hash=d787f09b6c6cc374a9ad21fe9120fc1ffamily=Khudanpur, familyi=K., given=Sanjeev, giveni=S.

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN - ĐIỆN TỬ

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

Đề tài:

TỐI ƯU THỜI GIAN VÀ NĂNG LƯỢNG TRONG VIỆC TRUYỀN DỮ LIỆU CHO HỆ THỐNG HỌC MÁY LIÊN KẾT ĐA TÁC NHIỆM SỬ DỤNG THUẬT TOÁN SOFT ACTOR CRITIC

Sinh viên thực hiện	:	ĐINH THỊ QUỲNH
MSSV	:	20193073
Lớp	:	CTTN ĐTVT K64
Giảng viên hướng dẫn	:	PGS. Nguyễn Tiến Hòa

Hà Nội, 8 - 2023

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN - ĐIỆN TỬ

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

TỐI ƯU THỜI GIAN VÀ NĂNG LƯỢNG TRONG VIỆC TRUYỀN DỮ LIỆU CHO HỆ THỐNG HỌC MÁY LIÊN KẾT ĐA NHIỆM ỨNG DỤNG THUẬT TOÁN SAC

Sinh viên thực hiện	:	Đinh Thị Quỳnh
MSSV	:	20193073
Lớp	:	CTTN - ĐTVT K64
Giảng viên hướng dẫn	:	PGS. Nguyễn Tiến Hòa
Cán bộ phản biện	:	

Hà Nội, 8 - 2023

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ

ĐÁNH GIÁ ĐỒ ÁN TỐT NGHIỆP
(DÀNH CHO CÁN BỘ HƯỚNG DẪN)

Tên đề tài: Tối ưu thời gian và năng lượng trong việc truyền dữ liệu cho hệ thống học máy liên kết đa nhiệm sử dụng thuật toán Soft Actor Critic

Họ tên SV: Đinh Thị Quỳnh

Cán bộ hướng dẫn: PGS. Nguyễn Tiến Hoà

STT	Tiêu chí (Điểm tối đa)	Hướng dẫn đánh giá tiêu chí	Điểm tiêu chí
1	**Thái độ làm việc (2,5 điểm)**	Nghiêm túc, tích cực và chủ động trong quá trình làm ĐATN	
		Hoàn thành đầy đủ và đúng tiến độ các nội dung được GVHD giao	2.5
2	**Kỹ năng viết quyển ĐATN (2 điểm)**	Trình bày đúng mẫu quy định, bố cục các chương logic và hợp lý: Bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy, có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn, v.v.	
		Kỹ năng diễn đạt, phân tích, giải thích, lập luận: Cấu trúc câu rõ ràng, văn phong khoa học, lập luận logic và có cơ sở, thuật ngữ chuyên ngành phù hợp, v.v.	2
3	**Nội dung và kết quả đạt được (5 điểm)**	Nêu rõ tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài, các vấn đề và các giả thuyết, phạm vi ứng dụng của đề tài. Thực hiện đầy đủ quy trình nghiên cứu: Đặt vấn đề, mục tiêu đề ra, phương pháp nghiên cứu/ giải quyết vấn đề, kết quả đạt được, đánh giá và kết luận.	
		Nội dung và kết quả được trình bày một cách logic và hợp lý, được phân tích và đánh giá thỏa đáng. Biện luận phân tích kết quả mô phỏng/ phần mềm/ thực nghiệm, so sánh kết quả đạt được với kết quả trước đó có liên quan.	
		Chỉ rõ phù hợp giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. Hàm lượng khoa học/ độ phức tạp cao, có tính mới/tính sáng tạo trong nội dung và kết quả đồ án.	5
4	**Điểm thành tích (1 điểm)**	Có bài báo KH được đăng hoặc chấp nhận đăng/ đạt giải SV NCKH giải 3 cấp Trường trở lên/ Các giải thưởng khoa học trong nước, quốc tế từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế. **(1 điểm)**	
		Được báo cáo tại hội đồng cấp Trường trong hội nghị SV NCKH nhưng không đạt giải từ giải 3 trở lên/ Đạt giải khuyến khích trong cuộc thi khoa học trong nước, quốc tế/ Kết quả đồ án là sản phẩm ứng dụng có tính hoàn thiện cao, yêu cầu khối lượng thực hiện lớn. **(0,5 điểm)**	1
		Điểm tổng các tiêu chí:	10.5
		Điểm hướng dẫn:	10

Cán bộ hướng dẫn

Nguyễn Tiến Hòa

Điểm từng tiêu chí cho lẻ đến 0,5. Nếu Điểm tổng các tiêu chí > 10 thì Điểm hướng dẫn làm tròn thành 10

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ

ĐÁNH GIÁ ĐỒ ÁN TỐT NGHIỆP
(DÀNH CHO CÁN BỘ PHẢN BIỆN)

Tên đề tài: Tối ưu thời gian và năng lượng trong việc truyền dữ liệu cho hệ thống học máy liên kết đa nhiệm sử dụng thuật toán Soft Actor Critic

Họ tên SV: Đinh Thị Quỳnh MSSV: 20193073

Cán bộ phản biện: .....

STT	Tiêu chí (Điểm tối đa)	Hướng dẫn đánh giá tiêu chí	Điểm tiêu chí
1	**Trình bày quyển ĐATN (4 điểm)**	Đồ án trình bày đúng mẫu quy định, bố cục các chương logic và hợp lý: Bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy, có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn, v.v.	
		Kỹ năng diễn đạt, phân tích, giải thích, lập luận: Cấu trúc câu rõ ràng, văn phong khoa học, lập luận logic và có cơ sở, thuật ngữ chuyên ngành phù hợp, v.v.	
2	**Nội dung và kết quả đạt được (5.5 điểm)**	Nêu rõ tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài, các vấn đề và các giả thuyết, phạm vi ứng dụng của đề tài. Thực hiện đầy đủ quy trình nghiên cứu: Đặt vấn đề, mục tiêu đề ra, phương pháp nghiên cứu/ giải quyết vấn đề, kết quả đạt được, đánh giá và kết luận.	
		Nội dung và kết quả được trình bày một cách logic và hợp lý, được phân tích và đánh giá thỏa đáng. Biện luận phân tích kết quả mô phỏng/ phần mềm/ thực nghiệm, so sánh kết quả đạt được với kết quả trước đó có liên quan.	
		Chỉ rõ phù hợp giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. Hàm lượng khoa học/ độ phức tạp cao, có tính mới/ tính sáng tạo trong nội dung và kết quả đồ án.	
3	**Điểm thành tích (1 điểm)**	Có bài báo KH được đăng hoặc chấp nhận đăng/ đạt giải SV NCKH giải 3 cấp Trường trở lên/ Các giải thưởng khoa học trong nước, quốc tế từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế. **(1 điểm)**	
		Được báo cáo tại hội đồng cấp Trường trong hội nghị SV NCKH nhưng không đạt giải từ giải 3 trở lên/ Đạt giải khuyến khích trong cuộc thi khoa học trong nước, quốc tế/ Kết quả đồ án là sản phẩm ứng dụng có tính hoàn thiện cao, yêu cầu khối lượng thực hiện lớn. **(0,5 điểm)**	
		Điểm tổng các tiêu chí:	
		Điểm phản biện:	

Cán bộ phản biện
(Ký và ghi rõ họ tên)

Điểm từng tiêu chí cho lẻ đến 0,5. Nếu Điểm tổng các tiêu chí > 10 thì Điểm phản biện làm tròn thành 10.

LỜI NÓI ĐẦU

Trong những năm cuối của tuổi học trò, ở lứa tuổi 18 đa số ai ai cũng đều cảm thấy lo lắng, chơi vơi vì chưa rõ định hướng của bản thân, một câu hỏi băn khoăn hiển hiện trong đầu mỗi cô cậu học trò lúc ấy rằng "Mình nên chọn trường Đại học nào?", "Liệu rằng ngành học này có phải lựa chọn đúng cho bản thân mình không?" và vô vàn các câu hỏi về tương lai khác. Em cũng vậy, em cũng mông lung và do dự rất nhiều. Tuy nhiên, vào những giây phút cuối cùng em đã quyết định lựa chọn theo học tại ngôi trường danh giá Đại học Bách khoa Hà Nội bởi tình yêu khoa học kỹ thuật và truyền thống gia đình. Và sau 5 năm được học tập tại đây, em chưa một lần phải hối hận vì quyết định ấy. Cảm ơn Bách khoa vì những tháng ngày tuổi trẻ sinh viên đã cháy hết mình với đam mê, với nhiệt huyết để rồi em có thể mỉm cười bởi tất cả.

Sau một quá trình học tập và nghiên cứu, em nhận thấy bản thân mình ưa thích chủ đề nghiên cứu về ứng dụng của mạng Cảm biến không dây, và em nhận ra một số hạn chế còn tồn tại của mạng khi triển khai vào thực tế. Vì vậy, em đã tìm hiểu và quyết định lựa chọn đề tài **"Tối ưu thời gian và năng lượng trong hệ thống học máy liên kết đa nhiệm ứng dụng thuật toán học tăng cường SAC"** làm đề tài nghiên cứu đồ án của bản thân.

Em xin gửi lời cảm ơn sâu sắc và chân thành nhất tới **Phó Giáo Sư Nguyễn Tiến Hòa** đã dạy bảo và bồi dưỡng em những kiến thức cơ bản nhất của ngành cũng như thầy đã truyền cho em ngọn lửa đam mê nghiên cứu suốt khoảng thời gian 3 năm em làm thành viên phòng thí nghiệm Xử lý băng tần gốc do thầy quản lí. Những bài giảng tận tình cũng như những lời dạy dỗ, định hướng cụ thể đã trực tiếp giúp em hoàn thành đề tài nghiên cứu một cách tốt nhất có thể.

Đồng thời, em xin gửi một lời cảm ơn, tri ân tới các người anh, người chị đã giúp đỡ tận tình em từ những ngày đầu tiên em xây dựng ý tưởng, khởi động nghiên cứu và phát triển đồ án. Em xin chân thành cảm ơn anh *Nguyễn Minh Dương* cựu sinh viên K54 viện Điện tử - Viễn thông, anh *Nguyễn Doãn Hiếu* cựu sinh viên K62 viện Điện tử - Viễn thông, chị *Lưu Ngọc Minh* cựu sinh viên K61 viện Điện tử - Viễn thông, và chị *Nguyễn Thị Hoài Linh* cựu sinh viên K61 viện Điện tử - Viễn thông.

Ngoài ra, mình cũng vô cùng cảm ơn những người bạn lớp CTTN ĐTVT K64, những thành viên của C9-417 đã đồng hành và hỗ trợ mình trong suốt khoảng thời gian học tập tại đây.

Mặc dù đã cố gắng và nỗ lực hết sức để hoàn thành đề tài mục tiêu với kết quả tốt nhất có thể nhưng em tin rằng mình không thể tránh khỏi những sai sót. Em kính mong nhận được sự chỉ bảo tận tình và sự cảm thông của các quý thầy cô và các bạn.

Em xin chân thành cảm ơn!

Hà Nội, ngày 04, tháng 08, năm 2023

Đinh Thị Quỳnh

LỜI CAM ĐOAN

Tôi là Đinh Thị Quỳnh, mã số sinh viên 20193073, sinh viên Chương trình tài năng Điện tử Viễn thông , khóa K64. Người hướng dẫn là PGS. Nguyễn Tiến Hòa. Tôi xin cam đoan toàn bộ nội dung được trình bày trong đồ án **"Tối ưu tối ưu kênh truyền mạng cảm biến không dây tầng ứng dụng thuật toán học tăng cường"** là kết quả quá trình tìm hiểu và nghiên cứu của tôi dưới sự hướng dẫn của cán bộ hướng dẫn. Các dữ liệu được nêu trong đồ án là hoàn toàn trung thực, phản ánh đúng kết quả đo đạc thực tế. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ; các tài liệu tham khảo được liệt kê rõ ràng. Tôi xin chịu hoàn toàn trách nhiệm với những nội dung được viết trong đồ án này.

Người cam đoan

Đinh Thị Quỳnh

Mục lục

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

ACN	Actor - Critic Network	Mạng tác nhân - đánh giá
AN	Actor Network	Mạng Tác nhân
AP	Access Point	Điểm truy cập
BS	Base station	Trạm phát gốc
CN	Critic Network	Mạng Đánh giá
CWSN	Cluster Wireless Sensor Network	Mạng cảm biến không dây phân cụm
DDPG	Deep Deterministic Policy Gradient	Thuật toán xác định chính sách dựa vào độ dốc
DQN	Deep Q-Network	Mạng học chất lượng sâu
DRL	Deep Reinforcement Learning	Học sâu tăng cường
FIFO	First-In-First-Out	Cấu trúc lưu trữ vào trước ra trước
FSPL	Free-Space Path Loss	Suy hao trong không gian tự do
FWSN	Flat Wireless Sensor Network	Mạng cảm biến không dây đồng cấp
HetNets	Heterogeneous network	Mạng không đồng nhất
HWSN	Hierarchical Wireless Sensor Network	Mạng cảm biến không dây nhiều tầng
IP	Internet Protocol	Giao thức mạng
LEACH	Low-energy adaptive clustering hierarchy	Hệ thống phân cấp phân nhóm thích ứng năng lượng thấp
M2M	Machine-to-Machine	Máy tới máy
MAC	Medium access control	Lớp kiểm soát truy nhập
NN	Neural Network	Mạng tế bào thần kinh
RL	Reinforcement Learning	Học tăng cường
SPIN	Sensor Protocols for Information via Negotiation	Giao thức liên kết cảm biến thỏa hiệp thông tin
TAN	Target Actor Network	Mạng Tác nhân mục tiêu
TCN	Target Critic Network	Mạng Đánh giá mục tiêu
TD	Temporary Difference	Sai khác tạm thời
TDMA	Time-division multiple access	Đa truy nhập phân chia theo thời gian
UAV	Unmanned aerial vehicle	Máy bay không người lái
WLAN	Wireless Local Area Network	Mạng liên kết không dây cục bộ
WPAN	Wireless Personal Area Network	Mạng không dây khu vực cá nhân
WSN	Wireless Sensor Network	Mạng cảm biến không dây

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

TÓM TẮT ĐỒ ÁN

Abstract: Học phân tán đa nhiệm là một mô hình phức tạp nhằm huấn luyện các mô hình học máy trên các thiết bị phân tán trong khi xử lý nhiều tác vụ đồng thời. Việc phân bổ tài nguyên hiệu quả và tối ưu hiệu suất mô hình trên các tác vụ khác nhau đang đối diện nhiều khó khăn trong ngữ cảnh này. Đồ án này đề xuất một phương pháp mới sử dụng thuật toán Soft Actor-Critic (SAC) để giải quyết vấn đề phân bổ tài nguyên trong bài toán học phân tán đa nhiệm.

Học phân tán (FL) cho phép huấn luyện mô hình trên các thiết bị phân tán mà vẫn bảo vệ quyền riêng tư dữ liệu bằng cách giữ dữ liệu cục bộ trên từng thiết bị. Tuy nhiên, trong các cài đặt đa nhiệm, các tác vụ thường có độ phức tạp và phân phối dữ liệu khác nhau, dẫn đến việc phân bổ tài nguyên không cân đối và hiệu suất kém cho một số tác vụ.

Để giải quyết vấn đề này, đồ án giới thiệu thuật toán Soft Actor-Critic, một kỹ thuật học tăng cường mạnh mẽ có khả năng xử lý không gian hành động liên tục. Bằng cách sử dụng SAC, chúng ta cho phép phân bổ tài nguyên động đến các tác vụ khác nhau trong quá trình học phân tán. Đại lý học cách thích ứng với chiến lược phân bổ tài nguyên dựa trên phản hồi từng tác vụ, tối ưu hóa việc phân phối tài nguyên và cải thiện hiệu quả học tập tổng thể.

Tôi tiến hành thực nghiệm một cách rộng rãi trên các tác vụ học phân tán đa nhiệm khác nhau để đánh giá hiệu quả của phương pháp đề xuất. Kết quả cho thấy tích hợp thuật toán SAC cho việc phân bổ tài nguyên cải thiện đáng kể kết quả học tập và đạt được hiệu suất tổng thể tốt hơn so với các chiến lược phân bổ tài nguyên tĩnh truyền thống.

Tóm lại, đồ án này đóng góp cho lĩnh vực học phân tán đa nhiệm bằng việc giới thiệu một phương pháp mới sử dụng thuật toán Soft Actor-Critic để phân bổ tài nguyên linh hoạt. Bằng cách phân bổ tài nguyên động, phương pháp đề xuất đảm bảo tác vụ nhận được sự chú ý tối ưu trong quá trình học phân tán, giúp cải thiện hiệu suất mô hình và sử dụng tài nguyên hiệu quả hơn trong môi trường phân tán. Kết quả nghiên cứu này nhấn mạnh tiềm năng của phân bổ tài nguyên dựa trên học tăng cường trong việc tăng tính mở rộng và hiệu quả của học phân tán đa nhiệm.

ABSTRACT

Multi-task federated learning is a challenging paradigm that aims to collaboratively train machine learning models on decentralized devices while handling multiple tasks simultaneously. Efficiently allocating resources and optimizing model performance across diverse tasks present significant challenges in this context. This thesis proposes a novel approach that utilizes the Soft Actor-Critic (SAC) algorithm to address the resource allocation problem in multi-task federated learning scenarios.

Federated Learning (FL) enables distributed model training while preserving data privacy by keeping data locally on individual devices. However, in multi-task settings, tasks often exhibit varying complexities and data distributions, leading to imbalanced resource allocation and suboptimal performance for certain tasks.

To address this issue, I introduce the Soft Actor-Critic algorithm, a powerful reinforcement learning technique capable of handling continuous action spaces. By employing SAC, we enable dynamic resource allocation to different tasks during the federated learning process. The agent learns to adapt its resource allocation strategies based on task-specific feedback, optimizing resource distribution and enhancing overall learning efficiency.

Extensive experiments are conducted on various multi-task federated learning setups to evaluate the effectiveness of the proposed approach. The results demonstrate that integrating the SAC algorithm for resource allocation significantly improves learning outcomes and achieves better overall performance compared to traditional static resource allocation strategies.

In conclusion, this thesis contributes to the field of multi-task federated learning by presenting a novel approach that leverages the Soft Actor-Critic algorithm for adaptive resource allocation. By dynamically allocating resources, our approach ensures optimal attention is given to different tasks during the federated learning process, leading to improved model performance and more efficient resource utilization in decentralized settings. The findings highlight the potential of reinforcement learning-based resource allocation in enhancing the scalability and effectiveness of multi-task federated learning.

CHƯƠNG 1: CHƯƠNG MỞ ĐẦU

1.1. Tổng quan đề tài:

Cách mạng công nghệ trong thế kỉ XXI đã ghi dấu một bước tiến vượt bậc trong sự phát triển và tiến bộ của nhân loại toàn cầu. Với sự ra đời và vươn mình mạnh mẽ của công nghệ mạng toàn cầu, đã mở ra một giai đoạn mới, được biết đến với tên gọi "cách mạng công nghệ 4.0". Trong cuộc cách mạng này, nhận thức và đời sống của mỗi cá nhân trong cộng đồng đã thay đổi một cách toàn diện nhờ vào những ứng dụng thực tiễn và quan trọng của công nghệ mạng. Truyền tin khoảng cách lớn, tra cứu thông tin, quản lý thông tin và giao tiếp không dây là những thành tựu mang tính đột phá trong cách mà chúng ta tiếp cận thông tin và kết nối với nhau.

Đặc biệt, các mô hình kết nối mạng của nhiều thiết bị đã dần trở nên phổ biến ở khắp các hệ thống cơ sở hạ tầng dân cư từ quy mô nhỏ đến lớn. Có nhiều mô hình kết nối mạng đa dạng như: Mạng kết nối không đồng nhất, Mạng kết nối khu vực cục bộ, Mạng diện rộng, Mạng kết nối toàn cầu, Mạng cảm biến không dây và nhiều mô hình mạng tiêu biểu khác đã được phát triển và triển khai trên khắp thế giới.

Cùng với sự phát triển của cuộc cách mạng công nghệ 4.0, sự quan tâm ngày càng tăng về bảo vệ quyền riêng tư dữ liệu người tiêu dùng [1] đã dẫn đến xuất hiện một nhóm kỹ thuật học máy mới khai thác sự tham gia của nhiều người dùng điện thoại di động. Một trong số những kỹ thuật phổ biến trong nhóm này là Federated Learning [2, 3]. Kỹ thuật học này cho phép người dùng cùng nhau xây dựng một mô hình học chung trong khi vẫn giữ toàn bộ dữ liệu huấn luyện trên thiết bị người dùng của họ (UE).

Cụ thể, mỗi thiết bị người dùng tính toán các cập nhật cho mô hình toàn cầu hiện tại dựa trên dữ liệu huấn luyện cục bộ của nó. Sau đó, các cập nhật này được tổng hợp và phản hồi bởi một máy chủ trung tâm, để mà tất cả các thiết bị người dùng đều có quyền truy cập vào cùng một mô hình toàn cầu để tính toán các cập nhật mới. Quá trình này được lặp lại cho đến khi đạt được mức độ chính xác của mô hình học. Như vậy, quyền riêng tư dữ liệu của người dùng được bảo vệ tốt vì dữ liệu huấn luyện cục bộ không được chia sẻ, do đó, phương pháp này loại bỏ việc học máy dựa trên việc thu thập, lưu trữ và huấn luyện dữ liệu tại các trung tâm dữ liệu như các phương pháp truyền thống.

1.2. Tính cấp thiết của đồ án

Với sự phát triển nhanh chóng của Internet of Things (IoT) và các thiết bị di động thông minh, việc thu thập và xử lý dữ liệu phân tán ngày càng trở nên quan trọng. Tuy nhiên, việc truyền toàn bộ dữ liệu từ các thiết bị cục bộ về máy chủ trung tâm có thể gây ra một số vấn đề như tốn kém về năng lượng, băng thông hạn chế, và mất quyền riêng tư. Do đó, nghiên cứu về tối ưu hóa thời gian và năng lượng trong việc truyền dữ liệu sử dụng Multi Task Federated

Learning là cực kỳ cấp thiết để cải thiện hiệu quả và tiện ích của học máy phân tán.

1.3. Động lực nghiên cứu:

Học Máy Liên Kết (Federated Learning - FL) đã thu hút sự chú ý đáng kể trong cộng đồng học thuật nhờ tính chất thu hút và ứng dụng đa dạng [4]. Phương pháp truyền thống của học máy, dựa vào máy chủ trung tâm để tính toán và thu thập dữ liệu người dùng, đối mặt với những thách thức như tăng tải thông tin giao tiếp và nguy cơ rò rỉ dữ liệu. FL giải quyết các vấn đề về quyền riêng tư này bằng cách đào tạo các mô hình học máy cục bộ, trong đó người dùng chỉ truyền thông số mô hình của họ đến Một Trạm Cơ Sở (Base Station - BS) [5]. Phương pháp này bảo vệ cả quyền riêng tư và khía cạnh hợp tác của khung công việc học máy, làm cho nó phù hợp cho các mạng Internet of Things (IoT). Do đó, FL đã được áp dụng trong các ứng dụng trí tuệ nhân tạo khác nhau trong mạng IoT, bao gồm đám mây xe hơi [6, 7, 8], ứng dụng từ xa [9, 10, 11], chăm sóc sức khỏe thông minh [12, 13, 14], và mạng cạnh di động [15, 16, 17, 18].

Mặc dù có những tiến bộ gần đây trong phần cứng tính toán và các mô hình như tính toán cạnh di động (MEC) và tính toán sương mù (fog computing), tuy nhiên, nguồn tài nguyên truyền thông giới hạn của các hệ thống IoT vẫn là một thách thức [19, 20, 21]. Sự gia tăng mũi nhọn của các thiết bị IoT và tài nguyên không dây hạn chế làm cản trở việc triển khai các kịch bản IoT quy mô lớn tích hợp hệ thống FL. Do đó, đã đề xuất một số giải pháp nhằm cải thiện hiệu suất truyền thông trong quá trình FL [22]. Các bài toán tối ưu đã được định dạng để giảm thiểu chi phí thời gian trong mỗi vòng truyền thông [23] và tối đa hóa tốc độ tổng hợp dữ liệu từ người dùng FL [24]. Các kỹ thuật như cắt tỉa mô hình và thưa thớt dữ liệu đã được áp dụng để giảm tải thông tin giao tiếp trong giai đoạn truyền thông.

Ngoài ra, nỗ lực đã được thực hiện để giải quyết thách thức giảm tính toán và năng lượng truyền thông trong FL [25]. Tuy nhiên, một số nghiên cứu thiếu phân tích hội tụ toàn diện cho FL [25]. Để giảm chi phí truyền thông lên, tác giả trong [26] đề xuất hai phương pháp, đánh giá chúng trong ngữ cảnh huấn luyện mạng thần kinh sâu để phân loại hình ảnh. Phương pháp tốt nhất giảm thiểu thông tin giao tiếp cần thiết để huấn luyện một mô hình hợp lý lên đến hai bậc. Tác giả trong [27] đề xuất một thuật toán mới (SCAFFOLD) yêu cầu ít vòng truyền thông và đạt được sự hội tụ nhanh hơn.

Tuy nhiên, hầu hết sự chú ý trong những nghiên cứu này đã tập trung vào việc phát triển thuật toán để tăng tốc sự hội tụ của quá trình học, trong khi ít chú trọng đến các yếu tố giới hạn hiệu suất khác có thể cản trở học máy liên kết, như truyền thông không dây và nguồn tài nguyên năng lượng hạn chế của thiết bị người dùng di động (UE).

1.4. Những đóng góp chính của đồ án

Nội dung của đồ án là thiết kế và triển khai một phương pháp tối ưu dựa theo mô hình thuật toán SAC với mục tiêu là tối ưu thời gian và năng lượng truyền đi của hệ thống. Những đóng góp quan trọng của đồ án được liệt kê qua những luận điểm sau:

1. Dựa vào thông tin khảo sát và thực tế của mô hình mạng thiết lập các phương trình ràng buộc về tài nguyên của hệ thống và yêu cầu về tốc độ dữ liệu truyền tải. Xây dựng bài

toán tối ưu phân bổ tài nguên, tối đa tô

2. Đề xuất xây dựng và áp dụng một mô hình thuật toán dựa vào thuật toán đã có DDPG để giải quyết vấn đề tối ưu. Thuật toán sẽ lần lượt được xây dựng và thay đổi tham số cũng như phương pháp xử lí để phù hợp với riêng vấn đề tối ưu. Thuật toán đề xuất được hi vọng sẽ xác định được giải pháp với độ phức tạp và thời gian tính toán phù hợp.

3. Mô phỏng thuật toán và trình bày trực quan kết quả tính toán của phương pháp đề xuất, đồng thời so sánh với kết quả của một số phương pháp xử lý khác. Nhận xét và khẳng định mức độ hiệu quả, tiến bộ của phương pháp được đề xuất về hiệu suất hoạt động trong mạng cảm biến không dây.

1.5. Bố cục của đồ án

Đồ án này được chia thành 6 chương truyền tải các nội dung chính như sau:

- Chương 1: Chương mở đầu
  Chương sẽ lần lượt nêu ra tổng quan đề tài, động lực thực hiện nghiên cứu đồ án này cùng với việc hoạch định ra được phạm vi và đối tượng nghiên cứu của đồ án.

- Chương 2: Tổng quan về Học máy liên kết đa tác nhiệm (Multi Task Federated Learning)
  Phần này nêu ra cơ sở lý thuyết cơ bản sẽ được sử dụng trong đồ án này.

- Chương 3: Tổng quan về thuật toán học tăng cường
  Phần này trình bày cơ sở lý thuật của thuật toán học tăng cường và thuật toán SAC được sử dụng trong đồ án.

- Chương 4: Mô hình hệ thống và Xây dựng bài toán tối ưu
  Với mô hình hệ thống được xây dựng và tính toán ở Chương 3, phần này sẽ trình bày phương án mô phỏng cũng như kết quả mô phỏng mà đồ án đạt được. Đây cũng là chương cuối cùng của đồ án.

- Chương 5: Mô hình thuật toán và kết quả
  Thuật toán sẽ nhận các tham số mô tả đặc điểm của môi trường truyền tin và những ràng buộc của bài toán tối ưu. Sau khi tính toán, thuật toán trả về giá trị và lựa chọn tối ưu cho hệ thống.

- Chương 6: Kết luận và định hướng phát triển
  Chương này sẽ tổng kết đồ án và thảo luận về hướng phát triển nội dung đồ án trong tương lai. Đây cũng là chương cuối cùng đồ án.

1.6. Kết luận chương

Mô hình truyền tin của các nút trong mạng cảm biến không dây nhiều tầng được mô phỏng trong đồ án. Một thuật toán dựa vào thuật toán SAC được đưa ra để giải quyết vấn đề tối ưu tổng dung lượng truyền tin của cả mạng với một số ràng buộc của vấn đề tối ưu. Kết quả cuối cùng

được mô tả và so sánh với một số phương pháp tối ưu khác. Chương mở đầu đã nêu ra được động lực nghiên cứu của đồ án này, từ đó đã giới hạn được phạm vi cũng như đối tượng nghiên cứu chính của đồ án, phân chia bố cục cấu trúc của đồ án. Nội dung chi tiết từng phần sẽ được trình bày ở các Chương tiếp theo.

CHƯƠNG 2: GIỚI THIỆU VỀ MULTI TASK FEDERATED LEARNING

2.1. Giới thiệu chương

Chương này sẽ đi sâu vào việc nêu tổng quan các cơ sơ lý thuyết tổng quan về phương pháp Federated Learning được sử dụng trong đồ án.

Bố cục của chương lần lượt phần 2.2 giới thiệu những khái niệm chung về phương pháp Federated Learning

2.2. Federated Learning

Điện thoại di động, thiết bị đeo được và xe tự hành chỉ là một vài trong số các mạng phân tán hiện đại tạo ra vô số dữ liệu mỗi ngày. Do sức mạnh tính toán ngày càng tăng của các thiết bị này, cùng với những lo ngại về việc truyền thông tin cá nhân việc lưu trữ dữ liệu cục bộ và đẩy tính toán mạng lên biên ngày càng hấp dẫn.

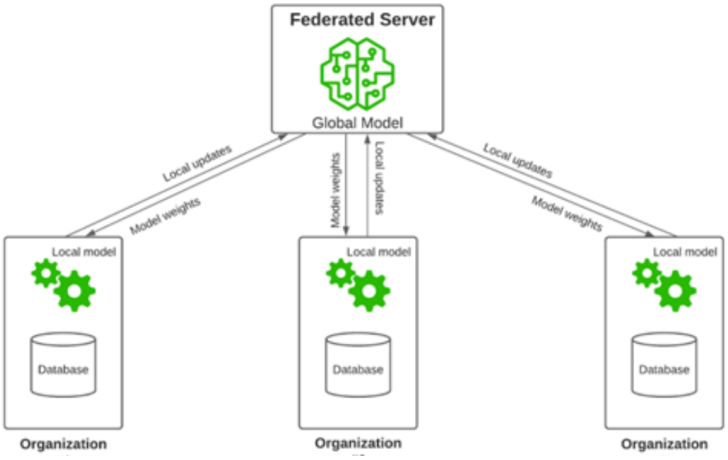
Khái niệm điện toán cạnh (Edge Computing) không phải là một khái niệm mới. Thật vậy, tính toán các truy vấn đơn giản trên các thiết bị phân tán, năng lượng thấp là một lĩnh vực nghiên cứu kéo dài hàng thập kỷ đã được khám phá dưới góc độ xử lý truy vấn trong mạng cảm biến, điện toán ở biên và điện toán sương mù [28, 29, 30]. Các công trình gần đây cũng đã xem xét việc đào tạo các mô hình máy học tập trung nhưng phục vụ và lưu trữ chúng cục bộ; ví dụ, đây là một cách tiếp cận phổ biến trong mô hình hóa và cá nhân hóa người dùng di động [31, 32].

Tuy nhiên, khi khả năng lưu trữ và tính toán của các thiết bị trong các mạng phân tán tăng lên, có thể tận dụng các tài nguyên cục bộ nâng cao trên mỗi thiết bị. Điều này đã dẫn đến sự quan tâm ngày càng tăng đối với học liên kết (Federated Learning) [4], khám phá các mô hình thống kê đào tạo trực tiếp trên các thiết bị từ xa. Việc học trong môi trường như vậy khác biệt đáng kể so với môi trường phân tán truyền thống đòi hỏi những tiến bộ cơ bản trong các lĩnh vực như quyền riêng tư, học máy quy mô lớn và tối ưu hóa phân tán, đồng thời đặt ra những câu hỏi mới về giao điểm của các lĩnh vực khác nhau, chẳng hạn như học máy và hệ thống [33].

Các phương pháp học liên kết đã được các nhà cung cấp dịch vụ lớn triển khai [34] và đóng vai trò quan trọng trong việc hỗ trợ các ứng dụng nhạy cảm với quyền riêng tư nơi dữ liệu đào tạo được phân phối ở biên, ví dụ: [35] [36]. Ví dụ về các ứng dụng tiềm năng bao gồm: tình cảm học tập, vị trí ngữ nghĩa hoặc hoạt động của người dùng điện thoại di động; thích ứng với hành vi của người đi bộ trong các phương tiện tự hành; và dự đoán các sự kiện sức khỏe như nguy cơ đau tim từ các thiết bị đeo được [37] [38].

2.2.1. Định nghĩa

Federated Learning (Học liên kết) là một phương pháp học máy phân tán, trong đó mô hình học được đào tạo trên nhiều thiết bị hoặc máy tính cá nhân mà không cần phải chuyển dữ



Hình 2.1: Mô hình Federated Learning

liệu về một trung tâm tập trung. Thay vì gửi dữ liệu từ tất cả các thiết bị về một trung tâm, chỉ có các thông số (như trọng số mô hình) được truyền về trung tâm và kết hợp từ tất cả các thiết bị tham gia.

Quá trình học liên kết bắt đầu bằng việc gửi mô hình ban đầu từ trung tâm đến các thiết bị. Sau đó, mỗi thiết bị sẽ sử dụng dữ liệu cục bộ của chính nó để đào tạo mô hình. Trong quá trình này, thông tin cụ thể về dữ liệu của từng thiết bị không được tiết lộ, giúp bảo vệ quyền riêng tư của người dùng.

Sau khi các thiết bị địa phương đã đào tạo mô hình của chúng, các thông số mô hình được truyền về trung tâm, nơi chúng được kết hợp và cập nhật mô hình toàn cầu. Quá trình này được lặp lại nhiều lần, mỗi lần cải thiện mô hình toàn cầu bằng cách học từ nhiều thiết bị.

Federated Learning mang lại nhiều lợi ích, bao gồm việc giảm thiểu việc truyền dữ liệu trực tiếp, bảo vệ quyền riêng tư và an ninh dữ liệu, tiết kiệm băng thông, và tăng tính khả dụng của mô hình học máy trên nhiều thiết bị. Phương pháp này thường được sử dụng trong các ứng dụng di động, y tế, IoT, và các lĩnh vực khác yêu cầu bảo mật và phân phối dữ liệu.

2.2.2. Nền tảng và nguyên tắc cơ bản của Học Máy Phân Tán (Federated Learning - FL)

FL liên quan đến việc huấn luyện cộng tác các mô hình Deep Neural Networks (DNN) trên các thiết bị cuối. Nó bao gồm hai bước chính trong quá trình huấn luyện FL, đó là:

- Huấn luyện mô hình cục bộ trên các thiết bị cuối.
- Tổng hợp toàn cầu các tham số được cập nhật trên máy chủ FL.

Trong phần này sẽ trình bày một giới thiệu ngắn về việc huấn luyện mô hình DNN, điều này cũng áp dụng cho việc huấn luyện mô hình cục bộ trong FL.

Hơn nữa, các mô hình DNN có thể dễ dàng được tổng hợp và vượt trội so với các kỹ thuật ML truyền thống, đặc biệt khi dữ liệu lớn. Việc triển khai FL trong các mạng lưới mobile edge có thể tận dụng tự nhiên sức mạnh tính toán ngày càng tăng cũng như dữ liệu dồi dào được thu thập bởi các thiết bị cuối phân tán, cả hai đều là những động lực đưa DL trỗi dậy [39]. Do đó, một sự giới thiệu ngắn gọn về việc huấn luyện mô hình DNN chung sẽ hữu ích cho các phần

tiếp theo. Sau đó, tôi tiếp tục cung cấp hướng dẫn về quá trình huấn luyện FL bao gồm cả tổng hợp toàn cầu và huấn luyện cục bộ. Ngoài ra, tôi cũng nhấn mạnh về các thách thức thống kê trong việc huấn luyện mô hình FL và trình bày các giao thức và khung công cộng mã nguồn của FL.

2.2.2.1. Deep Learning

Các thuật toán ML truyền thống dựa vào bộ trích xuất đặc trưng được xây dựng bằng tay để xử lý dữ liệu raw [40]. Do đó, kiến thức chuyên môn thường là một điều kiện tiên quyết để xây dựng một mô hình ML hiệu quả. Ngoài ra, việc lựa chọn đặc trưng phải được tùy chỉnh và bắt đầu lại cho mỗi bài toán mới. Ngược lại, các DNN dựa trên việc học biểu diễn, tức là DNN có thể tự động khám phá và học các đặc trưng này từ dữ liệu raw [40] và do đó thường vượt trội hơn so với các thuật toán ML truyền thống, đặc biệt khi có nhiều dữ liệu.

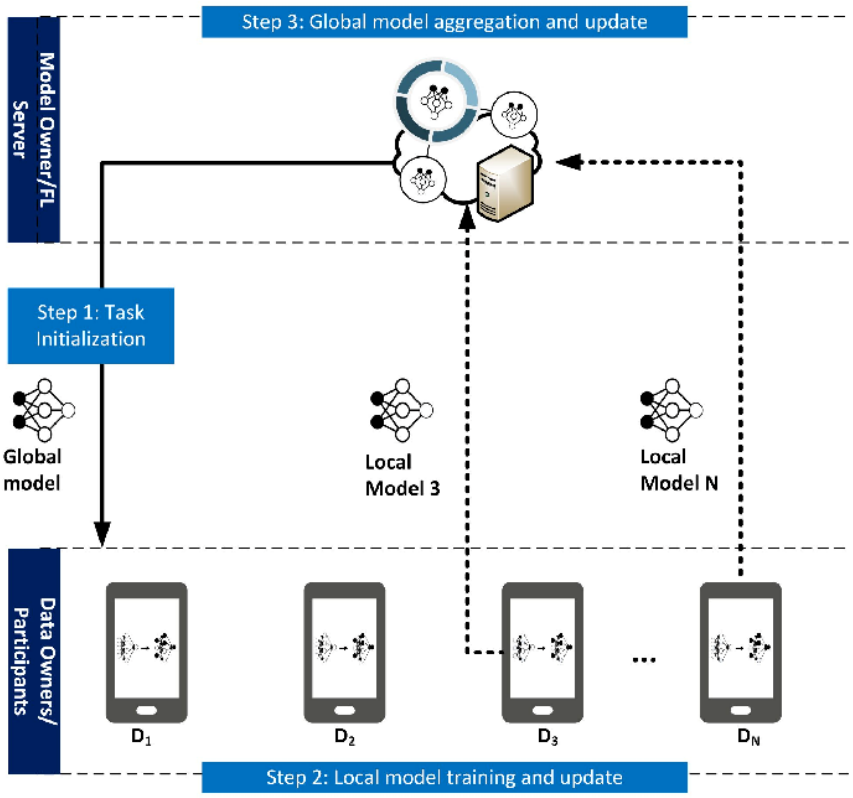
Deep Learning (DL) nằm trong lĩnh vực của mô hình tính toán được lấy cảm hứng từ não bộ, trong đó mạng nơ-ron là một phần quan trọng. Mạng nơ-ron bao gồm ba lớp: lớp đầu vào, lớp ẩn, lớp đầu ra. Trong một DNN truyền thống, giá trị đầu vào có trọng số và đã được điều chỉnh độ lệch (bias) trước khi được đi qua một hàm kích hoạt phi tuyến tính để tính ra giá trị đầu ra. Một số hàm kích hoạt bao gồm hàm ReLu và hàm softmax [40]. Một DNN điển hình bao gồm nhiều lớp ẩn mà ánh xạ một đầu vào thành một đầu ra. Ví dụ, mục tiêu của một DNN được huấn luyện cho phân loại hình ảnh là tạo ra một vector điểm số làm đầu ra, trong đó chỉ số vị trí có điểm số cao nhất tương ứng với lớp mà hình ảnh đầu vào được phân loại vào. Do đó, mục tiêu của việc huấn luyện một DNN là tối ưu hóa trọng số của mạng sao cho hàm mất mát (loss function) được tối thiểu hóa.

Trước khi huấn luyện, tập dữ liệu được chia thành hai phần: tập huấn luyện và tập kiểm tra. Sau đó, tập huấn luyện được sử dụng làm dữ liệu đầu vào để tối ưu hóa trọng số trong DNN. Trọng số được hiệu chỉnh thông qua stochastic gradient descent (SGD), trong đó trọng số được cập nhật bằng tích của tốc độ học lr, tức là bước giảm gradient descent trong mỗi lần lặp, và đạo hàm riêng của hàm mất mát L theo trọng số w. Công thức SGD như sau:

$$W = W - \mathrm{lr}\frac{\partial L}{\partial W} \tag{2.1}$$

$$\frac{\partial L}{\partial W} \approx \frac{1}{m}\sum_{i \in B}\frac{\partial l^{(i)}}{\partial W} \tag{2.2}$$

Công thức SGD được trình bày trong 2.1 là của mini-batch GD. Công thức 2.2 được suy ra như ma trận gradient trung bình trên các ma trận gradient của B batch, trong đó mỗi batch là một tập con ngẫu nhiên bao gồm m mẫu huấn luyện. Điều này ưu tiên hơn so với full batch GD, tức là khi toàn bộ tập huấn luyện được bao gồm trong việc tính toán đạo hàm riêng, vì full batch GD có thể dẫn đến quá trình huấn luyện chậm và ghi nhớ theo lô [41]. Các ma trận gradient được suy ra thông qua quá trình "backpropagation" từ đạo hàm đầu vào. Các vòng lặp huấn luyện sau đó được lặp lại qua nhiều epochs, tức là toàn bộ quá trình đi qua tập huấn luyện, để giảm thiểu mất mát. Một DNN được huấn luyện tốt tổng quát tốt, tức là đạt được độ chính xác suy luận cao khi áp dụng vào dữ liệu mà nó chưa từng thấy trước đây, chẳng hạn như bộ kiểm tra.



Hình 2.2: Kiến trúc hệ thống FL

Có các phương pháp học tập thay thế khác nhau, chẳng hạn như học tập bán giám sát [41], học tập không giám sát [42] và học tập tăng cường [43]. Ngoài ra, cũng có nhiều mạng và kiến trúc DNN được tùy chỉnh để xử lý các tính chất biến đổi của dữ liệu đầu vào, chẳng hạn như Multilayer Perceptron (MLP) [44], Convolutional Neural Network (CNN) [45] thường dùng cho các nhiệm vụ Thị giác máy tính, và Recurrent Neural Network (RNN) [46] thường được sử dụng cho các nhiệm vụ tuần tự.

2.2.2.2. Federated Learning

Được truyền cảm hứng từ lo ngại về quyền riêng tư của chủ sở hữu dữ liệu, khái niệm FL được giới thiệu trong [47]. FL cho phép người dùng cùng nhau huấn luyện một mô hình chung trong khi giữ dữ liệu cá nhân trên thiết bị của họ, từ đó giảm bớt mối quan ngại về quyền riêng tư. Như vậy, FL có thể phục vụ như một công nghệ hỗ trợ cho việc huấn luyện mô hình ML trên các mạng cạnh điện thoại di động (mobile edge networks).

Nhìn chung, có hai thực thể chính trong hệ thống FL, đó là chủ sở hữu dữ liệu (gọi là các người tham gia) và chủ sở hữu mô hình (gọi là máy chủ FL). Giả sử $\mathcal{N} = \{1, \ldots, N\}$ biểu thị tập hợp các chủ sở hữu dữ liệu, mỗi người trong số họ có một bộ dữ liệu riêng $D_{i \in \mathcal{N}}$. Mỗi chủ sở hữu dữ liệu i sử dụng bộ dữ liệu D_i của mình để huấn luyện một mô hình cục bộ $\mathbf{w_i}$ và chỉ gửi các tham số mô hình cục bộ đó tới máy chủ FL. Sau đó, tất cả các mô hình cục bộ thu thập được được tổng hợp lại thành mô hình toàn cầu $\mathbf{w_G}$, ký hiệu là $\mathbf{w} = \cup_{i \in \mathcal{N}} \mathbf{w}_i$

Điều này khác biệt so với quá trình huấn luyện truyền thống tập trung, trong đó sử dụng $\mathbf{D} = \cup_{i \in \mathcal{N}} D_i$ để huấn luyện mô hình $\mathbf{w_T}$, tức là dữ liệu từ mỗi nguồn cá nhân được tổng hợp

trước khi tiến hành huấn luyện mô hình ở trung tâm.

Với FL, dữ liệu không cần phải được gửi đến máy chủ tập trung, và các mô hình cục bộ được đào tạo trên thiết bị của chính người sở hữu dữ liệu. Điều này giúp giữ cho dữ liệu cá nhân ở trạng thái bảo mật trên các thiết bị và đảm bảo tính riêng tư trong quá trình huấn luyện mô hình toàn cầu. Qua đó, FL tạo điều kiện cho việc hợp tác trong huấn luyện mô hình mà không cần tiết lộ dữ liệu riêng tư.

Một kiến trúc và quy trình huấn luyện điển hình của một hệ thống FL được hiển thị trong Hình 2.2. Trong hệ thống này, các chủ sở hữu dữ liệu đóng vai trò là các người tham gia FL, họ cùng nhau huấn luyện một mô hình ML được yêu cầu bởi máy chủ tổng hợp. Một giả định cơ bản là các chủ sở hữu dữ liệu là trung thực, có nghĩa là họ sử dụng dữ liệu cá nhân thực sự của mình để huấn luyện và gửi các mô hình cục bộ thực sự đến máy chủ FL.

Nói chung, quá trình huấn luyện FL bao gồm ba bước sau đây:

- Local Model Training: Mỗi chủ sở hữu dữ liệu sử dụng dữ liệu của họ để huấn luyện một mô hình cục bộ ($\mathbf{w_i}$). Quá trình này xảy ra trên từng thiết bị tham gia, không cần gửi dữ liệu đến máy chủ tập trung.

- Local Model Update: Sau khi huấn luyện xong, các chủ sở hữu dữ liệu gửi các tham số của mô hình cục bộ của họ đến máy chủ FL.

- Global Model Aggregation: Máy chủ FL tổng hợp tất cả các mô hình cục bộ nhận được từ các người tham gia để tạo ra mô hình toàn cầu ($\mathbf{w_G}$). Quá trình này đảm bảo rằng mô hình toàn cầu đại diện cho thông tin được học từ tất cả các thiết bị tham gia mà không tiết lộ dữ liệu thực tế của bất kỳ chủ sở hữu dữ liệu nào.

Quá trình này lặp lại qua nhiều vòng lặp và trên nhiều epochs cho đến khi mô hình toàn cầu hội tụ và đạt được hiệu suất tốt trên dữ liệu mới. Quá trình này được lặp lại qua nhiều vòng lặp cho đến khi đạt được điều kiện dừng, chẳng hạn như số lượng vòng lặp tối đa hoặc đạt được độ hội tụ đủ. Qua đó, mô hình toàn cầu được cải thiện theo từng bước, kết hợp thông tin từ các mô hình cục bộ để đạt được hiệu suất tốt trên tập dữ liệu toàn cầu.

- Bước 1 (Khởi tạo nhiệm vụ): Máy chủ quyết định nhiệm vụ huấn luyện, tức là ứng dụng mục tiêu và yêu cầu dữ liệu tương ứng. Máy chủ cũng chỉ định các siêu tham số của mô hình toàn cầu và quá trình huấn luyện, chẳng hạn như tốc độ học. Sau đó, máy chủ phát sóng mô hình toàn cầu đã được khởi tạo $\mathbf{w_{G0}}$ và nhiệm vụ tới các người tham gia đã được lựa chọn.

- Bước 2 (Huấn luyện và cập nhật mô hình cục bộ): Dựa trên mô hình toàn cầu hiện tại w_{Gt}, trong đó t đại diện cho chỉ số vòng lặp hiện tại, mỗi người tham gia tương ứng sử dụng dữ liệu cục bộ và thiết bị của mình để cập nhật các tham số mô hình cục bộ wit. Mục tiêu của người tham gia i trong vòng lặp t là tìm các tham số tối ưu w_i^t mà giảm thiểu hàm mất mát $L(w_i^t)$, tức là:

$$\mathbf{w}_i^{t^*} = \arg\min_{\mathbf{w}_i^t} L\left(\mathbf{w}_i^t\right) \tag{2.3}$$

  Các tham số mô hình cục bộ đã được cập nhật sau đó được gửi đến máy chủ.

- Bước 3 (Bộ tổng hợp và cập nhật mô hình toàn cầu): Máy chủ tổng hợp các mô hình cục bộ từ các người tham gia và sau đó gửi lại các tham số mô hình toàn cầu đã được cập nhật $\mathbf{w}_G^{t+1}$ cho các chủ sở hữu dữ liệu.

Máy chủ giảm thiểu hàm mất mát toàn cầu $L\left(\mathbf{w}_G^t\right)$:

$$L\left(\mathbf{w}_G^t\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(\mathbf{w}_i^t\right) \tag{2.4}$$

Quá trình huấn luyện FL được lặp lại cho đến khi hàm mất mát toàn cầu hội tụ, hoặc đạt được độ chính xác mong muốn.

2.2.3. Các thách thức chính của Federated Learning (FL)

Tiếp theo, tôi mô tả bốn trong số những thách thức cốt lõi liên quan đến việc giải quyết vấn đề tối ưu phân tán được đưa ra trong 2.4. Những thách thức này làm cho khung cảnh liên kết khác biệt so với các vấn đề cổ điển khác, chẳng hạn như học phân tán trong môi trường trung tâm dữ liệu hoặc phân tích dữ liệu riêng truyền thống.

2.2.3.1. Giao tiếp Tốn kém.

Giao tiếp là một rào cản quan trọng trong các mạng liên kết, đặc biệt khi kết hợp với mối lo ngại về quyền riêng tư khi gửi dữ liệu gốc, làm cho việc đảm bảo dữ liệu được tạo ra trên mỗi thiết bị duy trì tại chỗ. Thực tế, các mạng liên kết có thể bao gồm một số lượng khổng lồ các thiết bị, chẳng hạn hàng triệu điện thoại thông minh, và tốc độ giao tiếp trong mạng có thể chậm hơn đáng kể so với tính toán cục bộ, thậm chí chậm hơn nhiều lần [48, 49]. Vì vậy, để phù hợp mô hình với dữ liệu được tạo ra bởi các thiết bị trong mạng liên kết, cần phải phát triển các phương pháp giao tiếp hiệu quả, mà gửi thông điệp nhỏ hoặc cập nhật mô hình theo cách lặp lại là một phần của quá trình đào tạo, thay vì gửi toàn bộ tập dữ liệu qua mạng. Để làm giảm giao tiếp hơn trong tình huống như vậy, có hai khía cạnh quan trọng cần xem xét: (i) giảm tổng số vòng giao tiếp, hoặc (ii) giảm kích thước các thông điệp truyền tải trong mỗi vòng.

2.2.3.2. Sự không đồng nhất trong Hệ thống.

Khả năng lưu trữ, tính toán và khả năng giao tiếp của mỗi thiết bị trong các mạng liên kết có thể khác nhau do sự biến đổi về phần cứng (CPU, bộ nhớ), kết nối mạng (3G, 4G, 5G, wifi) và nguồn điện (mức pin). Ngoài ra, kích thước mạng và ràng buộc liên quan đến hệ thống trên mỗi thiết bị thường dẫn đến chỉ có một phần nhỏ của các thiết bị được hoạt động cùng một lúc, chẳng hạn hàng trăm thiết bị hoạt động trong một mạng hàng triệu thiết bị [34]. Mỗi thiết bị cũng có thể không đáng tin cậy, và không phải là hiếm khi một thiết bị hoạt động có thể bị ngừng lại tại một vòng lặp cụ thể do ràng buộc về kết nối hoặc năng lượng. Những đặc điểm ở mức hệ thống này tác động mạnh mẽ làm tăng thêm các thách thức như khắc phục hiện tượng "straggler" và khả năng chống lỗi. Vì vậy, các phương pháp học liên kết được phát triển và phân tích phải: (i) dự đoán mức độ tham gia thấp, (ii) chịu đựng phần cứng không đồng nhất, và (iii) mạnh mẽ đối với các thiết bị bị ngừng hoạt động trong mạng.

2.2.3.3. Sự không đồng nhất thống kê.

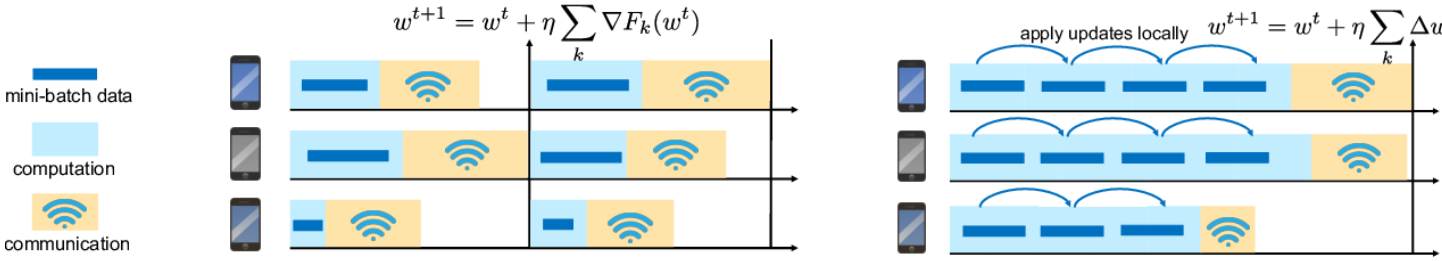
Các thiết bị thường tạo và thu thập dữ liệu theo cách không phân phối đồng nhất trên mạng, ví dụ, người dùng điện thoại di động sử dụng ngôn ngữ khác nhau trong ngữ cảnh của một nhiệm vụ dự đoán từ tiếp theo. Hơn nữa, số lượng điểm dữ liệu trên các thiết bị có thể biến đổi đáng kể, và có thể tồn tại một cấu trúc cơ bản mô tả mối quan hệ giữa các thiết bị và phân phối liên quan của chúng. Mô hình tạo dữ liệu này vi phạm các giả định phân phối độc lập và đồng nhất (Independent and Identically Distributed - I.I.D) thường được sử dụng trong tối ưu hóa phân tán, làm tăng khả năng xuất hiện các hiện tượng "straggler" và có thể gây thêm sự phức tạp trong việc mô hình hóa, phân tích và đánh giá. Thực sự, mặc dù vấn đề học liên kết cơ bản của [34] nhằm mục tiêu học một mô hình toàn cục duy nhất, nhưng còn tồn tại các phương án khác như học đồng thời các mô hình cục bộ riêng biệt thông qua các khung việc học đa nhiệm [50]. Có mối liên hệ mật thiết trong mặt này giữa các phương pháp tiên phong cho học liên kết và học thiết yếu [51]. Cả góc nhìn học đa nhiệm và học thiết yếu đều cho phép mô hình hóa cá nhân hoặc cụ thể cho từng thiết bị, đây thường là một phương pháp tự nhiên hơn để xử lý sự không đồng nhất thống kê của dữ liệu.

2.2.3.4. Mối lo ngại về Quyền riêng tư.

Cuối cùng, quyền riêng tư thường là một vấn đề quan trọng trong các ứng dụng học liên kết. Học liên kết thực hiện một bước tiến để bảo vệ dữ liệu được tạo ra trên mỗi thiết bị bằng cách chia sẻ cập nhật mô hình, chẳng hạn thông tin gradient, thay vì dữ liệu gốc [52, 53, 54]. Tuy nhiên, việc truyền thông tin cập nhật mô hình trong suốt quá trình đào tạo vẫn có thể tiết lộ thông tin nhạy cảm, cho một bên thứ ba, hoặc cho máy chủ trung tâm [55]. Trong khi các phương pháp gần đây nhằm nâng cao quyền riêng tư của học liên kết bằng cách sử dụng các công cụ như tính toán đa bên an toàn hoặc quyền riêng tư khác biệt, những phương pháp này thường cung cấp quyền riêng tư với mức độ hiệu suất mô hình giảm hoặc hiệu suất hệ thống bị giảm. Hiểu và cân nhắc những sự đánh đổi này, cả về mặt lý thuyết và thực nghiệm, là một thách thức đáng kể trong việc thực hiện các hệ thống học liên kết riêng tư.

2.2.4. Các công trình nghiên cứu liên quan và tiến bộ gần đây trong lĩnh vực Federated Learning

Các thách thức trong học liên kết ban đầu có vẻ giống như các vấn đề cổ điển trong các lĩnh vực như quyền riêng tư, học máy quy mô lớn và tối ưu hóa phân tán. Ví dụ, nhiều phương pháp đã được đề xuất để giải quyết vấn đề truyền thông đắt đỏ trong cộng đồng học máy, tối ưu hóa và xử lý tín hiệu. Tuy nhiên, những phương pháp này thường không thể xử lý hoàn toàn quy mô của các mạng học liên kết, chưa nói đến các thách thức về tính không đồng nhất hệ thống và thống kê. Tương tự, trong khi quyền riêng tư là một khía cạnh quan trọng đối với nhiều ứng dụng học máy, các phương pháp bảo vệ quyền riêng tư cho học liên kết có thể khó để khẳng định một cách chặt chẽ do biến thiên thống kê trong dữ liệu, và có thể còn khó khăn hơn trong việc thực hiện do các ràng buộc hệ thống trên từng thiết bị và trên cả mạng có thể rất lớn. Trong phần này, tôi trình bày chi tiết hơn về những thách thức được trình bày trong Phần 2.2.3, bao



Hình 2.3: Bên trái: Phân tán SGD (mini-batch). Mỗi thiết bị, k, tính toán gradient từ một nhóm dữ liệu nhỏ để xấp xỉ $\nabla F_k(w)$, và những cập nhật mini-batch tổng hợp được áp dụng trên máy chủ. Bên phải: Các phương pháp cập nhật cục bộ. Mỗi thiết bị thực hiện ngay lập tức các cập nhật cục bộ, chẳng hạn gradient, sau khi tính toán và máy chủ thực hiện tổng hợp toàn cầu sau một số lượng cập nhật cục bộ biến đổi. Các phương pháp cập nhật cục bộ có khả năng giảm thiểu giao tiếp bằng cách thực hiện công việc thêm tại cấp địa phương.

gồm cuộc thảo luận về các kết quả cổ điển cũng như các công việc gần đây tập trung đặc biệt vào học liên kết.

2.2.4.1. Hiệu suất truyền thông

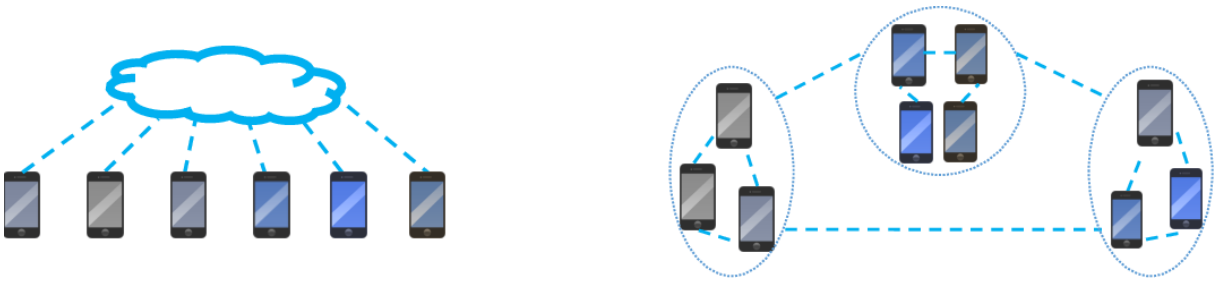
Truyền thông là một rào cản quan trọng cần xem xét khi phát triển các phương pháp cho các mạng học liên kết. Mặc dù việc cung cấp một bài đánh giá độc lập về các phương pháp học phân tán hiệu suất truyền thông vượt ra ngoài phạm vi của đồ án này, nên tôi chỉ ra một số hướng tổng quát, mà tôi chia thành (1) Phương pháp cập nhật cục bộ, (2) Các giải pháp nén, và (3) Huấn luyện phân tán.

*** Cập nhật cục bộ**

Các phương pháp tối ưu mini-batch đã trở thành một mô hình phổ biến trong việc học máy phân tán trong môi trường trung tâm dữ liệu [56, 57]. Các phương pháp này mở rộng các kỹ thuật ngẫu nhiên cổ điển để xử lý đồng thời nhiều điểm dữ liệu. Tuy nhiên, khả năng áp dụng của chúng đã thể hiện hạn chế trong việc quản lý cân bằng giữa giao tiếp và tính toán trong xử lý dữ liệu phân tán [58] . Như một phản ứng, những phương pháp gần đây đã xuất hiện để tăng cường hiệu quả giao tiếp trong các cài đặt phân tán bằng cách giới thiệu khái niệm số lượng cập nhật cục bộ biến đổi cho mỗi máy trong mỗi vòng giao tiếp, làm cho sự cân đối giữa tính toán và giao tiếp linh hoạt hơn.

Đối với các mục tiêu tối ưu hóa lồi, phương pháp cục bộ cập nhật phân tán dual cơ bản đã thu hút sự chú ý [59, 60]. Những chiến lược này khai thác cấu trúc chặn đối để phân tách mục tiêu toàn cầu thành các bài toán con có thể được giải quyết độc lập trong mỗi vòng giao tiếp. Các phương pháp cục bộ cập nhật cục bộ phân tán cũng đã xuất hiện, mang lại lợi ích bổ sung là khả năng áp dụng cho các mục tiêu không lồi [61]. Những tiến bộ này mang lại cải thiện hiệu suất đáng kể trong các tình huống thực tế, thể hiện tăng tốc đáng kể so với các phương pháp mini-batch cổ điển hoặc các kỹ thuật phân tán như ADMM [62] trong môi trường trung tâm dữ liệu thực tế. Hình 2.3 cung cấp hình dung trực quan về các phương pháp cập nhật cục bộ.

Trong các cài đặt phân tán, các phương pháp tối ưu cho phép cập nhật cục bộ linh hoạt và yêu cầu sự tham gia thấp từ các thiết bị đã trở thành các công cụ giải quyết de facto [63, 4, 50].



Hình 2.4: Thành phần tập trung và phi tập trung. Trong cài đặt học máy liên kết thông thường, giả sử một mạng sao (bên trái) trong đó máy chủ kết nối với tất cả các thiết bị từ xa. Các thành phần phi tập trung (bên phải) là một lựa chọn tiềm năng khi giao tiếp với máy chủ trở thành hạn chế.

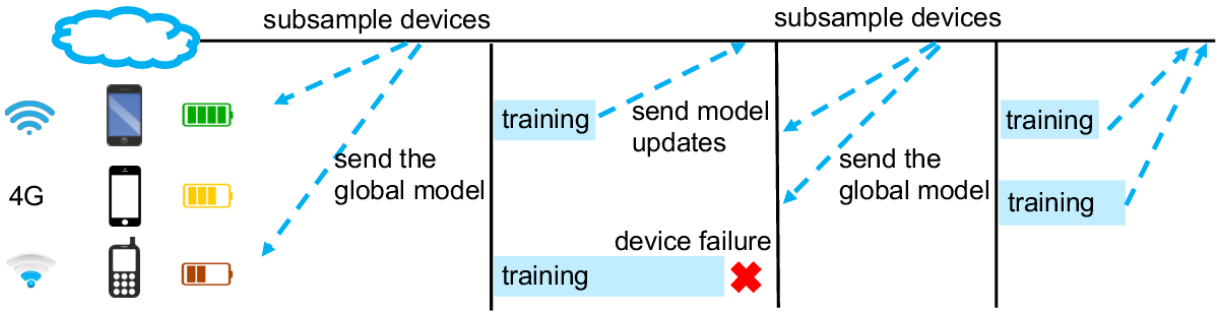
Phương pháp thường được sử dụng nhất cho việc học phân tán là Federated Averaging (FedAvg) [4], một phương pháp dựa trên việc lấy trung bình cập nhật của descent gradient ngẫu nhiên (SGD) cục bộ cho bài toán cơ bản. FedAvg đã được chứng minh hoạt động tốt thực nghiệm, đặc biệt là cho các vấn đề không lồi, nhưng không đảm bảo tính hội tụ và có thể bắt đầu biểu thức trong các tình huống thực tế khi dữ liệu không đồng nhất [63]. Tôi sẽ thảo luận về các phương pháp xử lý sự không đồng nhất thống kê như vậy một cách cụ thể hơn trong phần sau.

*** Các Phương pháp Nén** Trong khi các phương pháp cập nhật cục bộ có thể giảm số vòng giao tiếp tổng cộng, các phương pháp nén mô hình như làm thưa, lấy mẫu con và lượng tử hóa có thể giảm đáng kể kích thước của các thông điệp được truyền tải ở mỗi vòng giao tiếp. Các phương pháp này đã được nghiên cứu rộng rãi, cả về mặt thực nghiệm và lý thuyết, trong tài liệu trước đó về việc huấn luyện phân tán trong môi trường trung tâm dữ liệu; chúng tôi trì hoãn việc giới thiệu đến [64, 65] để có một bản đánh giá toàn diện hơn.

Trong các môi trường phân tán, việc tham gia thấp của các thiết bị, dữ liệu cục bộ không phân phối đồng nhất và các hệ thống cập nhật cục bộ đặt ra những thách thức mới đối với các phương pháp nén mô hình này. Ví dụ, các kỹ thuật bù lỗi thông thường được sử dụng rộng rãi trong việc học phân tán cổ điển [66] không thể được mở rộng trực tiếp sang các cài đặt phân tán khi lỗi tích luỹ cục bộ có thể trở nên lỗi lạc hậu nếu các thiết bị không được lấy mẫu thường xuyên. Tuy nhiên, một số nghiên cứu đã cung cấp các chiến lược thực tiễn trong các môi trường phân tán, chẳng hạn như buộc các mô hình cập nhật trở nên thưa và thấp-rank; thực hiện lượng tử hóa với các phép xoay ngẫu nhiên có cấu trúc [26]; sử dụng nén có mất mát và kỹ thuật dropout để giảm giao tiếp từ máy chủ đến thiết bị [67]; và áp dụng mã hóa không mất mát Golomb [68].

Từ góc độ lý thuyết, trong khi công trình trước đã khám phá tính hội tụ với huấn luyện có độ chính xác thấp trong bối cảnh dữ liệu không phân phối đồng nhất [69], các giả định được đưa ra không xem xét các đặc điểm chung của cài đặt phân tán, chẳng hạn như sự tham gia thấp của thiết bị hoặc các phương pháp tối ưu hóa cập nhật cục bộ.

*** Huấn luyện Phân tán** Trong việc học phân tán, mạng sao (trong đó máy chủ trung tâm được kết nối với mạng thiết bị, như trong bảng điều khiển bên trái của Hình 2.4) là mô hình tối ưu cho giao tiếp; do đó, chúng tôi tập trung vào cài đặt mạng sao trong bài viết này. Tuy nhiên, chúng tôi tóm tắt về các mô hình phi tập trung (trong đó thiết bị chỉ giao tiếp với các hàng xóm



Hình 2.5: Hình 4: Sự không đồng nhất về hệ thống trong học máy liên kết. Các thiết bị có thể khác nhau về kết nối mạng, năng lượng và phần cứng. Hơn nữa, một số thiết bị có thể ngừng hoạt động bất cứ lúc nào trong quá trình huấn luyện. Do đó, các phương pháp huấn luyện liên kết phải chịu đựng được môi trường hệ thống không đồng nhất và sự tham gia thấp của các thiết bị, tức là chúng phải cho phép chỉ một tập con nhỏ các thiết bị hoạt động ở mỗi vòng.

của nó, ví dụ như bảng điều khiển bên phải của Hình 2.4) như một phương án thay thế tiềm năng. Trong môi trường trung tâm dữ liệu, việc huấn luyện phi tập trung đã được chứng minh nhanh hơn so với việc huấn luyện tập trung khi hoạt động trên các mạng có băng thông thấp hoặc độ trễ cao; chúng tôi trì hoãn việc giới thiệu đến [70] để có một bản đánh giá toàn diện hơn. Tương tự, trong việc học phân tán, các thuật toán phi tập trung trong lý thuyết có thể giảm bớt chi phí giao tiếp cao trên máy chủ trung tâm. Một số công trình gần đây [70, 71] đã nghiên cứu việc huấn luyện phi tập trung trên dữ liệu không đồng nhất với các hệ thống cập nhật cục bộ. Tuy nhiên, chúng bị hạn chế hoặc chỉ áp dụng cho các mô hình tuyến tính [70] hoặc giả định về sự tham gia đầy đủ của các thiết bị [71]. Cuối cùng, các mô hình mẫu giao tiếp phân cấp cũng đã được đề xuất [72] để giảm gánh nặng trên máy chủ trung tâm, bằng cách sử dụng máy chủ cạnh đầu tiên để tổng hợp các cập nhật từ các thiết bị cạnh rồi dựa vào máy chủ đám mây để tổng hợp các cập nhật từ máy chủ cạnh. Mặc dù đây là một phương pháp hứa hẹn để giảm giao tiếp, nhưng nó không áp dụng cho tất cả các mạng, vì loại hình phân cấp vật lý này có thể không tồn tại hoặc không biết trước.

2.2.4.2. Sự không đồng nhất của Hệ thống

Trong cài đặt phân tán, có sự biến đổi đáng kể về đặc điểm của hệ thống trên toàn mạng, vì các thiết bị có thể khác nhau về phần cứng, kết nối mạng và nguồn năng lượng pin. Như được miêu tả trong Hình 2.5, các đặc điểm của hệ thống này làm cho các vấn đề như các thiết bị trễ trường hợp (stragglers) trở nên đáng kể hơn so với trong các môi trường trung tâm dữ liệu thông thường. Chúng tôi tổng quát chia thành một số hướng quan trọng để xử lý sự không đồng nhất của hệ thống: (i) giao tiếp không đồng bộ, (ii) lấy mẫu thiết bị hoạt động và (iii) khả năng chống lỗi. Như đã được đề cập trong Phần 2.2.4.1, chúng tôi giả định một mô hình mạng sao trong các thảo luận tiếp theo.

*** Giao Tiếp Bất đồng bộ** Trong các cài đặt trung tâm dữ liệu truyền thống, cả các kế hoạch đồng bộ và bất đồng bộ đều được sử dụng phổ biến để song song hóa các thuật toán tối ưu lặp lại, với mỗi phương pháp có những ưu điểm và nhược điểm riêng. Các kế hoạch đồng bộ đơn giản và đảm bảo một mô hình tính toán tương đương tuần tự, nhưng cũng dễ bị ảnh hưởng

bởi các thiết bị trễ trường hợp khi đối mặt với sự biến đổi của các thiết bị. Các kế hoạch bất đồng bộ là một phương pháp hấp dẫn để giảm thiểu sự trễ trường hợp trong các môi trường không đồng nhất, đặc biệt là trong các hệ thống bộ nhớ chung [73, 74]. Tuy nhiên, chúng thường dựa vào các giả định về trễ có giới hạn để kiểm soát mức độ lạc hậu, mà đối với thiết bị k phụ thuộc vào số lượng thiết bị khác đã cập nhật kể từ khi thiết bị k lấy từ máy chủ trung tâm. Trong khi máy chủ tham số bất đồng bộ đã thành công trong các trung tâm dữ liệu phân tán [73, 74], các giả định về trễ có giới hạn cổ điển có thể không thực tế trong các cài đặt phân tán, nơi trễ có thể trong khoảng từ vài giờ đến vài ngày, hoặc hoàn toàn không có giới hạn.

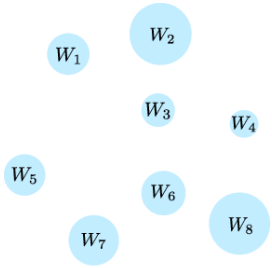
*** Lấy Mẫu Chủ Động (Active Sample)**

Trong các mạng phân tán, thường chỉ có một tập con nhỏ các thiết bị tham gia vào mỗi vòng huấn luyện. Tuy nhiên, phần lớn các phương pháp phân tán, ví dụ như những phương pháp được mô tả trong [63, 4, 50] là passively (chế độ thụ động) trong việc không cố gắng ảnh hưởng đến việc nào các thiết bị tham gia. Một phương pháp thay thế liên quan đến việc chọn lựa các thiết bị tham gia một cách tích cực ở mỗi vòng. Ví dụ, Nishio và Yonetani [75] khám phá các chính sách lấy mẫu thiết bị mới dựa trên tài nguyên hệ thống, với mục tiêu là cho máy chủ tổng hợp càng nhiều cập nhật từ thiết bị càng tốt trong một khoảng thời gian đã xác định trước. Tương tự, Kang el [76] xem xét các chi phí hệ thống phát sinh trên mỗi thiết bị khi thiết kế cơ chế khích lệ để khuyến khích các thiết bị có dữ liệu chất lượng cao tham gia vào quá trình học. Tuy nhiên, những phương pháp này giả định một mô hình tĩnh về các đặc điểm của hệ thống mạng; vấn đề vẫn còn mở là làm thế nào để mở rộng các phương pháp này để xử lý những biến đổi thời gian thực và độ trễ tính toán và giao tiếp của từng thiết bị. Hơn nữa, mặc dù những phương pháp này chủ yếu tập trung vào sự biến đổi của hệ thống để thực hiện lấy mẫu tích cực, chúng tôi chú ý rằng cũng đáng xem xét việc lấy mẫu tích cực một tập nhỏ nhưng đủ đại diện dựa trên cấu trúc thống kê cơ bản.

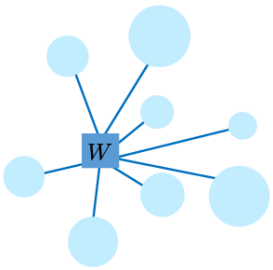
*** Khả Năng Chống Lỗi(Fault Tolerance)**

Khả năng chống lỗi đã được nghiên cứu rộng rãi trong cộng đồng hệ thống và là một yếu tố quan trọng trong các hệ thống phân tán cổ điển [77, 78]. Các công trình gần đây cũng đã nghiên cứu cụ thể về khả năng chống lỗi cho các tải công việc học máy trong môi trường trung tâm dữ liệu [79]. Tuy nhiên, khi học trên các thiết bị từ xa, khả năng chống lỗi trở nên quan trọng hơn vì thường xuyên một số thiết bị tham gia sẽ bỏ cuộc trước khi hoàn thành vòng lặp huấn luyện đã cho [34]. Một chiến lược thực tế là đơn giản là bỏ qua sự cố về thiết bị như vậy [34], điều này có thể mang lại sự thiên vị cho kế hoạch lấy mẫu thiết bị nếu các thiết bị gặp sự cố có đặc điểm dữ liệu cụ thể. Ví dụ, các thiết bị từ khu vực xa có thể có khả năng rơi ra khỏi vòng lặp vì kết nối mạng kém và do đó mô hình phân tán được huấn luyện sẽ có sự thiên vị hướng tới các thiết bị có điều kiện mạng thuận lợi. Về mặt lý thuyết, trong khi một số công trình gần đây đã nghiên cứu các đảm bảo hội tụ của các biến thể của các phương pháp học phân tán [80, 81, 82], ít phân tích cho phép tham gia thấp [63], hoặc nghiên cứu trực tiếp hiệu ứng của các thiết bị bị mất kết nối.

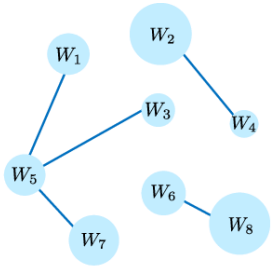
Tính toán mã hóa là một phương án khác để chống lại các sự cố về thiết bị bằng cách giới thiệu dư thừa thuật toán. Các công trình gần đây đã nghiên cứu việc sử dụng mã để tăng tốc quá trình huấn luyện học máy phân tán [83, 83, 84, 85] . Ví dụ, trong trường hợp gặp các thiết bị



(a) Learn personalized models for each device; do not learn from peers.



(b) Learn a global model; learn from peers.



(c) Learn personalized models for each device; learn from peers.

Hình 2.6: Các phương pháp mô hình hóa khác nhau trong mạng liên kết. Tùy thuộc vào các đặc tính của dữ liệu, mạng và ứng dụng cần quan tâm, người ta có thể chọn (a) Học các mô hình riêng biệt cho mỗi thiết bị, (b) Điều chỉnh một mô hình toàn cầu cho tất cả các thiết bị, hoặc (c) Học các mô hình liên quan nhưng khác biệt trong mạng.

trễ trường hợp, mã độ dốc và các biến thể của nó [83] [85] sao chép cẩn thận các khối dữ liệu (cũng như tính toán độ dốc trên các khối dữ liệu đó) trên các nút tính toán để đạt được phục hồi chính xác hoặc không chính xác của các độ dốc thực sự. Mặc dù đây là một phương pháp có vẻ hứa hẹn cho cài đặt phân tán, những phương pháp này đối mặt với các thách thức cơ bản trong các mạng phân tán khi chia sẻ dữ liệu bị sao chép qua các thiết bị thường không khả thi do ràng buộc về quyền riêng tư và quy mô của mạng.

2.2.4.3. Đa dạng thống kê

Có những thách thức phát sinh khi huấn luyện các mô hình phân tán từ dữ liệu không phân phối đồng nhất trên các thiết bị, cả về mặt mô hình hóa dữ liệu (được miêu tả trong Hình 2.6) và về mặt phân tích hành vi hội tụ của các quy trình huấn luyện liên quan. Chúng tôi sẽ thảo luận về công trình liên quan trong các hướng này dưới đây.

*** Mô hình hóa Dữ liệu Đa dạng**

Có một tập lớn các tài liệu trong lĩnh vực học máy đã mô hình hóa sự không đồng nhất thống kê thông qua các phương pháp như meta-learning [86] và học đa nhiệm [87]; những ý tưởng này đã được mở rộng gần đây vào cài đặt phân tán [88, 89, 90, 91, 92, 50]. Ví dụ, MOCHA [50], một khung tối ưu hóa được thiết kế cho cài đặt phân tán, có thể cho phép cá nhân hóa bằng cách học các mô hình riêng biệt nhưng liên quan cho mỗi thiết bị trong khi tận dụng một biểu diễn chia sẻ qua học đa nhiệm. Phương pháp này có các đảm bảo hội tụ lý thuyết cho các mục tiêu cân nhắc, nhưng bị hạn chế trong khả năng mở rộng lên các mạng lớn và bị hạn chế trong các mục tiêu lồi. Một phương pháp khác [90] mô hình hóa cấu trúc mạng sao như một mạng Bayes và thực hiện suy luận biến thể trong quá trình học. Mặc dù phương pháp này có thể xử lý các mô hình không lồi, nó tốn kém khi tổng quát hóa cho các mạng phân tán lớn. Khodak và đồng nghiệp [92] đã chứng minh rằng việc meta-học tốc độ học trong mỗi nhiệm vụ sử dụng thông tin học đa nhiệm (trong đó mỗi nhiệm vụ tương ứng với một thiết bị) có thể cải thiện hiệu suất thực nghiệm so với FedAvg thông thường. Eichner và đồng nghiệp [91] nghiên cứu một giải pháp phong phú (chọn linh hoạt giữa một mô hình toàn cục và các mô hình cụ thể

cho từng thiết bị) để giải quyết các mô hình chu kỳ trong các mẫu dữ liệu trong quá trình huấn luyện phân tán. Zhao và đồng nghiệp [88] khám phá việc học chuyển tiếp cho cá nhân hóa bằng cách chạy FedAvg sau khi huấn luyện mô hình toàn cục trên một số dữ liệu proxy chia sẻ. Mặc dù có những tiến bộ gần đây này, vẫn còn tồn tại những thách thức quan trọng trong việc tạo ra các phương pháp cho việc mô hình hóa không đồng nhất một cách mạnh mẽ, có khả năng mở rộng và tự động trong các cài đặt phân tán.

Khi mô hình hóa dữ liệu phân tán, cũng cần xem xét các vấn đề vượt qua khả năng chính xác, như sự công bằng. Đặc biệt, giải quyết một hàm mất mát tổng hợp như trong (1) một cách ngây thơ có thể ẩn dụ lợi thế hoặc bất lợi cho một số thiết bị, vì mô hình học sẽ có thể bị thiên lệch về các thiết bị có lượng dữ liệu lớn hơn, hoặc (nếu cân bằng trọng số của các thiết bị), đến các nhóm thiết bị thường xuyên xuất hiện. Các công trình gần đây đã đề xuất các phương pháp mô hình hóa sửa đổi nhằm giảm phương sai của hiệu suất mô hình trên các thiết bị. Một số cách làm đơn giản chỉ thực hiện một số lượng cập nhật cục bộ đa dạng dựa trên mất mát cục bộ [38]. Các phương pháp tiếp cận cơ bản hơn bao gồm Agnostic Federated Learning [93], tối ưu hóa mô hình tập trung cho bất kỳ phân phối mục tiêu nào được hình thành bằng sự kết hợp của các phân phối của khách hàng thông qua một kế hoạch tối ưu hóa minimax. Một phương pháp khác phổ quát hơn được đưa ra bởi Liel [94], đề xuất một mục tiêu được gọi là q-FFL trong đó các thiết bị với mất mát cao được gán trọng số tương đối cao hơn để khuyến khích giảm phương sai trong phân phối độ chính xác cuối cùng. Bên cạnh các vấn đề về sự công bằng, chúng tôi chú ý rằng các khía cạnh như sự chịu trách nhiệm và khả năng giải thích trong việc học phân tán cũng đáng được khám phá, nhưng có thể gặp khó khăn do quy mô và tính không đồng nhất của mạng.

*** Đảm bảo Hội tụ cho Dữ liệu không đồng nhất (Non-IID)**

Sự không đồng nhất thống kê cũng tạo ra những thách thức mới trong việc phân tích hành vi hội tụ trong các cài đặt phân tán, ngay cả khi đang học một mô hình toàn cục duy nhất. Thực sự, khi dữ liệu không phân phối đồng nhất trên các thiết bị trong mạng, các phương pháp như FedAvg đã được chứng minh là có thể phân kỳ trong thực tế [63, 4]. Parallel SGD và các biến thể liên quan, thực hiện các cập nhật cục bộ tương tự như FedAvg, đã được phân tích trong cài đặt I.I.D [72, 61, 95, 58, 96]. Tuy nhiên, kết quả dựa trên giả định rằng mỗi giải thuật cục bộ là một bản sao của quá trình ngẫu nhiên giống nhau (do giả định I.I.D), điều này không đúng trong các cài đặt phân tán thông thường. Để hiểu hiệu suất của FedAvg trong các cài đặt không đồng nhất thống kê, gần đây đã được đề xuất phương pháp FedProx [63]. FedProx thực hiện một điều chỉnh nhỏ cho phương pháp FedAvg để đảm bảo sự hội tụ, cả về mặt lý thuyết và thực nghiệm. FedProx cũng có thể được hiểu là phiên bản tổng quát, được điều chỉnh lại của FedAvg, có tác động thực tế trong ngữ cảnh tính đến sự không đồng nhất của hệ thống trên các thiết bị. Một số công trình khác [80, 81, 97, 82] cũng đã khám phá các đảm bảo hội tụ trong bối cảnh dữ liệu không đồng nhất với các giả định khác nhau, ví dụ như tính lồi [81]hoặc độ dốc giới hạn đồn đẳng [80]. Cũng có các phương pháp heuristics nhằm giải quyết sự không đồng nhất thống kê, bằng cách chia sẻ dữ liệu thiết bị cục bộ hoặc một số dữ liệu proxy trên máy chủ [38, 98]. Tuy nhiên, những phương pháp này có thể không thực tế: ngoài việc đặt gánh nặng cho băng thông mạng, gửi dữ liệu cục bộ lên máy chủ [98] vi phạm giả định quan trọng về quyền riêng tư của học phân tán, và việc gửi dữ liệu proxy chia sẻ toàn cầu đến tất cả thiết bị [38] đòi hỏi nỗ lực để

tạo ra hoặc thu thập cẩn thận những dữ liệu phụ trợ như vậy.

2.2.4.4. Quyền riêng tư

Mối lo ngại về quyền riêng tư thường thúc đẩy nhu cầu giữ dữ liệu thô trên mỗi thiết bị cục bộ trong các cài đặt phân tán. Tuy nhiên, việc chia sẻ thông tin khác như cập nhật mô hình như một phần của quá trình huấn luyện cũng có thể rò rỉ thông tin nhạy cảm của người dùng [52, 99, 100]. Ví dụ, Carlini và đồng nghiệp [52] đã chứng minh rằng có thể trích xuất các mẫu văn bản nhạy cảm, chẳng hạn như một số thẻ tín dụng cụ thể, từ một mạng nơ-ron tái phát được huấn luyện trên dữ liệu ngôn ngữ của người dùng. Do sự quan tâm ngày càng tăng về các phương pháp học bảo vệ quyền riêng tư, trong phần 2.2.4.4, tôi sẽ trước tiên tổng quan lại các công trình trước đây về việc nâng cao quyền riêng tư trong bối cảnh học máy tổng quát (phân tán). Sau đó, tôi sẽ xem xét các phương pháp bảo vệ quyền riêng tư gần đây được thiết kế đặc biệt cho các cài đặt phân tán trong phần 2.2.4.1

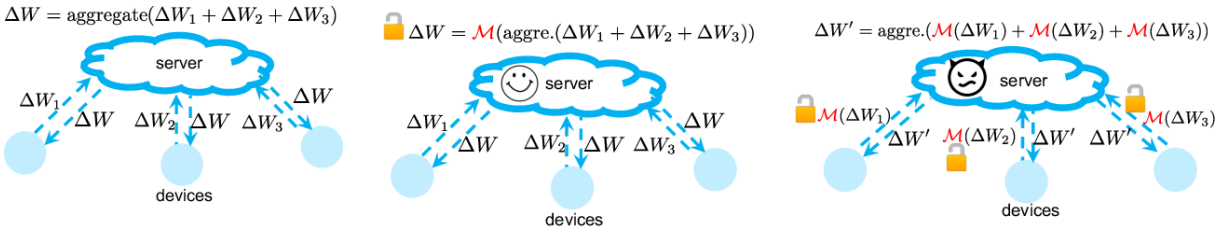
Quyền riêng tư trong Học máy

Học bảo vệ quyền riêng tư đã được nghiên cứu rộng rãi bởi cộng đồng học máy [55], hệ thống [101] và lý thuyết [102, 101]. Ba chiến lược chính, mỗi chiến lược chúng tôi sẽ đánh giá ngắn gọn, bao gồm sự riêng tư khác biệt để truyền tải các bản phác thảo dữ liệu nhiễu, mã hóa đồng nhất để hoạt động trên dữ liệu được mã hóa, và đánh giá chức năng an toàn hoặc tính toán đa bên.

Trong các phương pháp bảo vệ quyền riêng tư khác nhau này, riêng tư khác biệt [103, 54, 104] được sử dụng rộng rãi nhất do cam kết về thông tin lý thuyết mạnh, đơn giản về thuật toán và hệ thống tương đối nhỏ. Đơn giản thì một cơ chế ngẫu nhiên được coi là có đặc tính riêng tư khác biệt nếu việc thay đổi một phần tử đầu vào sẽ không gây ra sự khác biệt quá lớn trong phân phối đầu ra; điều này có nghĩa là không thể rút ra bất kỳ kết luận nào về việc mẫu cụ thể nào được sử dụng trong quá trình học. Quyền riêng tư cấp mẫu như vậy có thể được đạt được trong nhiều nhiệm vụ học [105, 106, 107, 108, 109]. Đối với các phương pháp học dựa trên độ dốc, một phương pháp phổ biến là áp dụng quyền riêng tư khác biệt bằng cách nhiễu ngẫu nhiên đầu ra trung gian ở mỗi vòng lặp [105, 106]. Trước khi áp dụng nhiễu, ví dụ như thông qua nhiễu Gaussian [105], nhiễu Laplace [77] hoặc nhiễu Binomial [110], thông thường sẽ cắt cấp độ dốc để giới hạn ảnh hưởng của mỗi ví dụ đối với cập nhật tổng thể.

Có sự đánh đổi cơ bản giữa quyền riêng tư khác biệt và độ chính xác của mô hình, khi thêm nhiều nhiễu sẽ dẫn đến quyền riêng tư lớn hơn, nhưng có thể làm tổn hại đáng kể đến độ chính xác. Mặc dù việc bảo vệ quyền riêng tư khác biệt đã trở thành thước đo de facto cho quyền riêng tư trong học máy, vẫn còn nhiều định nghĩa riêng tư khác như k-anonymity [111], d-presence [112] và distance correlation [113], có thể áp dụng cho các vấn đề học khác nhau [114].

*** Quyền riêng tư trong học liên kết** Cài đặt phân tán đặt ra các thách thức mới đối với các thuật toán bảo vệ quyền riêng tư hiện có. Ngoài việc cung cấp cam kết về quyền riêng tư chặt chẽ, cần phải phát triển các phương pháp có tính toán thấp, hiệu quả về giao tiếp và chịu được các thiết bị bị bỏ lỡ - tất cả mà không làm tổn hại quá nhiều đến độ chính xác. Mặc dù có nhiều định nghĩa về quyền riêng tư trong học máy phân tán [51, 115] thường chúng có thể được phân loại thành hai loại chính: quyền riêng tư toàn cầu và quyền riêng tư cục bộ. Như được minh



(a) Federated learning without additional privacy protection mechanisms.

(b) Global privacy, where a trusted server is assumed.

(c) Local privacy, where the central server might be malicious.

Hình 2.7: Minh họa về các cơ chế tăng cường quyền riêng tư khác nhau trong một vòng lặp của học máy liên kết. Ký hiệu "M" đại diện cho một cơ chế ngẫu nhiên được sử dụng để bảo vệ dữ liệu. Với quyền riêng tư toàn cầu (b), các cập nhật mô hình được bảo mật đối với tất cả các bên thứ ba ngoại trừ một bên đáng tin cậy duy nhất (máy chủ trung tâm). Với quyền riêng tư cục bộ (c), các cập nhật mô hình cá nhân cũng được bảo mật đối với máy chủ.

họa trong Hình 2.7, quyền riêng tư toàn cầu yêu cầu các cập nhật mô hình được tạo ra ở mỗi vòng lặp là riêng tư đối với tất cả các bên thứ ba không đáng tin cậy ngoại trừ máy chủ trung tâm, trong khi quyền riêng tư cục bộ yêu cầu thêm rằng các cập nhật cũng phải riêng tư đối với máy chủ.

Các công trình hiện tại nhằm cải thiện tính riêng tư của học máy phân tán thường xây dựng trên các giao thức mật mã cổ điển trước đây như SMC [116] và quyền riêng tư khác biệt [117, 110]. Tác giả trong [116] giới thiệu một giao thức SMC để bảo vệ các cập nhật mô hình cá nhân. Máy chủ trung tâm không thể thấy bất kỳ cập nhật nào cục bộ, nhưng vẫn có thể quan sát kết quả tổng hợp chính xác ở mỗi vòng. SMC là một phương pháp không mất dữ liệu và có thể duy trì độ chính xác ban đầu với cam kết quyền riêng tư rất cao. Tuy nhiên, phương pháp kết quả này gây thêm chi phí giao tiếp đáng kể. Các công trình khác [47, 117] áp dụng quyền riêng tư khác biệt vào học máy phân tán và cung cấp quyền riêng tư khác biệt toàn cầu. Những tiếp cận này có một số siêu tham số ảnh hưởng đến giao tiếp và độ chính xác phải được chọn cẩn thận, mặc dù các công trình theo sau [115] đề xuất các chiến lược cắt độ dốc biến đổi để giúp giảm thiểu vấn đề này. Trong trường hợp cần cam kết quyền riêng tư mạnh hơn, Bhowmick et al. [99] giới thiệu phiên bản nới lỏng của quyền riêng tư cục bộ bằng cách giới hạn sức mạnh của các kẻ đối thủ tiềm năng. Nó mang lại cam kết quyền riêng tư mạnh hơn so với quyền riêng tư toàn cầu và có hiệu suất mô hình tốt hơn so với quyền riêng tư cục bộ nghiêm ngặt. Li et al. [51] đề xuất các thuật toán có tính riêng tư khác biệt cục bộ trong bối cảnh học bất biến, có thể áp dụng vào học máy phân tán với cá nhân hóa, đồng thời cung cấp cam kết học có thể chứng minh trong các cài đặt lồi. Ngoài ra, quyền riêng tư khác biệt cũng có thể kết hợp với các kỹ thuật nén mô hình để giảm giao tiếp và đồng thời thu được lợi ích về quyền riêng tư [110].

2.2.5. Những hướng phát triển tiềm năng

Học máy liên kết là một lĩnh vực nghiên cứu đang hoạt động và tiếp tục phát triển. Mặc dù công trình gần đây đã bắt đầu giải quyết những thách thức được đề cập trong phần 2.2.3, nhưng vẫn còn nhiều hướng nghiên cứu quan trọng chưa được khám phá. Trong phần này, tôi tóm tắt một số hướng nghiên cứu triển vọng liên quan đến những thách thức đã thảo luận trước đó (giao

tiếp đất đỏ, sự đa dạng trong hệ thống, sự đa dạng thống kê và vấn đề riêng tư), và giới thiệu những thách thức bổ sung liên quan đến các vấn đề như triển khai thực tế và việc đánh giá trong môi trường học máy liên kết.

- **Các phương pháp truyền thông cực đoan.** Vẫn còn cần phải xem xét xem cần bao nhiêu truyền thông trong học máy liên kết. Thực tế là đã được biết đến rằng các phương pháp tối ưu hóa cho học máy có thể chịu được sự thiếu chính xác; thậm chí lỗi này có thể giúp cải thiện khả năng tổng quát [118]. Trong khi các phương pháp truyền thông một lần hoặc phân chia và chinh phục đã được nghiên cứu trong các môi trường trung tâm dữ liệu truyền thống [119, 120], hành vi của những phương pháp này chưa được hiểu rõ trong các mạng lớn hoặc thống kê đa dạng. Tương tự, các heuristics một lần hoặc vài lần [119, 121, 122] đã được đề xuất gần đây cho học máy liên kết, nhưng vẫn chưa được phân tích lý thuyết hoặc đánh giá ở quy mô lớn.

- **Giảm thiểu truyền thông và biên Pareto.** Tôi đã thảo luận về một số cách để giảm truyền thông trong quá trình đào tạo học máy liên kết, chẳng hạn như cập nhật cục bộ và nén mô hình. Để tạo ra một hệ thống thực tế cho học máy liên kết, điều quan trọng là hiểu cách các kỹ thuật này tương tác với nhau và phân tích hệ thống nhất quán giữa độ chính xác và truyền thông cho mỗi phương pháp. Cụ thể, các kỹ thuật hữu ích nhất sẽ thể hiện sự cải thiện tại biên Pareto - đạt được độ chính xác cao hơn bất kỳ phương pháp nào khác trong cùng ngân sách truyền thông, và lý tưởng nhất, trên một loạt rộng các hồ sơ truyền thông/độ chính xác. Các phân tích toàn diện tương tự đã được thực hiện cho việc suy diễn mạng neural hiệu quả [123], và cần thiết để so sánh các kỹ thuật giảm thiểu truyền thông cho học máy liên kết một cách có ý nghĩa.

- **Các mô hình không đồng bộ mới.** Như đã thảo luận trong phần 2.2.4.2, hai phương pháp truyền thông thường được nghiên cứu nhiều nhất trong tối ưu hóa phân tán là các phương pháp đồng bộ hàng loạt và không đồng bộ (trong đó giả định rằng độ trễ được giới hạn). Những phương pháp này thể hiện sự thực tế hơn trong môi trường trung tâm dữ liệu - nơi các nút làm việc thường được dành riêng cho khối công việc, tức là chúng sẵn sàng để 'kéo' công việc tiếp theo của họ từ nút trung tâm ngay sau khi họ 'đẩy' kết quả của công việc trước đó. Ngược lại, trong các mạng liên kết, mỗi thiết bị thường không dành riêng cho công việc cụ thể và hầu hết các thiết bị không hoạt động trong bất kỳ vòng lặp cụ thể nào. Do đó, đáng xem xét tác động của mô hình truyền thông tập trung vào thiết bị này một cách thực tế hơn - trong đó mỗi thiết bị có thể quyết định khi nào 'đánh thức' và tương tác với máy chủ trung tâm theo cách kích hoạt bởi sự kiện.

- **Chẩn đoán sự không đồng nhất.** Các công trình gần đây đã cố gắng định lượng sự không đồng nhất thống kê thông qua các chỉ số như sự không tương đồng cục bộ (như được định nghĩa trong ngữ cảnh của học máy liên kết trong [88] và được sử dụng cho các mục đích khác trong các công trình như [124, 125] và khoảng cách của máy đào đất [88]. Tuy nhiên, những chỉ số này không thể tính toán dễ dàng trên mạng liên kết trước khi quá trình đào tạo diễn ra. Tầm quan trọng của những chỉ số này thúc đẩy những câu hỏi mở sau đây: (i)

Liệu có tồn tại các chẩn đoán đơn giản để xác định nhanh mức độ không đồng nhất trong mạng liên kết trước? (ii) Có thể phát triển các chẩn đoán tương tự để định lượng lượng không đồng nhất liên quan đến hệ thống không? (iii) Có thể tận dụng các định nghĩa hiện tại hoặc mới về sự không đồng nhất để cải thiện thêm sự hội tụ của các phương pháp tối ưu hóa học máy liên kết không?

- **Ràng buộc riêng tư chi tiết** Các định nghĩa về riêng tư đã được trình bày trong Phần 2.2.4.4 bao gồm riêng tư ở mức cục bộ hoặc toàn cầu đối với tất cả các thiết bị trong mạng. Tuy nhiên, trong thực tế, có thể cần định nghĩa riêng tư ở mức chi tiết hơn, vì các ràng buộc riêng tư có thể khác nhau giữa các thiết bị hoặc thậm chí giữa các điểm dữ liệu trên cùng một thiết bị. Ví dụ, gần đây, Li et al. [51] đã đề xuất các cam kết về riêng tư cụ thể cho từng mẫu dữ liệu (không phải riêng cho người dùng), từ đó cung cấp một hình thức yếu hơn của riêng tư để đổi lấy các mô hình chính xác hơn. Phát triển các phương pháp để xử lý các ràng buộc riêng tư hỗn hợp (riêng tư cụ thể cho thiết bị hoặc cho từng mẫu) là một hướng nghiên cứu thú vị và tiếp tục trong tương lai.

- **Vượt ra ngoài học có giám sát.** Cần lưu ý rằng các phương pháp đã được thảo luận cho đến nay đã được phát triển với nhiệm vụ của học có giám sát trong tâm trí, tức là chúng giả định rằng tồn tại nhãn cho tất cả dữ liệu trong mạng liên kết. Trong thực tế, nhiều dữ liệu được tạo ra trong các mạng liên kết thực tế có thể là không được gán nhãn hoặc được gán nhãn yếu. Hơn nữa, vấn đề đang xem xét có thể không phải là đưa một mô hình về dữ liệu như được trình bày trong (1), mà thay vào đó là thực hiện một số phân tích dữ liệu thám hiểm, xác định thống kê tổng hợp hoặc thực hiện một nhiệm vụ phức tạp hơn như học củng cố. Giải quyết các vấn đề vượt ra ngoài học có giám sát trong các mạng liên kết có thể sẽ đòi hỏi giải quyết những thách thức tương tự về khả năng mở rộ, không đồng nhất và riêng tư.

- Triển khai sản xuất hóa học máy liên kết. Vượt qua những thách thức chính đã được thảo luận trong bài viết này, có một số vấn đề thực tế phát sinh khi triển khai học máy liên kết trong sản xuất. Đặc biệt, các vấn đề như thay đổi khái niệm (khi mô hình tạo dữ liệu cơ bản thay đổi theo thời gian); biến đổi hàng ngày (khi các thiết bị thể hiện hành vi khác nhau vào các thời điểm khác nhau trong ngày hoặc tuần) [91]; và vấn đề khởi đầu lạnh (khi các thiết bị mới tham gia vào mạng) cần phải được xử lý cẩn thận.

- **Thực nghiệm** Cuối cùng, vì học máy liên kết là một lĩnh vực mới nổi, chúng ta đang ở giai đoạn quan trọng để hình thành các phát triển trong lĩnh vực này và đảm bảo rằng chúng dựa trên các tình huống, giả định và tập dữ liệu thực tế. Điều quan trọng là cộng đồng nghiên cứu rộng lớn cần tiếp tục xây dựng trên các phiên bản hiện có và công cụ đánh giá, chẳng hạn như LEAF [126] và TensorFlow Federated [34], để hỗ trợ cả sự tái sản xuất kết quả thực nghiệm và việc phổ biến các giải pháp mới cho học máy liên kết.

CHƯƠNG 3: TỔNG QUAN PHƯƠNG PHÁP HỌC TĂNG CƯỜNG

3.1. Giới thiệu chương

Chương này tập trung giơi thiệu tổng quan khái niệm cơ bản, tính chất đặc trưng và ứng dụng của phương pháp học tăng cường và thuật toán SAC.

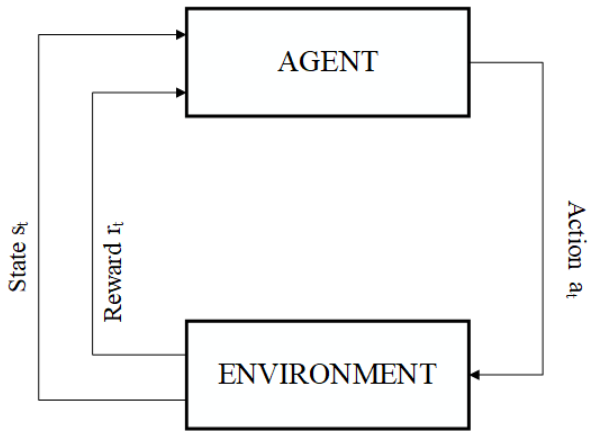
Bố cục của chương được chia lần lượt, mục 3.2 đi vào giới thiệu những kiến thức cơ bản về thuật toán Học tăng cường bao gồm: định nghĩa, phân loại, ưu điểm, nhược điểm và ứng dụng; mục 3.3 tập trung giới thiệu về mạng tế bào thần kinh nhân tạo; mục 3.4 giới thiệu về mạng Tác nhân - Đánh giá; cuối cùng mục 3.8 sẽ giới thiệu về thuật toán SAC với mô hình tổ chức và cách thức hoạt động của nó.

3.2. Thuật toán Học tăng cường - Reinforcement Learning

Ý tưởng cơ bản của thuật toán học tăng cường là dựa vào ý tưởng rằng con người học bằng cách tương tác với môi trường xung quanh, đó có lẽ là điều đầu tiên xảy ra với mỗi người khi nghĩ về bản chất của việc học. Khi một đứa trẻ chơi đùa, vẫy tay hoặc nhìn xung quanh, nó không có chỉ dẫn rõ ràng, nhưng nó có mối liên hệ trực tiếp về cảm giác với môi trường của nó. Thực hiện mối liên hệ này tạo ra nhiều thông tin về nguyên nhân và kết quả, về hậu quả của các hành động và về những việc cần làm để đạt được mục tiêu. Trong suốt cuộc đời của mỗi người, những tương tác như vậy chắc chắn là một nguồn kiến thức chính về môi trường và bản thân. Học từ sự tương tác là một ý tưởng nền tảng cơ bản cho gần như tất cả các lý thuyết về học tập và trí thông minh.

3.2.1. Định nghĩa

Học tăng cường là một quá trình huấn luyện để định rõ các hành động tương ứng với trạng thái của môi trường nhằm tối đa giá trị độ lợi tích lũy trong cả quá trình. Các vấn đề học tập củng cố liên quan đến việc học những gì cần làm - cách ánh xạ tình huống thành hành động - để tối đa hóa tín hiệu phần thưởng dưới dạng giá trị. Về mặt cơ bản, chúng là những vấn đề vòng lặp kín bởi vì các hành động của hệ thống học tập ảnh hưởng đến các đầu vào sau này của nó. Hơn nữa, người học không được cho biết phải thực hiện hành động nào, như trong nhiều hình thức học máy, mà thay vào đó, họ phải khám phá ra hành động nào mang lại phần thưởng nhiều nhất bằng cách thử chúng. Trong những trường hợp cụ thể và phức tạp, các hành động có thể ảnh hưởng không chỉ đến phần thưởng trước mắt mà còn ảnh hưởng đến tình huống tiếp theo và thông qua đó, tất cả các phần thưởng tiếp theo. Ba đặc điểm, khép kín theo một cách thiết yếu - không có hướng dẫn trực tiếp về những hành động cần thực hiện - độ hiệu quả của các hành động, bao gồm cả tín hiệu khen thưởng, phát ra trong khoảng thời gian dài, là ba đặc điểm phân biệt quan trọng nhất của vấn đề Học tập tăng cường. Cách thức hoạt động cơ bản của một mô



Hình 3.1: Cấu trúc cơ bản một hệ thống Học tăng cường

hình Học tăng cường được mô tả ở Hình 3.1.

Một tác nhân có khả năng nhận diện thông tin trạng thái của môi trường ở một mức độ nhất định và có khả năng thực hiện các hành động ảnh hưởng đến trạng thái đó. Tác nhân cũng phải có một mục tiêu hoặc các mục tiêu liên quan đến tình trạng của môi trường. Học tăng cường khác với học có giám sát, phương pháp học tập từ một tập hợp các ví dụ được dán nhãn được cung cấp bởi một giám sát viên bên ngoài có kiến thức [127]. Đối tượng của loại học tập này là để hệ thống ngoại suy hoặc tổng quát hóa các phản ứng của nó để nó hoạt động chính xác trong các tình huống không có trong tập huấn luyện. Đây là một hình thức học tập quan trọng, nhưng nếu chỉ học từ tương tác thì không đủ. Trong các bài toán tương tác, thường không thực tế để có được các mô phỏng về hành vi mong muốn vừa đúng vừa đại diện cho tất cả các tình huống mà tác nhân phải hành động. Trong phạm vi chưa được khám phá - nơi mà người ta mong đợi việc học sẽ có lợi nhất - một tác nhân phải có khả năng học hỏi từ kinh nghiệm của chính mình.

Học tăng cường cũng khác với học không giám sát, thường là về việc tìm kiếm cấu trúc ẩn trong các bộ sưu tập dữ liệu không được gắn nhãn [127]. Các thuật ngữ học tập có giám sát và học tập không giám sát dường như để phân loại đầy đủ các mô hình học máy, nhưng chúng thì không. Mặc dù nhiều tài liệu cho rằng học tập tăng cường như một loại học tập không có giám sát vì nó không dựa trên các ví dụ về hành vi đúng, RL đang cố gắng tối đa giá trị khen thưởng thay vì cố gắng tìm ra cấu trúc ẩn. Việc khám phá cấu trúc trong trải nghiệm của tác nhân chắc chắn có thể hữu ích trong việc học tăng cường, nhưng bản thân nó không giải quyết được vấn đề tối đa hóa giá trị độ thưởng của tác nhân học tăng cường. Do đó, phương pháp học tăng cường được công nhận là một mô hình học máy thứ ba, cùng với học có giám sát, học không giám sát và có lẽ cả các mô hình khác.

Một trong những thách thức nảy sinh trong Học tăng cường, chứ không phải trong các phương pháp huấn luyện máy tính khác, là sự đánh đổi giữa khám phá và khai thác. Để nhận được nhiều phần thưởng,Tác nhân trong Học tập tăng cường phải thích các hành động mà họ đã thử trong quá khứ và thấy có hiệu quả trong việc tạo ra phần thưởng. Nhưng để phát hiện ra các hành động như vậy, nó phải thử các hành động mà nó chưa chọn trước đó. Tác nhân phải khai thác những gì nó đã biết để nhận được phần thưởng, nhưng nó cũng phải khám phá để đưa ra các lựa chọn hành động tốt hơn trong tương lai. Một khó khăn của học tăng cường là hệ thống

không thể chỉ theo đuổi việc thăm dò hay khai thác. Tác nhân phải thử nhiều hành động khác nhau và dần dần xác định và có xu hướng chọn những hành động tốt nhất. Đối với một nhiệm vụ ngẫu nhiên, mỗi hành động phải được thử nhiều lần để đạt được giá trị thưởng mong đợi đáng tin cậy.

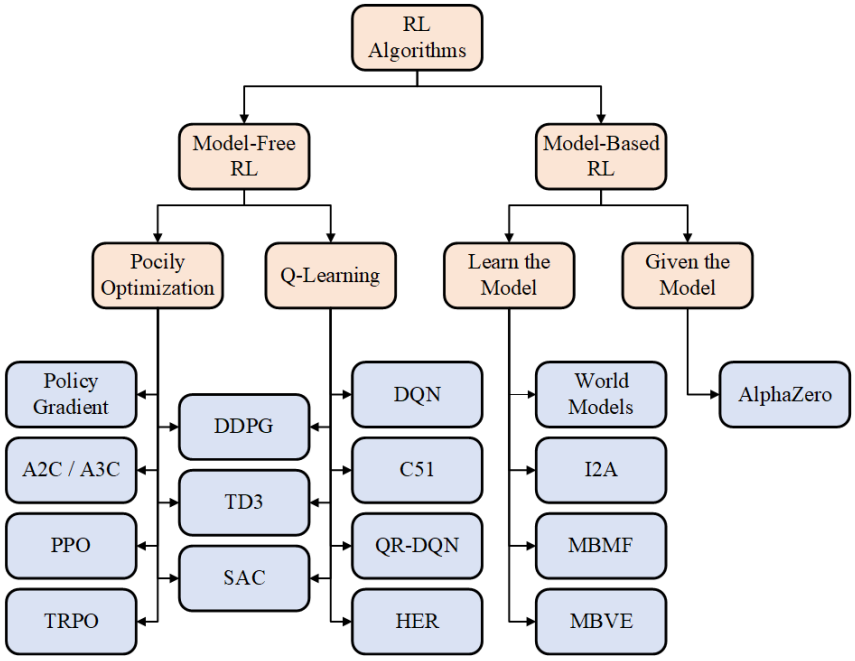
Một đặc điểm chính khác của học tăng cường là nó xem xét một cách rõ ràng toàn bộ vấn đề của tác nhân hướng đến mục tiêu tương tác với một môi trường không chắc chắn. Điều này trái ngược với nhiều cách tiếp cận xem xét các vấn đề con mà không giải quyết cách chúng có thể phù hợp với một bức tranh lớn hơn. Các nhà nghiên cứu khác đã phát triển lý thuyết về lập kế hoạch với các mục tiêu chung, nhưng không xem xét vai trò của lập kế hoạch trong việc ra quyết định theo thời gian thực hoặc câu hỏi về nguồn gốc của các mô hình dự báo cần thiết cho việc lập kế hoạch. Mặc dù những cách tiếp cận này đã mang lại nhiều kết quả hữu ích, nhưng việc tập trung vào các vấn đề con cô lập là một hạn chế đáng kể. Tất cả các tác nhân học tập củng cố đều có mục tiêu rõ ràng, có thể cảm nhận các khía cạnh của môi trường của họ và có thể chọn các hành động để tác động đến môi trường của họ. Hơn nữa, ngay từ đầu người ta thường giả định rằng tác nhân phải hoạt động mặc dù có sự không chắc chắn đáng kể về môi trường mà tác nhân phải đối mặt.

Một trong những khía cạnh thú vị nhất của việc học tăng cường hiện đại là sự tương tác thực chất và hiệu quả của nó với các ngành kỹ thuật và khoa học khác. Học tăng cường là một phần của xu hướng kéo dài nhiều thập kỷ trong trí tuệ nhân tạo và máy học hướng tới tích hợp nhiều hơn với thống kê, tối ưu hóa và các môn toán học khác. Đặc biệt hơn, việc học tăng cường cũng tương tác mạnh mẽ với tâm lý học và khoa học thần kinh, với những lợi ích đáng kể theo cả hai cách. Trong tất cả các hình thức học máy, học tăng cường là hình thức học gần nhất với hình thức học tập mà con người và các loài động vật khác làm, và nhiều thuật toán cốt lõi của học tăng cường ban đầu được lấy cảm hứng từ các hệ thống học tập sinh học. Và quá trình học tăng cường cũng đã đem lại hiệu quả, cả thông qua mô hình tâm lý học của động vật phù hợp hơn với một số dữ liệu thực nghiệm và thông qua mô hình có ảnh hưởng của các bộ phận trong hệ thống phần thưởng của não bộ.

3.2.2. Phân loại

Các mô hình thuật toán đại biểu được mô tả phân loại tổng quá qua Hình 3.2 bên dưới. Tuy nhiên, thực sự khá khó để vẽ ra một phân loại chính xác, toàn diện của các thuật toán trong không gian RL hiện đại, bởi vì tính mô-đun của các thuật toán không được thể hiện tốt bằng cấu trúc cây.

Một trong những yếu tố ảnh hưởng tới việc phân loại trong thuật toán Học tăng cường là thắc mắc liệu tác nhân có quyền truy cập (hoặc học) một mô hình của môi trường hay không. Ưu điểm chính của việc có một mô hình là nó cho phép tác nhân lập kế hoạch bằng cách suy nghĩ trước, xem điều gì sẽ xảy ra cho một loạt các lựa chọn có thể và quyết định rõ ràng giữa các lựa chọn của nó. Sau đó, các tác nhân có thể chắt lọc kết quả từ việc lập kế hoạch trước thành một chính sách đã học. Một ví dụ đặc biệt nổi tiếng của phương pháp này là AlphaZero. Khi điều này hoạt động, nó có thể dẫn đến sự cải thiện đáng kể về hiệu quả của mô hình so với các phương pháp không có mô hình.



Hình 3.2: Phân loại các thuật toán của thuật toán Học tăng cường

Nhược điểm chính là mô hình đáng tin cậy của môi trường thường không có sẵn cho tác nhân. Nếu tác nhân muốn sử dụng một mô hình trong trường hợp này, họ phải học hỏi mô hình đó hoàn toàn từ kinh nghiệm, điều này tạo ra một số thách thức. Thách thức lớn nhất là tác nhân có thể khai thác sự thiên vị trong mô hình, dẫn đến tác nhân hoạt động tốt so với mô hình đã học, nhưng lại hoạt động kém tối ưu trong môi trường thực. Học mô hình về cơ bản là khó, vì vậy ngay cả nỗ lực cao độ — sẵn sàng bỏ nhiều thời gian và tính toán — cũng không thể thành công.

Các thuật toán sử dụng mô hình được gọi là các phương pháp dựa trên mô hình và những thuật toán không được gọi là không có mô hình. Mặc dù các phương pháp không có mô hình cho thấy khả năng đạt được hiệu quả của việc lấy mẫu từ việc sử dụng một mô hình, nhưng chúng có xu hướng dễ thực hiện và dễ điều chỉnh hơn.

3.2.2.1. Phương pháp không dựa vào mô hình

Trong phương pháp dựa vào mô hình, hai cách tiếp cận chính để biểu diễn và huấn luyện tác nhân chính: Tối ưu hóa chính sách và Học theo Chất lượng. **Họ phương pháp Tối ưu hóa chính sách** được mô tả bằng một chính sách cụ thể, kí hiệu là $\pi_{theta}(a|s)$. Các chính sách tối ưu trực tiếp các tham số θ bằng cách tiến dần độ dốc hàm mục tiêu, kí hiệu là $J(\pi_{\theta})$ hoặc tối ưu gián tiếp bằng cách tính giá trị xấp xỉ cục bộ của $J(\pi_{\theta})$. Quá trình này hầu như được thực hiện theo chính sách, có nghĩa là mỗi bản cập nhật chỉ sử dụng dữ liệu được thu thập trong theo phiên bản mới nhất của chính sách. Tối ưu hóa chính sách cũng thường liên quan đến việc học một công cụ xấp xỉ $V_{phi}(s)$ cho hàm giá trị trên chính sách $V^{\pi}(s)$, được sử dụng để tìm ra cách cập nhật chính sách. Một số ví dụ của phương pháp Tối ưu hóa chính sách là:

- Thuật toán A2C / A3C sử dụng phương pháp độ dốc tiến dần để trực tiếp tối đa hiệu năng làm việc.

- Thuật toán PPO, trong đó các bản cập nhật gián tiếp tối đa hiệu suất, tối đa một hàm mục tiêu thay thế và đưa ra một ước tính chi tiết, do đó hàm $J(\pi_\theta)$ sẽ thay đổi.

Họ phương pháp Học theo Chất lượng tính toán một giá trị xấp xỉ $Q_\theta(s,a)$ của hàm giá trị hành động tối ưu $Q^*(s,a)$. Thông thường, hàm mục tiêu dược chọn dựa trên phương trình Bellman. Việc tối ưu hóa được thực hiện ngoài chính sách, có nghĩa là mỗi bản cập nhật có thể sử dụng dữ liệu được thu thập tại bất kỳ thời điểm nào trong quá trình huấn luyện. Chính sách tương ứng có được thông qua kết nối giữa Q^* và π^*: hành động toou ưu tính toán được quá trình Học Q của tác nhân được tính toán theo phương trình:

$$a(s) = arg\max_a Q_\theta(s,a) \tag{3.1}$$

Trong đó, a là hành động mang lại giá trị độ lợi tối ưu nhất và s là trạng thái tương ứng của hành động. Một số ví dụ về phương pháp học Q bao gồm:

- Thuật toán DQN, sử dụng hàm giá trị độ lợi Q với mạng , một trong những ứng dụng đầu tiên của thuật toán học sâu tăng cường.

- Thuật toán C51, một biến thể hoạc phân phối trên độ lợi có kì vọng Q^*.

3.2.2.2. Phương pháp dựa vào mô hình

Không giống như họ các phương pháp không dựa vào mô hình, các phương pháp huấn luyện dựa theo mô hình sử dụng một mô hình dựa đoán của môi trường để xác định hành động dựa vào nguyên lý "Điều gì sẽ xảy ra nếu tác nhân lựa chọn hành động này?", sau đó tác nhân sẽ lựa chọn hành động tốt nhất. Bởi trọng tâm nội dung đồ án là phương pháp DDPG thuộc nhóm các phương pháp không dựa vào mô hình nên họ các thuật toán dựa vào mô hình không được tập trung tìm hiểu.

3.2.3. Ưu thế

Mô hình học tập này rất giống với việc học tập của con người, đó là hoạc tập bằng cách phạm lỗi và sửa sai. Do đó, nó gần đạt được kết quả tối ưu. Học tập tăng cường mục tiêu là đạt được hành vi lý tưởng của một mô hình trong một bối cảnh cụ thể, để tối đa hóa hiệu suất của nó. Nó có thể tạo ra mô hình hoàn hảo để giải quyết một vấn đề cụ thể.

Thuật toán học tăng cường có thể được sử dụng để giải quyết các vấn đề rất phức tạp mà các kỹ thuật thông thường không thể giải quyết được. Một trong những kết quả nghiên cứu và ứng dụng của thuật toán là giúp robot có thể học cách đi bộ bằng cách triển khai các thuật toán học tăng cường.

Kỹ thuật này được ưa chuộng để đạt được kết quả lâu dài, điều mà đã từng rất khó đạt được bằng các phương pháp học máy trước. Mô hình có thể sửa chữa các lỗi xảy ra trong quá trình đào tạo, một khi lỗi được cập nhật vào mô hình, khả năng xảy ra cùng vấn đề là rất thấp. Do đó, càng huấn luyện, mô hình càng tối ưu và tránh được nhiều sai phạm đã mắc trong quá khứ hơn, và kết quả sẽ tiến dần đến kết quả tối ưu.

Trong trường hợp không có tập dữ liệu đào tạo, nó nhất định phải học hỏi kinh nghiệm từ tập dữ liệu đó. Các mô hình học tập củng cố có thể làm tốt hơn con người trong nhiều nhiệm vụ. Chương trình DeepMind's AlphaGo, một mô hình học tăng cường, đã đánh bại nhà vô địch thế giới Lee Sedol tại ván cờ vây vào tháng 3 năm 2016.

Nó có thể hữu ích khi cách duy nhất để thu thập thông tin về môi trường là tương tác với nó. Các thuật toán học tập tăng cường đảm bảo sự cân bằng giữa khám phá và khai thác. Khám phá là quá trình thử nghiệm những hành động khác nhau để xem liệu chúng có tốt hơn những thứ đã từng thử trước đó hay không. Khai thác là quá trình thử những thứ đã hoạt động tốt nhất trong quá khứ. Các thuật toán học tập khác không thực hiện sự cân bằng này.

3.2.4. Hạn chế

Mặc dù RL đã đạt được những thành tựu đáng kể nhưng không thể phủ nhận rằng thuật toán vẫn còn tồn tại nhiều hạn chế. Thuật toán RL với các hàm xấp xỉ, đặc biệt với mạng tế bào thần kinh sâu (Deep Neural Network), gặp phải bộ ba ràng buộc thỏa mãn, bao gồm tính không ổn định và tính phân kì gây ra bởi sự tích hợp của tính chất "ngoài chính sách", hàm xấp xỉ và tính tự khởi tạo của hệ thống.

Bản chất của kỹ thuật học tăng cường là nó sẽ "học" lại liên tục từ những hậu quả gây bởi những hành vi chưa phù hợp, do đó, thuật toán sẽ yêu cầu tài nguyên máy tính về bộ nhớ và chip xử lý cao. Ngoài ra vấn đề thời gian xử lý cũng sẽ lâu hơn so với những thuật toán khác. Việc học tăng cường quá nhiều có thể dẫn đến trạng thái quá tải, làm giảm kết quả.

Vì phương pháp học tăng cường khá phức tạp để khởi tạo và triển khai, do đó nó thường không được ưu tiên sử dụng để giải các bài toán đơn giản.

Học tăng cường cần nhiều dữ liệu và tính toán nhiều. Nó luôn cần nạp thêm dữ liệu thực tế và đó chính là lý do tại sao nó hoạt động thực sự tốt trong các trò chơi điện tử vì người ta có thể chơi trò chơi này đi chơi lại nhiều lần, vì vậy việc nhận được nhiều dữ liệu có vẻ khả thi.

Để giải quyết nhiều vấn đề của việc học tăng cường, chúng ta có thể sử dụng kết hợp việc học tăng cường với các kỹ thuật khác thay vì loại bỏ nó hoàn toàn. Một sự kết hợp phổ biến là Học tăng cường với Học sâu.

Sự khác biệt chính giữa học tăng cường và học sâu là: Học sâu là quá trình học từ một tập dữ liệu đào tạo và sau đó áp dụng học tập đó vào một tập dữ liệu mới. Nhưng học tập củng cố là quá trình học tập năng động bằng cách điều chỉnh các hành động dựa trên phản hồi liên tục để tối đa hóa giá trị độ lợi của hành động được chọn.

3.2.5. Ứng dụng

Ngày nay, thuật toán RL ngày càng được ứng dụng rộng rãi trên nhiều lĩnh vực. Một số ứng dụng của thuật toán RL có thể nhắc đến như: Xe tự hành, Hệ thống gợi ý ..., các ứng dụng tiêu biểu được mô tả ở Hình 3.3

Trong chăm sóc sức khỏe, bệnh nhân có thể được điều trị từ các chính sách học được từ hệ thống RL. RL có thể tìm ra các chính sách tối ưu bằng cách sử dụng các kinh nghiệm trước đó mà không cần thông tin trước đó về mô hình toán học của các hệ thống sinh học. Nó làm cho cách tiếp cận này áp dụng hơn so với các hệ thống dựa trên kiểm soát khác trong chăm sóc sức

Hình 3.3: Ứng dụng của thuật toán Học sâu tăng cường

khỏe.

Hệ thống giới thiệu có thể gợi ý các sản phẩm, dịch vụ, thông tin cho người dùng theo sở thích của họ. Nó giúp giảm bớt các vấn đề với thông tin tràn lan ngày nay bằng các đề xuất được cá nhân hóa, ví dụ: về tin tức, phim, nhạc, nhà hàng. Sở thích của người dùng có thể thay đổi thường xuyên, do đó, việc giới thiệu tin tức cho người dùng dựa trên các bài đánh giá và lượt thích có thể trở nên lỗi thời nhanh chóng. Với tính năng học tập tăng cường, lấy người dùng làm trung tâm, hệ thống RL có thể nhận biết được đặc điểm, tính cách, sở thích và hành vi hàng ngày của khách hàng bằng cách tương tác tự nhiên trong thời gian dài với họ. Horizon là nền tảng RL ứng dụng mã nguồn mở của Facebook.

Nhiều bài báo đã đề xuất Học sâu tăng cường để vận hành xe tự động [128]. Đối với ô tô tự lái, có nhiều khía cạnh khác nhau cần xem xét, chẳng hạn như giới hạn tốc độ ở những nơi khác nhau, khu vực có thể lái được, tránh va chạm [129]. Một số mục tiêu lái xe tự hành có thể áp dụng học tăng cường bao gồm tối ưu hóa quỹ đạo, lập kế hoạch chuyển động, đỗ xe tự động [130], tối ưu hóa bộ điều khiển và các chính sách học tập dựa trên thực tế của đường cao tốc.

Trong các xí nghiệp công nghiệp, các thiết bị robot dựa trên học tăng cường được sử dụng để thực hiện các nhiệm vụ khác nhau. Ngoài thực tế là những robot này hiệu quả hơn con người, chúng cũng có thể thực hiện các nhiệm vụ nguy hiểm cho con người. Việc sử dụng học sâu và học tăng cường có thể đào tạo robot có khả năng cầm nắm các vật thể khác nhau - ngay cả những vật thể không nhìn thấy được trong quá trình huấn luyện [131]. Ví dụ, điều này có thể được sử dụng để xây dựng các sản phẩm trong một dây chuyền lắp ráp.

Một ứng dụng trong lĩnh vực trò chơi của thuật toán RL là AlphaGo Zero. Sử dụng phương pháp học tăng cường, AlphaGo Zero đã có thể học trò chơi cờ vây từ đầu. Nó học được bằng cách chơi với chính nó. Sau 40 ngày tự luyện tập, Alpha Go Zero đã có thể vượt trội hơn phiên bản Alpha Go được biết đến với cái tên Master đã đánh bại kì thủ số một thế giới Ke Jie. Nó chỉ

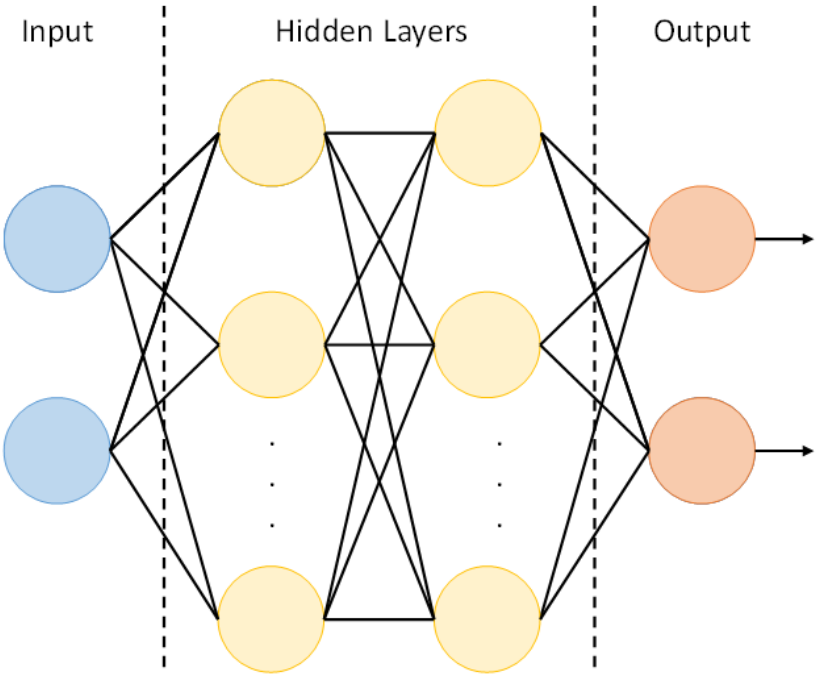
sử dụng màu sắc đen và trắng từ bảng làm các tính năng đầu vào và một mạng nơ-ron duy nhất kết hợp với một thuật toán cây tìm kiếm đơn giản để đánh giá các bước di chuyển vị trí và các bước di chuyển mẫu mà không cần sử dụng bất kỳ đợt triển khai nào của Monte Carlo.

Trong quy trình xử lý ngôn ngữ tự nhiên, RL có thể được sử dụng trong tóm tắt văn bản, trả lời câu hỏi và dịch máy. Sự kết hợp giữa học tập có giám sát và tăng cường được sử dụng để tóm tắt văn bản trừu tượng trong [132], mục tiêu của tác giả là giải quyết vấn đề gặp phải trong quá trình tóm tắt trong khi sử dụng các mô hình mã hóa-giải mã dựa trên RNN.

3.3. Mạng nơ-ron nhân tạo: Neural Network

Mạng tế bào thần kinh là chuỗi những thuật toán được đưa ra để hỗ trợ tìm kiếm những mối quan hệ cơ bản của một tập hợp dữ liệu dựa vào việc bắt chước cách thức hoạt động từ não bộ con người. Nói cách khác, mạng nơ ron nhân tạo được xem là hệ thống của các tế bào thần kinh nhân tạo có khả năng thích ứng được với mọi thay đổi từ đầu vào. Do vậy, nó có thể đưa ra được mọi kết quả một cách tốt nhất có thể mà không cần phải thiết kế lại những tiêu chí đầu ra.

Mạng tế bào thần kinh nhân tạo có thể hoạt động như mạng nơ-ron của con người. Mỗi một nút trong mạng nơ-ron nhân tạo là một hàm toán học với chức năng thu thập và phân loại các thông tin dựa theo cấu trúc cụ thể. Mạng nơ-ron nhân tạo có sự tương đồng chuẩn mạnh vối những phương pháp thống kê như đồ thị đường cong và phân tích hồi quy. Mạng nơ ron nhân tạo có chứa những lớp bao hàm các nút được liên kết lại với nhau. Mỗi nút lại là một tri giác có cấu tạo tương tự với hàm hồi quy đa tuyến tính.Bên trong một lớp tri giác đa lớp, chúng sẽ được sắp xếp dựa theo các lớp liên kết với nhau. Lớp đầu vào sẽ thu thập các mẫu đầu vào và lớp đầu ra sẽ thu nhận các phân loại hoặc tín hiệu đầu ra mà các mẫu đầu vào có thể phản ánh lại.Mỗi một mạng Mạng nơ ron nhân tạo thường bao gồm 3 tầng xử lí tín hiệu, được mô tả ở Hình 3.4



Hình 3.4: Cấu trúc mạng tế bào thần kinh

Tầng đầu vào nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng. Tầng đầu ra, ngoài cùng bên phải cùng và nó thể hiện cho những đầu ra của mạng. Tầng ẩn nằm giữa tầng vào và tầng ra nó thể hiện cho quá trình suy luận logic của mạng.

Trong mạng tế bào thần kinh nhân tạo, mỗi nút mạng là một sigmoid nơ-ron nhưng hàm kích hoạt của chúng có thể khác nhau. Tuy nhiên trong thực tế người ta thường để chúng cùng dạng với nhau để tính toán cho thuận lợi. Ở mỗi tầng, số lượng các nút mạng (nơ-ron) có thể khác nhau tuỳ thuộc vào bài toán và cách giải quyết. Nhưng thường khi làm việc người ta để các tầng ẩn có số lượng nơ-ron bằng nhau. Ngoài ra, các nơ-ron ở các tầng thường được liên kết đôi một với nhau tạo thành mạng kết nối đầy đủ. Khi đó ta có thể tính được kích cỡ của mạng dựa vào số tầng và số nơ-ron.

Với mạng mạng nơ ron nhân tạo thì mỗi nút mạng là một sigmoid nơ-ron nhưng chúng lại có hàm kích hoạt khác nhau. Thực tế, người ta thường sử dụng có cùng loại với nhau để việc tính toán thuận lợi hơn. Tại mỗi tầng, số lượng nút mạng có thể khác nhau còn tùy vào bài toán hoặc cách giải quyết. Tuy nhiên, khi làm việc người ta sẽ để các tầng ẩn số với số lượng nơ-ron khác nhau. Ngoài ra, những nơron nằm ở tầng thường sẽ liên kết đôi với nhau để tạo thành mạng kết nối đầy đủ nhất. Khi đó, người dùng có thể tính toán được kích cỡ của mạng dựa vào tầng và số lượng nơ ron. Trọng số liên kết của các nơ-ron giữa các lớp cũng sẽ có sự thay đổi khác nhau.

Thuật toán SAC sử dụng mạng Học theo Chất lượng sâu (DQN) trong việc huấn luyện mạng Đánh giá. Mạng DQN là sự kết hợp giữa mạng NN và một hàm tính giá trị Q, xác định bằng công thức:

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q(s^{'},a) \tag{3.2}$$

Trong đó $Q(s,a)$ là giá trị độ lợi tương ứng khi thực hiện hành động a tại trạng thái s, $r(s,a)$ là giá trị thưởng mà môi trường tính toán với chính cặp trạng thái - hành động này. Hệ số giảm độ lợi γ có vai trò đảm bảo càng về các lần lặp sau, sức ảnh hưởng của giá trị Q càng nhỏ.

Công thức 3.2 cho thấy giá trị độ lợi của hành động a ở trạng thái s bằng độ thưởng $r(s,a)$ cộng với giá trị Q lớn nhất của các trạng thái tiếp theo $s^{'}$ khi thực hiện các hành động khác. Sau các lần lặp, một ma trận giá trị độ lợi của các cặp hành động - trạng thái được tạo nên, khi này tác nhân chỉ cần chọn một hành động với mỗi trạng thái sao cho giá trị độ lợi là lớn nhất. Tuy nhiên, do thuật toán SAC là một quá trình ngẫy nhiên nên giá trị Q ở thời điểm trước và sau khi thực hiện hành động sẽ khác nhau, khác biệt này gọi sự khác biệt tạm thời, kí hiệu là $TD(a,s)$:

$$TD(a,s) = R(s,a) + \gamma \max_{a} Q(s^{'},a^{'}) - Q_{t-1}(s,a) \tag{3.3}$$

Do đó, ma trận giá trị độ lợi phải cập nhật cả trọng số TD theo công thức:

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha TD_t(a,s) \tag{3.4}$$

Trong đó α được gọi là tốc độ học tập, qua các lần lặp và lựa chọn hành động, giá trị $Q(s,a)$ sẽ dần hội tụ, đây chính là quá trình học Q.

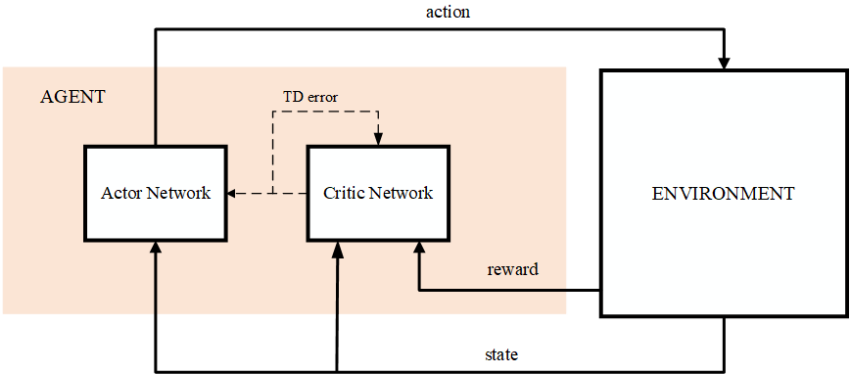
3.4. Thuật toán Actor - Critic

Thuật toán (AC) được phát triển từ những thập niên 90 của thế kỉ XX. Thuật toán được phát triển từ sự kết hợp điểm mạnh và khắc phục điểm yếu của mạng thuần Tác nhân và mạng thuần Đánh giá đã có từ trước.

Mạng thuần Tác nhân hoạt động với một họ chính sách được tham số hóa. Độ dốc (gradient) của hiệu suất huấn luyện được tính toán trực tiếp từ quá trình mô phỏng và trọng số của mạng được cập nhật để kết quả huấn luyện cải thiện hơn [133]. Một hạn chế có thể có của các phương pháp như vậy là các công cụ ước lượng gradient có thể có một phương sai lớn. Hơn nữa, khi chính sách thay đổi, một gradient mới được tính toán và độc lập với các kết quả trước đấy; do đó, không có "học", theo nghĩa là tích lũy và củng cố dữ liệu cũ.

Mạng thuần Đánh giá hoạt động dựa vào phép tính xấp xỉ hàm giá trị, mục đích tìm hiểu giải pháp gần đúng cho phương trình Bellman, sau đó hy vọng sẽ đưa ra một chính sách gần như tối ưu [133]. Các phương pháp như vậy là gián tiếp theo nghĩa là chúng không cố gắng tối ưu hóa trực tiếp trên một không gian chính sách. Các phương pháp thuộc nhóm này có thể thành công trong việc đạt được sự chấp thuận "tốt" của hàm giá trị, nhưng thiếu đảm bảo đáng tin cậy về mặt gần như tối ưu của chính sách kết quả.

Do đặc điểm của hai mạng thuần Tác nhân và Đánh giá, mạng AC có thể đạt được sự ổn định độ tin cậy tương đối tốt kể cả khi chính sách của mạng Đánh giá được thay đổi. Kiến trúc của một thuật toán AC được mô tả ở Hình 3.5:



Hình 3.5: Kiến trúc thuật toán AC

Trong thuật toán AC, mạng Tác nhân được gọi là mạng chính sách còn mạng Đánh giá được gọi là mạng thẩm định độ lợi. Trong khi mạng Tác nhân được dùng để lựa chọn các hành động tương ứng, mạng Đánh giá sẽ đánh giá độ lợi, hại của hành động được chọn bằng cách tính toán một hàm giá trị.

Theo như Hình 3.5, mạng Tác nhân và mạng Đánh giá là hai mạng riêng biệt nhưng dùng dùng chung thông tin trạng thái. Đầu tiên, môi trường sẽ gửi thông tin trạng thái hiện tại cho mạng Tác nhân. Sau đó, mạng Tác nhân sẽ phản hồi lại một hành động tương ứng, môi trường sẽ tính toán giá trị thưởng ứng với cặp trạng thái - hành động và gửi thông tin trạng thái cũng như thưởng cho mạng Đánh giá. Mạng Đánh giá sử dụng thông tin về trạng thái và thưởng để ước tính giá trị của hành động hiện tại và nó cũng liên tục điều chỉnh hàm giá trị. Trong khi đó,

mạng Tác nhân cập nhật chính sách hành động của mình theo hướng nâng cao giá trị của hành động. Trong một chu kỳ, mạng Đánh giá thẩm định việc lựa chọn hành động bằng hàm giá trị, cung cấp cho mạng Tác nhân ước tính "gradient" và cuối cùng, ta có được chiến lược hành động tối ưu. Việc đánh giá chính sách trong mạng Đánh giá là rất quan trọng, điều này có lợi hơn cho sự hội tụ và ổn định của mạng Actor hiện tại. Các đặc điểm trên đảm bảo rằng thuật toán AC có thể có được chiến lược hành động tối ưu với ước tính gradient ở phương sai thấp hơn.

3.5. Thuật toán học tăng cường điều chỉnh theo entropy

Entropy-Regularized Reinforcement Learning là một phương pháp trong học tăng cường, trong đó mục tiêu không chỉ là tối đa hóa tổng thưởng thu được từ việc thực hiện hành động, mà còn bao gồm việc tối đa hóa entropy của chính sách hành động. Entropy trong ngữ cảnh này được hiểu là mức độ "ngẫu nhiên" hoặc "không chắc chắn" của chính sách, tức là khả năng mà các hành động được chọn có thể thay đổi một cách ngẫu nhiên. Ví dụ, nếu một đồng xu được cân bằng để hầu như luôn luôn ra mặt trên, nó có entropy thấp; nếu nó được cân bằng và có một nửa cơ hội cho cả hai kết quả, nó có entropy cao.

Cho x là một biến ngẫu nhiên với hàm khối lượng xác suất hoặc hàm mật độ P. Entropy H của x được tính từ phân phối P theo công thức:

$$H(P) = \underset{x\sim P}{\mathrm{E}}[-\log P(x)] \tag{3.5}$$

Trong học tăng cường được điều chỉnh bằng entropy, tại mỗi bước thời gian, người học nhận được một phần thưởng bổ sung tỉ lệ thuận với entropy của chính sách hành động tại bước thời gian đó. Điều này làm thay đổi bài toán học tăng cường thành:

$$\pi^* = \arg\max_{\pi} \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{\infty} \gamma^t \left(R\left(s_t, a_t, s_{t+1}\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right)\right] \tag{3.6}$$

Ở đây, $\alpha > 0$ là hệ số cân đối. (Lưu ý: chúng ta đang giả sử ở đây là mô hình vô hạn với giá trị chiết khấu, và tôi sẽ tiếp tục như vậy cho phần còn lại của trang này.) Bây giờ chúng ta có thể định nghĩa các hàm giá trị có chút khác biệt trong bối cảnh này. V^{π} được thay đổi để bao gồm các phần thưởng entropy từ mọi bước thời gian:

$$V^{\pi}(s) = \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{\infty} \gamma^t \left(R\left(s_t, a_t, s_{t+1}\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right) \mid s_0 = s\right] \tag{3.7}$$

Q^{π} được thay đổi để bao gồm các phần thưởng entropy từ mọi bước thời gian trừ bước thời gian đầu tiên:

$$Q^{\pi}(s,a) = \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t, s_{t+1}\right) + \alpha \sum_{t=1}^{\infty} \gamma^t H\left(\pi\left(\cdot \mid s_t\right)\right) \mid s_0 = s, a_0 = a\right] \tag{3.8}$$

Với các định nghĩa này, V^{π} và Q^{π} được kết nối bởi:

$$V^{\pi}(s) = \underset{a\sim\pi}{\mathrm{E}}[Q^{\pi}(s,a)] + \alpha H(\pi(\cdot \mid s)) \tag{3.9}$$

Và phương trình Bellman cho Q^π là

$$\begin{aligned} Q^\pi(s,a) &= \mathop{\mathrm{E}}_{\substack{s'\sim P \\ a'\sim\pi}} \left[R\left(s,a,s'\right) + \gamma\left(Q^\pi\left(s',a'\right) + \alpha H\left(\pi\left(\cdot \mid s'\right)\right)\right)\right] \\ &= \mathop{\mathrm{E}}_{s'\sim P} \left[R\left(s,a,s'\right) + \gamma V^\pi\left(s'\right)\right]. \end{aligned} \tag{3.10}$$

3.6. Thuật toán DDPG

3.6.1. Giới thiệu chung

Deep Deterministic Policy Gradient (DDPG) là một thuật toán đồng thời học một hàm Q và một chính sách. Nó sử dụng dữ liệu ngoại tuyến và phương trình Bellman để học hàm Q, và sử dụng hàm Q để học chính sách.

Phương pháp này chặt chẽ liên quan đến Q-learning và có cùng động cơ: nếu bạn biết hàm giá trị hành động tối ưu $Q^*(s,a)$, thì ở bất kỳ trạng thái nào, hành động tối ưu $a^*(s)$ có thể được tìm bằng cách giải

$$a^*(s) = \arg\max_a Q^*(s,a). \tag{3.11}$$

DDPG xen kẽ việc học bộ xấp xỉ cho $Q^*(s,a)$ với việc học bộ xấp xỉ cho $a^*(s)$, và làm điều này theo cách được điều chỉnh đặc biệt cho môi trường có không gian hành động liên tục. Nhưng điều gì làm cho DDPG được điều chỉnh đặc biệt cho môi trường có không gian hành động liên tục? Điều này liên quan đến cách chúng ta tính giá trị lớn nhất qua các hành động trong $\max_a Q^*(s,a)$.

Khi có một số hữu hạn hành động rời rạc, việc tìm giá trị lớn nhất không gây vấn đề, vì chúng ta có thể tính toán giá trị Q cho mỗi hành động riêng lẻ và so sánh chúng trực tiếp. (Điều này cũng ngay lập tức cung cấp cho chúng ta hành động tối ưu hóa giá trị Q). Nhưng khi không gian hành động là liên tục, chúng ta không thể đánh giá toàn bộ không gian một cách thấu đáo và giải quyết vấn đề tối ưu là rất phức tạp. Sử dụng một thuật toán tối ưu thông thường sẽ làm cho việc tính toán $\max_a Q^*(s,a)$ trở thành một phần tử đắt đỏ. Và vì nó cần được thực hiện mỗi khi đại lý muốn thực hiện một hành động trong môi trường, điều này không thể chấp nhận được.

Bởi vì không gian hành động là liên tục, hàm $Q^*(s,a)$ được giả định có tính khả vi đối với đối số hành động. Điều này cho phép chúng ta thiết lập một quy tắc học dựa trên đạo hàm hiệu quả cho một chính sách $\mu(s)$ mà tận dụng sự thực này. Thay vì chạy một phần tử tối ưu đắt đỏ mỗi khi chúng ta muốn tính toán $\max_a Q(s,a)$, chúng ta có thể xấp xỉ nó bằng $\max_a Q(s,a) \approx Q(s,\mu(s))$.

3.6.2. Các phương trình quan trọng

3.6.2.1. Phần học hàm Q của DDPG

Đầu tiên, hãy tóm tắt phương trình Bellman mô tả hàm giá trị hành động tối ưu, $Q^*(s,a)$. Nó được cho bởi

$$Q^*(s,a) = \mathop{\mathrm{E}}_{s'\sim P}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a') \right] \tag{3.12}$$

trong đó $s' \sim P$ được viết tắt để nói rằng trạng thái kế tiếp, s', được lấy mẫu từ một phân phối $P(\cdot|s,a)$.

Phương trình Bellman này là điểm khởi đầu để học một ước lượng cho $Q^*(s,a)$. Giả sử ước lượng là một mạng neural $Q_\phi(s,a)$, với các tham số ϕ, và chúng ta đã thu thập một tập $\mathscr{D}$ các chuyển tiếp (s,a,r,s',d) (trong đó d chỉ ra xem trạng thái s' có là trạng thái kết thúc không). Chúng ta có thể thiết lập một hàm sai số trung bình bình phương Bellman (MSBE), cho biết độ chính xác gần như Q_ϕ đạt được phương trình Bellman:

$$L(\phi,\mathscr{D}) = \underset{(s,a,r,s',d)\sim\mathscr{D}}{\mathrm{E}}\left[\left(Q_\phi(s,a) - \left(r+\gamma(1-d)\max_{a'} Q_\phi(s',a')\right)\right)^2\right] \tag{3.13}$$

Ở đây, trong việc đánh giá $(1-d)$, chúng ta đã sử dụng một quy ước Python để đánh giá True thành 1 và False thành 0. Do đó, khi d==True, nghĩa là, khi s' là một trạng thái kết thúc, hàm Q phải thể hiện rằng người chơi không nhận được thêm phần thưởng sau trạng thái hiện tại.

Các thuật toán Q-learning cho các bộ xấp xỉ hàm, như DQN (và tất cả các biến thể của nó) và DDPG, chủ yếu dựa trên việc tối thiểu hóa hàm mất mát MSBE này. Có hai kỹ thuật chính được sử dụng bởi tất cả chúng, và sau đó là một chi tiết cụ thể cho DDPG.

Kỹ thuật Một: Hệ thống Lưu trữ (Replay Buffers). Tất cả các thuật toán tiêu chuẩn để huấn luyện mạng neural sâu xấp xỉ $Q^*(s,a)$ sử dụng một hệ thống lưu trữ trải nghiệm. Đó là tập hợp $\mathscr{D}$ của những trải nghiệm trước đó. Để thuật toán có hành vi ổn định, hệ thống lưu trữ trải nghiệm nên đủ lớn để chứa một loạt trải nghiệm rộng, nhưng không phải lúc nào cũng nên giữ tất cả. Nếu chỉ sử dụng dữ liệu gần đây nhất, sẽ quá khớp với nó và mọi thứ sẽ bị hỏng; nếu sử dụng quá nhiều trải nghiệm, có thể làm chậm quá trình học. Điều này có thể cần một chút điều chỉnh để đạt được sự cân đối. Kỹ thuật Hai: Mạng Mục tiêu. Các thuật toán Q-learning sử dụng mạng mục tiêu. Biểu thức:

$$r+\gamma(1-d)\max_{a'} Q_\phi(s',a') \tag{3.14}$$

Được gọi là mục tiêu, bởi vì khi chúng ta tối thiểu hóa hàm mất mát MSBE, chúng ta đang cố gắng làm cho hàm Q trở nên giống hơn với mục tiêu này. Vấn đề ở đây là mục tiêu phụ thuộc vào các tham số mà chúng ta đang cố gắng huấn luyện: ϕ. Điều này làm cho việc tối thiểu hóa MSBE trở nên không ổn định. Giải pháp là sử dụng một tập hợp tham số gần với ϕ, nhưng với một độ trễ thời gian - tức là một mạng thứ hai, được gọi là mạng mục tiêu, trễ hơn mạng chính. Các tham số của mạng mục tiêu được ký hiệu là ϕ_{targ}.

Trong các thuật toán dựa trên DQN, mạng mục tiêu chỉ được sao chép từ mạng chính sau mỗi một số bước cố định. Trong các thuật toán dạng DDPG, mạng mục tiêu được cập nhật một lần cho mỗi cập nhật mạng chính bằng cách lấy trung bình polyak:

$$\phi_{\text{targ}} \leftarrow \rho\phi_{\text{targ}} + (1-\rho)\phi, \tag{3.15}$$

Trong đó ρ là một siêu tham số nằm giữa 0 và 1 (thường gần với 1). (Siêu tham số này được gọi là polyak trong mã nguồn).

Chi tiết DDPG: Tính toán Max qua Các hành động trong Mục tiêu. Như đã đề cập trước đó: việc tính toán giá trị lớn nhất qua các hành động trong mục tiêu là một thách thức trong

không gian hành động liên tục. DDPG giải quyết vấn đề này bằng cách sử dụng mạng chính sách mục tiêu để tính toán một hành động gần đạt tới việc tối đa hóa $Q_{\phi_{\text{targ}}}$. Mạng chính sách mục tiêu được tìm thấy bằng cách lấy trung bình polyak các tham số chính sách trong suốt quá trình huấn luyện.

Tóm lại, việc học Q-learning trong DDPG được thực hiện bằng cách tối thiểu hóa hàm mất mát MSBE sau đây với gradient ngẫu nhiên:

$$L(\phi, \mathscr{D}) = \underset{(s,a,r,s',d)\sim\mathscr{D}}{\mathrm{E}} \left[\left(Q_\phi(s,a) - \left(r + \gamma(1-d) Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s')) \right) \right)^2 \right], \tag{3.16}$$

Trong đó $\mu_{\theta_{\text{targ}}}$ là chính sách mục tiêu.

3.6.3. Phần Học chính sách của DDPG

Học chính sách trong DDPG khá đơn giản. Chúng ta muốn học một chính sách xác định $\mu_\theta(s)$ mà cho ra hành động tối đa hóa $Q_\phi(s,a)$. Bởi vì không gian hành động là liên tục và chúng ta giả định rằng hàm Q liên quan đến hành động có khả năng khả vi, chúng ta có thể thực hiện lên đỉnh gradient (chỉ liên quan đến tham số chính sách) để giải quyết.

$$\max_\theta \underset{s\sim\mathscr{D}}{\mathrm{E}} \left[Q_\phi(s, \mu_\theta(s)) \right]. \tag{3.17}$$

Lưu ý rằng các tham số hàm Q được xem xét như là hằng số ở đây.

3.7. Twin Delayed DDPG (TD3)

3.7.1. Giới thiệu chung

Twin Delayed DDPG (TD3) là một thuật toán học tăng cường được phát triển từ DDPG (Deep Deterministic Policy Gradient). Mặc dù DDPG có thể đạt được hiệu suất tốt trong một số trường hợp, nhưng thường thì nó dễ bị thiếu ổn định đối với siêu tham số và các loại điều chỉnh khác. Một trường hợp thất bại phổ biến của DDPG là hàm Q học được học bắt đầu đánh giá quá cao giá trị Q, dẫn đến chính sách bị phá vỡ, vì nó tận dụng sai sót trong hàm Q. Twin Delayed DDPG (TD3) là một thuật toán giải quyết vấn đề này bằng cách giới thiệu ba điều quan trọng:

- Học Double-Q bị cắt (Clipped Double-Q Learning). TD3 học hai hàm Q thay vì một (do đó được gọi là "twin"), và sử dụng giá trị Q nhỏ hơn trong hai giá trị Q để tạo thành các mục tiêu trong hàm lỗi Bellman.

- Cập nhật Chính sách "Được Trì Hoãn" (“Delayed” Policy Updates). TD3 cập nhật chính sách (và mạng mục tiêu) ít thường xuyên hơn so với hàm Q. Bài báo khuyến nghị cập nhật chính sách một lần cho mỗi hai lần cập nhật hàm Q.

- Làm mịn Chính sách Mục tiêu (Target Policy Smoothing). TD3 thêm nhiễu vào hành động mục tiêu, để làm cho việc chính sách tận dụng sai sót hàm Q khó hơn bằng cách làm mịn Q theo các thay đổi trong hành động.

Cùng với ba điều này, TD3 đem lại hiệu suất cải thiện đáng kể so với DDPG cơ bản.

3.7.2. Các phương trình quan trọng

TD3 đồng thời học hai hàm Q: Q_{ϕ_1} và Q_{ϕ_2}, thông qua việc giảm thiểu sai số Bellman bình phương, tương tự như cách DDPG học một hàm Q duy nhất. Để hiển thị cách TD3 thực hiện điều này và cách nó khác biệt so với DDPG thông thường, chúng ta sẽ làm việc từ phần bên trong nhất của hàm mất mát ra ngoài.

Đầu tiên: việc làm mịn chính sách mục tiêu. Các hành động được sử dụng để hình thành mục tiêu học Q dựa trên chính sách mục tiêu, $\mu_{\theta_{\text{targ}}}$, nhưng với việc thêm nhiễu bị cắt từng chiều của hành động. Sau khi thêm nhiễu bị cắt, hành động mục tiêu sau đó được cắt để nằm trong phạm vi hành động hợp lệ (tất cả các hành động hợp lệ, a, thoả mãn $a_{Low} \leq a \leq a_{High}$. Các hành động mục tiêu do đó là:

$$a'\left(s'\right) = \text{clip}\left(\mu_{\theta_{\text{targ}}}\left(s'\right) + \text{clip}(\varepsilon, -c, c), a_{\text{Low}}, a_{\text{High}}\right), \quad \varepsilon \sim \mathcal{N}(0, \sigma) \tag{3.18}$$

Việc làm mịn chính sách mục tiêu về cơ bản hoạt động như một bộ điều chỉnh đối với thuật toán. Nó giải quyết một chế độ thất bại cụ thể có thể xảy ra trong DDPG: nếu bộ xấp xỉ hàm Q phát triển một đỉnh sắc nét không chính xác cho một số hành động, chính sách sẽ nhanh chóng khai thác đỉnh đó và sau đó có hành vi dễ vỡ hoặc không chính xác. Điều này có thể được tránh bằng cách làm mịn hàm Q trên các hành động tương tự, mà việc làm mịn chính sách mục tiêu được thiết kế để thực hiện.

Tiếp theo: học double-Q bị cắt. Cả hai hàm Q đều sử dụng một mục tiêu duy nhất, được tính bằng cách sử dụng một trong hai hàm Q cho mục tiêu tạo ra giá trị mục tiêu nhỏ hơn:

$$y\left(r, s', d\right) = r + \gamma(1-d) \min_{i=1,2} Q_{\phi_{i,\,\text{targ}}}\left(s', a'\left(s'\right)\right) \tag{3.19}$$

Và sau đó cả hai hàm Q được học bằng cách hồi quy đến mục tiêu này:

$$\begin{aligned} L(\phi_1, \mathscr{D}) &= \underset{(s,a,r,s',d)\sim\mathscr{D}}{\text{E}}\left[\left(Q_{\phi_1}(s,a) - y\left(r,s',d\right)\right)^2\right], \\ L(\phi_2, \mathscr{D}) &= \underset{(s,a,r,s',d)\sim\mathscr{D}}{\text{E}}\left[\left(Q_{\phi_2}(s,a) - y\left(r,s',d\right)\right)^2\right]. \end{aligned} \tag{3.20}$$

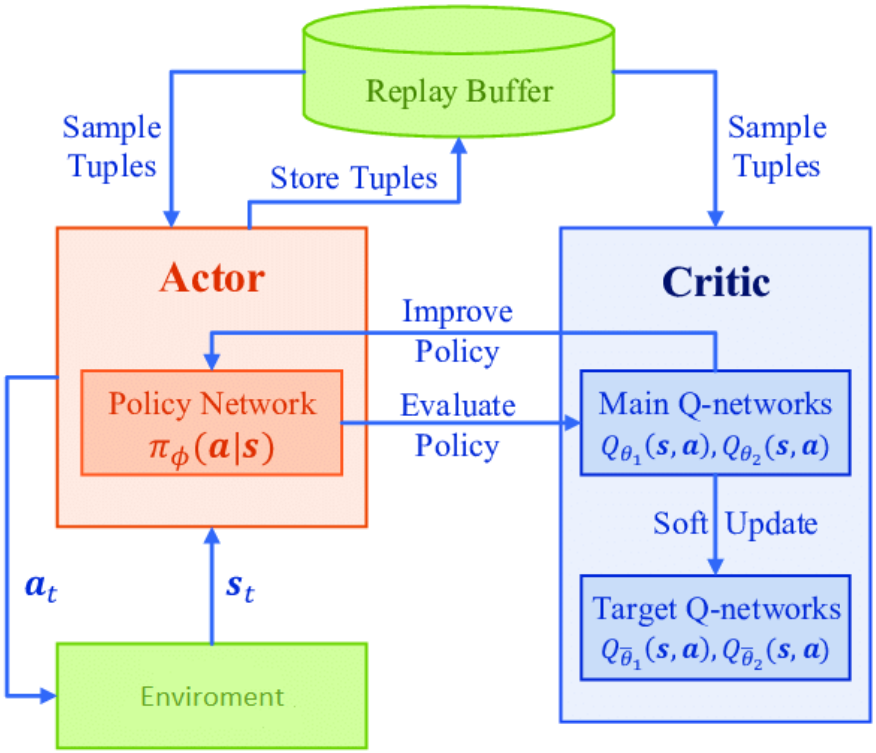
Cuối cùng: chính sách được học bằng cách tối đa hóa Q_{ϕ_1}:

$$\max_{\theta} \underset{s\sim\mathscr{D}}{\text{E}}\left[Q_{\phi_1}\left(s, \mu_\theta(s)\right)\right] \tag{3.21}$$

Điều này không thay đổi nhiều so với DDPG. Tuy nhiên, trong TD3, chính sách được cập nhật ít thường xuyên hơn so với việc cập nhật các hàm Q. Điều này giúp làm giảm độ biến động thường xuất hiện trong DDPG do cách cập nhật chính sách ảnh hưởng đến mục tiêu.

3.8. Thuật toán Soft Actor and Critic

Soft Actor Critic (SAC) là một thuật toán tối ưu hóa chính sách ngẫu nhiên theo cách ngoại tuyến (Off policy), tạo nên một cầu nối giữa việc tối ưu hóa chính sách ngẫu nhiên và các phương pháp theo kiểu DDPG 3.6. Nó không phải là một phiên bản kế thừa trực tiếp của TD3 3.7 (được xuất bản đồng thời), nhưng nó tích hợp kỹ thuật "clipped double-Q trick", và do tính



Hình 3.6: Cấu trúc thuật toán SAC

ngẫu nhiên bẩm sinh trong chính sách của SAC, nó cũng có lợi từ một điều gì đó tương tự như việc làm mịn chính sách mục tiêu.

Một đặc điểm trung tâm của SAC là việc sử dụng entropy regularization 3.5. Chính sách được huấn luyện để tối đa hóa sự cân đối giữa lợi tức kỳ vọng và entropy, một đo lường về sự ngẫu nhiên trong chính sách. Điều này có mối liên hệ mật thiết với sự cân bằng giữa việc khám phá và khai thác: tăng entropy dẫn đến việc khám phá nhiều hơn, điều này có thể gia tăng quá trình học sau này. Điều này cũng có thể ngăn chính sách sớm hội tụ đến một cực tiểu cục bộ xấu.

SAC đồng thời học một chính sách π_θ và hai hàm Q: Q_{ϕ_1} và Q_{ϕ_2}. Hiện nay, có hai biến thể của SAC được coi là tiêu chuẩn: một biến thể sử dụng hệ số điều chỉnh entropy cố định α, và một biến thể khác thực thi ràng buộc entropy bằng cách thay đổi α trong quá trình huấn luyện. Cấu trúc thuật toán SAC được mô tả cụ thể ở Hình 3.6 chứa chủ yếu 2 phần diễn xuất (Actor) và phê bình (Critic). Phần diễn xuất chứa mạng chính sách, chịu trách nhiệm đưa ra quyết định về việc giao nhiệm vụ dựa trên việc quan sát trạng thái hệ thống và đồng thời cải thiện chính sách. Nhà phê bình chứa các mạng Q chính và mạng Q mềm mục tiêu, có trách nhiệm đánh giá chính sách. Ngoài ra, bộ đệm lưu trữ trải nghiệm được sử dụng để lưu trữ các trải nghiệm giao nhiệm vụ, có thể được sử dụng để huấn luyện các mạng trong phần diễn xuất và nhà phê bình.

Để minh hoạ cách SAC hoạt động, tôi xin trình bày hàm chính sách và hàm giá trị. Sau đó, sẽ trình bày chi tiết phần nhà phê bình và phần diễn xuất của SAC.

3.8.1. Hàm chính sách và hàm giá trị

Trong học tăng cường, chính sách (policy) là một chiến lược chọn hành động xác định hoặc ngẫu nhiên để xác định giá trị kỳ vọng trong dài hạn, có thể là xác định hoặc ngẫu nhiên. Trong thuật toán, chính sách là ngẫu nhiên và có thể được biểu diễn là $\pi(a|s)$, có nghĩa là phân phối xác suất của các hành động dựa trên một trạng thái quan sát cụ thể. Mục tiêu của tác nhân là học một chính sách tối ưu $\pi^*(\mathbf{a} \mid \mathbf{s})$ mà tối đa hóa giá trị kỳ vọng tương ứng với phần thưởng được định nghĩa trong , và việc đánh giá và cải thiện chính sách được trình bày trong các phần tiếp theo. Chúng ta sau đó định nghĩa hai hàm, hàm giá trị hành động và hàm giá trị trạng thái. Vì hành động hiện tại có thể ảnh hưởng đến giá trị trả lại trong tương lai, hàm giá trị hành động $Q_\pi(s,a)$ được định nghĩa là giá trị trả lại đã giảm dần kỳ vọng của một chuỗi hành động bắt đầu từ thời điểm 0 với trạng thái s và lựa chọn hành động a, được biểu diễn như sau:

$$Q^\pi(\mathbf{s},\mathbf{a}) = \mathop{\mathbb{E}}_{\mu\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t R_t \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}\right] \tag{3.22}$$

Ở đây, R_t là tên ngắn gọn cho $R(s_t,a_t)$, $\gamma \in (0,1)$ là hệ số giảm dần, μ biểu thị cho dãy trạng thái-hành động $\{s_0,a_0,s_1,a_1,\ldots,s_{T-1},a_{T-1},s_T\}$. Tương tự, chúng ta định nghĩa $V_\pi(s)$ là hàm giá trị trạng thái, có nghĩa là giá trị trả về kỳ vọng đã giảm dần, bắt đầu từ trạng thái s và thực hiện các hành động theo chính sách π, được trình bày như sau:

$$V^\pi(\mathbf{s},\mathbf{a}) = \mathop{\mathbb{E}}_{\mu\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t R_t \mid \mathbf{s}_0 = \mathbf{s} = \mathbf{a}\right] \tag{3.23}$$

Trong thuật toán Soft Actor-Critic (SAC), mục tiêu tối ưu hóa không chỉ là tối đa hóa phần thưởng kỳ vọng, mà còn là tối đa hóa thông entropi của chính sách đồng thời. Do đó, hàm giá trị hành động và hàm giá trị trạng thái cần được điều chỉnh phù hợp với mục tiêu tối ưu hóa của thuật toán. Tương tự như hàm giá trị có trọng số được định nghĩa trong [134] và dựa vào 3.22, ta có thể biểu diễn hàm giá trị hành động mềm như sau:

$$Q_h^\pi(\mathbf{s},\mathbf{a}) = \mathop{\mathbb{E}}_{\mu\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t R_t + \alpha\sum_{t=1}^{T-1}\gamma^t H\left(\pi\left(\cdot \mid \mathbf{s}_t\right)\right) \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}\right] \tag{3.24}$$

Trong đó, α là hệ số nhiệt độ xác định mức độ quan trọng của entropi chính sách trong mục tiêu tối ưu hóa. Tương tự, theo 3.23, ta định nghĩa hàm giá trị trạng thái mềm như sau:

$$V_h^\pi(\mathbf{s}) = \mathop{\mathbb{E}}_{\mu\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t\left(R_t + \alpha H\left(\pi\left(\cdot \mid \mathbf{s}_t\right)\right)\right) \mid \mathbf{s}_0 = \mathbf{s}\right] \tag{3.25}$$

Vì số chiều của không gian trạng thái và không gian hành động có thể cực kỳ lớn, và quá trình lặp giá trị cho đến khi hội tụ có độ phức tạp tính toán quá cao, việc sử dụng bộ xấp xỉ hàm cho hàm giá trị hành động mềm và chính sách là cần thiết. Trong thuật toán, mạng nơ-ron sâu (DNN) được sử dụng để biểu diễn hàm giá trị hành động mềm và chính sách. Hàm giá trị hành động mềm $Q_h^\pi(s,a)$ có thể được tham số hóa thành $Q_\phi(s,a)$ bằng cách sử dụng một DNN kết nối đầy đủ chứa nhiều lớp ẩn, và ϕ biểu thị các tham số của mạng. Tương tự, chính sách $\pi(a|s)$ được tham số hóa thành $\pi_\phi(a|s)$ bằng một DNN kết nối đầy đủ, và ϕ biểu thị các tham số của mạng.

3.8.2. Đánh giá chính sách (Policy Evaluation)

Trong thuật toán, các hàm giá trị mềm có thể được tính theo cách lặp với một chính sách π đã cho. Theo phương trình Bellman, mối quan hệ giữa hàm giá trị hành động mềm và hàm giá trị trạng thái mềm được thể hiện như sau:

$$Q_h^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = R_t + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim \rho_x}\left[V_h^{\pi}(\mathbf{s}_{t+1})\right] \tag{3.26}$$

$$V_h^{\pi}(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q_h^{\pi}(\mathbf{s}_t, \mathbf{a}_t) - \alpha \log \pi(\mathbf{a}_t \mid \mathbf{s}_t)\right] \tag{3.27}$$

Trong đó, ρ_π là phân phối margin của trạng thái được tạo ra bởi chính sách π. Để ổn định quá trình huấn luyện trong việc lặp của hàm giá trị hành động mềm, chúng ta sử dụng một hàm giá trị hành động mềm mục tiêu với tham số ϕ, có thể được thu được thông qua trung bình trượt mũi tên của ϕ. Chúng ta định nghĩa hàm giá trị hành động mềm mục tiêu như sau:

$$\hat{Q}_{\delta}(\mathbf{s}_t, \mathbf{a}_t) = R_t + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim \rho_\pi}\left[V_{\bar{\partial}}(\mathbf{s}_{t+1})\right]. \tag{3.28}$$

Để phá vỡ các tương quan thời gian trong quá trình huấn luyện, chúng ta thiết lập một bộ đệm trải nghiệm $\mathscr{M}$ với kích thước cố định. Trong mỗi khung thời gian, sự chuyển tiếp của trạng thái trong môi trường xe cộ, hành động thực hiện và phần thưởng ngay lập tức tạo thành một bộ ba (s_t, a_t, R_t, s_{t+1}), sau đó được lưu trữ trong $\mathscr{M}$. Bằng cách lấy mẫu một lô nhỏ các bộ ba từ bộ đệm, bộ điều khiển và nhà phê bình được cập nhật. Sau đó, hàm mất mát của nhà phê bình trở thành:

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathscr{M}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right] \tag{3.29}$$

và các tham số của hàm giá trị hành động mềm được huấn luyện bằng cách tối thiểu hóa hàm mất mát $J_Q(\theta)$.

3.8.3. Cải thiện chính sách (Implement Policy)

Trong thuật toán, nếu một chính sách của việc chuyển giao nhiệm vụ là tối ưu, thì tất cả các nhiệm vụ chuyển giao trong một khoảng thời gian sẽ được hoàn thành với thời gian và giá thấp nhất, và tổng tiện ích cũng sẽ tối đa. Ngược lại, nếu một số nhiệm vụ chuyển giao được hoàn thành trong thời gian dài hoặc không thể hoàn thành trước thời hạn, giá trị tiện ích sẽ nhỏ hơn, điều này phản ánh rằng chính sách hiện tại không tốt. Do đó, chính sách cần được cải thiện. Trong SAC, các tham số chính sách có thể được học bằng cách tối thiểu hóa độ lệch KL kỳ vọng, được thể hiện như sau:

$$\pi_t = \arg\min_{\pi' \in \mathrm{II}} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\cdot \mid \mathbf{s}_t) \| \frac{\exp\left(\frac{1}{\alpha} Q^{\pi}(\mathbf{s}_t, \cdot)\right)}{Z^{\pi}(\mathbf{s}_t)}\right), \tag{3.30}$$

trong đó II biểu thị một tập hợp các chính sách tương ứng với một họ phân phối có tham số như phân phối Gauss. $Z^{\pi}(\mathbf{s}_t$ là một hàm thực hiện chuẩn hóa phân phối và không ảnh hưởng đến độ dốc theo chính sách mới. Chúng tôi tiếp tục biến đổi độ lệch KL trong 3.30 bằng cách

nhân α và bỏ qua thuật ngữ chuẩn hóa hằng số $\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[\alpha \log Z^{\pi}\left(\mathbf{s}_t\right)\right.$ Sau đó, các tham số chính sách có thể được huấn luyện bằng cách tối thiểu hóa hàm sau:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathscr{M}}\left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[\alpha \log\left(\pi_\phi\left(\mathbf{a}_t \mid \mathbf{s}_t\right)\right) - Q_\theta\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]\right] \tag{3.31}$$

Trong quá trình lặp chính sách, quá trình đánh giá chính sách mềm và cải tiến chính sách mềm luân phiên cho đến khi lặp hội tụ đến một chính sách tối ưu với entropy tối đa trong các chính sách trong Π .

3.8.4. Thiết kế Thuật toán Dựa trên SAC

Trong quá trình đánh giá chính sách và cải tiến chính sách đã được mô tả ở trên, tham số nhiệt độ được xem xét như một hằng số. Ở đây, chúng tôi sẽ thể hiện cách chọn nhiệt độ tối ưu tự động trong SAC. Bởi vì entropy có thể biến đổi không thể dự đoán được cả trong các nhiệm vụ đào tạo khác nhau và trong quá trình đào tạo, việc điều chỉnh nhiệt độ trở nên khó khăn. Để giải quyết vấn đề này, một vấn đề tối ưu hóa bị ràng buộc được xây dựng. Trong khi tối đa hóa lợi tức kỳ vọng, entropy của chính sách cần đáp ứng một ràng buộc tối thiểu, như được thể hiện dưới đây:

$$\begin{aligned} &\max \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T'-1} R\left(\mathbf{s}_t, \mathbf{a}_t\right)\right], \\ &\text{s.t. } \mathbb{E}_{\left(\mathbf{s}_t, \mathbf{a}_t\right) \sim \rho_\pi}\left[-\log\left(\pi_t\left(\mathbf{a}_t \mid \mathbf{s_t}\right)\right)\right] \geq \mathscr{H}_0, \quad \forall t, \end{aligned} \tag{3.32}$$

trong đó $\mathscr{H}_0$ là một ngưỡng entropy tối thiểu của chính sách được xác định trước. Tham số nhiệt độ có thể được học trong mỗi khe thời gian bằng cách tối thiểu hóa hàm mục tiêu sau đây:

$$J(\alpha) = \mathbb{E}_{\mathbf{a}_t \sim \pi_t}\left[-\alpha \log \pi_t\left(\mathbf{a}_t \mid \mathbf{s}_t\right) - \alpha \mathscr{H}_0\right]. \tag{3.33}$$

Ngoài ra, để giảm thiểu sự lệch tích cực trong việc cải thiện chính sách, trong thuật toán được sử dụng hai hàm giá trị hành động mềm. Như được đề xuất trong [135], chúng ta tham số hóa hai hàm giá trị hành động mềm và huấn luyện chúng độc lập. Sau đó, chúng ta sử dụng giá trị nhỏ nhất của các hàm giá trị hành động mềm để tính gradient ngẫu nhiên của $J_Q(\theta)$ và gradient chính sách của $J_\pi(\phi)$.

Ngoài ra, tôi trình bày thuật toán SAC trong Algorithm 1 như sau:

3.8.5. Phần học hàm Q của SAC

Các hàm Q được học theo cách tương tự như TD3, nhưng có một số khác biệt quan trọng. Trước hết, tương tự như TD3:

- Cả hai hàm Q được học thông qua việc tối thiểu hóa lỗi MSBE bằng cách hồi quy đến một mục tiêu chia sẻ duy nhất.
- Mục tiêu chia sẻ được tính toán bằng cách sử dụng mạng Q mục tiêu, và các mạng Q mục tiêu được thu được bằng cách lấy trung bình polyak các tham số mạng Q qua quá trình huấn luyện.

- Mục tiêu chia sẻ sử dụng kỹ thuật double-Q bị cắt.

Khác với TD3, SAC có:

- Mục tiêu cũng bao gồm một thành phần xuất phát từ việc sử dụng đạo hàm entropy của SAC.

- Các hành động của trạng thái kế tiếp sử dụng trong mục tiêu đến từ chính sách hiện tại thay vì từ một chính sách mục tiêu.

- Không có sự rõ ràng của chính sách mục tiêu. TD3 huấn luyện một chính sách xác định, và do đó nó thực hiện việc mịn hoặc bằng cách thêm nhiễu ngẫu nhiên vào các hành động của trạng thái kế tiếp. SAC huấn luyện một chính sách ngẫu nhiên, và do đó nhiễu từ tính ngẫu nhiên đó là đủ để đạt được hiệu ứng tương tự.

Trước khi đưa ra biểu thức cuối cùng của hàm Q-loss, hãy dành một chút thời gian để thảo luận về cách đóng góp từ việc chính quy hóa entropy được thể hiện. Chúng tôi sẽ bắt đầu bằng cách lấy phương trình Bellman đệ quy của Q^π có chứa chính quy entropy từ trước đó, và viết lại một chút bằng cách sử dụng định nghĩa của entropy:

$$\begin{aligned} Q^\pi(s,a) &= \mathop{\mathrm{E}}_{\substack{s'\sim P\\ a'\sim\pi}}\left[R\left(s,a,s'\right)+\gamma^\pi\left(Q^\pi\left(s',a'\right)+\alpha H\left(\pi\left(\cdot \mid s'\right)\right)\right)\right] \\ &= \mathop{\mathrm{E}}_{\substack{s'\sim P\\ a'\sim\pi}}\left[R\left(s,a,s'\right)+\gamma\left(Q^\pi\left(s',a'\right)-\alpha\log\pi\left(a' \mid s'\right)\right)\right] \end{aligned} \tag{3.34}$$

Phía bên phải là kì vọng qua trạng thái tiếp theo (được lấy từ bộ nhớ tái tạo) và hành động tiếp theo (được lấy từ chính sách hiện tại, không phải từ bộ nhớ tái tạo). Vì nó là kì vọng, chúng ta có thể xấp xỉ nó bằng các mẫu:

$$Q^\pi(s,a)\approx r+\gamma\left(Q^\pi\left(s',\tilde{a}'\right)-\alpha\log\pi\left(\tilde{a}' \mid s'\right)\right),\quad \tilde{a}'\sim\pi\left(\cdot \mid s'\right). \tag{3.35}$$

SAC thiết lập lỗi MSBE cho mỗi Q-function bằng cách sử dụng loại xấp xỉ mẫu này cho mục tiêu. Điều duy nhất vẫn chưa xác định ở đây là Q-function nào được sử dụng để tính toán sao lưu mẫu: tương tự như TD3, SAC sử dụng double-Q bị cắt, và lấy giá trị Q nhỏ nhất giữa hai bộ xấp xỉ Q.

Kết hợp tất cả lại, hàm mất mát cho các mạng Q trong SAC là:

$$L\left(\phi_i,\mathscr{D}\right)=\mathop{\mathrm{E}}_{(s,a,r,s',d)\sim\mathscr{D}}\left[\left(Q_{\phi_i}(s,a)-y\left(r,s',d\right)\right)^2\right], \tag{3.36}$$

Trong đó mục tiêu được cho bởi

$$y\left(r,s',d\right)=r+\gamma(1-d)\left(\min_{j=1,2}Q_{\phi_{\mathrm{tar},j}},\left(s',\tilde{a}'\right)-\alpha\log\pi_\theta\left(\tilde{a}' \mid s'\right)\right),\quad \tilde{a}'\sim\pi_\theta\left(\cdot \mid s'\right) \tag{3.37}$$

3.8.6. Học chính sách

Trong mỗi trạng thái, chính sách hoạt động để tối đa hóa tổng lợi ích tương lai kỳ vọng cộng với entropy tương lai kỳ vọng. Vì vậy, nó nên tối đa hóa $V^{\pi}(s)$, và chúng ta triển khai thành:

$$\begin{aligned} V^{\pi}(s) &= \underset{a\sim\pi}{\mathrm{E}}[Q^{\pi}(s,a)] + \alpha H(\pi(\cdot \mid s)) \\ &= \underset{a\sim\pi}{\mathrm{E}}[Q^{\pi}(s,a) - \alpha \log \pi(a \mid s)]. \end{aligned} \tag{3.38}$$

Cách chúng ta tối ưu hóa chính sách sử dụng kỹ thuật đánh lừa reparameterization, trong đó một mẫu từ $\pi_{\theta}(\cdot|s)$ được vẽ bằng cách tính toán một hàm xác định của trạng thái, tham số chính sách và nhiễu độc lập. Để minh họa: theo tác giả của bài báo SAC, chúng ta sử dụng chính sách Gaussian squashed, có nghĩa là các mẫu được thu được theo

$$\tilde{a}_{\theta}(s,\xi) = \tanh\left(\mu_{\theta}(s) + \sigma_{\theta}(s) \odot \xi\right), \quad \xi \sim \mathcal{N}(0,I) \tag{3.39}$$

Kỹ thuật reparameterization cho phép chúng ta viết lại kỳ vọng qua các hành động (mà chứa một điểm đau: phân phối phụ thuộc vào các tham số chính sách) thành kỳ vọng qua nhiễu (loại bỏ điểm đau: phân phối hiện không phụ thuộc vào các tham số):

$$\underset{a\sim\pi_{\theta}}{\mathrm{E}}[Q^{\pi_{\theta}}(s,a) - \alpha \log \pi_{\theta}(a \mid s)] = \underset{\xi\sim\mathcal{N}}{\mathrm{E}}[Q^{\pi_{\theta}}(s,\tilde{a}_{\theta}(s,\xi)) - \alpha \log \pi_{\theta}(\tilde{a}_{\theta}(s,\xi) \mid s)] \tag{3.40}$$

Để có được hàm mất mát của chính sách, bước cuối cùng là chúng ta cần thay thế $Q^{\pi_{\theta}}$ bằng một trong các bộ xấp xỉ hàm của chúng ta. Không giống như trong TD3, mà sử dụng Q_{ϕ_1} (chỉ là xấp xỉ Q đầu tiên), SAC sử dụng $\min_{j=1,2} Q_{\phi_j}$ (giá trị nhỏ nhất của hai xấp xỉ Q). Chính sách được tối ưu hóa theo cách sau:

$$\max_{\theta} \underset{\substack{s\sim\mathcal{D} \\ \xi\sim\mathcal{N}}}{\mathrm{E}} \left[\min_{j=1,2} Q_{\phi_j}(s,\tilde{a}_{\theta}(s,\xi)) - \alpha \log \pi_{\theta}(\tilde{a}_{\theta}(s,\xi) \mid s) \right], \tag{3.41}$$

Điều này gần giống với việc tối ưu hóa chính sách trong DDPG và TD3, ngoại trừ phần min-double-Q trick, tính ngẫu nhiên và thuật ngữ entropy.

3.9. Kết luận chương

Trong chương 3, một số khái niệm và hiểu biết cơ bản về phương pháp học máy và thuật toán SAC đã được mô tả cụ thể. Chương tiếp theo sẽ trình bày về vấn đề tối ưu, từ đó xây dựng bài toán tối ưu. Một thuật toán tối ưu dựa trên thuật toán SAC được thiết kế để phù hợp với vấn đề tối ưu khảo sát.

Thuật toán 1: Thuật toán SAC

Khởi tạo mạng chính Q hành động mềm $Q_{\theta 1}(s,a)$ và $Q_{\theta 2}(s,a)$ với trọng số $\theta 1$ và $\theta 2$.
Khởi tạo mạng mềm Q hành động mục tiêu $Q_{\theta 1}(s,a)$ và $Q_{\theta 2}(s,a)$ với trọng số $\theta_{\text{target 1}} = \theta 1$ và $\theta_{\text{target 2}} = \theta 2$.
Khởi tạo chính sách $\pi(a|s)$ với trọng số ϕ.
Khởi tạo bộ đệm tái trải nghiệm $\mathscr{M} = \emptyset$.
for vòng lặp $= 1, 2, ..., \text{ep}$ **do**
    Thu thập quan sát trạng thái ban đầu s_0
    for bước $= 1, 2, ..., \text{step}$ **do**
        if Số bước nhỏ hơn số bước khởi động **then**
            Lựa chọn hành động a_t ngẫu nhiên.
        else
            Tính toán hành động a_t.
        end if
        Tính giá trị thưởng r_t và xác định trạng thái tiếp theo s_{t+1}.
        Lưu tập dữ liệu (s_t, a_t, r_t, s_{t+1}) vào bộ nhớ M.
    end for
    if C đã đầy **then**
        Lựa chọn ngẫu nhiên b tập dữ liệu (s_t, a_t, r_t, s_{t+1}) từ bộ nhớ M.
        Cập nhật θ_i bằng cách tính gradient của hàm $J_Q(\theta_i)$ được định nghĩa trong 3.29,

$$\theta_i = \theta_i - \delta_Q \nabla_\theta \frac{1}{|\mathscr{B}|} \sum_{\mathscr{B}} J_Q(\theta_i), \forall i = 1, 2$$

.
        Cập nhật tham số chính sách ϕ bằng cách tính gradient của hàm $J_\pi(\phi)$3.31,

$$\alpha = \alpha - \delta_\alpha \nabla_\alpha \frac{1}{|\mathscr{B}|} \sum_{\mathscr{B}} J(\alpha)$$

        Cập nhật tham số nhiệt độ α bằng cách tính gradient của hàm $J(\alpha)$ 3.33,

$$\alpha = \alpha - \delta_\alpha \nabla_\alpha \frac{1}{|\mathscr{B}|} \sum_{\mathscr{B}} J(\alpha)$$

        Cập nhật tham số của hàm giá trị hành động mềm mục tiêu $\theta_{\text{target i}}$ bằng

$$\theta_i = \omega\theta_i + (1-\omega)\theta_i, \quad \forall i = 1, 2. \forall i = 1, 2.$$

    end if
end for
Trả về chính sách lựa chọn hành động của mạng Tác nhân π_a.
Lựa chọn hành động tối ưu a_t tại thời điểm t

CHƯƠNG 4: MÔ HÌNH HỆ THỐNG VÀ PHÂN TÍCH

4.1. Giới thiệu chương

Chương này tập trung giải quyết hai vấn đề: Mô tả hệ thống multi-task federated learning được khảo sát trong đồ án, từ đó tính toán và xây dựng bài toán tối ưu và thiết lập thuật toán SAC phù hợp để giải quyết bài toán.

4.2. Mô hình hệ thống và Xây dựng bài toán tối ưu

4.2.1. Mô hình hệ thống

Trong hệ thống không dây này, tồn tại một tập hợp các thiết bị U giao tiếp thông qua một trạm cơ sở (BS). Mỗi thiết bị được trang bị hai mô hình Trí tuệ Nhân tạo (AI) được thiết kế để xử lý các nhiệm vụ khác nhau, trong đó một nhiệm vụ quan trọng hơn nhiệm vụ còn lại. Các thiết bị này, được gọi là Thiết bị Trang bị Người dùng (UE), duy trì các bộ dữ liệu cục bộ riêng biệt được ký hiệu là D_n, trong đó D_n đại diện cho kích thước của mỗi bộ dữ liệu. Tổng kích thước của tất cả các bộ dữ liệu kết hợp có thể được biểu thị là $D = \sum_{n=1}^{N} D_n$. Trong ngữ cảnh của một hệ thống học có giám sát, xem xét một UE cụ thể i. Tập hợp các cặp đầu vào-đầu ra, được biểu thị dưới dạng $\{x_i, y_i\}_{i=1}^{D_n}$, tạo thành tập dữ liệu cho UE này. Trong trường hợp này, một vectơ mẫu đầu vào có d đặc trưng được ký hiệu là x_i, trong khi giá trị đầu ra được gắn nhãn tương ứng cho mẫu x_i được biểu thị bởi y_i. Các tập dữ liệu này có thể được tạo ra thông qua các tương tác của UE, chẳng hạn như thông qua việc sử dụng ứng dụng di động. Tận dụng dữ liệu thu thập từ những UE này, các mạng không dây có thể sử dụng các ứng dụng học máy khác nhau. Ví dụ bao gồm dự đoán tải trạm cơ sở để hỗ trợ cân bằng tải động hoặc dự đoán vị trí bay không ngừng của drone để tối đa hóa phạm vi phủ sóng.

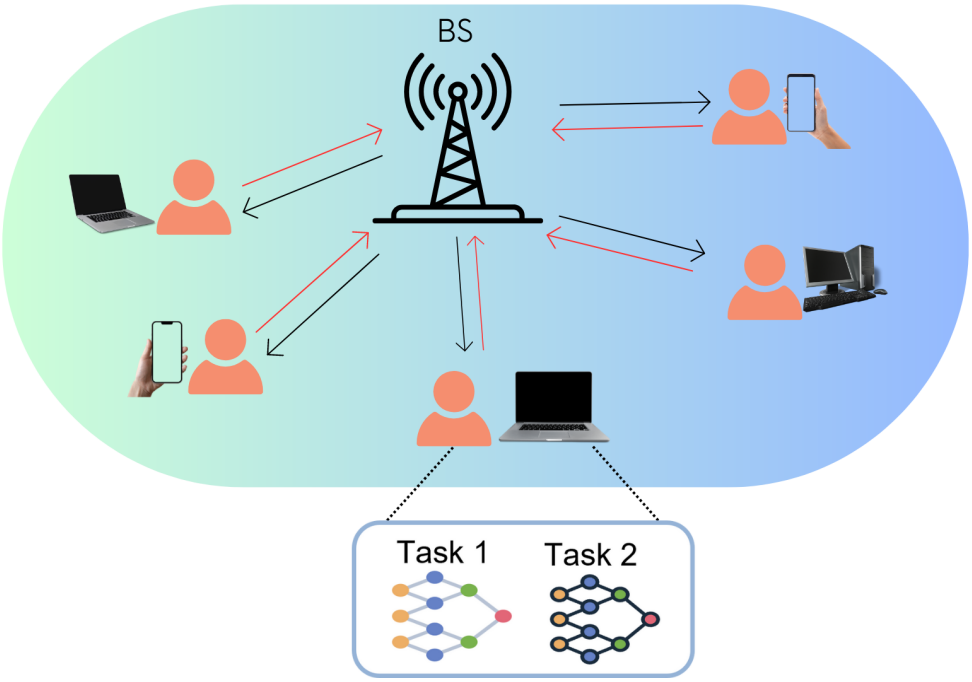
4.2.1.1. Xử lý Cục bộ

Số chu kỳ CPU cần thiết cho UE i để thực hiện một mẫu dữ liệu được biểu thị là C_i. Do đó, số chu kỳ CPU cần thiết cho UE i để thực hiện một vòng lặp cục bộ được biểu thị bởi $C_i D_i$. Khả năng tính toán của người dùng i được định nghĩa là f_i, đại diện cho số chu kỳ CPU mỗi giây mà người dùng i có thể thực hiện. Biểu thức sau xác định năng lượng tiêu thụ để thực hiện tổng cộng $C_i D_i$ chu kỳ CPU tại người dùng i:

$$E_i^{\mathrm{Cmp}} = I_i E_{i1}^{\mathrm{Cmp}} = \kappa I_i C_i D_i f_i^2 \tag{4.1}$$

Ở đây, κ đại diện cho điện dung chuyển đổi hiệu quả, phụ thuộc vào kiến trúc vi mạch. Thời gian tính toán cần thiết cho xử lý dữ liệu tại thiết bị i có thể được tính toán như sau:

$$\tau_i = \frac{I_i C_i D_i}{f_i}, \quad \forall i \in \mathscr{I} \tag{4.2}$$



Hình 4.1: Mô hình hệ thống multitask federated learning

4.2.1.2. Truyền Dẫn Không Dây

Sau khi tính toán cục bộ, mỗi người dùng áp dụng phương pháp truyền tần số nhiều truy cập (FDMA) để truyền mô hình FL cục bộ của họ đến trạm cơ sở (BS). Tốc độ truyền đạt được (nats/s) cho người dùng i có thể được định nghĩa như sau:

$$r_i = b_i \log_2 \left(1 + \frac{g_i p_i}{N_0 b_i}\right) \tag{4.3}$$

Ở đây, độ gia tăng kênh giữa người dùng i và BS được biểu thị bởi b_i, công suất phát trung bình của người dùng i được ký hiệu là p_i, băng thông được phân bổ cho người dùng i được chỉ ra bởi B_i, và mật độ phổ công suất của tạp âm Gaussian được biểu thị bởi N_0.

Trong quá trình này, người dùng i cần tải lên mô hình FL cục bộ của họ lên BS. Mỗi người dùng được yêu cầu truyền cùng một lượng dữ liệu, được biểu thị bởi ký hiệu s. Để đảm bảo rằng dữ liệu kích thước s có thể được truyền trong thời gian truyền τ_i đã cho, chúng ta cần $\tau_i r_i \geq s$. Năng lượng tiêu thụ bởi người dùng i để truyền dữ liệu kích thước s trong khoảng thời gian τ_i có thể được biểu thị như sau:

$$E_i^{\text{com}} = \tau_i p_i \left(\frac{s_i}{\tau_i}\right) \tag{4.4}$$

4.2.1.3. Truyền Thông Tin

Trong giai đoạn này, BS thu thập và tích hợp mô hình Học liên đoàn (FL) toàn cầu. BS sau đó phát sóng mô hình toàn cầu đến tất cả người dùng trong hướng xuống. Vì BS có băng thông dư dả và công suất truyền tải cao để phát sóng, thời gian hướng xuống tương đối ngắn hơn so

với thời gian truyền dữ liệu hướng lên. Để bảo vệ quyền riêng tư của người dùng, một yêu cầu cơ bản trong FL, BS không có quyền truy cập vào dữ liệu cục bộ (D_i).

Theo mô hình FL được trình bày ở trên, sự tiêu thụ năng lượng của mỗi thiết bị bao gồm hai phần: năng lượng tính toán cục bộ (E_i^{Cmp}) và năng lượng truyền không dây (E_i^{Com}). Hãy đặt I_0 cho tổng số lượt lặp. Tổng năng lượng tiêu thụ của tất cả các thiết bị tham gia trong FL có thể được biểu thị như sau:

$$E = I_0 \sum_{i=1}^{\mathscr{I}} \left(E_i^{\mathrm{Cmp}} + E_i^{\mathrm{Com}} \right) \tag{4.5}$$

Thời gian tổng cộng cần để hoàn thành việc thực thi thuật toán FL được gọi là thời gian hoàn thành. Thời gian hoàn thành của mỗi người dùng bao gồm cả thời gian tính toán cục bộ và thời gian truyền. Thời gian hoàn thành T_i của người dùng i được xác định như sau:

$$T_i = I_0 \left(\tau_i + t_i \right) = I_0 \left(\frac{I_i C_i D_i}{f_i} + t_i \right) \tag{4.6}$$

Chúng ta có điều sau nếu T là thời gian hoàn thành tối đa cho toàn bộ thuật toán FL:

$$T_i \leq T, \quad \forall i \in \mathscr{I} \tag{4.7}$$

4.3. Xây dựng bài toán tối ưu

Các vấn đề tối thiểu hóa được giải quyết bằng học liên đoàn, trong đó các mô hình học máy được huấn luyện trên các nguồn dữ liệu phi tập trung mà không chia sẻ dữ liệu gốc, đảm bảo quyền riêng tư và an ninh dữ liệu. Cập nhật mô hình hiệu quả được kích hoạt bằng cách truyền chỉ gradient thay vì toàn bộ tập dữ liệu, giảm sử dụng băng thông. Ngoài ra, việc cải thiện khái quát hóa có thể được đạt được thông qua việc huấn luyện trên các dữ liệu đa dạng. Học Liên đoàn (FL) là một phương pháp tiên tiến trong Học Máy. Thay vì thu thập dữ liệu và huấn luyện mô hình tập trung tại một vị trí duy nhất, FL xử lý dữ liệu theo cách phân tán tại các máy khách cá nhân. Sự hợp tác giữa máy chủ và các máy khách đảm bảo rằng mất mát mô hình được xác nhận tại máy chủ, và các cải tiến được cập nhật dần dần trên các máy khách phân tán, như được mô tả trong Hình 4.1.

Trong ngữ cảnh của FL, chúng ta giới thiệu một vectơ ϕ để biểu thị các tham số mô hình FL toàn cầu. Hàm mất mát được biểu thị như sau: $f(\phi, x_{il}, y_{il})$, biểu thị hiệu suất FL qua vectơ đầu vào x_{il} và vectơ đầu ra y_{il}. Hàm tổng mất mát của người dùng i được xác định bởi nhiệm vụ học cụ thể và khác nhau cho mỗi nhiệm vụ. Xem xét rằng người dùng i sở hữu một tập dữ liệu D_i, hàm mất mát cho người dùng i có thể được biểu thị như sau:

$$F_i(\phi, x_{i1}, y_{i1}, \cdots, x_{iD_i}, y_{iD_i}) = \frac{1}{D_i} \sum_{l=1}^{D_i} f(\phi, x_{il}, y_{il}) \tag{4.8}$$

Hơn nữa, hàm $f(\phi, x_{il}, y_{il})$ là hàm mất mát của người dùng i với một mẫu dữ liệu và hàm $F_i(\phi, x_{i1}, y_{i1}, \cdots, x_{iD_i}, y_{iD_i})$ là hàm tổng mất mát của người dùng i với toàn bộ tập dữ liệu cục bộ. Sau đó, mô hình học là bộ giảm thiểu của bài toán giảm thiểu hàm mất mát toàn cầu sau đây:

$$\min_{\phi} F(\phi) \triangleq \sum_{i=1}^{\mathscr{I}} \frac{D_i}{D} F_i(\phi) = \frac{1}{D} \sum_{i=1}^{I} \sum_{l=1}^{D_i} f(\phi, x_{il}, y_{il}) \tag{4.9}$$

Để giải quyết vấn đề (4.9), FEDL sử dụng một phương pháp lặp mà liên quan đến nhiều lần lặp toàn cầu, còn được gọi là vòng tròn giao tiếp, để đạt được một mức độ chính xác toàn cầu mong muốn ε. Trong mỗi vòng lặp toàn cầu, có sự tương tác giữa các thiết bị cuối (UEs) và cơ sở (BS).

Trong mỗi giai đoạn tính toán, mục tiêu cụ thể $F_i(\phi_i)$ được tối thiểu hóa một cách tích cực bởi một UE tham gia, sử dụng dữ liệu đào tạo cục bộ D_i của nó.

$$\phi_i^{(t)} = \arg\min_{\phi_i \in \mathbb{R}^d} F_i \left(\phi_i \mid \phi^{(t-1)}, \nabla J^{(t-1)} \right) \tag{4.10}$$

4.3.1. Phát biểu vấn đề

Mục tiêu của chúng ta là tối thiểu hóa cả tiêu thụ năng lượng của các thiết bị cuối (UEs) và thời gian Học Liên đoàn trong hai nhiệm vụ của mô hình Trí tuệ Nhân tạo (AI):

$$\min_{\tau, \mathbf{T}_{\text{com}}, \mathbf{T}_{\text{cmp}}} \left[(1-\alpha)E_1 + (1-\alpha)\kappa T_1 + \kappa\alpha T_2 + \alpha E_2 \right] \tag{4.11a}$$

thỏa mãn

$$0 \leq \alpha \leq 1 \tag{4.11b}$$

$$\sum_{i=1}^{\mathscr{I}} \tau_i \leq T_{\text{com}}, \tag{4.11c}$$

$$\max_i \frac{c_i D_i}{f_i} = T_{cmp}, \tag{4.11d}$$

$$f_i^{\min} \leq f_i \leq f_i^{\max}, \forall i \in \mathscr{I}, \tag{4.11e}$$

$$p_i^{\min} \leq p_i(s_i/\tau_i) \leq p_i^{\max}, \forall i \in \mathscr{I}. \tag{4.11f}$$

Tuy nhiên, việc tối thiểu hóa cả tiêu thụ năng lượng của UEs và thời gian Học Liên đoàn đang xảy ra mâu thuẫn. Ví dụ, UEs có thể tiết kiệm năng lượng bằng cách thiết lập mức tần số thấp nhất suốt thời gian, nhưng điều này chắc chắn sẽ làm tăng thời gian học. Hơn nữa, chúng ta phải tối đa hóa năng lượng và thời gian của hai nhiệm vụ của mô hình AI, trong đó một nhiệm vụ quan trọng hơn, để đảm bảo sự cân bằng trong phân bổ tài nguyên giữa hai nhiệm vụ, chúng ta thêm hệ số α cho nhiệm vụ quan trọng hơn (4.11a).

Trong khi ràng buộc (4.11c) bắt giữ khía cạnh chia sẻ thời gian của việc truyền dữ liệu lên giữa các Thiết bị Cuối (UEs). Ràng buộc này đảm bảo thời gian truyền của mỗi UE đáp ứng yêu cầu cụ thể. Trên một khía cạnh khác, ràng buộc (4.11d) xác định thời gian tính toán trong một vòng lặp cục bộ và được xác định bởi UE có khả năng xử lý chậm nhất, được gọi là UE "chặn đầu". Ràng buộc này đảm bảo thời gian tính toán của mỗi UE không vượt quá giới hạn do UE "chặn đầu" đặt ra.

Ràng buộc (4.11e) và (4.11f) đặt ra các vùng khả thi cho tần số CPU và công suất truyền của UEs, tương ứng. Các ràng buộc này đảm bảo việc tồn tại sự đa dạng giữa các UEs có các loại CPU và bộ truyền khác nhau. Bằng cách đặt các ràng buộc này, mô hình xem xét các khả năng biến đổi của các UEs trong việc tính toán và truyền.

Cuối cùng, phạm vi khả thi của độ chính xác cục bộ được hạn chế bởi ràng buộc cuối cùng. Ràng buộc này đảm bảo rằng độ chính xác cục bộ được đạt được bởi mỗi UE nằm trong khoảng xác định, từ đó đảm bảo chất lượng của các mô hình cục bộ.

4.3.2. Quá trình triển khai

Bằng cách sử dụng không gian trạng thái, không gian hành động và phần thưởng đã xác định, SAC có thể được triển khai nhanh chóng và dễ dàng. Quá trình thực hiện thử nghiệm và huấn luyện bao gồm nhiều vòng lặp, mỗi vòng bao gồm nhiều bước. Trong mỗi vòng lăp, trước tiên thuật toán SAC tính toán một hành động theo trạng thái hiện tại, từ đó lựa chọn hành động tương ứng. Sau khi có được hành động, thuật toán sẽ tính toán giá trị thưởng và tìm trạng thái tiếp theo theo hành động đã chọn. Sau đó tác nhân SAC bắt đầu học bằng cách cập nhật 2 mạng Q chính và 1 mạng mục tiêu như đã trình bày trong phần 3.8

4.3.2.1. Khởi tạo môi trường

Môi trường được khởi tạo từ các tham số: Số user N; Số lượng kênh truyền L; Băng thông B của hệ thống; Tần số trung tâm f_c;Số chu kỳ CPU cần thiết cho UE i để thực hiện một mẫu dữ liệu được biểu thị là C_i; số chu kỳ CPU cần thiết cho UE i để thực hiện một vòng lặp cục bộ được biểu thị bởi C_iD_i. Môi trường sẽ thực hiện tính toán theo các công thức đã trình bày tại phần 4.2.1

4.3.2.2. Khởi tạo mạng tế bào thần kinh

Trong quá trình này, hai mạng tế bào được khởi tạo với hai lớp ẩn, lớp ẩn đầu tiên gồm 400 nút và lớp ẩn thứ hai gồm 300 nút. Mạng Tác nhân được khởi tạo để sử dụng cho các mạng Tác nhân và mạng Tác nhân mục tiêu trong mô hình SAC. Mạng được khởi tạo với số chiều đầu vào là số chiều của một trạng thái và trả về đầu ra có số chiều là số chiều của một hành động. Mạng Đánh giá được khởi tạo để sử dụng cho mạng Đánh giá và mạng Đánh giá mục tiêu trong mô hình SAC. Mạng được khởi tạo với đầu vào gồm số chiều của trạng thái và hành động, đầu ra của mạng là một giá trị cụ thể.

4.3.2.3. Khởi tạo bộ nhớ

Nội dung của quá trình khởi tạo tương tự quá trình khởi tạo bộ nhớ của thuật toán SAC được đề cập ở mục 3.8. Bộ nhớ của hệ thống được khởi tạo với kích thước bộ nhớ rất lớn, kí hiệu là C. Nhiệm vụ của bộ nhớ là lưu trữ tất cả các thông tin từng trạng thái của hệ thống từ thời điểm bắt đầu trong toàn hệ thống. Bộ nhớ lưu trữ sử dụng cấu trúc dữ liệu hàng đợi với phương pháp lưu trữ FIFO. Vì kích thước của bộ lớn quá lớn, việc đưa tất cả dữ liệu được lưu trữ vào quá trình huấn luyện sẽ là một thách thức lớn. Quá trình "học" sẽ yêu cầu tài nguyên phần cứng cao cũng như thời gian xử lí lâu, do đó cần một bộ phận chia nhỏ tập dữ liệu được lưu thành những tập nhỏ hơn.

Bộ chia nhỏ dữ liệu huấn luyện được khởi tạo với nhiệm vụ lựa chọn các tập dữ liệu huấn luyện có kích thước nhỏ hơn một cách ngẫu nhiên từ bộ nhớ. Kích thước của tập dữ liệu huấn luyện được xác định tùy thuộc vào yêu cầu của hệ thống.

4.3.2.4. Khởi tạo mô hình thuật toán SAC

Thuật toán DDPG được khởi tạo với một số nhiệm vụ: Thiết lập hàm khám phá môi trường khảo sát; Thiết lập hàm lựa chọn hành động thông qua mạng tế bào thần kinh dựa vào trạng thái môi trường tương ứng; Thiết lập hàm lưu trữ các thông tin trạng thái vào bộ nhớ; Cập nhật trọng số mạng tế bào thần kinh theo phương pháp cập nhật mềm.

Mô hình thuật toán SAC được khởi tạo với bốn mạng tế bào thần kinh lần lượt là mạng Tác nhân, mạng Q soft mục tiêu, 2 mạng Q chính. Chi tiết chức năng và nhiệm vụ của mạng trong mô hình thuật toán đã được trình bày ở phần 3.8.

4.3.2.5. Quá trình vận hành thuật toán

Đầu tiên, tham số dữ liệu môi trường được khởi tạo, hình thành môi trường truyền tin khảo sát. Sau đó, hệ thống sẽ khởi tạo mô hình mạng NN với tham số định dạng dữ liệu đầu vào và đầu ra tương ứng của các mạng. Bộ nhớ và bộ chia nhỏ dữ liệu được hình thành với các tham số: kích thước bộ nhớ, kích thước tập dữ liệu huấn luyện. Bước tiếp theo, hệ thống sẽ khởi tạo mô hình thuật toán SAC và chuẩn bị bước vào giai đoạn Huấn luyện. Giai đoạn Huấn luyện gồm nhiều vòng lặp, mỗi vòng lặp gồm nhiều bước nhỏ hơn.

Thuật toán 1 mô tả lại thuật toán tối ưu đề xuất dựa theo thuật toán SAC dưới dạng mã giả.

Trong giai đoạn huấn luyện, thuật toán sử dụng một biến "khởi động" để phân chia quá trình huấn luyện làm hai giai đoạn Khám phá và Khai thác. Trong mỗi vòng lặp, giai đoạn Khám phá chỉ hoạt động khi số bước của mô hình nhỏ hơn giá trị "khởi động" định trước. Trong giai đoạn này, hệ thống chỉ lựa chọn ngẫu nhiên một hành động và tính toán giá trị thưởng và trạng thái tiếp theo tương ứng. Dữ liệu thông tin trạng thái, hành động được lưu vào bộ nhớ, giai đoạn này hệ thống không cập nhật trọng số mạng NN.

Sau quá trình Khám phá, hệ thống sẽ bắt đầu quá trình Khai thác, hệ thống sẽ lựa chọn hành động tương ứng với trạng thái hệ thống thông qua mạng Đánh giá. Hệ thống sẽ tính toán giá trị thưởng, trạng thái tiếp theo tương ứng và lưu vào bộ nhớ. Tiếp theo, hệ thống tính toán giá trị độ lợi và cập nhật giá trị trọng số các mạng NN. Giai đoạn này được gọi là giai đoạn Khai thác vì hệ thống sẽ sử dụng tập dữ liệu mẫu có kích thước định trước để huấn luyện và cập nhật trọng sốc các mạng thay vì chỉ lưu dữ liệu mới vào bộ nhớ của hệ thống. Tuy nhiên nếu kích thước bộ nhớ đã đầy, hệ thống sẽ chỉ còn trạng thái Khai thác, không còn giai đoạn Khám phá nữa.

Toàn bộ chu trình hoạt động của mô hình thuật toán DDPG được mô tả ở Hình 3.6

4.4. Kết luận chương

Như vậy, trong phần này của đồ án, mô hình thuật toán đề xuất dựa vào thuật toán SAC được xây dựng nhằm giải quyết bài toán tối ưu. Chương tiếp theo sẽ thực hiện quá trình khai báo tham số đầu vào, mô phỏng và phân tích kết quả tối ưu thu được của thuật toán đồng thời so sánh hiệu suất của thuật toán với một số phương pháp tối ưu khác.

CHƯƠNG 5: KẾT QUẢ MÔ PHỎNG VÀ THẢO LUẬN

5.1. Giới thiệu chương

Sau khi đã xây dựng được bài toán tối ưu, phương pháp xử lý sử dụng thuật toán SAC được lựa chọn để phù hợp với mục tiêu của bài toán từ chương 4.1, chương này sẽ bắt đầu triển khai tối ưu bài toán (4.3) với các tham số thực tế từ khảo sát môi trường. Kết quả của quá trình huấn luyện được mô phỏng trên biểu đồ.

5.2. Thông số bài toán mô phỏng

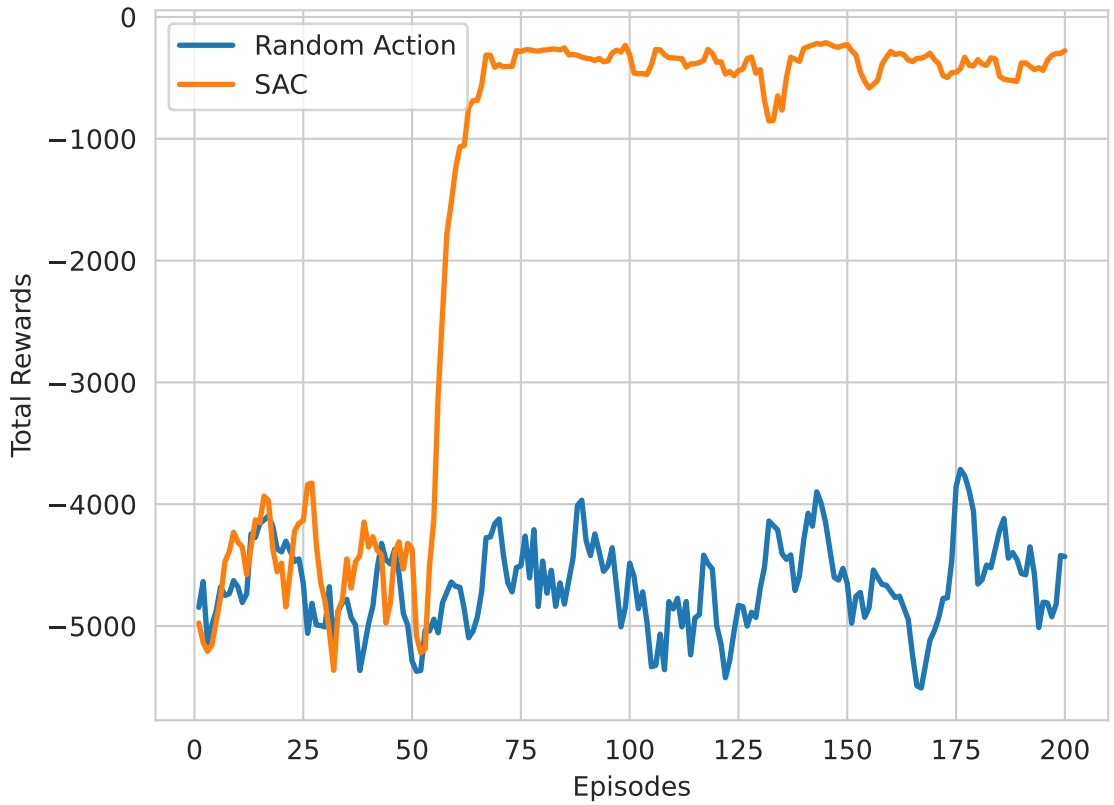
chúng tôi triển khai $K = 50$ người dùng theo phân bố đồng đều trong một khu vực vuông có kích thước $500\,\mathrm{m} \times 500\,\mathrm{m}$ với Trạm cơ sở (BS) được đặt ở trung tâm. Mô hình mất mát đường truyền là $128.1 + 37.6\log_{10} d$ (d được tính bằng km) và độ lệch chuẩn của sự suy giảm ánh sáng bóng là 8 dB. Ngoài ra, mật độ phổ điện nhiễu là $N0 = -174, \mathrm{dBm/Hz}$. Chúng tôi sử dụng tập dữ liệu thực tế từ blog mở để phản hồi trong [39]. Tập dữ liệu này với tổng số 60.000 mẫu dữ liệu xuất phát từ các bài đăng trên blog và số chiều của mỗi mẫu dữ liệu là 281. Nhiệm vụ dự đoán liên quan đến dữ liệu là dự đoán số lượng bình luận trong 24 giờ tới. Tham số C_k được phân phối đều trong khoảng $[1,3]104 chuk/mu$. Dung tích chuyển mạch hiệu quả trong tính toán cục bộ là $\kappa = 10^{-28}$.chúng tôi chọn một công suất truyền tải trung bình tối đa bằng nhau $p_{\max,1} = \ldots = p_{\max,k} = 10dB$, thông lượng tính toán tối đa bằng nhau $f_{1,\max} = \ldots = f_{\kappa,\max} = f_{\max} = 2\,\mathrm{GHz}$, một kích thước dữ liệu truyền tải $s = 28.1 kbits$ và một băng thông $B = 20MHz$. Mỗi người dùng có $D_k = 500$ mẫu dữ liệu, được chọn ngẫu nhiên từ tập dữ liệu với xác suất bằng nhau. Tất cả các kết quả thống kê được tính trung bình qua 1000 lần chạy độc lập.

5.3. Phân tích kết quả mô phỏng

Với N = 50, kết quả hoạt động của ba phương pháp tối ưu: phương pháp ứng dụng thuật toán SAC - phương pháp lựa chọn Ngẫu nhiên, qua các vòng lặp được biểu diễn qua đồ thị Hình 5.1. Dễ dàng nhận thấy kết quả của thuật toán tối ưu theo cơ sở thuật toán SAC cho kết quả cuối cùng tốt hơn so với phương pháp còn lại.

Tại những vòng lặp đầu tiên, trước vòng lặp thứ 20, thuật toán được đề xuất có kết quả kém hơn phương pháp còn lại, với tổng . Tại một số vòng lặp, kết quả của thuật toán SAC có giá trị bằng -4000 do hành động a_t đã vi phạm các điều kiện ràng buộc. Trong khi đó, thuật toán ngẫu nhiên cũng tương tự. Điều này xảy ra do hệ thống học máy vẫn đang trong quá trình tìm kiếm chính sách lựa chọn tối ưu π_a, các tập dữ liệu dược tính toán và cập nhật về hệ thống.

Từ vòng lặp 20 cho đến vòng lặp 60, hiệu quả của thuật toán được đề xuất tăng cao đáng kể, kết quả tổng reward của hệ thống là khoảng -1000. Hiện tượng này xảy ra do lúc này hệ



Hình 5.1: Kết quả

thống đã được huấn luyện đủ, và dần định hình được chính sách hành động tối ưu hơn. Trong khi đó, kết quả của phương pháp còn lại không đươc ổn đinh.

Thuật toán tối ưu được đề xuất bắt đầu hội tụ và ổn định từ vòng lặp 80 trở đi. Giá trị hội tụ xấp xỉ -100.

Tại nhiều vòng lặp, kết quả của thuật toán lựa chọn Ngẫu nhiên hành động có giá trị bằng -5000 do hành động vi phạm các điều kiện ràng buộc.

5.4. Tổng kết

Như vậy phương pháp được đề xuất nhằm tối thiểu giá trị năng lượng và thời gian truyền đi trong hệ thống học liên kết đa tác vụ sử dụng thuật toán SAC đem lại hiệu quả tốt hơn, cùng độ ổn định cao hơn so với phương pháp ngẫu nhiên. Ở những bước đầu, mô hình thuật toán đạt hiệu quả thấp và thiếu độ ổn định, tuy nhiên càng về những vòng lặp cuối, kết quả càng được cải thiện và duy trì ở giá trị hội tụ

5.5. Kết luận chương

Chương đã cung cấp những tham số mô phỏng của môi trường và tham số đầu vào của phương pháp đề xuất. Kết quả mô phỏng và tính toán của phương pháp tối ưu năng lượng và thời gian dựa theo thuật toán SAC đã được trình bày và so sánh với hiệu quả hoạt động của phương pháp ngãu nhiên.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Trong đề tài này, một phương pháp sử dụng mô hình thuật toán SAC nhằm tối ưu thời gian và năng lượng trong bài toán multi task federated learning được đề xuất, thiết kế, mô phỏng và phân tích kết quả thực nghiệm. Kết quả của phương pháp được so sánh với phương pháp phân bổ tài nguyên là phương pháp Ngẫu nhiên. Sự so sánh cho thấy mức độ hiệu quả, tối ưu hơn của thuật toán được đề xuất. Như vậy, phương pháp phân bổ tài nguyên mạng được đề xuất rất hiệu quả và thích hợp với bài toán tối ưu thời gian và năng lượng trong hệ thống multi task federated learning.

Với kết quả thu được, thuật toán SAC có thể được ứng dụng trong nhiều bài toán tối ưu khác trong hệ thống multi task federated như Tối ưu năng lượng tiêu thụ của hệ thống, hay có thể ứng dụng cho các mạng khác. Ngoài ra, mô hình thuật toán SAC rất thích hợp để xử lý các bài toán có liên tục, trong nhiều lĩnh vực như Điều khiển robot, Xe tự hành, Phân bổ tài nguyên trong hệ thống UAV.

Tài liệu tham khảo

[1] A Anonymous. “Consumer Data Privacy in a Networked World: A Framework for Protecting Privacy and Promoting Innovation in the Global Digital Economy”. In: *J. Priv. Confidentiality* 4 (2013). URL: https://api.semanticscholar.org/CorpusID:169956848.

[2] Brendan McMahan et al. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[3] A AbhishekV et al. “Federated Learning: Collaborative Machine Learning without Centralized Training Data”. In: *international journal of engineering technology and management sciences* (2022). URL: https://api.semanticscholar.org/CorpusID:251659795.

[4] H. B. McMahan et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *International Conference on Artificial Intelligence and Statistics*. 2016. URL: https://api.semanticscholar.org/CorpusID:14955348.

[5] Dinh C. Nguyen et al. “Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges”. In: *IEEE Internet of Things Journal* 8 (2021), pp. 12806–12825.

[6] Anran Du, Yicheng Shen, and Lewis Tseng. “CarML: distributed machine learning in vehicular clouds”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020).

[7] Wiseborn Manfe Danquah and Deniz Turgay Altilar. “Data partitioning and scheduling schemes for federated platoon-based vehicular cloud”. In: *Veh. Commun.* 38 (2022), p. 100529.

[8] Wiseborn Manfe Danquah and Deniz Turgay Altilar. “UniDRM: Unified Data and Resource Management for Federated Vehicular Cloud Computing”. In: *IEEE Access* 9 (2021), pp. 157052–157067.

[9] Huanhuan Zhang et al. “OnRL: improving mobile video telephony via online reinforcement learning”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020).

[10] Xin Liu et al. “Federated Remote Physiological Measurement with Imperfect Data”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2022), pp. 2154–2163.

[11] Woonghee Lee. “Federated Reinforcement Learning-Based UAV Swarm System for Aerial Remote Sensing”. In: *Wireless Communications and Mobile Computing* (2022).

[12] Yiqiang Chen et al. “FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare”. In: *IEEE Intelligent Systems* 35 (2019), pp. 83–93.

[13] Xiaohang Xu et al. “Privacy-Preserving Federated Depression Detection From Multisource Mobile Health Data”. In: *IEEE Transactions on Industrial Informatics* 18 (2022), pp. 4788–4797.

[14] Akhil Vaid et al. “Federated Learning of Electronic Health Records to Improve Mortality Prediction in Hospitalized Patients With COVID-19: Machine Learning Approach”. In: *JMIR Medical Informatics* 9 (2021).

[15] Chaoqun You et al. “Semi-Synchronous Personalized Federated Learning Over Mobile Edge Networks”. In: *IEEE Transactions on Wireless Communications* 22 (2022), pp. 2262–2277.

[16] Long Luo et al. “Joint Client Selection and Resource Allocation for Federated Learning in Mobile Edge Networks”. In: *2022 IEEE Wireless Communications and Networking Conference (WCNC)* (2022), pp. 1218–1223.

[17] Bing Luo et al. “Cost-Effective Federated Learning in Mobile Edge Networks”. In: *IEEE Journal on Selected Areas in Communications* 39 (2021), pp. 3606–3621.

[18] Wei Yang Bryan Lim et al. “Federated Learning in Mobile Edge Networks: A Comprehensive Survey”. In: *IEEE Communications Surveys & Tutorials* 22 (2019), pp. 2031–2063.

[19] Quoc-Viet Pham et al. “Coalitional Games for Computation Offloading in NOMA-Enabled Multi-Access Edge Computing”. In: *IEEE Transactions on Vehicular Technology* 69 (2020), pp. 1982–1993.

[20] Thanh Tung Vu et al. “Cell-Free Massive MIMO for Wireless Federated Learning”. In: *IEEE Transactions on Wireless Communications* 19 (2019), pp. 6377–6392.

[21] Yuxuan Sun, Sheng Zhou, and Deniz Gündüz. “Energy-Aware Analog Aggregation for Federated Learning with Redundant Data”. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)* (2019), pp. 1–7.

[22] Amirhossein Reisizadeh et al. “FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization”. In: *ArXiv* abs/1909.13014 (2019).

[23] Yunlong Lu et al. “Communication-Efficient Federated Learning and Permissioned Blockchain for Digital Twin Edge Networks”. In: *IEEE Internet of Things Journal* 8 (2021), pp. 2276–2288.

[24] Yunlong Lu et al. “Communication-Efficient Federated Learning for Digital Twin Edge Networks in Industrial IoT”. In: *IEEE Transactions on Industrial Informatics* 17 (2021), pp. 5709–5718.

[25] Nguyen H. Tran et al. “Federated Learning over Wireless Networks: Optimization Model Design and Analysis”. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications* (2019), pp. 1387–1395.

[26] Jakub Konecný et al. “Federated Learning: Strategies for Improving Communication Efficiency”. In: *ArXiv* abs/1610.05492 (2016).

[27] Sai Praneeth Karimireddy et al. “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning”. In: *International Conference on Machine Learning*. 2019.

[28] Flavio Bonomi et al. “Fog computing and its role in the internet of things”. In: *MCC '12*. 2012. URL: https://api.semanticscholar.org/CorpusID:207196503.

[29] Amol Deshpande et al. “Model-based approximate querying in sensor networks”. In: *The VLDB Journal* 14 (2005), pp. 417–443. URL: https://api.semanticscholar.org/CorpusID:15732931.

[30] Pedro García López et al. “Edge-centric Computing: Vision and Challenges”. In: *Comput. Commun. Rev.* 45 (2015), pp. 37–42. URL: https://api.semanticscholar.org/CorpusID:207232279.

[31] Mohammad Rastegari et al. “XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks”. In: *European Conference on Computer Vision*. 2016. URL: https://api.semanticscholar.org/CorpusID:14925907.

[32] Tsvi Kuflik, Judy Kay, and Bob Kummerfeld. “Challenges and Solutions of Ubiquitous User Modeling”. In: *Ubiquitous Display Environments*. 2012. URL: https://api.semanticscholar.org/CorpusID:16737899.

[33] Alexander J. Ratner et al. “MLSys: The New Frontier of Machine Learning Systems”. In: *arXiv: Learning* (2019). URL: https://api.semanticscholar.org/CorpusID:212881896.

[34] Keith Bonawitz et al. “Towards Federated Learning at Scale: System Design”. In: *ArXiv* abs/1902.01046 (2019). URL: https://api.semanticscholar.org/CorpusID:59599820.

[35] Muhammad Ammad-ud-din et al. “Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System”. In: *ArXiv* abs/1901.09888 (2019). URL: https://api.semanticscholar.org/CorpusID:59336224.

[36] Andrew Hard et al. “Federated Learning for Mobile Keyboard Prediction”. In: *ArXiv* abs/1811.03604 (2018). URL: https://api.semanticscholar.org/CorpusID:53207681.

[37] D. Anguita et al. “A Public Domain Dataset for Human Activity Recognition using Smartphones”. In: *The European Symposium on Artificial Neural Networks*. 2013. URL: https://api.semanticscholar.org/CorpusID:6975432.

[38] Li Huang et al. “LoAdaBoost: Loss-Based AdaBoost Federated Machine Learning on medical Data”. In: *ArXiv* abs/1811.12629 (2018). URL: https://api.semanticscholar.org/CorpusID:54195170.

[39] Xue-wen Chen and Xiaotong Lin. “Big Data Deep Learning: Challenges and Perspectives”. In: *IEEE Access* 2 (2014), pp. 514–525. URL: https://api.semanticscholar.org/CorpusID:10158224.

[40] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural networks : the official journal of the International Neural Network Society* 61 (2014), pp. 85–117. URL: https://api.semanticscholar.org/CorpusID:11715509.

[41] Xiaojin Zhu. “Semi-Supervised Learning Literature Survey”. In: 2005. URL: https://api.semanticscholar.org/CorpusID:2731141.

[42] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *CoRR* abs/1511.06434 (2015). URL: https://api.semanticscholar.org/CorpusID:11758569.

[43] Volodymyr Mnih et al. “Playing Atari with Deep Reinforcement Learning”. In: *ArXiv* abs/1312.5602 (2013). URL: https://api.semanticscholar.org/CorpusID:15238391.

[44] Hervé Bourlard and Yves Kamp. “Auto-association by multilayer perceptrons and singular value decomposition”. In: *Biological Cybernetics* 59 (1988), pp. 291–294. URL: https://api.semanticscholar.org/CorpusID:206775335.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (2012), pp. 84–90. URL: https://api.semanticscholar.org/CorpusID:195908774.

[46] Tomas Mikolov et al. “Recurrent neural network based language model”. In: *Interspeech*. 2010. URL: https://api.semanticscholar.org/CorpusID:17048224.

[47] H. B. McMahan et al. “Federated Learning of Deep Networks using Model Averaging”. In: *ArXiv* abs/1602.05629 (2016). URL: https://api.semanticscholar.org/CorpusID:16861557.

[48] Junxian Huang et al. “An in-depth study of LTE: effect of network protocol and application behavior on performance”. In: *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013). URL: https://api.semanticscholar.org/CorpusID:5408357.

[49] C.H. van Berkel. “Multi-core for mobile phones”. In: *2009 Design, Automation & Test in Europe Conference & Exhibition* (2009), pp. 1260–1265. URL: https://api.semanticscholar.org/CorpusID:17116263.

[50] Virginia Smith et al. “Federated Multi-Task Learning”. In: *ArXiv* abs/1705.10467 (2017). URL: https://api.semanticscholar.org/CorpusID:3586416.

[51] Jeffrey Li et al. “Differentially Private Meta-Learning”. In: *ArXiv* abs/1909.05830 (2019). URL: https://api.semanticscholar.org/CorpusID:202566017.

[52] Nicholas Carlini et al. “The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets”. In: *ArXiv* abs/1802.08232 (2018). URL: https://api.semanticscholar.org/CorpusID:3459837.

[53] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. “Privacy Aware Learning”. In: *JACM*. 2012. URL: https://api.semanticscholar.org/CorpusID:2692439.

[54] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Found. Trends Theor. Comput. Sci.* 9 (2014), pp. 211–407. URL: https://api.semanticscholar.org/CorpusID:207178262.

[55] H. B. McMahan et al. “Learning Differentially Private Recurrent Language Models”. In: *International Conference on Learning Representations*. 2017. URL: https://api.semanticscholar.org/CorpusID:3461939.

[56] Shai Shalev-Shwartz and Tong Zhang. “Accelerated Mini-Batch Stochastic Dual Coordinate Ascent”. In: *NIPS*. 2013. URL: https://api.semanticscholar.org/CorpusID:7316764.

[57] Ohad Shamir and Nathan Srebro. “Distributed stochastic optimization and learning”. In: *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2014), pp. 850–857. URL: https://api.semanticscholar.org/CorpusID:7248464.

[58] Sebastian U. Stich. “Local SGD Converges Fast and Communicates Little”. In: *ArXiv* abs/1805.09767 (2018). URL: https://api.semanticscholar.org/CorpusID:43964415.

[59] Tianbao Yang. “Trading Computation for Communication: Distributed Stochastic Dual Coordinate Ascent”. In: *NIPS*. 2013. URL: https://api.semanticscholar.org/CorpusID:6699442.

[60] Virginia Smith et al. “CoCoA: A General Framework for Communication-Efficient Distributed Optimization”. In: *ArXiv* abs/1611.02189 (2016). URL: https://api.semanticscholar.org/CorpusID:7716538.

[61] Sashank J. Reddi et al. “AIDE: Fast and Communication Efficient Distributed Optimization”. In: *ArXiv* abs/1608.06879 (2016). URL: https://api.semanticscholar.org/CorpusID:10575496.

[62] Stephen P. Boyd et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Found. Trends Mach. Learn.* 3 (2011), pp. 1–122. URL: https://api.semanticscholar.org/CorpusID:51789432.

[63] Anit Kumar Sahu et al. “Federated Optimization in Heterogeneous Networks”. In: *arXiv: Learning* (2018). URL: https://api.semanticscholar.org/CorpusID:59316566.

[64] Hongyi Wang et al. “ATOMO: Communication-efficient Learning via Atomic Sparsification”. In: *Neural Information Processing Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:47020104.

[65] Hantian Zhang et al. “ZipML: Training Linear Models with End-to-End Low Precision, and a Little Bit of Deep Learning”. In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:344319.

[66] Frank Seide et al. “1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs”. In: *Interspeech*. 2014. URL: https://api.semanticscholar.org/CorpusID:2189412.

[67] Sebastian Caldas et al. “Expanding the Reach of Federated Learning by Reducing Client Resource Requirements”. In: *ArXiv* abs/1812.07210 (2018). URL: https://api.semanticscholar.org/CorpusID:56169829.

[68] Felix Sattler et al. “Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31 (2019), pp. 3400–3413. URL: https://api.semanticscholar.org/CorpusID:71147030.

[69] Hanlin Tang et al. “Communication Compression for Decentralized Training”. In: *Neural Information Processing Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:52891696.

[70] Lie He, An Bian, and Martin Jaggi. “COLA: Decentralized Linear Learning”. In: *Neural Information Processing Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:53111830.

[71] Anusha Lalitha et al. “Decentralized Bayesian Learning over Graphs”. In: *ArXiv* abs/1905.10466 (2019). URL: https://api.semanticscholar.org/CorpusID:166227792.

[72] Tao Lin, Sebastian U. Stich, and Martin Jaggi. “Don’t Use Large Mini-Batches, Use Local SGD”. In: *ArXiv* abs/1808.07217 (2018). URL: https://api.semanticscholar.org/CorpusID:52071640.

[73] Wei Dai et al. “High-Performance Distributed ML at Scale through Parameter Server Consistency Models”. In: *ArXiv* abs/1410.8043 (2014). URL: https://api.semanticscholar.org/CorpusID:6898610.

[74] Qirong Ho et al. “More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server”. In: *Advances in neural information processing systems* 2013 (2013), pp. 1223–1231. URL: https://api.semanticscholar.org/CorpusID:2221833.

[75] Takayuki Nishio and Ryo Yonetani. “Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* (2018), pp. 1–7. URL: https://api.semanticscholar.org/CorpusID:5062760.

[76] Jiawen Kang et al. “Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach”. In: *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)* (2019), pp. 1–5. URL: https://api.semanticscholar.org/CorpusID:159041730.

[77] Miguel Castro. “Practical Byzantine fault tolerance”. In: *USENIX Symposium on Operating Systems Design and Implementation*. 1999. URL: https://api.semanticscholar.org/CorpusID:221599614.

[78] Yang Liu et al. “Data Center Networks: Topologies, Architectures and Fault-Tolerance Characteristics”. In: 2013. URL: https://api.semanticscholar.org/CorpusID:113887524.

[79] Aurick Qiao et al. “Fault Tolerance in Iterative-Convergent Machine Learning”. In: *ArXiv* abs/1810.07354 (2018). URL: https://api.semanticscholar.org/CorpusID:52985688.

[80] Hao Yu, Sen Yang, and Shenghuo Zhu. “Parallel Restarted SGD for Non-Convex Optimization with Faster Convergence and Less Communication”. In: *ArXiv* abs/1807.06629 (2018). URL: https://api.semanticscholar.org/CorpusID:125100665.

[81] Shiqiang Wang et al. “Adaptive Federated Learning in Resource Constrained Edge Computing Systems”. In: *IEEE Journal on Selected Areas in Communications* 37 (2018), pp. 1205–1221. URL: https://api.semanticscholar.org/CorpusID:51921962.

[82] Peng Jiang and Gagan Agrawal. “A Linear Speedup Analysis of Distributed Deep Learning with Sparse and Quantized Communication”. In: *Neural Information Processing Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:54014664.

[83] Zachary B. Charles and Dimitris Papailiopoulos. “Gradient Coding Using the Stochastic Block Model”. In: *2018 IEEE International Symposium on Information Theory (ISIT)* (2018), pp. 1998–2002. URL: https://api.semanticscholar.org/CorpusID:52017106.

[84] Zachary B. Charles, Dimitris Papailiopoulos, and Jordan S. Ellenberg. “Approximate Gradient Coding via Sparse Random Graphs”. In: *ArXiv* abs/1711.06771 (2017). URL: https://api.semanticscholar.org/CorpusID:20707789.

[85] Rashish Tandon et al. “Gradient Coding: Avoiding Stragglers in Distributed Learning”. In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:33632433.

[86] Susan A. J. Birch. “Learning to Learn”. In: *The Journal of practical nursing* 29 11 (2017), pp. 25–6, 38. URL: https://api.semanticscholar.org/CorpusID:11974053.

[87] Theodoros Evgeniou and Massimiliano Pontil. “Regularized multi–task learning”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004). URL: https://api.semanticscholar.org/CorpusID:719551.

[88] Yue Zhao et al. “Federated Learning with Non-IID Data”. In: *ArXiv* abs/1806.00582 (2018). URL: https://api.semanticscholar.org/CorpusID:46936175.

[89] Fei Chen et al. “Federated Meta-Learning for Recommendation”. In: *ArXiv* abs/1802.07876 (2018). URL: https://api.semanticscholar.org/CorpusID:3451040.

[90] Luca Corinzia and Joachim M. Buhmann. “Variational Federated Multi-Task Learning”. In: *ArXiv* abs/1906.06268 (2019). URL: https://api.semanticscholar.org/CorpusID:189898379.

[91] Hubert Eichner et al. “Semi-Cyclic Stochastic Gradient Descent”. In: *ArXiv* abs/1904.10120 (2019). URL: https://api.semanticscholar.org/CorpusID:128361946.

[92] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. “Adaptive Gradient-Based Meta-Learning Methods”. In: *Neural Information Processing Systems*. 2019. URL: https://api.semanticscholar.org/CorpusID:174802574.

[93] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. “Agnostic Federated Learning”. In: *ArXiv* abs/1902.00146 (2019). URL: https://api.semanticscholar.org/CorpusID:59553531.

[94] Tian Li, Maziar Sanjabi, and Virginia Smith. “Fair Resource Allocation in Federated Learning”. In: *ArXiv* abs/1905.10497 (2019). URL: https://api.semanticscholar.org/CorpusID:166227978.

[95] Ohad Shamir, Nathan Srebro, and Tong Zhang. “Communication-Efficient Distributed Optimization using an Approximate Newton-type Method”. In: *ArXiv* abs/1312.7853 (2013). URL: https://api.semanticscholar.org/CorpusID:16103184.

[96] Jianyu Wang and Gauri Joshi. “Adaptive Communication Strategies to Achieve the Best Error-Runtime Trade-off in Local-Update SGD”. In: *ArXiv* abs/1810.08313 (2018). URL: https://api.semanticscholar.org/CorpusID:53043345.

[97] Hao Yu, Rong Jin, and Sen Yang. “On the Linear Speedup Analysis of Communication Efficient Momentum SGD for Distributed Non-Convex Optimization”. In: *International Conference on Machine Learning*. 2019. URL: https://api.semanticscholar.org/CorpusID:150373664.

[98] Eunjeong Jeong et al. “Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data”. In: *ArXiv* abs/1811.11479 (2018). URL: https://api.semanticscholar.org/CorpusID:53820846.

[99] Abhishek Bhowmick et al. “Protection Against Reconstruction and Its Applications in Private Federated Learning”. In: *ArXiv* abs/1812.00984 (2018). URL: https://api.semanticscholar.org/CorpusID:54440037.

[100] Luca Melis et al. “Exploiting Unintended Feature Leakage in Collaborative Learning”. In: *2019 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 691–706. URL: https://api.semanticscholar.org/CorpusID:53099247.

[101] Yehuda Lindell and Benny Pinkas. “Privacy Preserving Data Mining”. In: *Journal of Cryptology* 15 (2000), pp. 177–206. URL: https://api.semanticscholar.org/CorpusID:657888.

[102] Vitaly Feldman et al. “Privacy Amplification by Iteration”. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (2018), pp. 521–532. URL: https://api.semanticscholar.org/CorpusID:52055752.

[103] Cynthia Dwork. “A firm foundation for private data analysis”. In: *Communications of the ACM* 54 (2011), pp. 86–95. URL: https://api.semanticscholar.org/CorpusID:14270685.

[104] Cynthia Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography Conference*. 2006. URL: https://api.semanticscholar.org/CorpusID:2468323.

[105] Martín Abadi et al. “Deep Learning with Differential Privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016). URL: https://api.semanticscholar.org/CorpusID:207241585.

[106] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. “Differentially Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds”. In: *arXiv: Learning* (2014). URL: https://api.semanticscholar.org/CorpusID:206656426.

[107] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. “Differentially Private Empirical Risk Minimization”. In: *Journal of machine learning research : JMLR* 12 (2009), pp. 1069–1109. URL: https://api.semanticscholar.org/CorpusID:1578541.

[108] Nicolas Papernot et al. “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data”. In: *ArXiv* abs/1610.05755 (2016). URL: https://api.semanticscholar.org/CorpusID:8696462.

[109] Nicolas Papernot et al. “Scalable Private Learning with PATE”. In: *ArXiv* abs/1802.08908 (2018). URL: https://api.semanticscholar.org/CorpusID:3544583.

[110] Naman Agarwal et al. “cpSGD: Communication-efficient and differentially-private distributed SGD”. In: *Neural Information Processing Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:44113205.

[111] Khaled El Emam and Fida Kamal Dankar. “Research Paper: Protecting Privacy Using k-Anonymity”. In: *Journal of the American Medical Informatics Association : JAMIA* 15 5 (2008), pp. 627–37. URL: https://api.semanticscholar.org/CorpusID:9685032.

[112] Mehmet Ercan Nergiz and Chris Clifton. “δ-Presence without Complete World Knowledge”. In: *IEEE Transactions on Knowledge and Data Engineering* 22 (2010), pp. 868–883. URL: https://api.semanticscholar.org/CorpusID:17191894.

[113] Praneeth Vepakomma et al. “R EDUCING LEAKAGE IN DISTRIBUTED DEEP LEARNING FOR SENSITIVE HEALTH DATA”. In: 2019. URL: https://api.semanticscholar.org/CorpusID:201635379.

[114] Isabel Wagner and David Eckhoff. “Technical Privacy Metrics”. In: *ACM Computing Surveys (CSUR)* 51 (2015), pp. 1–38. URL: https://api.semanticscholar.org/CorpusID:1083483.

[115] Om Thakkar, Galen Andrew, and H. B. McMahan. “Differentially Private Learning with Adaptive Clipping”. In: *Neural Information Processing Systems*. 2019. URL: https://api.semanticscholar.org/CorpusID:150373827.

[116] Keith Bonawitz et al. “Practical Secure Aggregation for Privacy-Preserving Machine Learning”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017). URL: https://api.semanticscholar.org/CorpusID:3833774.

[117] Robin C. Geyer, Tassilo Klein, and Moin Nabi. “Differentially Private Federated Learning: A Client Level Perspective”. In: *ArXiv* abs/1712.07557 (2017). URL: https://api.semanticscholar.org/CorpusID:3630366.

[118] Y. Yao, Lorenzo Rosasco, and Andrea Caponnetto. “On Early Stopping in Gradient Descent Learning”. In: *Constructive Approximation* 26 (2007), pp. 289–315. URL: https://api.semanticscholar.org/CorpusID:8323954.

[119] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. “Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates”. In: *J. Mach. Learn. Res.* 16 (2013), pp. 3299–3340. URL: https://api.semanticscholar.org/CorpusID:14832724.

[120] Lester W. Mackey, Ameet Talwalkar, and Michael I. Jordan. “Divide-and-Conquer Matrix Factorization”. In: *ArXiv* abs/1107.0789 (2011). URL: https://api.semanticscholar.org/CorpusID:8084514.

[121] Neel Guha and Virginia Smith. “Model Aggregation via Good-Enough Model Spaces”. In: *arXiv: Learning* (2018). URL: https://api.semanticscholar.org/CorpusID:195570265.

[122] Neel Guha, Ameet Talwalkar, and Virginia Smith. “One-Shot Federated Learning”. In: *ArXiv* abs/1902.11175 (2019). URL: https://api.semanticscholar.org/CorpusID:67856019.

[123] Tolga Bolukbasi et al. “Adaptive Neural Networks for Efficient Inference”. In: *International Conference on Machine Learning*. 2017. URL: https://api.semanticscholar.org/CorpusID:6750751.

[124] Mark W. Schmidt and Nicolas Le Roux. “Fast Convergence of Stochastic Gradient Descent under a Strong Growth Condition”. In: *arXiv: Optimization and Control* (2013). URL: https://api.semanticscholar.org/CorpusID:11535680.

[125] Sharan Vaswani, Francis R. Bach, and Mark W. Schmidt. “Fast and Faster Convergence of SGD for Over-Parameterized Models and an Accelerated Perceptron”. In: *ArXiv* abs/1810.07288 (2018). URL: https://api.semanticscholar.org/CorpusID:52988335.

[126] Sebastian Caldas et al. “LEAF: A Benchmark for Federated Settings”. In: *ArXiv* abs/1812.01097 (2018). URL: https://api.semanticscholar.org/CorpusID:53701546.

[127] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[128] Ahmad EL Sallab et al. “Deep reinforcement learning framework for autonomous driving”. In: *Electronic Imaging* 2017.19 (2017), pp. 70–76.

[129] B Ravi Kiran et al. “Deep reinforcement learning for autonomous driving: A survey”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021).

[130] Peizhi Zhang et al. “Reinforcement learning-based end-to-end parking for automatic parking system”. In: *Sensors* 19.18 (2019), p. 3996.

[131] Hai Nguyen and Hung La. “Review of deep reinforcement learning for robot manipulation”. In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2019, pp. 590–595.

[132] Romain Paulus, Caiming Xiong, and Richard Socher. “A deep reinforced model for abstractive summarization”. In: *arXiv preprint arXiv:1705.04304* (2017).

[133] Vijay Konda and John Tsitsiklis. “Actor-Critic Algorithms”. In: *Advances in neural information processing systems* 12 (1999).

[134] Wei Wang et al. “Safe Off-Policy Deep Reinforcement Learning Algorithm for Volt-VAR Control in Power Distribution Systems”. In: *IEEE Transactions on Smart Grid* 11 (2020), pp. 3008–3018. URL: https://api.semanticscholar.org/CorpusID:210172573.

[135] Scott Fujimoto, Herke van Hoof, and David Meger. “Addressing Function Approximation Error in Actor-Critic Methods”. In: *International Conference on Machine Learning*. 2018. URL: https://api.semanticscholar.org/CorpusID:3544558.

[136] George Trigeorgis et al. “Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), pp. 5200–5204. URL: https://api.semanticscholar.org/CorpusID:206742471.

[137] Zhaohui Yang et al. “Energy Efficient Federated Learning Over Wireless Communication Networks”. In: *IEEE Transactions on Wireless Communications* 20 (2019), pp. 1935–1949.

[138] Walid Saad, Mehdi Bennis, and Mingzhe Chen. “A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems”. In: *IEEE Network* 34 (2019), pp. 134–142.

[139] Mingzhe Chen et al. “A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks”. In: *IEEE Transactions on Wireless Communications* 20 (2019), pp. 269–283.

[140] Dinh Thai Hoang et al. “A survey of mobile cloud computing: architecture, applications, and approaches”. In: *Wirel. Commun. Mob. Comput.* 13 (2013), pp. 1587–1611.

[141] Chenxin Ma et al. “Distributed optimization with arbitrary local solvers”. In: *Optimization Methods and Software* 32 (2015), pp. 813–848.

[142] Guangxu Zhu, Yong Wang, and Kaibin Huang. “Low-Latency Broadband Analog Aggregation for Federated Edge Learning”. In: *ArXiv* abs/1812.11494 (2018).

[143] Kevin Hsieh et al. “The Non-IID Data Quagmire of Decentralized Machine Learning”. In: *ArXiv* abs/1910.00189 (2019).

[144] Anastasia Koloskova, Tao Lin, and Sebastian U. Stich. “An Improved Analysis of Gradient Tracking for Decentralized Machine Learning”. In: *Neural Information Processing Systems*. 2022.

[145] Minh-Duong Nguyen et al. “HCFL: A High Compression Approach for Communication-Efficient Federated Learning in Very Large Scale IoT Networks”. In: *ArXiv* abs/2204.06760 (2022).

[146] Fan Meng et al. “Power Allocation in Multi-user Cellular Networks: Deep Reinforcement Learning Approaches”. In: *IEEE Transactions on Wireless Communications* 19.10 (2020), pp. 6255–6267.

[147] Scott M. Diamond and Marion G. Ceruti. “Application of Wireless Sensor Network to Military Information Integration”. In: *2007 5th IEEE International Conference on Industrial Informatics*. Vol. 1. 2007, pp. 317–322. DOI: 10.1109/INDIN.2007.4384776.

[148] Lalatendu Muduli, Devi Prasad Mishra, and Prasanta K Jana. “Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review”. In: *Journal of Network and Computer Applications* 106 (2018), pp. 48–67.

[149] Th Arampatzis, John Lygeros, and Stamatis Manesis. “A survey of applications of wireless sensors and wireless sensor networks”. In: *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.* IEEE. 2005, pp. 719–724.

[150] Jianzhong Li et al. *Wireless sensor networks*. Springer, 2018.

[151] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[152] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless sensor networks: technology, protocols, and applications*. John wiley & sons, 2007.

[153] Tanveer Zia and Albert Zomaya. “Security issues in wireless sensor networks”. In: *2006 International Conference on Systems and Networks Communications (ICSNC’06)*. IEEE. 2006, pp. 40–40.

[154] Bahareh Gholamzadeh and Hooman Nabovati. “Concepts for designing low power wireless sensor network”. In: *International Journal of Electronics and Communication Engineering* 2.9 (2008), pp. 1869–1875.

[155] Cauligi S Raghavendra, Krishna M Sivalingam, and Taieb Znati. *Wireless sensor networks*. Springer, 2006.

[156] Pouya Bolourchi and Sener Uysal. “Forest fire detection in wireless sensor network using fuzzy logic”. In: *2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2013, pp. 83–87.

[157] Edward N Udo and Etebong B Isong. “Flood monitoring and detection system using wireless sensor network”. In: *Asian journal of computer and information systems* 1.4 (2013).

[158] TL Alumona, VE Idigo, and KP Nnoli. “Remote monitoring of patients health using wireless sensor networks (WSNs)”. In: *IPASJ International Journal of Electronics & Communication (IIJEC)* 2.9 (2014), pp. 90–95.

[159] Uttara Gogate and Jagdish Bakal. “Healthcare monitoring system based on wireless sensor network for cardiac patients”. In: *Biomedical & Pharmacology Journal* 11.3 (2018), p. 1681.

[160] Mustafa Kocakulak and Ismail Butun. “An overview of Wireless Sensor Networks towards internet of things”. In: *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*. Ieee. 2017, pp. 1–6.

[161] Karwan Muheden, Ebubekir Erdem, and Sercan Vançin. “Design and implementation of the mobile fire alarm system using wireless sensor networks”. In: *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE. 2016, pp. 000243–000246.

[162] Baowei Wang et al. “Temperature error correction based on BP neural network in meteorological wireless sensor network”. In: *International Journal of Sensor Networks* 23.4 (2017), pp. 265–278.

[163] Juan Aponte-Luis et al. “An efficient wireless sensor network for industrial monitoring and control”. In: *Sensors* 18.1 (2018), p. 182.

[164] Beom-Su Kim et al. “A survey on real-time communications in wireless sensor networks”. In: *Wireless communications and mobile computing* 2017 (2017).

[165] Fabian Nack. “An overview on wireless sensor networks”. In: *Institute of Computer Science (ICS), Freie Universität Berlin* 6 (2010).

[166] Amit Sarkar and T Senthil Murugan. “Cluster head selection for energy efficient and delay-less routing in wireless sensor network”. In: *Wireless Networks* 25.1 (2019), pp. 303–320.

[167] Ljubica Blazevic, J-Y Le Boudec, and Silvia Giordano. “A location-based routing method for mobile ad hoc networks”. In: *IEEE Transactions on mobile computing* 4.2 (2005), pp. 97–110.

[168] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. “An application-specific protocol architecture for wireless microsensor networks”. In: *IEEE Transactions on wireless communications* 1.4 (2002), pp. 660–670.

[169] Amandeep Kaur and Amit Grover. “LEACH and extended LEACH protocols in wireless sensor network-a survey”. In: *International Journal of Computer Applications* 116.10 (2015).

[170] Sunil Kumar Singh, Prabhat Kumar, and Jyoti Prakash Singh. “A survey on successors of LEACH protocol”. In: *Ieee Access* 5 (2017), pp. 4298–4328.

[171] Zhizhou Wu et al. “Bus priority control system based on wireless sensor network (WSN) and zigbee”. In: *2006 IEEE International Conference on Vehicular Electronics and Safety*. IEEE. 2006, pp. 148–151.

[172] Wanjing Ma and Xiaoguang Yang. “Design and evaluation of an adaptive bus signal priority system base on wireless sensor network”. In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE. 2008, pp. 1073–1077.

[173] Hieu Khac Le, Dan Henriksson, and Tarek Abdelzaher. “A control theory approach to throughput optimization in multi-channel collection sensor networks”. In: *Proceedings of the 6th international conference on Information processing in sensor networks*. 2007, pp. 31–40.

[174] Yan Wu, Sonia Fahmy, and Ness B Shroff. *On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm*. Tech. rep. PURDUE UNIV LAFAYETTE IN DEPT OF COMPUTER SCIENCES, 2008.

[175] Dijun Luo et al. “Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks”. In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 1566–1574.

[176] Imad S AlShawi et al. “Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm”. In: *IEEE Sensors journal* 12.10 (2012), pp. 3010–3018.

[177] AS Poornima and BB Amberker. “Logical ring based key management for clustered sensor networks with changing cluster head”. In: *2010 International Conference on Signal Processing and Communications (SPCOM)*. IEEE. 2010, pp. 1–5.

[178] Zhi Ren et al. “Energy-efficient ring-based multi-hop clustering routing for WSNs”. In: *2012 Fifth International Symposium on Computational Intelligence and Design*. Vol. 1. IEEE. 2012, pp. 14–17.

[179] Roberto Riggio, Tinku Rasheed, and Sabrina Sicari. “Performance evaluation of an hybrid mesh and sensor network”. In: *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE. 2011, pp. 1–6.

[180] Thuy Tran Vinh, Thu Ngo Quynh, and Mai Banh Thi Quynh. “Emrp: Energy-aware mesh routing protocol for wireless sensor networks”. In: *The 2012 International Conference on Advanced Technologies for Communications*. IEEE. 2012, pp. 78–82.

[181] A Vijayalakshmi and P Vanaja Ranjan. “Slot Management based Energy Aware routing (SMEAR) for wireless sensor networks”. In: *2012 International Conference on Computing, Communication and Applications*. IEEE. 2012, pp. 1–5.

[182] Zhibin Li and Peter X Liu. “Priority-based congestion control in multi-path and multi-hop wireless sensor networks”. In: *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2007, pp. 658–663.

[183] Omar Banimelhem and Samer Khasawneh. “Grid-based multi-path with congestion avoidance routing (GMCAR) protocol for wireless sensor networks”. In: *2009 International Conference on Telecommunications*. IEEE. 2009, pp. 131–136.

[184] Wei-dong Liu et al. “A low power grid-based cluster routing algorithm of wireless sensor networks”. In: *2010 International Forum on Information Technology and Applications*. Vol. 1. IEEE. 2010, pp. 227–229.

[185] Nguyen Doan Hieu, Dao Le Thu Thao, and Tran Manh Hoang. *Energy efficient deployment solutions in high density heterogeneous networks*. REV, 2021.

[186] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. “Wireless sensor network survey”. In: *Computer networks* 52.12 (2008), pp. 2292–2330.

[187] Eiko Yoneki and Jean Bacon. “A survey of Wireless Sensor Network technologies”. In: *UCAM-CL-TR-646* (2005).

[188] Jon S Wilson. *Sensor technology handbook*. Elsevier, 2004.

[189] Sasha Slijepcevic and Miodrag Potkonjak. “Power efficient organization of wireless sensor networks”. In: *ICC 2001. IEEE international conference on communications. Conference record (Cat. No. 01CH37240)*. Vol. 2. IEEE. 2001, pp. 472–476.

[190] EN Barron and H Ishii. “The Bellman equation for minimizing the maximum cost”. In: *Nonlinear Analysis: Theory, Methods & Applications* 13.9 (1989), pp. 1067–1090.

[191] M Ángeles Serna et al. “Distributed forest fire monitoring using wireless sensor networks”. In: *International Journal of Distributed Sensor Networks* 11.10 (2015), p. 964564.

[192] Ittipong Khemapech, Ishbel Duncan, and Alan Miller. “A survey of wireless sensor networks technology”. In: *6th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*. Vol. 13. University of St Andrews St Andrews. 2005.

[193] Tanvi Sood and Kanika Sharma. “P-LUET: A Prolong Lines of Uniformity Based Enhanced Threshold Algorithm for Heterogeneous Wireless Sensor Network Enabled Internet of Things Framework”. In: *Wireless Personal Communications* 120.4 (2021), pp. 2935–2970.

[194] Yuan Zhang et al. “Ubiquitous WSN for healthcare: Recent advances and future prospects”. In: *IEEE Internet of Things Journal* 1.4 (2014), pp. 311–318.

[195] Syed Kamrul Islam and Mohammad Rafiqul Haider. *Sensors and low power signal processing*. Springer Science & Business Media, 2009.

[196] Harald T Friis. “A note on a simple transmission formula”. In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256.

[197] Vũ Văn Yêm. *Giáo trình Hệ thống Viễn thông*. NXB Bách khoa Hà Nội, 2016.

[198] Nguyễn Văn Đức. *Kênh vô tuyến - Radio Channels*. NXB Khoa học và Kỹ thuật, 2017.

[199] Noman Shabbir and Syed Rizwan Hassan. “Routing Protocols for Wireless Sensor Networks (WSNs)”. In: *Wireless Sensor Networks*. Ed. by Philip Sallis. Rijeka: IntechOpen, 2017. Chap. 2.

[200] Noman Shabbir and Syed Rizwan Hassan. “Routing protocols for wireless sensor networks (WSNs)”. In: *Wireless Sensor Networks-Insights and Innovations* (2017), pp. 36–40.

[201] Abror Abduvaliyev et al. “On the vital areas of intrusion detection systems in wireless sensor networks”. In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1223–1237.

[202] Imran Amin and Atif Saeed. “5.10 Wireless Technologies in Energy Management”. In: *Comprehensive Energy Systems*. Ed. by Ibrahim Dincer. Elsevier, 2018, pp. 389–422. ISBN: 978-0-12-814925-6.

[203] Stanislav Safaric and Kresimir Malaric. “ZigBee wireless standard”. In: *Proceedings ELMAR 2006*. IEEE. 2006, pp. 259–262.

[204] Omar Yahya, Haider Alrikabi, and Ibtisam Aljazaery. “Reducing the data rate in internet of things applications by using wireless sensor network”. In: (2020).

[205] Nanhao Zhu et al. “Research on high data rate wireless sensor networks”. In: *14eme Journees Nationales du Reseau Doctoral de Micro et Nanoelectronique* (2011), p. 61.

[206] Özgecan Özdogan, Emil Björnson, and Erik G. Larsson. “Intelligent Reflecting Surfaces: Physics, Propagation, and Pathloss Modeling”. In: *IEEE Wireless Communications Letters* 9.5 (2020), pp. 581–585.

[207] Darian Pérez-Adán et al. “Intelligent Reflective Surfaces for Wireless Networks: An Overview of Applications, Approached Issues, and Open Problems”. In: *Electronics* 10.19 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10192345. URL: https://www.mdpi.com/2079-9292/10/19/2345.

[208] Chongwen Huang et al. “Achievable Rate maximization by Passive Intelligent Mirrors”. In: (2018). arXiv: 1807.07196 [cs.IT].

[209] Emil Björnson, Luca Sanguinetti, and Marios Kountouris. “Deploying Dense Networks for Maximal Energy Efficiency: Small Cells Meet Massive MIMO”. In: *IEEE Journal on Selected Areas in Communications* 34.4 (2016), pp. 832–847. DOI: 10.1109/JSAC.2016.2544498.

[210] Emil Björnson, Özgecan Özdogan, and Erik G. Larsson. “Reconfigurable Intelligent Surfaces: Three Myths and Two Critical Questions”. In: *IEEE Communications Magazine* 58.12 (2020), pp. 90–96. DOI: 10.1109/MCOM.001.2000407.

[211] Yuanwei Liu et al. “Reconfigurable Intelligent Surfaces: Principles and Opportunities”. In: *IEEE Communications Surveys Tutorials* 23.3 (2021), pp. 1546–1577. DOI: 10.1109/COMST.2021.3077737.