hóa có liên kết, điều này không thể được đảm bảo. Dữ liệu thuộc sở hữu của những người tham gia khác nhau có thể tuân theo các bản phân phối hoàn toàn khác nhau, nghĩa là chúng tôi không thể đưa ra các giả định IID về bộ dữ liệu phi tập trung trong học tập liên kết.

Số điểm dữ liệu không cân bằng: để tối ưu hóa phân tán trong một trung tâm dữ liệu, có thể chia đều dữ liệu giữa các nút điện toán. Tuy nhiên, trong các tình huống thực tế, những người tham gia khác nhau thường có khối lượng tập dữ liệu đào tạo rất khác nhau.

Số lượng người tham gia đông đảo: Để tối ưu hóa phân tán trong trung tâm dữ liệu, có thể dễ dàng kiểm soát số lượng nút tính toán song song. Tuy nhiên, vì ML hoặc DL thường yêu cầu nhiều dữ liệu nên các ứng dụng sử dụng phương pháp học liên kết có thể cần có nhiều người tham gia, đặc biệt là với thiết bị di động.

Liên kết giao tiếp chậm và không đáng tin cậy: Trong một trung tâm dữ liệu, các nút có thể giao tiếp nhanh chóng với nhau và các gói gần như không bao giờ bị mất trong một trung tâm dữ liệu. Tuy nhiên, trong học tập liên kết, giao tiếp giữa máy khách và máy chủ dựa trên các kết nối Internet hiện có.

Để giải quyết những thách thức trên phải đối mặt trong tối ưu hóa liên kết, McMahan et al. lần đầu tiên áp dụng thuật toán FedAvg để tối ưu hóa có liên kết [6]. Trọng tâm của FedAvg là các hàm mục tiêu không lồi thường thấy khi huấn luyện DNNs. FedAvg có thể áp dụng cho bất kỳ hàm mục tiêu tổng hữu hạn nào có dạng sau:

$$\min_{w \in R^d} f(w) - \frac{1}{n}\sum_{i=1}^{n} f_i(w),$$

Trong đó: n là số lượng dữ liệu cục bộ, w thuộc R^d đại diện cho tham số mô hình của tập dữ liệu cục bộ d.

Giả sử rằng có K người tham gia trong hệ thống HFL, với D_k biểu thị tập dữ liệu thuộc sở hữu của người tham gia thứ k, với P_k là tập chỉ mục của các điểm dữ liệu trên máy khách k. Định nghĩa $n_k = P_k$. Nghĩa là, giả định rằng người tham gia

thứ i có n_k điểm dữ liệu huấn luyện. Kết quả là, xem xét có K người tham gia, phương trình trên có thể được viết lại thành

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad where \quad F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$$

Có thể áp dụng SGD một cách đơn giản cho tối ưu hóa có liên kết, trong đó một phép tính độ dốc lô nhỏ duy nhất được thực hiện trong mỗi vòng đào tạo liên kết. Ở đây, "một vòng" đề cập đến các hoạt động gửi thông tin cập nhật từ những người tham gia đến máy chủ và từ máy chủ trở lại những người tham gia. Cách tiếp cận này hiệu quả về mặt tính toán, nhưng yêu cầu số lượng vòng đào tạo giao tiếp rất lớn để tạo ra các mô hình thỏa mãn yêu cầu.

Đối với DML, với việc đào tạo song song trong các trung tâm dữ liệu hoặc cụm máy tính, chi phí giao tiếp tương đối nhỏ và chi phí tính toán chiếm ưu thế. Các cách tiếp cận gần đây tập trung vào việc áp dụng các đơn vị xử lý đồ họa (GPU) để giảm các chi phí này. Ngược lại, trong học tập liên kết, chi phí giao tiếp chiếm ưu thế khi giao tiếp diễn ra qua Internet hoặc mạng diện rộng (WAN), ngay cả với mạng không dây và mạng di động. Trong học tập liên kết, một tập dữ liệu tại chỗ thường nhỏ so với tổng kích thước tập dữ liệu, và các thiết bị đầu cuối hiện đại (chẳng hạn như điện thoại thông minh) có bộ xử lý tương đối nhanh, thậm chí bao gồm cả GPU. Do đó, chi phí tính toán không đáng kể so với chi phí giao tiếp đối với nhiều loại mô hình trong học liên kết. Do đó, chúng tôi có thể sử dụng tính toán bổ sung để giảm số vòng giao tiếp cần thiết để đào tạo một mô hình. Sau đây là hai cách chính để thêm tính toán:

Tăng tính song song: Chúng tôi có thể thu hút thêm nhiều người tham gia làm việc độc lập giữa các vòng giao tiếp máy khách-máy chủ.

Tăng tính toán trên mỗi người tham gia: Thay vì thực hiện một phép tính đơn giản như phép tính độ dốc, mỗi máy khách thực hiện một phép tính phức tạp hơn giữa các vòng giao tiếp, chẳng hạn như thực hiện cập nhật trọng lượng nhiều mô hình trong một kỷ nguyên đào tạo.

Khởi tạo các tham số chung ở server sau đó gửi chúng cho các client cũng là một cách cho mô hình hoạt động tốt hơn so với việc mỗi client tự khởi tạo tham số ban đầu. [2]

2.2.4. THUỘC TÍNH CỦA FEDERATED LEARNING

2.2.4.1. TÍNH RIÊNG TƯ

Ưu điểm chính của việc sử dụng các phương pháp liên kết để học máy là đảm bảo quyền riêng tư hoặc bí mật dữ liệu. Thật vậy, không có dữ liệu nào của người dùng được tải lên hoặc trao đổi với máy chủ trung tâm. Vì toàn bộ cơ sở dữ liệu được phân đoạn thành các bit cục bộ, điều này khiến việc xâm nhập vào cơ sở dữ liệu trở nên khó khăn hơn.

Với học liên kết, chỉ có các tham số học máy được trao đổi. Ngoài ra, các tham số như vậy có thể được mã hóa trước khi chia sẻ giữa các vòng học để mở rộng quyền riêng tư và các lược đồ mã hóa đồng hình có thể được sử dụng để trực tiếp thực hiện các tính toán trên dữ liệu được mã hóa mà không cần giải mã chúng trước. Bất chấp các biện pháp bảo vệ như vậy, các tham số này vẫn có thể rò rỉ thông tin về các mẫu dữ liệu cơ bản, chẳng hạn như bằng cách thực hiện nhiều truy vấn cụ thể trên các tập dữ liệu cụ thể. Do đó, khả năng truy vấn của các nút là một điểm chú ý chính, có thể được giải quyết bằng cách sử dụng quyền riêng tư khác biệt và tổng hợp an toàn.

Người ta nhận thấy rằng các vấn đề về quyền riêng tư của học liên kết thường là do các ước tính đang chạy, điều này cản trở việc sử dụng các mô hình học sâu nâng cao. Một chuẩn hóa hàng loạt tĩnh (sBN - Batch Normalization) để tối ưu hóa các mạng thần kinh sâu bị hạn chế về quyền riêng tư đã được phát triển. Trong giai đoạn đào tạo, sBN không theo dõi các ước tính đang chạy mà chỉ chuẩn hóa dữ liệu theo lô. Chỉ thống kê các biểu diễn ẩn từ dữ liệu cục bộ sau khi mô hình hội tụ được tính toán. Phương pháp này phù hợp với khung FL vì các mô hình cục bộ không cần tải lên các ước tính đang chạy trong quá trình đào tạo. Các mô hình cục bộ chỉ tải lên số liệu thống kê của họ một lần sau khi tối ưu hóa, giúp giảm đáng kể rủi ro rò rỉ dữ liệu. [4]

2.2.4.2. TÍNH CÁ NHÂN HÓA

Mô hình được tạo cung cấp thông tin chi tiết dựa trên các mẫu nút toàn cầu. Tuy nhiên, nếu một nút tham gia muốn học hỏi từ các mẫu toàn cầu nhưng cũng điều chỉnh kết quả theo trạng thái đặc biệt của nó, thì phương pháp học tập liên kết có thể được điều chỉnh để tạo hai mô hình cùng một lúc trong khung học tập đa tác vụ. Ngoài ra, các kỹ thuật phân cụm có thể được áp dụng cho các nút tổng hợp có chung một số điểm tương đồng sau khi quá trình học tập hoàn tất. Điều này cho phép khái quát hóa các mô hình được học bởi các nút cũng theo dữ liệu cục bộ của chúng.

Trong trường hợp mạng nơ-ron sâu, có thể chia sẻ một số lớp trên các nút khác nhau và giữ một số lớp trên mỗi nút cục bộ. Thông thường, các lớp đầu tiên thực hiện nhận dạng mẫu chung được chia sẻ và đào tạo tất cả các bộ dữ liệu. Các lớp cuối cùng sẽ vẫn còn trên mỗi nút cục bộ và chỉ được đào tạo trên tập dữ liệu của nút cục bộ.

Các phương pháp cá nhân hóa ban đầu thường đưa ra chi phí tính toán và giao tiếp bổ sung có thể không cần thiết. Để giảm đáng kể chi phí tính toán và truyền thông trong FL, một phương pháp "Masking Trick" đã được phát triển. “Masking Trick” cho phép khách hàng địa phương đóng góp một cách thích ứng vào việc đào tạo các mô hình toàn cầu linh hoạt và hiệu quả hơn nhiều so với học tập liên kết cổ điển. [4]

2.2.4.3. TÍNH CÁ NHÂN HÓA THÔNG QUA HỌC META

Trong Học liên kết, chúng tôi mong muốn đào tạo các mô hình trên nhiều đơn vị máy tính (người dùng), trong khi người dùng chỉ có thể giao tiếp với một máy chủ trung tâm chung mà không trao đổi mẫu dữ liệu của họ. Cơ chế này khai thác sức mạnh tính toán của tất cả người dùng và cho phép người dùng có được mô hình phong phú hơn khi các mô hình của họ được đào tạo trên một tập hợp các điểm dữ liệu lớn hơn. Tuy nhiên, sơ đồ này chỉ phát triển một đầu ra chung cho tất cả người dùng và do đó, nó không điều chỉnh mô hình cho từng người dùng. Đây là một tính năng quan trọng còn thiếu, đặc biệt là do tính không đồng nhất của việc phân phối dữ liệu cơ bản cho nhiều người dùng khác nhau. [4]

Meta learning có thể được kết hợp trong việc cá nhân hóa các phương pháp học liên kết cho người dùng biên. Gần đây, phương pháp PFL(Personalized Federated Learning) đã được giới thiệu như một cách để kết hợp các khung học meta MAML(Model-Agnostic Meta-Learning) và Proto phổ biến với các phương pháp học liên kết bất khả tri không đồng nhất. Các thử nghiệm mở rộng cho thấy rằng các phương pháp PFL vượt trội hơn so với việc kết hợp các khung học tập meta với FedAvg. [7]

2.2.4.4. TÍNH PHÁP LÝ

Các khung pháp lý phương Tây ngày càng nhấn mạnh hơn vào việc bảo vệ dữ liệu và truy xuất nguồn gốc dữ liệu. Báo cáo năm 2012 của Nhà Trắng khuyến nghị áp dụng nguyên tắc giảm thiểu dữ liệu, được đề cập trong GDPR (General Data Protection Regulation) của Châu Âu. Trong một số trường hợp, việc chuyển dữ liệu từ quốc gia này sang quốc gia khác (ví dụ: dữ liệu bộ gen) là bất hợp pháp, tuy nhiên,

các tập đoàn quốc tế đôi khi cần thiết cho những tiến bộ khoa học. Trong những trường hợp như vậy, học tập liên kết mang đến các giải pháp để đào tạo một mô hình toàn cầu trong khi vẫn tôn trọng các ràng buộc bảo mật. [4]

2.2.5. ƯU, NHƯỢC ĐIỂM VÀ GIẢI PHÁP CỦA FEDERATED LEARNING

2.2.5.1. ƯU ĐIỂM

Học tập liên kết đào tạo mô hình từ nhiều nguồn dữ liệu khác nhau mà không gây rò rỉ thông tin quan trọng hay nhạy cảm của người tham gia. Nó tăng tính bảo mật cho dữ liệu của người dùng, dữ liệu không di chuyển mà chỉ có mô hình di chuyển giữa các bên. Do tính chất nổi bật này giúp mô hình có thể tiếp cận được thêm nhiều nguồn dữ liệu nhạy cảm hơn của người dùng, góp phần cải thiện mô hình và trải nghiệm của người dùng mà không mất đi tính riêng tư của họ. Federated learning cho phép một số bên hợp tác đào tạo mô hình ML để mỗi bên có thể tận hưởng một mô hình tốt hơn những gì nó có thể đạt được khi chỉ có một mình. Ví dụ, học liên kết có thể được sử dụng bởi các ngân hàng thương mại tư nhân để phát hiện việc vay nợ nhiều bên, vốn luôn là vấn đề đau đầu trong ngành ngân hàng.

Với học liên kết, không cần thiết lập cơ sở dữ liệu trung tâm và bất kỳ tổ chức tài chính nào tham gia học liên kết đều có thể bắt đầu truy vấn người dùng mới đến các cơ quan khác trong liên kết. Các cơ quan khác chỉ cần trả lời các câu hỏi về cho vay tại địa phương mà không cần biết thông tin cụ thể của người sử dụng. Điều này không chỉ bảo vệ quyền riêng tư của người dùng và tính toàn vẹn của dữ liệu mà còn đạt được mục tiêu kinh doanh quan trọng là xác định hoạt động cho vay đa bên (multi-party lending).

2.2.5.2. NHƯỢC ĐIỂM

Liên kết giao tiếp giữa chủ sở hữu dữ liệu cục bộ và máy chủ tổng hợp có thể chậm và không ổn định. Có thể có một số lượng rất lớn chủ sở hữu dữ liệu cục bộ (ví dụ: người dùng di động) khiến hệ thống không ổn định và không thể đoán trước được.

Dữ liệu từ những người tham gia khác nhau trong học tập liên kết có thể tuân theo các phân phối không giống nhau và những người tham gia khác nhau có thể có số lượng mẫu dữ liệu không cân bằng, điều này có thể dẫn đến mô hình thiên vị hoặc thậm chí thất bại trong việc đào tạo một mô hình.

Vì những người tham gia bị phân tán và khó xác thực, các cuộc tấn công đầu độc mô hình học liên kết, trong đó một hoặc nhiều người tham gia độc hại gửi các

bản cập nhật mô hình hỏng để làm cho mô hình liên kết trở nên vô dụng, có thể diễn ra và làm rối loạn toàn bộ hoạt động.

Học liên kết không áp dụng cho tất cả các ứng dụng học máy. Nếu mô hình quá lớn để chạy trên thiết bị của người dùng, thì nhà phát triển sẽ cần tìm các giải pháp khác để bảo vệ quyền riêng tư của người dùng.

Khi dữ liệu đào tạo nằm trên thiết bị của người dùng, các kỹ sư dữ liệu không có cách nào để đánh giá dữ liệu và đảm bảo rằng nó sẽ có lợi cho ứng dụng. Vì lý do này, việc học liên kết phải được giới hạn trong các ứng dụng mà dữ liệu người dùng không cần xử lý trước.

Một giới hạn khác của học máy liên kết là ghi nhãn dữ liệu. Hầu hết các mô hình học máy là học giám sát, có nghĩa là chúng yêu cầu các ví dụ đào tạo được gắn nhãn thủ công bởi người chú thích cụ thể là người dùng. Ví dụ: tập dữ liệu ImageNet là một kho lưu trữ có nguồn lực từ cộng đồng chứa hàng triệu hình ảnh và các lớp tương ứng của chúng.

Trong học tập liên kết, trừ khi kết quả có thể được suy ra từ các tương tác của người dùng (ví dụ: dự đoán từ tiếp theo mà người dùng đang nhập), các nhà phát triển không thể mong đợi người dùng bỏ qua việc gắn nhãn dữ liệu đào tạo cho mô hình học máy. [8]

2.2.5.3. GIẢI PHÁP

Mặc dù việc gửi các tham số mô hình được đào tạo đến máy chủ ít nhạy cảm về quyền riêng tư hơn gửi dữ liệu người dùng, nhưng điều đó không có nghĩa là các tham số mô hình hoàn toàn sạch dữ liệu riêng tư. Trên thực tế, nhiều thử nghiệm đã chỉ ra rằng các mô hình học máy được đào tạo có thể ghi nhớ dữ liệu người dùng và Membership Inference Attacks [9] có thể tạo lại dữ liệu đào tạo trong một số mô hình thông qua thử và sai.

Một biện pháp khắc phục quan trọng đối với những lo ngại về quyền riêng tư của học liên kết là loại bỏ các mô hình do người dùng đào tạo sau khi chúng được tích hợp vào mô hình trung tâm. Máy chủ đám mây không cần lưu trữ các mô hình riêng lẻ sau khi cập nhật mô hình cơ sở của nó.

Cuối cùng, bằng cách thêm một chút nhiễu vào các tham số được đào tạo và sử dụng các kỹ thuật chuẩn hóa, các nhà phát triển có thể giảm đáng kể khả năng ghi nhớ dữ liệu của người dùng của mô hình.

Học liên kết đang trở nên phổ biến vì nó giải quyết một số vấn đề cơ bản của trí tuệ nhân tạo hiện đại. Các nhà nghiên cứu đang liên tục tìm kiếm những cách mới

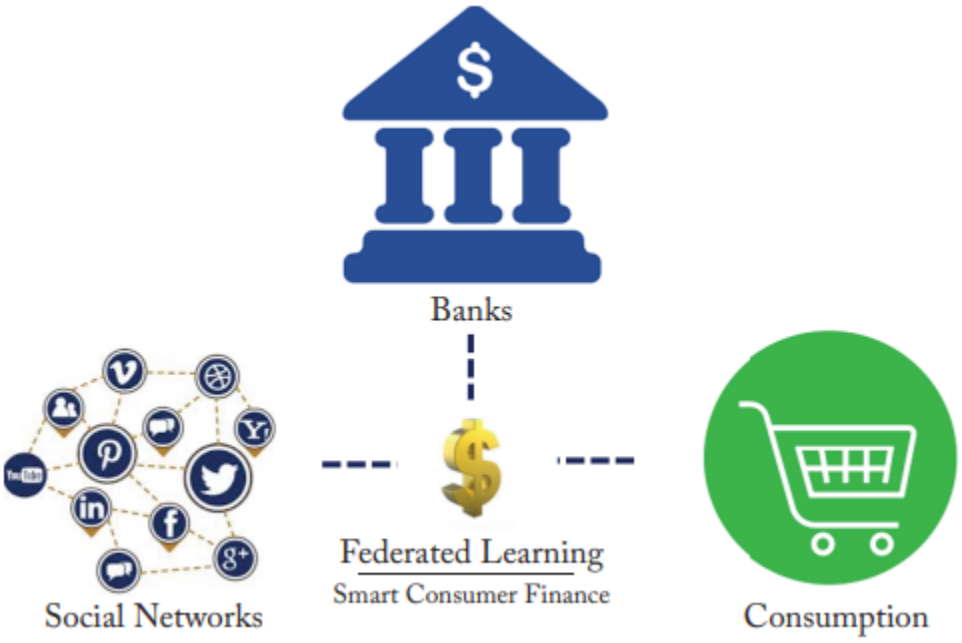
để áp dụng học liên kết vào các ứng dụng AI mới và vượt qua các giới hạn của nó. Sẽ rất thú vị để xem lĩnh vực này phát triển như thế nào trong tương lai. [8]

CHƯƠNG 3: ỨNG DỤNG

3.1. LĨNH VỰC ỨNG DỤNG

3.1.1. TÀI CHÍNH NGÂN HÀNG

Ngành tài chính bị ảnh hưởng rất nhiều bởi các quy định của chính phủ theo nhiều cách để bảo vệ các nhà đầu tư khỏi sự quản lý yếu kém và gian lận, duy trì sự ổn định của ngành tài chính, bảo vệ quyền riêng tư và bảo mật của dữ liệu người dùng, v.v. Để tiết kiệm chi phí và khối lượng công việc từ các quy định của chính phủ, nhiều công ty tài chính và ngân hàng đã khai thác các công nghệ hiện đại như AI, dịch vụ đám mây và công nghệ Internet di động để cung cấp dịch vụ tài chính hiệu quả và tuân thủ các quy định nghiêm ngặt của chính phủ.



Hình 11: Học liên kết trong tài chính tiêu dùng thông minh [2]

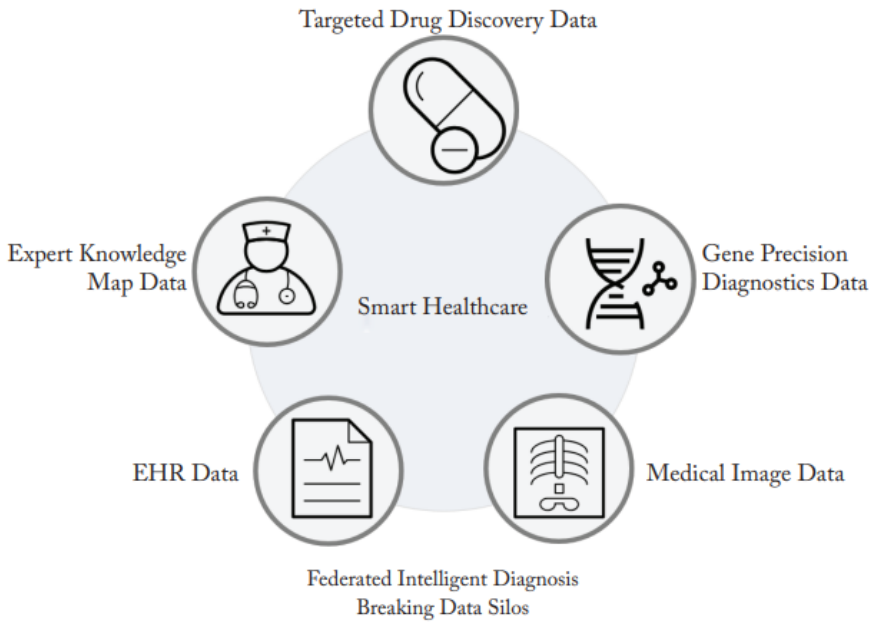
Lấy tài chính tiêu dùng thông minh làm ví dụ. Mục đích là tận dụng các kỹ thuật ML để cung cấp các dịch vụ tài chính được cá nhân hóa cho những người tiêu dùng đáng tin cậy để khuyến khích tiêu dùng. Các tính năng dữ liệu liên quan đến tài chính tiêu dùng chủ yếu bao gồm thông tin về trình độ của người tiêu dùng, sức mua và sở thích mua hàng, cũng như các đặc tính của sản phẩm. Trong các ứng dụng thực tế, các tính năng dữ liệu này có thể được thu thập bởi các phòng ban hoặc công

ty khác nhau. Ví dụ (Hình 11), thông tin về trình độ và sức mua của người tiêu dùng có thể được suy ra từ khoản tiết kiệm ngân hàng của cô ấy và sở thích mua hàng của cô ấy đối với các sản phẩm hoặc dịch vụ khác nhau có thể được phân tích từ các mạng xã hội của cô ấy. Các đặc điểm của sản phẩm được ghi lại bởi một cửa hàng điện tử. Trong kịch bản này, chúng ta phải đối mặt với hai vấn đề. Đầu tiên, để bảo vệ quyền riêng tư và bảo mật dữ liệu của người dùng, các rào cản dữ liệu giữa các ngân hàng, trang mạng xã hội và trang mua sắm điện tử rất khó bị phá vỡ. Do đó, dữ liệu không thể được tổng hợp trực tiếp. Thứ hai, dữ liệu được lưu trữ bởi ba bên thường không đồng nhất và các mô hình ML truyền thống không thể hoạt động trực tiếp trên dữ liệu không đồng nhất. Hiện tại, những vấn đề này vẫn chưa được giải quyết hiệu quả bằng các phương pháp ML truyền thống.

Học liên kết và học chuyển là chìa khóa để giải quyết những vấn đề này. Đầu tiên, dựa trên học tập liên kết, chúng tôi có thể xây dựng các mô hình được cá nhân hóa cục bộ cho ba bên mà không làm lộ dữ liệu của họ. Trong khi đó, chúng ta có thể tận dụng học chuyển đổi để giải quyết vấn đề về tính không đồng nhất của dữ liệu và khắc phục những hạn chế của các kỹ thuật AI truyền thống. Do đó, học tập liên kết cung cấp hỗ trợ kỹ thuật tuyệt vời để chúng tôi xây dựng một hệ sinh thái liên doanh nghiệp, dữ liệu chéo và miền chéo cho dữ liệu lớn và AI. [2]

3.1.2. Y HỌC: CHĂM SÓC SỨC KHỎE

Với sự tiến bộ của công nghệ AI, nhiều ứng dụng AI đã được phát triển trong lĩnh vực y tế với hy vọng giảm chi phí lao động và sai sót của con người. Ví dụ, các chương trình AI dành cho tim mạch và X quang đã được phát triển để giúp chẩn đoán bệnh tim và xác định tế bào ung thư ở giai đoạn đầu. Với những ứng dụng đầy hứa hẹn của AI về sức khỏe, ngày càng có nhiều nhà cung cấp dịch vụ chăm sóc sức khỏe tận dụng AI để tạo ra hiệu quả và cải thiện việc chăm sóc bệnh nhân (Hình 12).



Hình 12: Học liên kết trong chẩn đoán thông minh [2]

Tuy nhiên, việc áp dụng các công nghệ AI trong ngành y tế vẫn còn ở giai đoạn sơ khai. Các hệ thống y tế thông minh hiện tại còn lâu mới thực sự là "trí thông minh" và một số đang bị nghi ngờ vì đưa ra các khuyến nghị điều trị không an toàn và không chính xác. Nhiều yếu tố có thể góp phần vào sự thiếu hụt của các hệ thống y tế thông minh hiện có. Một vấn đề quan trọng là khó khăn trong việc thu thập một lượng dữ liệu đủ lớn với các tính năng phong phú có thể mô tả toàn diện triệu chứng của bệnh nhân. Ví dụ, để chẩn đoán chính xác một căn bệnh, chúng tôi có thể cần các đặc điểm từ nhiều nguồn khác nhau bao gồm các triệu chứng bệnh, trình tự gen, báo cáo y tế, kết quả kiểm tra và bài báo học thuật. Tuy nhiên, không có nguồn dữ liệu ổn định để điền giá trị của tất cả các tính năng đó. Bên cạnh đó, nhãn của phần lớn dữ liệu đào tạo bị thiếu. Các nhà nghiên cứu ước tính rằng sẽ mất 10 năm với 10.000 chuyên gia để thu thập một bộ dữ liệu đủ hữu ích để phát triển AI trong chăm sóc sức khỏe. Việc thiếu dữ liệu và nhãn dẫn đến hiệu suất kém của các mô hình ML trở thành nút cổ chai của các hệ thống y tế thông minh.

Để vượt qua nút cổ chai này, các tổ chức y tế có thể đoàn kết với nhau bằng cách chia sẻ dữ liệu của họ tuân thủ các quy định bảo vệ quyền riêng tư. Sau đó, chúng ta có thể sở hữu một tập dữ liệu đủ lớn để đào tạo một mô hình có thể hoạt động tốt hơn nhiều so với mô hình được đào tạo dựa trên dữ liệu từ một tổ chức y tế duy nhất. Kết hợp học liên kết với học chuyển là một giải pháp hứa hẹn để đạt được mục tiêu này. Đầu tiên, dữ liệu từ các tổ chức y tế rất nhạy cảm với các vấn đề về

quyền riêng tư và bảo mật. Việc thu thập trực tiếp dữ liệu như vậy tại một địa điểm là không khả thi. Học liên kết cho phép tất cả các bên tham gia hợp tác đào tạo một mô hình dùng chung mà không cần trao đổi hoặc tiết lộ dữ liệu bệnh nhân riêng tư của họ. Thứ hai, các kỹ thuật học chuyển giao có thể giúp mở rộng không gian mẫu và đặc trưng của dữ liệu huấn luyện, từ đó cải thiện hiệu suất của mô hình được chia sẻ. Do đó, học chuyển giao liên kết có thể đóng một vai trò quan trọng trong việc phát triển các hệ thống y tế thông minh. Nếu một số lượng kha khá các tổ chức y tế có thể thiết lập một liên minh dữ liệu với nhau thông qua học tập liên kết trong tương lai, AI về sức khỏe có thể mang lại nhiều lợi ích hơn cho nhiều bệnh nhân hơn. [2]

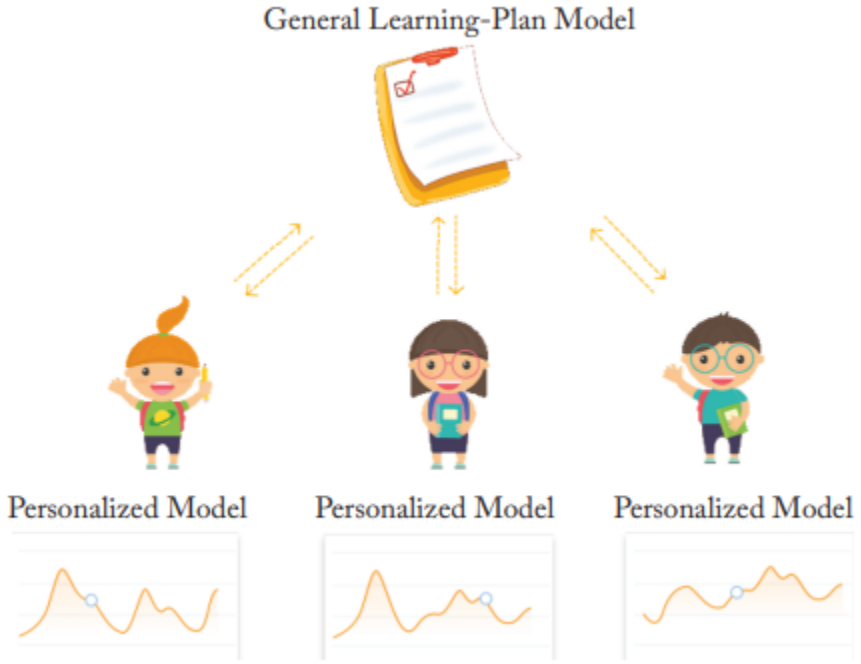
3.1.3. GIÁO DỤC

Các nhà giáo dục từ lâu đã kêu gọi các hệ thống giảng dạy tích hợp các môn học ngoại khóa (ví dụ: giữa các môn khoa học, công nghệ, kỹ thuật và toán học (STEM) cũng như giữa STEM và nhân văn). Tuy nhiên, các hệ thống giảng dạy hiếm khi có thể xử lý các kỹ năng tiên quyết, nền tảng kiến thức và kinh nghiệm cần thiết để cung cấp trải nghiệm học tập tích hợp như vậy. Một hệ thống hướng dẫn thích ứng điển hình (AIS) giải quyết một chủ đề duy nhất tại một thời điểm và nó thường có một bản thể luận nội dung, công cụ thích ứng và phương pháp quản lý dữ liệu duy nhất. Ví dụ: bản thể luận AIS toán học thường bao gồm một sơ đồ tri thức về các mục tiêu học tập chi tiết trong toán học, nhưng nó có thể có nhiều kết nối với các mục tiêu trong vật lý và hóa học. Ví dụ, kiến thức tính toán của học sinh có thể cung cấp thông tin cho trải nghiệm học tập của họ về vật lý hoặc hóa học. Do đó, việc tích hợp các bản thể luận trong các hệ thống giảng dạy sẽ không chỉ mở rộng phạm vi của nhiều AIS mà còn hỗ trợ trải nghiệm học tập thích ứng đa chương trình, phong phú hơn cho học sinh.

Để đạt được điều này, chúng ta có thể mã hóa biểu đồ tri thức của từng AIS dưới dạng một mạng lưới thần kinh biểu đồ được chứng minh là có sức mạnh biểu diễn cao. Sau đó, chúng ta có thể sử dụng các phương pháp tiếp cận dựa trên học tập có liên kết để xây dựng một mô hình toàn diện tích hợp các biểu đồ thần kinh kiến thức của nhiều AIS khác nhau, từ đó mở rộng kiến thức ngoại khóa, mô hình người học và tiếp cận dữ liệu từ AIS này sang AIS khác. Theo cách này, mỗi AIS đều được hưởng lợi từ việc đồng bộ hóa dữ liệu, giảm độ trễ và các tính năng bảo mật của một hệ thống được liên kết trong khi vẫn duy trì bản thể luận, công cụ thích ứng và dữ liệu của riêng mình.

Ngoài việc tích hợp các tài nguyên giáo dục, học tập liên kết có thể giúp đạt được giáo dục cá nhân hóa (Hình 13). Cụ thể hơn, các tổ chức giáo dục có thể sử

dụng học tập liên kết để cộng tác xây dựng mô hình lập kế hoạch học tập chung dựa trên dữ liệu được lưu trữ trên các thiết bị di động cá nhân của học sinh như điện thoại thông minh, iPad và máy tính xách tay. Mô hình chung có thể xây dựng một kế hoạch học tập tiêu chuẩn cho những sinh viên có nền tảng tương tự. Ngoài mô hình chung đó, có thể xây dựng một mô hình cá nhân hóa có thể cung cấp các hướng dẫn học tập được cá nhân hóa cho từng học sinh dựa trên điểm mạnh, nhu cầu, kỹ năng và sở thích của học sinh đó. [2]



Hình 13: Học liên kết trong giáo dục [2]

3.1.4. MẠNG DI ĐỘNG 5G

Học tập liên kết cũng đã trở thành một chủ đề nghiên cứu tích cực tại giao điểm của ML và mạng không dây hơn, và đặc biệt đối với mạng di động 5G và thậm chí xa hơn nữa. Chẳng hạn, dữ liệu trong các mạng không dây thường được đặt ở người dùng và ở rìa mạng, điều này làm cho ML truyền thống dựa trên việc thu thập dữ liệu tập trung không thể áp dụng được. Học liên kết là một giải pháp để giải quyết không chỉ các mối lo ngại về quyền riêng tư dữ liệu mà còn cả các thách thức về băng thông, độ tin cậy và độ trễ của giao tiếp. Học liên kết cũng có thể giúp xây dựng một mạng không dây tốt hơn. Ví dụ, cung cấp thông tin tổng quan về cách chúng ta có thể tận dụng học tập liên kết để giải quyết các thách thức chính và cải thiện hiệu suất của mạng di động 5G. [2]

3.2. TÌNH HÌNH PHÁT TRIỂN HIỆN TẠI

Ý tưởng về học tập liên kết đã xuất hiện dưới nhiều hình thức khác nhau trong suốt lịch sử khoa học máy tính, chẳng hạn như: privacy-preserving ML, privacy-preserving DL, distributed ML, federated optimization, …

Học liên kết đã được Google nghiên cứu trong một bài báo nghiên cứu xuất bản năm 2016 trên arXiv (một kho lưu trữ các bản in điện tử được lưu trữ bởi Đại học Cornell). Kể từ đó, đây là một lĩnh vực nghiên cứu tích cực trong cộng đồng AI, bằng chứng là khối lượng bản in tăng nhanh khi xuất hiện trên arXiv.

Công trình nghiên cứu gần đây về học liên kết chủ yếu tập trung vào việc cải thiện các thách thức về bảo mật và thống kê. Trình bày một khung học tập chuyển giao được liên kết linh hoạt có thể được điều chỉnh một cách hiệu quả cho các tác vụ ML đa bên an toàn khác nhau. Trong khuôn khổ này, liên kết cho phép chia sẻ kiến thức mà không ảnh hưởng đến quyền riêng tư của người dùng và cho phép truyền kiến thức bổ sung trong mạng thông qua học tập chuyển giao.

Trong một hệ thống học tập liên kết, chúng ta có thể cho rằng các bên tham gia là trung thực, bán trung thực hoặc độc hại. Khi một bên là độc hại, một mô hình có thể làm hỏng dữ liệu của nó trong quá trình đào tạo. Khả năng các cuộc tấn công đầu độc mô hình vào học tập liên kết do một tác nhân độc hại. Một số chiến lược để thực hiện tấn công đầu độc mô hình đã được nghiên cứu. Nó đã chỉ ra rằng ngay cả một đối thủ bị hạn chế cao cũng có thể thực hiện các cuộc tấn công đầu độc mô hình đồng thời duy trì khả năng tàng hình.

Việc kiểm tra lại các mô hình ML hiện có trong cài đặt học liên kết đã trở thành một hướng nghiên cứu mới.Ví dụ, kết hợp học liên kết với học tăng cường.

Học liên kết cũng đã được áp dụng trong các lĩnh vực thị giác máy tính (CV), ví dụ: phân tích hình ảnh y tế, xử lý ngôn ngữ tự nhiên (NLP), và hệ thống khuyến nghị (RS). Về các ứng dụng của học liên kết, các nhà nghiên cứu tại Google đã áp dụng học liên kết trong dự đoán bàn phím di động, đã đạt được sự cải thiện đáng kể về độ chính xác của dự đoán mà không làm lộ dữ liệu người dùng di động. Các nhà nghiên cứu tại Firefox đã sử dụng phương pháp học liên kết để dự đoán từ tìm kiếm. [2]

3.3. TƯƠNG LAI CỦA FEDERATED LEARNING

Trong thời đại dữ liệu người dùng được xem trọng, nó sinh ra một cách nhanh chóng với số lượng rất lớn. Với việc ứng dụng của AI càng ngày càng nhiều, len lỏi khắp các lĩnh vực của đời sống con người thì các điều luật liên quan đến bảo vệ dữ liệu người dùng cũng được sinh ra và nghiêm ngặt hơn.

Federated Learning ra đời do mối lo ngại ngày càng tăng về phân mảnh dữ liệu, kho lưu trữ dữ liệu, rò rỉ quyền riêng tư của người dùng và các vấn đề thiếu dữ liệu mà máy học phải đối mặt. Chúng ta đang nhận thức được những tác động nghiêm trọng của việc các tập đoàn lớn vi phạm quyền riêng tư của người dùng và các cơ quan quản lý đang thắt chặt luật quản lý việc chia sẻ dữ liệu riêng tư, chẳng hạn như các yêu cầu nghiêm ngặt nhất của GDPR về bảo mật dữ liệu. Như vậy, rõ ràng kỹ thuật Federated Learning đang trở thành một phương pháp tốt, giải quyết bài toán đau đầu hiện nay của nhân loại là thu thập dữ liệu để phục vụ nghiên cứu nhưng làm sao để những dữ liệu ấy không bị khai thác bất hợp pháp và ảnh hưởng đến quyền riêng tư của các cá nhân người dùng. Các phương pháp học máy truyền thống dựa trên việc thu thập dữ liệu tập trung không còn tuân thủ các luật bảo vệ dữ liệu nghiêm ngặt, nên để lĩnh vực AI tiếp tục phát triển, một giải pháp sáng tạo có thể bảo vệ quyền riêng tư của dữ liệu là rất cần thiết.

Do đó các phương pháp đào tạo mô hình tương tự như Federated Learning đang dần trở thành xu thế và tương lai của lĩnh vực AI. Các học viên sẽ quen với việc xây dựng các giải pháp có tất cả các khía cạnh cần thiết mà xã hội mong đợi và học tập liên kết sẽ trở thành một ví dụ điển hình về "AI for Social Good" (AI vì lợi ích xã hội). [2]

CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG THUẬT TOÁN

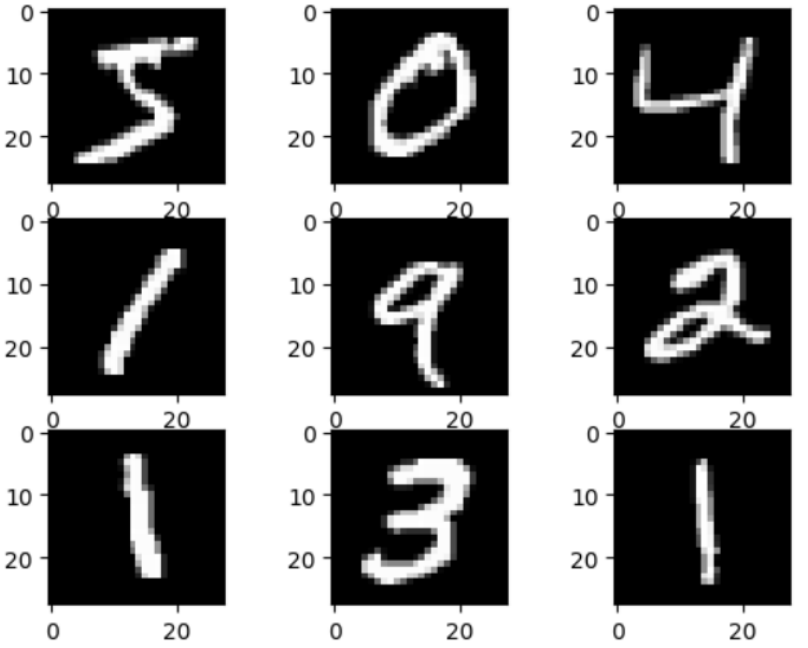
4.1. MỤC TIÊU

Xây dựng mô hình đào tạo dữ liệu kết hợp cách triển khai học liên kết cho tập dữ liệu MNIST để nhận dạng chữ số viết tay.

4.2. DỮ LIỆU

MNIST là một bộ dữ liệu bao gồm các hình ảnh viết tay được chuẩn hóa và cắt ở giữa. Nó có hơn 60.000 hình ảnh đào tạo và 10.000 hình ảnh thử nghiệm. Đây là một trong những bộ dữ liệu được sử dụng nhiều nhất cho mục đích học tập và thử nghiệm. Để tải và sử dụng tập dữ liệu, chúng ta có thể nhập bằng cú pháp bên dưới sau khi cài đặt gói torchvision:

>>> torchvision.datasets.MNIST()



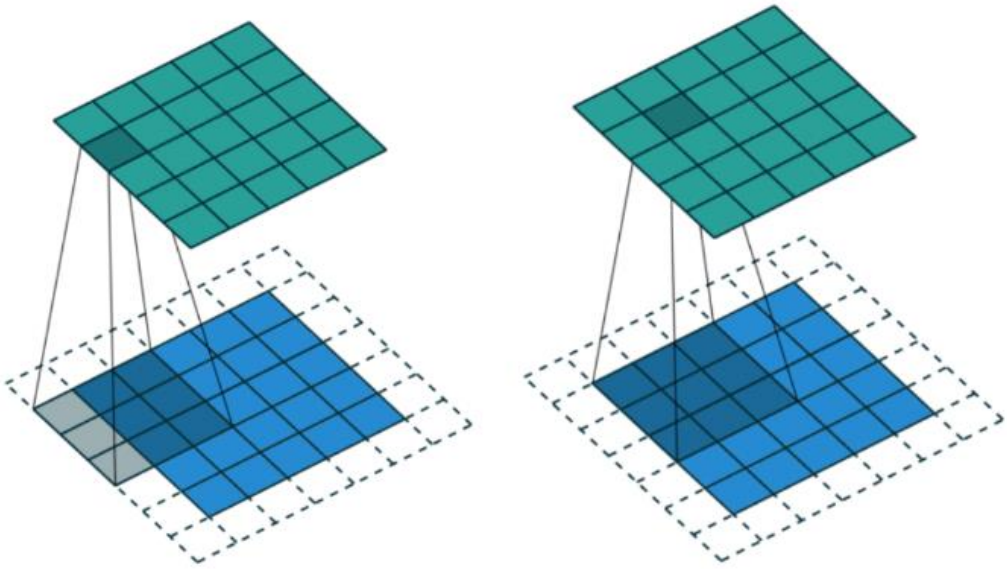
Hình 14: Hình ảnh ở trên mô tả chữ viết tay của tập dữ liệu MNIST

4.3. THUẬT TOÁN CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network (CNN) là một trong những mô hình học sâu được dùng phổ biến nhất để phân tích hình ảnh cho các tác vụ thị giác máy tính. Mô hình tính toán neural này chứa nhiều lớp tích chập có thể kết nối hoàn toàn hoặc được gộp chung. Các lớp tích chập này được thực hiện trích xuất các đặc trưng nổi trội của dữ liệu đầu vào, chính từ ưu điểm đó chúng thường được sử dụng để xử lý ảnh và cho thấy sự chính xác rất cao, mô hình được sử dụng trong nhiều ứng dụng

tiến tiến của trí tuệ nhân tạo, bao gồm nhận dạng khuôn mặt , xử lý ngôn ngữ tự nhiên và còn nhiều lĩnh vực khác. CNN có các lớp ẩn được gọi là lớp tích chập và các lớp này là thứ tạo nên CNN.

Cũng giống như bất kỳ lớp nào khác, lớp tích chập nhận đầu vào, biến đổi đầu vào theo một cách nào đó, sau đó xuất đầu vào đã chuyển đổi sang lớp tiếp theo. Với lớp tích chập, phép biến đổi xảy ra được gọi là phép tích chập (convolution operation). Đây là thuật ngữ được sử dụng bởi cộng đồng deep learning. Về mặt toán học, các hoạt động tích chập được thực hiện bởi các lớp tích chập thực sự được gọi là tương quan chéo (cross-correlations). Với mỗi lớp tích chập, chúng ta cần chỉ định số lượng bộ lọc mà lớp đó nên có. Những bộ lọc này thực sự là những gì phát hiện các mẫu (các cạnh, góc, hình dạng, kết cấu, đối tượng, màu sắc). Mạng càng đi sâu, các bộ lọc càng tinh vi. Ở các lớp sau, thay vì các cạnh và hình dạng đơn giản, bộ lọc của chúng tôi có thể phát hiện các đối tượng cụ thể như mắt, tai, tóc hoặc lông thú, lông vũ, vảy và mỏ. Ở các lớp sâu hơn nữa, các bộ lọc có thể phát hiện các vật thể phức tạp hơn như chó, mèo, thằn lằn và chim. Về mặt kỹ thuật, một bộ lọc có thể được coi là một ma trận tương đối nhỏ.



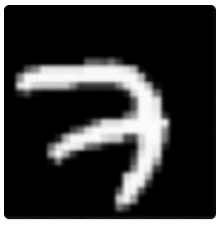
Hình 15: Quá trình tích chập không có số [10]

Hình ảnh ở trên giới thiệu quá trình tích chập không có số. Chúng tôi có một kênh đầu vào màu xanh lam ở phía dưới. Bộ lọc tích chập 3x3 được tô bóng ở phía dưới đang trượt qua kênh đầu vào và kênh đầu ra màu xanh lục. Đối với mỗi vị trí trên kênh đầu vào màu xanh lam, bộ lọc 3 x 3 thực hiện phép tính ánh xạ phần bóng mờ của kênh đầu vào màu xanh dương sang phần bóng mờ tương ứng của kênh đầu

ra màu xanh lá cây. Lớp màn trập này nhận một kênh đầu vào và bộ lọc sẽ trượt qua từng bộ 3 x 3 pixel của đầu chính cho đến khi nó trượt qua từng khối 3 x 3 pixel từ toàn bộ hình ảnh.

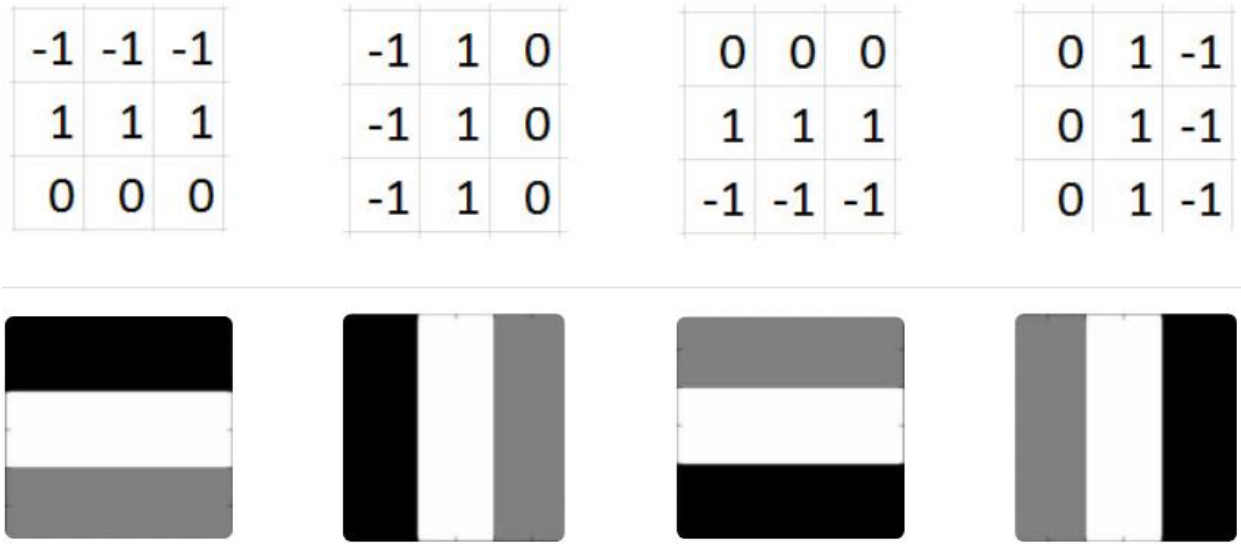
Sau khi bộ lọc này đã kết hợp toàn bộ đầu vào, chúng sẽ để lại một biểu diễn mới cho đầu vào của chúng ta, hiện được lưu trữ trong kênh đầu ra. Kênh đầu ra này được gọi là bản đồ đặc trưng (feature map). Kênh đầu ra màu xanh lục này trở thành kênh đầu vào cho lớp tiếp theo làm đầu vào và sau đó quá trình mà chúng ta vừa thực hiện với bộ lọc này sẽ xảy ra với kênh đầu ra mới này bằng các bộ lọc của lớp tiếp theo.

Giả sử rằng hình ảnh số 7 thang độ xám từ tập dữ liệu MNIST là đầu vào của chúng ta:



Hình 16: Hình ảnh số 7 [10]

Giả sử rằng chúng ta có bốn bộ lọc 3 x 3 cho lớp tích chập đầu tiên và những bộ lọc này chứa đầy các giá trị. Các giá trị này có thể được biểu diễn trực quan bằng cách -1 tương ứng với màu đen, 1 tương ứng với màu trắng và 0 tương ứng với màu xám.



Hình 17: Bộ lọc cho lớp tích chập đầu tiên [10]

Nếu chúng ta kết hợp hình ảnh ban đầu của chúng ta với từng bộ lọc trong số bốn bộ lọc này, thì kết quả đầu ra sẽ giống như hình bên dưới cho mỗi bộ lọc:

Hình 18: Kết quả sau khi áp dụng bộ lọc [10]

Chúng ta có thể thấy rằng cả bốn bộ lọc này đều đang phát hiện ra các cạnh. Trong các kênh đầu tiên, các điểm ảnh sáng nhất có thể hiểu được là những thứ mà bộ lọc đã phát hiện. Trong lần đầu tiên, chúng ta có thể thấy các cạnh nằm ngang trên cùng của cạnh thứ bảy đã được phát hiện và điều đó được biểu thị bằng các pixel sáng nhất (màu trắng).Thứ hai phát hiện các cạnh dọc bên trái, một lần nữa được hiển thị với các pixel sáng nhất. Cái thứ ba phát hiện các cạnh ngang dưới cùng và cái thứ tư phát hiện các cạnh dọc bên phải.

4.4. TRIỂN KHAI MÔ HÌNH

Tạo một đối tượng parser để lưu các tham số cho việc đào tạo mô hình giúp cho việc dễ điều chỉnh tham số:

```
parser = argparse.ArgumentParser()
# federated arguments
parser.add_argument('--epochs', type=int, default=10, help="rounds of training")
parser.add_argument('--num_users', type=int, default=100, help="number of users: K")
parser.add_argument('--frac', type=float, default=0.1, help="the fraction of clients: C")
parser.add_argument('--local_ep', type=int, default=5, help="the number of local epochs: E")
parser.add_argument('--local_bs', type=int, default=10, help="local batch size: B")
parser.add_argument('--lr', type=float, default=0.01, help="learning rate")
```

Hình 19: Khai báo các tham số

Đối tượng parser để lưu các tham số như: epochs (số lần lặp cho việc đào tạo ở server), num_users (số lượng client tham gia: K), frac (số client tham gia mỗi vòng lặp ở server: C), local_ep (số vòng lặp ở client: E), local_bs (kích thước các lô dữ liệu ở client), lr (tốc độ học tập, bước nhảy) và các tham số khác liên quan đến mô hình.

Hàm thực hiện chia nhỏ dữ liệu ở máy client, sau đó đào tạo, cập nhật các tham số của mô hình cho dữ liệu ở client và trả về các tham số cho server:

```python
class LocalUpdate(object):
    def __init__(self, args, dataset=None, idxs=None):
        self.args = args
        #CrossEntropyLoss: Tiêu chí này tính toán tổn thất entropy chéo giữa đầu vào và mục tiêu
        self.loss_func = nn.CrossEntropyLoss()
        self.selected_clients = []
        self.ldr_train = DataLoader(DatasetSplit(dataset, idxs), batch_size=self.args.local_bs, shuffle=True)

    def train(self, net):
        # train and update
        net.train()
        # Thực hiện giảm độ dốc ngẫu nhiên (tùy chọn với động lượng(momentum))
        optimizer = torch.optim.SGD(net.parameters(), lr=self.args.lr, momentum=self.args.momentum)
        epoch_loss = []
        for iter in range(self.args.local_ep):
            batch_loss = []
            for batch_idx, (images, labels) in enumerate(self.ldr_train):
                images, labels = images.to(self.args.device), labels.to(self.args.device)
                net.zero_grad()
                log_probs = net(images)
                loss = self.loss_func(log_probs, labels)
                loss.backward()
                optimizer.step()
                if self.args.verbose and batch_idx % 10 == 0:
                    print('Update Epoch: {} [{}/{} ({:.0f}%)]\tLoss: {:.6f}'.format(
                        iter, batch_idx * len(images), len(self.ldr_train.dataset),
                                100. * batch_idx / len(self.ldr_train), loss.item()))
                batch_loss.append(loss.item())
            epoch_loss.append(sum(batch_loss)/len(batch_loss))
        return net.state_dict(), sum(epoch_loss) / len(epoch_loss)
```

Hình 20: Chia dữ liệu và thực hiện đào tạo ở client

Hàm tính toán trung bình các trọng số FedAvg, thực hiện tính tổng các trọng số nhận được từ client sau đó tính trung bình các trọng số này:

```python
def FedAvg(w):
    w_avg = copy.deepcopy(w[0])
    for k in w_avg.keys():
        for i in range(1, len(w)):
            w_avg[k] += w[i][k]
        w_avg[k] = torch.div(w_avg[k], len(w))
    return w_avg
```

Hình 21: Thuật toán FedAvg

Lưu mô hình để sử dụng cho việc phân loại chữ số viết tay:

```python
model_scripted = torch.jit.script(net_glob) # Export to TorchScript
model_scripted.save('model_scripted.pt')
```

Hình 22: Lưu mô hình

4.5. KẾT QUẢ

4.5.1. THỬ NGHIỆM MÔ HÌNH VỚI CÁC GIÁ TRỊ KHÁC NHAU

Ở dưới chúng tôi thử với số lần lặp ở server (epochs) là 40 lần lặp, kích thước mini-batch ở client (local_bs) là 10, 20 và số lần lặp ở client (local_epochs) là 5, 10. Kết quả của việc training thể hiện trong hình dưới.

```
PS C:\Users\84947\Downloads\federated-learning-master> py -3.10 .\main_fed.py --epochs 40

Training with: epochs=40, local_bs=10, local_epochs=5
Validation accuracy: 97.81

Training with: epochs=40, local_bs=20, local_epochs=5
Validation accuracy: 97.27

Training with: epochs=40, local_bs=10, local_epochs=10
Validation accuracy: 98.12

Training with: epochs=40, local_bs=20, local_epochs=10
Validation accuracy: 98.01

============Select Best Model================
Best model have: epochs=40, local_bs=10, local_epochs=10
Test accuracy: 98.51
Saved model!!!
PS C:\Users\84947\Downloads\federated-learning-master>
```

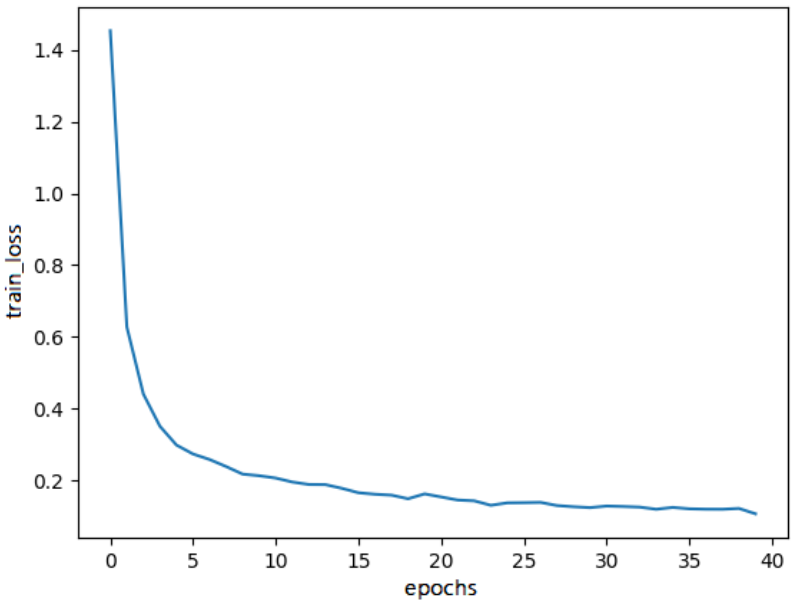
Hình 23: Kết quả mô hình sau khi đào tạo

Dựa vào kết quả hiện tại thì mô hình cho kết quả tốt nhất với local_bs=10 và local_epochs=10

```
Round  25, Average loss 0.139
Round  26, Average loss 0.137
Round  27, Average loss 0.139
Round  28, Average loss 0.137
Round  29, Average loss 0.126
Round  30, Average loss 0.124
Round  31, Average loss 0.119
Round  32, Average loss 0.132
Round  33, Average loss 0.119
Round  34, Average loss 0.135
Round  35, Average loss 0.120
Round  36, Average loss 0.119
Round  37, Average loss 0.119
Round  38, Average loss 0.124
Round  39, Average loss 0.106
Validation accuracy: 98.14
```

Hình 24: Giá trị trung bình hàm loss trong khi đào tạo mô hình

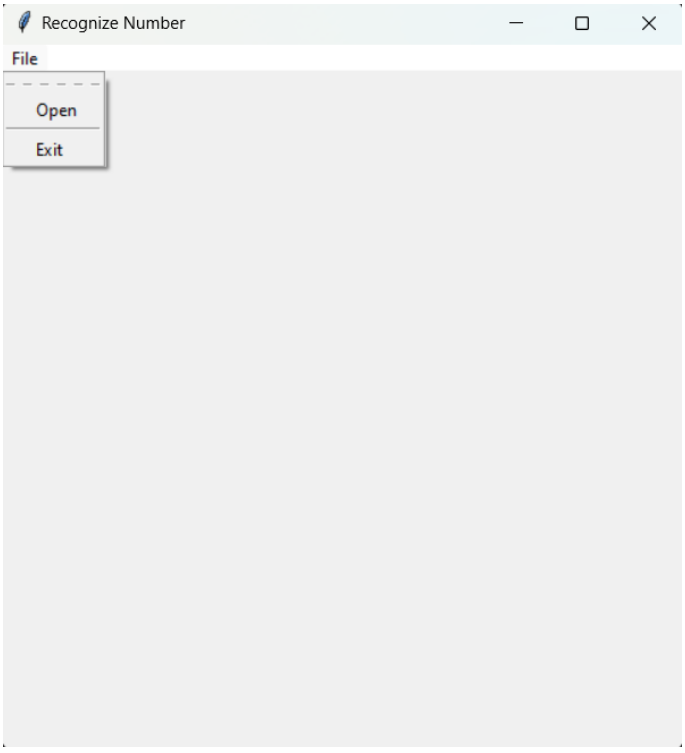
Mô hình đạt độ chính xác ≈ 98.1% trên tập validation và ≈ 98.5% trên tập test. Kết quả đạt được khá là tốt.



Hình 25: Biểu đồ thay đổi của hàm mất mát trong quá trình đào tạo (với các tham số epochs=40, local_bs=10 và local_epochs=10)

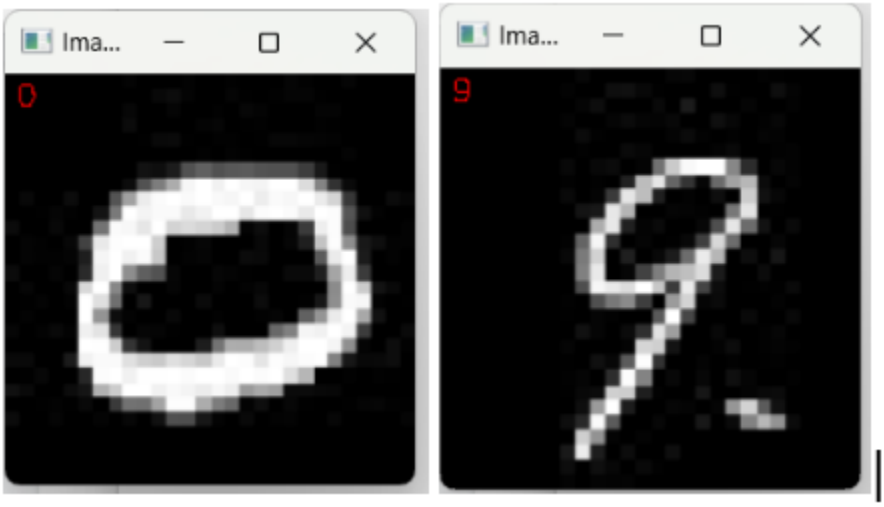
4.5.2. THỰC HIỆN NHẬN DẠNG ẢNH

Giao diện ứng dụng như hình dưới, sẽ có menu open để chúng ta có thể chọn ảnh cho việc phân loại chữ số viết tay. Ứng dụng này sử dụng mô hình chúng ta lưu sau khi đã đào tạo trước đó.



Hình 26: Giao diện app nhận dạng chữ số viết tay

Chọn hình ảnh để nhận dạng, dưới đây là kết quả nhận dạng ảnh đã chọn. Mô hình đã nhận dạng đúng với ảnh số 0 và số 9 mà chúng ta đã chọn:



Hình 27: Kết quả nhận dạng chữ số viết tay số 0 và 9

4.6. KẾT LUẬN

Mô hình được xây dựng dựa theo thuật toán CNN kết hợp học liên kết cho ra được độ chính xác khá tốt, hơn 98%.

CHƯƠNG 5: KẾT LUẬN

5.1. KẾT QUẢ ĐẠT ĐƯỢC

5.1.1. Ý NGHĨA KHOA HỌC

Báo cáo đã nêu lên một trong những thách thức bao trùm của kỷ nguyên kỹ thuật số là quyền riêng tư của dữ liệu. Các điều luật về bảo vệ thông tin của người dùng ngày càng được thắt chặt, trong khi các mô hình học máy hiện nay chưa đảm bảo tính riêng tư của dữ liệu nên xu hướng của trí tuệ nhân tạo sẽ phát triển theo hướng đáp ứng được các điều luật về bảo vệ thông tin của người dùng. Nội dung chính của báo cáo là đưa ra và trình bày về một giải pháp giúp bảo vệ thông tin người dùng trong quá trình đào tạo mô hình mà kết quả đào tạo không bị ảnh hưởng quá nhiều, giải pháp đó là federated learning. Thông qua đề tài này, chúng tôi nắm bắt được khái niệm của federated learning, cách hoạt động cũng như triển khai mô hình để nhận dạng chữ số viết tay. Thông qua việc triển khai mô hình, chúng tôi cũng nâng thêm hiểu biết và kỹ năng sử dụng python và các thư viện hỗ trợ việc triển khai mô hình. Bên cạnh đó, chúng tôi còn nâng cao thêm được khả năng đọc hiểu tài liệu, khả năng làm việc nhóm và khả năng trình bày báo cáo khoa học.

5.1.2. Ý NGHĨA THỰC TIỄN

Chúng tôi biết được biết được tầm quan trọng của bảo vệ dữ liệu trong thời đại kỹ thuật số hiện nay. Được tìm hiểu về một mô hình mới nổi đang vượt lên trên cả hệ thống tập trung (centralized systems) và phân tích tại chỗ (on-site analysis) đó là federated learning. Hiểu về thuật toán được sử dụng trong federated learning cũng như cách hoạt động của nó. Các mô hình tương tự federated learning sẽ là tương lai và hướng phát triển của lĩnh vực trí tuệ nhân tạo. Bên cạnh đó, sau khi thực hiện xây dựng mô hình phân loại chữ số viết tay kết hợp federated learning giúp chúng tôi biết nhiều hơn về thư viện torch, biết rõ hơn về cách thức hoạt động của mô hình CNN trong phân loại hình ảnh.

5.2. HẠN CHẾ

Federated learning là một lĩnh vực chưa được phổ cập, chưa có nhiều tài liệu về việc triển khai thực tế nên còn gặp khó khăn trong quá trình triển khai mô hình. Do sự hạn chế về nguồn lực và thời gian, chúng tôi tập trung vào nghiên cứu lý thuyết của federated learning hơn.Về phần triển khai mô hình thực tế trên tập dữ liệu MNIST, chúng tôi cũng gặp nhiều vấn đề khi triển khai mô hình vì chưa có nhiều tài liệu cho việc triển khai cụ thể cũng như xung đột giữa các phiên bản khi cài đặt

các module trong python. Do giới hạn về cơ sở vật chất, không có một cấu hình máy đủ tốt nên nhóm chúng tôi chỉ thử nghiệm trên một vài siêu tham số. Chưa có kinh nghiệm về giao diện app bằng python nên giao diện chưa được tối ưu. Giới hạn kiến thức toán học và khả năng đọc hiểu tiếng anh cũng là một yếu tố cản trở việc nghiên cứu của chúng tôi.

5.3. HƯỚNG PHÁT TRIỂN

Lý thuyết được trình bày trong báo cáo có thể được áp dụng cho một số lĩnh vực nhất định như ngân hàng, y tế, giáo dục, v.v. Vì nó sẽ hạn chế được việc rò rỉ dữ liệu và có thể bảo mật được dữ liệu của người dùng một cách tốt nhất, mà những lĩnh vực này là những lĩnh vực mà việc bảo mật thông tin người dùng là hết sức quan trọng.

Báo cáo cũng ra nhiều hướng nghiên cứu trong tương lai như:

- Nghiên cứu sâu hơn về hướng quản lý và thắt chặt quyền bảo vệ quyền riêng tư dữ liệu của người dùng.
- Nghiên cứu sâu hơn về các thuật toán khác được sử dụng trong federated learning.
- Nghiên cứu về các thuật toán deep learning khác sử dụng cho việc xử lý và phân tích hình ảnh.

Về phần thực nghiệm, ta có thể cải thiện giao diện của ứng dụng nhận dạng. Có thể xây dựng thêm một trang web để có thể dễ dàng tương tác hơn, thử nghiệm nhiều thuật toán khác hơn, dùng thêm các bộ siêu tham số hơn để cải thiện độ chính xác của mô hình. Ngoài ra có thể nghiên cứu áp dụng federated learning cho một ứng dụng hữu ích, gần gũi hơn trong các lĩnh vực như y tế, đời sống, v.v.

TÀI LIỆU THAM KHẢO

[1] S. Abdulrahman, "A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond," 10 / 2020. [Online]. Available: https://www.researchgate.net/publication/344871928_A_Survey_on_Federated_Learning_The_Journey_From_Centralized_to_Distributed_On-Site_Learning_and_Beyond#pf3. [Accessed 11 / 12 / 2022].

[2] Qiang Yang et al., "Federated learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*, MORGAN & CLAYPOOL, 2019, pp. 1-207.

[3] H. Association, "NVIDIA Clara Federated Learning – Điện toán AI cho tổ chức y tế," 02 / 12 / 2019. [Online]. Available: https://migovi.com/2019/12/02/nvidia-clara-federated-learning-ai-agx-suc-khoe/. [Accessed 15 / 11 / 2022].

[4] Wikipedia, "Federated Learning," 2021. [Online].

[5] D. Xuân, "Tự học ML | Stochastic Gradient Descent (SGD)," 18 / 12 / 2021. [Online]. Available: https://cafedev.vn/tu-hoc-ml-stochastic-gradient-descent-sgd/. [Accessed 21 / 11 / 2022].

[6] H. Brendan McMahan et al., "Federated Averaging," in *Communication-Efficient Learning of Deep Networks*, USA, 2016, pp. 0-10.

[7] Alireza Fallah et al., "Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach," 2020. [Online]. Available: https://paperswithcode.com/paper/personalized-federated-learning-with. [Accessed 11 / 12 / 2022].

[8] "Mọi thứ bạn cần biết về học liên kết," Tin Mỗi Giờ, 14 / 08 / 2021. [Online]. Available: https://www.tinmoiz.com/moi-thu-ban-can-biet-ve-hoc-lien-ket-584372/. [Accessed 14 / 11 / 2022].

[9] B. Dickson, "Machine learning: What are membership inference attacks?," 23 / 04 / 2021. [Online]. Available: https://bdtechtalks.com/2021/04/23/machine-learning-membership-inference-attacks/. [Accessed 11 / 12 / 2022].

[1 DEEPLIZARD, "Convolutional Neural Networks (CNNs) Explained," 10 / 11 / 0] 2017. [Online]. Available: https://deeplizard.com/learn/video/YRhxdVk_sIs. [Accessed 18 / 12 / 2022].