TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Federated Dynamic Impurity

Nghiên cứu và đề xuất thuật toán tổng hợp mô hình trong Học kết hợp nhằm cải thiện tốc độ hội tụ trong môi trường mất cân bằng phân phối dữ liệu đối với bài toán phân loại ảnh

NGUYỄN HUY HIỆU
hieu1520065@huce.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: Ths. Hoàng Nam Thắng _____

Chữ ký

Khoa: Công nghệ thông tin

School: Trường đại học Xây Dựng Hà Nội

HANOI, 12/2024

LỜI CAM KẾT

Tôi – Nguyễn Huy Hiệu – cam kết Đồ án Tốt nghiệp (ĐATN) với đề tài: "Federated Dynamic Impurity: Nghiên cứu và đề xuất thuật toán tổng hợp mô hình trong Học kết hợp nhằm cải thiện tốc độ hội tụ trong môi trường mất cân bằng phân phối dữ liệu đối với bài toán phân loại ảnh" là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của Thạc sĩ Hoàng Nam Thắng. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép từ bất kỳ công trình nào khác. Tất cả các tài liệu tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo.

Tôi xin hoàn toàn chịu trách nhiệm nếu có bất kỳ sao chép nào vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm

Hiệu

Nguyễn Huy Hiệu

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành tới giảng viên hướng dẫn trong quá trình thực hiện đồ án này, Thầy Hoàng Nam Thắng. Bên cạnh đó, em xin bày tỏ lòng biết ơn sâu sắc tới Tiến sĩ Hải Anh Trần và Tiến sĩ Trương X. Trần, những người đã tận tình hướng dẫn và hỗ trợ em rất nhiều trong suốt hành trình nghiên cứu này. Kiến thức sâu sắc, sự phản hồi quý báu và khích lệ của họ chính là những yếu tố then chốt định hình quá trình làm việc của em. Em thực sự may mắn khi có cơ hội được học hỏi và trưởng thành dưới sự dẫn dắt của họ.

Em cũng xin gửi lời cảm ơn chân thành nhất tới gia đình vì sự hỗ trợ, thấu hiểu và động viên không ngừng nghỉ trong suốt chặng đường này. Ngoài ra, em xin cảm ơn những người bạn của mình – tình bạn, sự động viên và niềm đam mê học tập của họ đã trở thành nguồn cảm hứng to lớn. Sự nhiệt tình của họ đã khiến cho hành trình này trở nên ý nghĩa và bổ ích hơn. Em thật sự trân trọng những mối quan hệ quý giá được hình thành trong quá trình nỗ lực nghiên cứu này.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Để giải quyết các lo ngại về bảo mật cũng như tài nguyên mạng cần thiết để huấn luyện một mô hình học máy hiệu quả, vào năm 2016, Google đã giới thiệu một mô hình học phân tán mới, gọi là Học kết hợp (Federated Learning). Mô hình học kết hợp tối ưu hóa khả năng của các thiết bị phía người dùng để thực hiện việc học phân tán, khi các thiết bị này cùng nhau huấn luyện một mô hình học máy. Học kết hợp đã mang đến một giải pháp tiềm năng cho các công ty và tập đoàn trong việc huấn luyện các mô hình học máy hiệu quả mà không phải lo ngại về các yếu tố như bảo mật thông tin dữ liệu người dùng, chi phí truyền thông và lưu trữ. Điều này đặc biệt quan trọng trong bài toán phân loại ảnh, khi dữ liệu đến từ nhiều nguồn khác nhau và có tính nhạy cảm cao hơn so với các loại dữ liệu khác. Học kết hợp cho phép mô hình có thể tiếp cận và học trên các dữ liệu đa dạng, nhạy cảm một cách hiệu quả.

Tuy nhiên, mô hình học kết hợp lại gặp phải một thách thức to lớn khi lượng dữ liệu trên các thiết bị thường có phân phối không đồng đều. Điều này xuất phát từ sự khác nhau trong cường độ sử dụng và mục đích sử dụng của mỗi thiết bị. Việc học từ các phân phối dữ liệu khác nhau dẫn đến các mô hình cục bộ khó có thể đưa ra kết quả tốt, gây khó khăn trong việc đạt được hội tụ của mô hình toàn cầu và làm giảm độ chính xác mà mô hình có thể đạt được. Hiện nay, đã có nhiều thuật toán khác nhau được đề xuất nhằm giải quyết vấn đề này, trong đó quá trình tổng hợp các mô hình cục bộ là một hướng nghiên cứu được quan tâm hàng đầu. Các thuật toán này đã đạt được những kết quả hứa hẹn, giúp cải thiện đáng kể độ chính xác và tốc độ hội tụ so với thuật toán tổng hợp gốc. Tuy nhiên, các phương pháp này vẫn chưa hoàn toàn giải quyết triệt để vấn đề phân phối dữ liệu không đồng đều.

Trong đồ án này, một thuật toán tổng hợp mô hình mới trong học kết hợp, cải tiến từ thuật toán tổng hợp FedImp và được gọi là DyFedImp, được đề xuất nhằm giải quyết vấn đề phân phối dữ liệu không đồng đều giữa các thiết bị. Thuật toán này bổ sung các yếu tố động trong quá trình tính toán trọng số cho các mô hình cục bộ. Mục tiêu chính của thuật toán đề xuất là cải thiện tốc độ hội tụ của mô hình toàn cầu mà vẫn đảm bảo tính tổng quát của mô hình. Các kết quả thực nghiệm được thực hiện thông qua việc giả lập mô hình học kết hợp trên các bộ dữ liệu thực tế cho bài toán phân loại ảnh đã cho thấy thuật toán đề xuất thành công trong việc giảm thiểu ảnh hưởng của phân phối dữ liệu không đồng đều lên quá trình hội tụ của mô hình. Cụ thể, DyFedImp đã cải thiện tốc độ hội tụ từ 10 đến 50% so với FedImp và từ 30 đến 75% so với các thuật toán tổng hợp mô hình khác.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong kỷ nguyên ra quyết định dựa trên dữ liệu, vai trò của các mô hình học máy đã trở nên nổi bật trong nhiều ứng dụng khác nhau. Từ việc cung cấp các gợi ý cá nhân hóa đến hỗ trợ phân tích dự đoán, các mô hình này đã hòa nhập liền mạch vào nền tảng của những tiến bộ công nghệ. Tuy nhiên, các phương pháp truyền thống để huấn luyện mô hình luôn yêu cầu dữ liệu được tập trung tại một vị trí duy nhất, điều này đã gây ra nhiều thách thức đáng kể trong việc xây dựng các mô hình học máy hiệu quả. Trong mô hình tập trung này, các lo ngại về quyền riêng tư người dùng, bảo mật và băng thông truyền thông ngày càng trở nên rõ nét, cho thấy nhu cầu cấp thiết về một giải pháp sáng tạo và thích ứng tốt hơn. Đặc biệt, khi ngày càng nhiều đạo luật và chính sách được ban hành nhằm ngăn chặn việc di chuyển dữ liệu ra khỏi lãnh thổ, chẳng hạn như GDPR (Europe), CCPA (California), PIPEDA (Canada), v.v., vấn đề này càng trở nên quan trọng hơn. Do đó, Học kết hợp (Federated Learning - FL) nổi lên như một phương pháp mang tính cách mạng, có tiềm năng vượt qua các thách thức trên và mở ra một kỷ nguyên mới cho việc huấn luyện mô hình học máy phi tập trung [1].

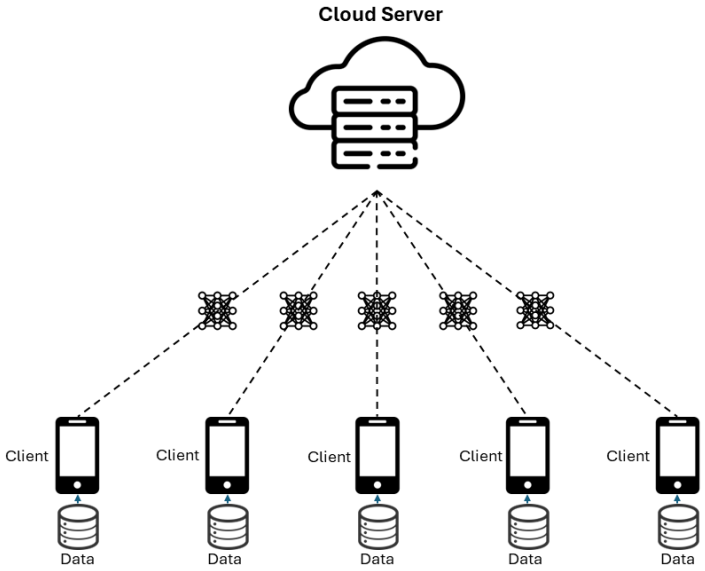
Phương pháp học máy kết hợp này đã cách mạng hóa cách tiếp cận huấn luyện truyền thống bằng cách cho phép nhiều thiết bị kết hợp cùng nhau để huấn luyện một mô hình chung mà không cần trao đổi dữ liệu thô. Điều này giúp loại bỏ nguy cơ rò rỉ dữ liệu nhạy cảm và cho phép mô hình học từ lượng dữ liệu đa dạng hơn. Không giống như phương pháp huấn luyện tập trung, nơi dữ liệu được tập hợp tại một máy chủ trung tâm, Học kết hợp (FL) khai thác khả năng tính toán song song của các thiết bị cục bộ, từ điện thoại thông minh đến các thiết bị biên hoặc các thiết bị IoT khác. Phương pháp này không chỉ bảo vệ quyền riêng tư của dữ liệu bằng cách giữ thông tin nhạy cảm tại chỗ mà còn giảm thiểu đáng kể chi phí và nhu cầu chuyển giao dữ liệu. Tính phi tập trung của FL đặc biệt phù hợp cho các ứng dụng trong các lĩnh vực mà bảo mật dữ liệu là mối quan tâm hàng đầu, như chăm sóc sức khỏe và tài chính. Bằng cách giảm thiểu các rủi ro liên quan đến việc tập trung dữ liệu, FL mở ra con đường cho một hệ sinh thái học máy an toàn và tôn trọng quyền riêng tư hơn. Hơn nữa, tính linh hoạt của phương pháp này đáp ứng tốt các yêu cầu động của nhiều lĩnh vực ứng dụng khác nhau, mang lại giải pháp vững chắc để khắc phục các hạn chế mà phương pháp huấn luyện tập trung truyền thống còn tồn tại.

Sau khi được giới thiệu lần đầu tiên, Học kết hợp đã nhanh chóng trở thành một

trong các chủ đề được nghiên cứu rộng rãi trong nhiều lĩnh vực nhờ tính ứng dụng cao của mô hình này [2]. Các lĩnh vực nghiên cứu có thể kể đến bao gồm học máy, bảo mật, IoT và hệ thống phân tán. Mô hình Học kết hợp cũng được nhiều công ty và tập đoàn ứng dụng thành công trong các lĩnh vực khác nhau. Google đã áp dụng Học kết hợp để tích hợp trí tuệ nhân tạo vào các sản phẩm như bàn phím Gboard [3], điện thoại Pixel [4] và ứng dụng Android Messages [5]. Apple cũng triển khai Học kết hợp trên các thiết bị chạy iOS 13, đặc biệt là cho các ứng dụng như bàn phím QuickType và bộ phân loại giọng nói “Hey Siri" [6]. Công ty doc.ai đang phát triển các giải pháp Học kết hợp trên nhiều thiết bị phục vụ nghiên cứu y tế [7]. Gần đây, với sự phổ biến rộng rãi của các mô hình ngôn ngữ lớn (LLMs), nhu cầu sử dụng Học kết hợp ngày càng gia tăng do các mô hình này yêu cầu được huấn luyện trên lượng dữ liệu khổng lồ. Một số mô hình tiêu biểu đã áp dụng Học kết hợp trong quá trình huấn luyện có thể kể đến như OpenAI GPTs và DALL-Es, Google PaLMs và Geminis, hay Meta LLaMAs [8]. Ngoài ra, Học kết hợp còn được ứng dụng trong nhiều lĩnh vực quan trọng khác như y tế và tài chính. Đối với bài toán phân loại ảnh, tiềm năng ứng dụng của Học kết hợp là vô cùng lớn, bởi tính chất của dữ liệu ảnh đòi hỏi quyền riêng tư cao. Điều này đặc biệt đúng với các loại ảnh nhạy cảm như ảnh y tế (chẩn đoán qua X-quang, CT scan) từ các bệnh viện, ảnh phục vụ nhận diện khuôn mặt hoặc giám sát an ninh, hay việc phân loại nội dung ảnh trên các nền tảng mạng xã hội. Mặt khác, dữ liệu ảnh thường được thu thập từ nhiều nguồn khác nhau như camera, thiết bị IoT và điện thoại di động. Điều này đặt ra nhu cầu cấp thiết trong việc nghiên cứu và phát triển các phương pháp Học kết hợp hiệu quả cho bài toán phân loại ảnh.

Quy trình Học kết hợp diễn ra qua một chuỗi cập nhật mô hình lặp đi lặp lại, bao gồm một máy chủ tập trung điều phối quá trình huấn luyện và các thiết bị đầu cuối, còn được gọi là các máy khách (clients) hoặc các nút [9], như được minh họa trong Hình 1.1. Ban đầu, các mô hình cục bộ được cập nhật dựa trên tập dữ liệu riêng biệt được lưu trữ trên từng thiết bị. Các cập nhật này phản ánh đặc trưng và ngữ cảnh thông tin độc lập của từng thiết bị. Tiếp theo, các mô hình cục bộ sẽ được gửi về máy chủ trung tâm để thực hiện quá trình tổng hợp. Trong quá trình này, các thuật toán như trung bình có trọng số thường được sử dụng để kết hợp các mô hình cục bộ thành một mô hình toàn cục thống nhất. Phương pháp tổng hợp này đảm bảo tích hợp một cách hài hòa và hiệu quả thông tin đa dạng được đóng góp từ mỗi thiết bị. Khung học máy phân tán này đặc biệt nổi bật nhờ khả năng thu thập và tận dụng kiến thức từ các tập dữ liệu không đồng nhất trên các thiết bị tham gia, đồng thời vẫn duy trì được tính bảo mật và quyền riêng tư của dữ liệu.

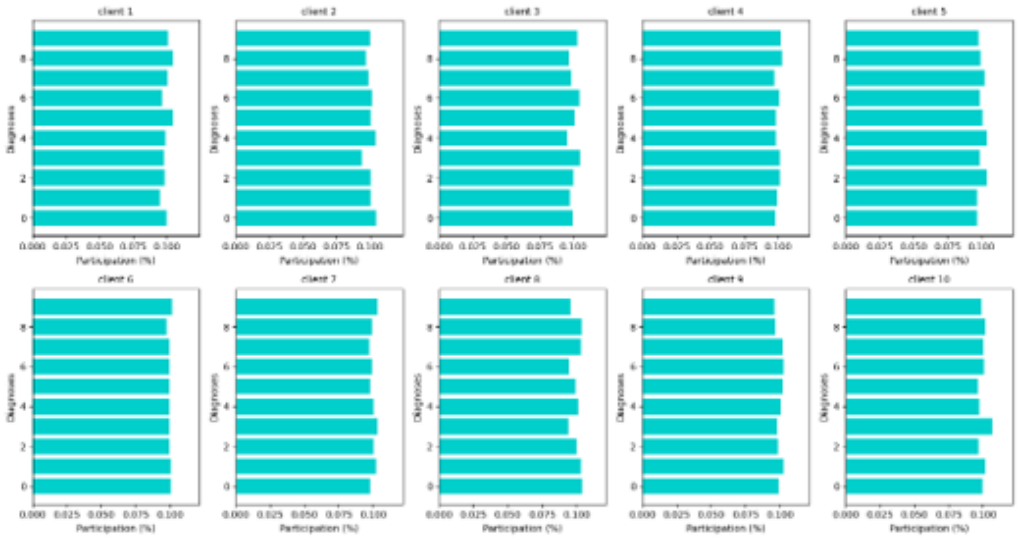
Một trong những thách thức chính mà các thuật toán Học kết hợp (FL) phải đối



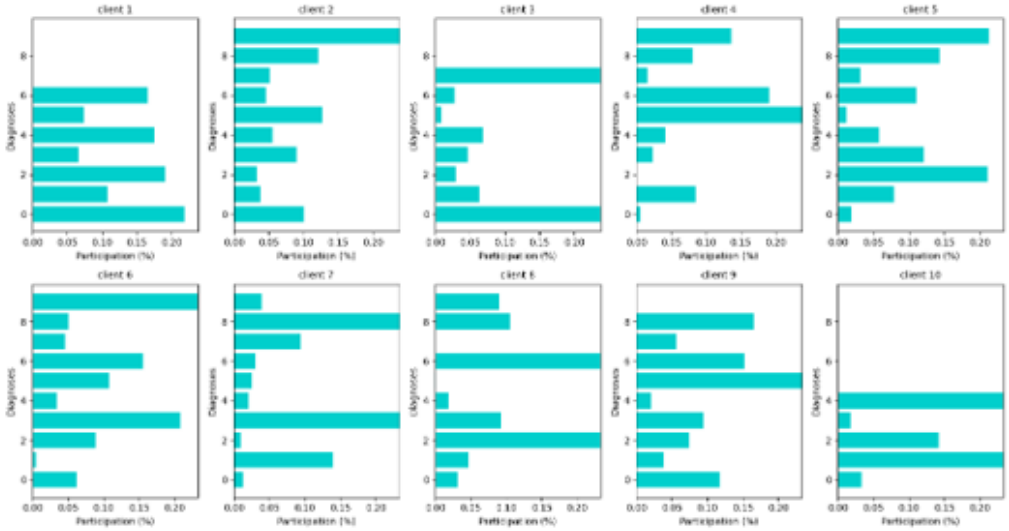
Hình 1.1: Mô tả một hệ thống học sử dụng Học kết hợp

mặt là sự tồn tại của dữ liệu không phân phối độc lập và đồng nhất (non-IID) trên các thiết bị phi tập trung [10]. Trong các tình huống học máy truyền thống, giả định IID (Independent and Identically Distributed) thường được áp dụng, tức là dữ liệu huấn luyện có cùng phân phối trên tất cả các thiết bị tham gia. Tuy nhiên, trong FL, giả định này thường không đúng vì phân phối dữ liệu có thể khác nhau giữa các thiết bị do các yếu tố như vị trí địa lý, hành vi người dùng hoặc đặc điểm của từng thiết bị. Điều này dẫn đến tình trạng dữ liệu trên một số thiết bị có thể chứa nhiều mẫu của một nhãn cụ thể hơn so với các nhãn khác, hoặc thậm chí một số nhãn hoàn toàn không xuất hiện. Dữ liệu non-IID đặt ra một trở ngại đáng kể cho việc hợp tác hiệu quả giữa các thiết bị trong FL. Khi mỗi thiết bị huấn luyện mô hình trên dữ liệu cục bộ có các đặc tính thống kê khác nhau, mô hình cục bộ có xu hướng chuyên biệt hóa vào các mẫu của phân phối dữ liệu đó. Kết quả là, việc tổng hợp các mô hình cục bộ thành một mô hình toàn cục thống nhất trở nên khó khăn. Tỷ lệ hội tụ kém phát sinh từ sự khác biệt giữa các mô hình cục bộ, khiến cho quá trình hợp nhất vào mô hình toàn cục bị chậm lại. Đồng thời, hiệu suất tổng quát hóa không tối ưu càng làm trầm trọng thêm vấn đề, khi mô hình toàn cục gặp khó khăn trong việc học được các thông tin đa dạng từ nhiều phân phối dữ liệu khác nhau. Điều này làm giảm hiệu quả của mô hình khi áp dụng cho các tập dữ liệu chưa thấy trước. Ví dụ minh họa về dữ liệu non-IID trong FL được mô tả trong Hình 4.4.

Giải quyết hiệu quả thách thức của dữ liệu non-IID trong Học kết hợp (FL) đã trở thành mục tiêu trọng tâm của nhiều thuật toán mới được đề xuất. Các thuật toán này áp dụng các chiến lược tinh vi nhằm giảm thiểu tác động của sự mất cân bằng



(a) Dữ liệu gần IID



(b) Dữ liệu non-IID

Hình 1.2: Ví dụ về sự khác biệt về phân phối dữ liệu

phân phối dữ liệu. Một trong những phương pháp phổ biến là sử dụng kỹ thuật tổng hợp có trọng số, trong đó mức độ đóng góp của từng thiết bị vào mô hình toàn cục được xác định dựa trên các yếu tố như chất lượng và mức độ liên quan của dữ liệu. Bên cạnh đó, các kỹ thuật gia tăng dữ liệu cũng được triển khai như một giải pháp bổ trợ hiệu quả, nhằm mở rộng và làm phong phú thêm bộ dữ liệu cục bộ một cách nhân tạo. Bằng cách tạo ra các biến thể từ dữ liệu hiện có, phương pháp này giúp mô hình toàn cục phát triển khả năng khái quát hóa tốt hơn, từ đó nắm bắt được các mẫu và đặc trưng tiềm ẩn một cách toàn diện hơn. Một chiến lược đáng chú ý khác là điều chỉnh tốc độ học thích ứng, cho phép điều chỉnh linh hoạt tốc độ cập nhật mô hình dựa trên các đặc điểm cụ thể của phân phối dữ liệu trên từng thiết bị. Cơ chế này tạo điều kiện để mô hình phản ứng linh hoạt với các tình huống non-IID đa dạng, đảm bảo quá trình huấn luyện được tối ưu hóa phù hợp với đặc trưng của

từng thiết bị đóng góp. Đồ án này giới thiệu một thuật toán mới nhằm cải thiện tốc độ hội tụ của FL, đặc biệt trong các trường hợp phân phối dữ liệu có mức độ mất cân bằng cao. Thuật toán đề xuất không chỉ đảm bảo tốc độ hội tụ nhanh hơn mà còn giữ được tính tổng quát của mô hình toàn cục khi xử lý dữ liệu non-IID.

1.2 Các giải pháp hiện tại và hạn chế

Một số thuật toán đã được đề xuất nhằm tổng hợp các mô hình cục bộ thành một mô hình toàn cục duy nhất với mục tiêu đạt được kết quả tối ưu [11]. Trong số đó, thuật toán Federated Averaging (FedAvg) [12] nổi bật như một phương pháp cơ sở và được sử dụng rộng rãi trong Học kết hợp. FedAvg hoạt động dựa trên nguyên tắc tính trung bình các tham số mô hình từ các nút tham gia và cập nhật mô hình toàn cục trên máy chủ trung tâm. Tuy nhiên, FedAvg tồn tại một số hạn chế đáng chú ý, đặc biệt là khi đối mặt với dữ liệu không đồng nhất (non-IID). Cụ thể, tốc độ hội tụ của FedAvg khá chậm và độ chính xác thấp khi mức độ không đồng nhất dữ liệu giữa các nút lớn. Một vấn đề cốt lõi là đóng góp của các nút tham gia vào quá trình huấn luyện không chỉ phụ thuộc vào kích thước của tập dữ liệu cục bộ mà còn liên quan đến đặc điểm phân phối dữ liệu. Do đó, việc đơn giản tính trung bình các tham số mô hình cục bộ không phải lúc nào cũng hợp lý, đặc biệt trong các trường hợp dữ liệu non-IID.

Nhằm khắc phục những hạn chế này, nhiều thuật toán cải tiến đã được đề xuất, tiêu biểu như FedProx [1] và FedAvgM [13]. FedProx giới thiệu một thuật toán tối ưu hóa nhằm xử lý sự không đồng nhất giữa các nút bằng cách bổ sung một ràng buộc chuẩn hóa vào quá trình huấn luyện. Trong khi đó, FedAvgM cải tiến FedAvg bằng cách kết hợp thêm động lượng (momentum) vào quá trình cập nhật mô hình toàn cục để tăng tốc độ hội tụ. Tuy nhiên, mặc dù những phương pháp này đã đạt được một số cải thiện, chúng vẫn chưa thể nâng cao đáng kể hiệu suất tổng thể của mô hình FL trong điều kiện dữ liệu non-IID. Điều này cho thấy cần có các giải pháp mới hiệu quả hơn nhằm giải quyết triệt để các thách thức về tốc độ hội tụ và khả năng tổng quát hóa của mô hình trong bối cảnh dữ liệu phi tập trung.

Bên cạnh các hướng tiếp cận truyền thống, các phương pháp tập trung vào việc điều chỉnh trọng số của từng mô hình cục bộ trong quá trình tổng hợp mô hình đã nhận được sự quan tâm đáng kể. Đây được xem là lời giải trực tiếp cho các hạn chế của thuật toán Federated Averaging (FedAvg) trong việc xử lý dữ liệu không đồng nhất (non-IID). Để khắc phục nhược điểm về tốc độ hội tụ của FedAvg, thuật toán Federated Adaptive Weighting (FedAdp) [14] đã được đề xuất. FedAdp cải thiện quá trình tổng hợp mô hình toàn cục bằng cách gán trọng số động cho từng nút tham gia, dựa trên mức độ đóng góp của chúng trong mỗi vòng giao tiếp. Cụ thể, đóng

góp của một nút được đánh giá thông qua góc giữa vector gradient cục bộ và vector gradient toàn cục. Giá trị này sau đó được ánh xạ qua một hàm phi tuyến được thiết kế để xác định trọng số thích hợp cho mỗi nút. Hiệu quả của FedAdp đặc biệt nổi bật trong trường hợp dữ liệu non-IID. Kết quả thực nghiệm cho thấy FedAdp giảm đáng kể số vòng giao tiếp cần thiết để đạt được các ngưỡng độ chính xác cụ thể so với FedAvg. Trên bộ dữ liệu MNIST, FedAdp rút ngắn số vòng giao tiếp để đạt 95% độ chính xác lên đến 54,1%, trong khi trên bộ dữ liệu FashionMNIST, số vòng giao tiếp cần thiết để đạt 80% độ chính xác giảm đến 45,4%. Những kết quả này minh chứng cho khả năng cải thiện tốc độ hội tụ đáng kể của FedAdp trong môi trường học kết hợp. Tuy nhiên, FedAdp vẫn tồn tại một nhược điểm quan trọng. Trong trường hợp dữ liệu trên các nút thiếu tính đại diện đầy đủ cho tất cả các nhãn (ví dụ: một số nhãn hoàn toàn không có mặt), các gradient cục bộ có thể lệch hướng so với gradient toàn cục mong đợi. Khi điều này xảy ra trên một lượng lớn các nút, gradient toàn cục — vốn được tính trung bình từ các gradient cục bộ — có thể không phản ánh đúng hướng học tập mong muốn. Hậu quả là, việc gán trọng số cao hơn cho các gradient cục bộ có hướng gần với gradient toàn cục có thể vô tình làm chậm tốc độ hội tụ và ảnh hưởng tiêu cực đến hiệu suất tổng thể của mô hình.

Tuy nhiên, FedAdp gặp phải một vấn đề đáng kể khi môi trường FL tồn tại nhiều nút non-IID có phân phối tương đồng nhau. Trong trường hợp này, thuật toán có thể gặp khó khăn trong việc điều chỉnh trọng số một cách chính xác, dẫn đến mô hình toàn cục bị kéo gần về các nút này, gây ra hiệu suất học tập không tối ưu. Để giải quyết vấn đề này, nghiên cứu gần đây đã đề xuất một thuật toán đánh trọng số mới, Federated Impurity Weighting (FedImp) [15]. FedImp sử dụng công thức entropy để ước lượng mức độ non-IID trên mỗi thiết bị và từ đó tính trọng số cho từng mô hình cục bộ. Ý tưởng cốt lõi của thuật toán này là: một mô hình cục bộ được huấn luyện trên bộ dữ liệu gần với IID sẽ có hướng gradient gần với cực trị toàn cục hơn. Trong bối cảnh này, entropy — một độ đo phổ biến để đánh giá sự hỗn loạn của một phân phối dữ liệu — được áp dụng để định lượng mức độ không đồng nhất của dữ liệu trên từng nút. Kết quả thực nghiệm của FedImp cho thấy hiệu suất vượt trội so với FedAdp và FedAvg trong phần lớn các tình huống thử nghiệm. Cụ thể, FedImp cải thiện tốc độ hội tụ nhanh hơn 60% trên bộ dữ liệu EMNIST và 40% trên bộ dữ liệu CIFAR-10. Đáng chú ý, trong trường hợp môi trường FL có mức độ mất cân bằng cao (ví dụ: 3 nút IID và 7 nút non-IID), FedImp hoàn toàn áp đảo các thuật toán còn lại về hiệu quả hội tụ và độ chính xác của mô hình toàn cục.

Mặc dù vậy, một hạn chế của FedImp là chưa thể hiện được sự linh hoạt cần thiết để xử lý tốt các trường hợp đa dạng và phức tạp hơn. Điều này đặt ra yêu cầu

về các nghiên cứu tiếp theo nhằm cải thiện khả năng thích ứng của thuật toán trong nhiều môi trường dữ liệu khác nhau.

1.3 Mục tiêu và định hướng giải pháp

Đồ án này nhằm giải quyết các hạn chế còn tồn tại trong các thuật toán tổng hợp mô hình của FL nói chung và thuật toán FedImp nói riêng bằng cách đề xuất một thuật toán cải tiến mới mang tên Federated Dynamic Impurity Weighting (DyFedImp). Thuật toán FedImp đã cho thấy hiệu suất vượt trội trong nhiều trường hợp, nhưng vẫn còn tiềm năng cải tiến để phù hợp hơn với các môi trường dữ liệu non-IID đa dạng. Một trong những hạn chế chính của FedImp là cách chọn tham số và huấn luyện còn mang tính tĩnh và cố định, khiến thuật toán thiếu khả năng thích ứng linh hoạt trong suốt quá trình huấn luyện. Dựa trên quan sát này, DyFedImp được thiết kế với ý tưởng cải tiến cơ chế tính toán trọng số bằng cách tích hợp các hàm động nhằm điều chỉnh tham số một cách linh hoạt theo từng giai đoạn của quá trình huấn luyện. Trọng số của các mô hình cục bộ sẽ được ước lượng và điều chỉnh phù hợp, không chỉ dựa trên mức độ non-IID hiện tại mà còn tính đến sự thay đổi trong phân phối dữ liệu và đóng góp của từng thiết bị qua thời gian. Với các cải tiến này, DyFedImp đặt mục tiêu mang lại một phương pháp tổng hợp mô hình toàn cục cân bằng và toàn diện hơn trong Học kết hợp. Thuật toán không chỉ cải thiện tốc độ hội tụ mà còn đảm bảo tính tổng quát cao cho mô hình toàn cục, đặc biệt trong các trường hợp dữ liệu không đồng nhất và có mức độ mất cân bằng lớn.

1.4 Đóng góp của đồ án

Đồ án có các đóng góp chính như sau:

- **Xác định các nhược điểm và chưa tối ưu của thuật toán FedImp**: Đồ án tiến hành phân tích sâu về công thức và hiệu năng thực nghiệm của FedImp, nhằm chỉ ra những hạn chế của thuật toán trong các kịch bản dữ liệu không đồng nhất (non-IID) đa dạng, cũng như sự thiếu linh hoạt trong việc thích ứng với các giai đoạn khác nhau của quá trình huấn luyện.
- **Đề xuất một thuật toán cải tiến của FedImp mới để cải thiện tốc độ hội tụ bằng cách thêm vào các yếu tố động trong công thức tính trọng số**: Dựa trên những đánh giá chi tiết về thuật toán FedImp và FedAdp, đồ án đề xuất và phát triển một thuật toán học kết hợp mới mang tên DyFedImp. Thuật toán này được thiết kế nhằm khắc phục các hạn chế của FedImp, đồng thời cung cấp một phương pháp linh hoạt và hiệu quả hơn để cải thiện FL trong các kịch bản dữ liệu không đồng nhất, đặc biệt là trong các trường hợp có mức độ non-IID cao.
- **Thực hiện các thực nghiệm trên bộ dữ liệu thực để đánh giá hiệu suất**

của thuật toán đề xuất: Các thực nghiệm được tiến hành trên các bộ dữ liệu ảnh thực tế như EMNIST và CIFAR-10, trong đó dữ liệu được chia nhỏ thành các phần tương ứng cho từng nút nhằm mô phỏng quá trình huấn luyện học kết hợp. Nhiều kịch bản thực nghiệm khác nhau được thiết lập để đánh giá tổng quan hiệu suất hội tụ của mô hình DyFedImp so với các thuật toán SOTA khác.

1.5 Bố cục đồ án

Bố cục của đồ án được sắp xếp như sau:

- Chương 2 tập trung vào việc cung cấp một tổng quan tài liệu toàn diện. Chương này nhằm thảo luận về bối cảnh của vấn đề và đi sâu vào các nghiên cứu hiện có.
- Chương 3 cung cấp các thông tin nền tảng cần thiết về các khái niệm chính liên quan đến Học kết hợp và khám phá các thuật toán khác nhau, làm rõ chức năng và thành phần của chúng. Bằng cách này, chương 3 xây dựng nền tảng cần thiết cho sự phát triển và hiểu biết của phương pháp được đề xuất ở các phần sau.
- Trong chương 4, phương pháp đề xuất được mô tả chi tiết hơn cùng với các đánh giá tổng quan. Chương này bắt đầu với mục 4.1, phân tích kỹ lưỡng những nhược điểm nhận thấy của thuật toán FedImp, cung cấp động lực cho việc giới thiệu một thuật toán mới toàn diện hơn. Mục 4.2 bao gồm mô tả các đề xuất được sử dụng nhằm giải quyết các nhước điểm đã được nêu ra trong phần 4.1, với 2 nhược điểm được thảo luận chính.
- Chương 5 đi sâu vào các kết quả số liệu thu được từ việc thực hiện phương pháp đề xuất. Chương này bắt đầu với việc xem xét toàn diện về các tập dữ liệu, các chỉ số đánh giá, và phương pháp mô phỏng được sử dụng trong quá trình thực nghiệm. Sau đó, phần tiếp theo trình bày các kết quả số liệu, mang đến những thông tin quan trọng về hiệu suất và tính hiệu quả của phương pháp đề xuất.
- Trong chương 6, chương kết luận, tóm tắt các phát hiện chính được rút ra từ phân tích các kết quả số liệu.

CHƯƠNG 2. KHẢO SÁT ĐỀ TÀI

Chương này đi sâu vào phạm vi nghiên cứu, nhằm cung cấp cái nhìn toàn diện về các giới hạn và các khía cạnh mà đồ án sẽ tập trung. Đồng thời, chương cũng khám phá các nghiên cứu hiện có liên quan đến đề tài. Thông qua việc đánh giá có hệ thống, chương này sẽ xây dựng nền tảng cho đồ án bằng cách xác định các khoảng trống cũng như các cơ hội nghiên cứu mở rộng.

2.1 Ngữ cảnh của bài toán

Đồ án này khám phá các thách thức mà các thuật toán Học kết hợp (FL) phải đối mặt, đặc biệt là những hạn chế gặp phải khi có dữ liệu không đồng nhất (non-IID). Thông qua việc đánh giá và tổng hợp các kết quả nghiên cứu trước đây, đồ án cung cấp một phương pháp cân bằng và toàn diện hơn để tổng hợp mô hình toàn cục trong Học kết hợp.

2.2 Các kết quả nghiên cứu tương tự

Nhiều phương pháp đã được đề xuất nhằm hợp nhất các mô hình cục bộ thành một mô hình toàn cục duy nhất để đạt được kết quả tối ưu. H. Brendan McMahan và cộng sự [12] đã giới thiệu thuật toán Federated Averaging (FedAvg), trong đó các nút thực hiện nhiều chu kỳ thuật toán SGD trên các tập dữ liệu cục bộ và gửi các mô hình đã cập nhật về máy chủ, nơi chúng được trung bình để tạo thành mô hình toàn cục mới. FedAvg có khả năng huấn luyện mô hình chất lượng cao với số vòng giao tiếp tương đối ít, được chứng minh qua các kết quả trên nhiều kiến trúc mô hình khác nhau, bao gồm perceptron đa lớp, hai mạng nơ-ron tích chập khác nhau, LSTM ký tự hai lớp, và LSTM cấp từ quy mô lớn. Tuy nhiên, sự xuất hiện của dữ liệu không đồng nhất (non-IID) trong hệ thống FL có thể ảnh hưởng tiêu cực đến hiệu suất của FedAvg, bao gồm tốc độ hội tụ chậm và độ chính xác kém. Việc giải quyết thách thức này là rất quan trọng để cải thiện hiệu quả của phương pháp FL, và đã có nhiều giải pháp được đề xuất để khắc phục vấn đề này.

SGD với momentum đã chứng minh thành công trong việc tăng tốc độ huấn luyện mạng trong phương pháp học máy tập trung, thông qua việc tích lũy lịch sử gradient qua thời gian để giảm dao động. Dựa trên ý tưởng này, Tzu-Ming Harry Hsu và cộng sự [13] đã đề xuất thuật toán Federated Averaging with Server Momentum (FedAvgM). Thuật toán này đặc biệt phù hợp cho Học kết hợp (FL), nơi các nút tham gia chỉ sở hữu một tập con nhỏ các nhãn và phân phối dữ liệu thưa thớt. Các thử nghiệm trên bộ dữ liệu CIFAR-10 cho thấy hiệu suất phân loại được cải thiện đáng kể với FedAvgM so với FedAvg, đặc biệt là khi đối mặt với các mức độ không đồng nhất khác nhau. Độ chính xác phân loại đã được cải thiện từ 30,1%

lên 76,9% trong các thiết lập có độ lệch dữ liệu cao nhất.

Reddi và cộng sự [16] đã giới thiệu ba phương pháp tối ưu hóa học liên kết thích ứng: FedAdagrad, FedYogi và FedAdam, nhằm giải quyết các vấn đề hội tụ không thuận lợi của FedAvg. Kết quả thực nghiệm của họ nhấn mạnh mối quan hệ giữa tính không đồng nhất của các nút và hiệu quả truyền thông. Đánh giá các phương pháp này cũng chỉ ra rằng việc sử dụng các bộ tối ưu hóa thích ứng có thể cải thiện đáng kể hiệu suất của Học kết hợp.

Yousef Yeganeh và cộng sự [17] đã đề xuất IDA (Inverse Distance Aggregation), một phương pháp trọng số thích ứng mới cho các nút dựa trên thông tin meta, nhằm xử lý dữ liệu không cân bằng và không đồng nhất. Phương pháp này sử dụng khoảng cách của các tham số mô hình như một chiến lược để giảm thiểu tác động của các ngoại lệ và cải thiện tốc độ hội tụ của mô hình. Kết quả trong bài báo cho thấy phương pháp IDA vượt trội hơn FedAvg về độ chính xác phân loại trong các kịch bản non-IID. IDA cũng có khả năng chống chịu tốt trước dữ liệu chất lượng thấp hoặc dữ liệu độc hại từ các nút.

Nhận thấy rằng tối thiểu hóa hàm mất mát cục bộ không đồng nghĩa với tối thiểu hóa hàm mất mát toàn cục, Durmus Alp Emre Acar và cộng sự đã giải quyết vấn đề cơ bản trong dữ liệu không đồng nhất khi giới thiệu thuật toán FL with Dynamic Regularization (FedDyn) [18]. FedDyn động điều chỉnh hàm mất mát ở từng nút để hàm mất mát đã được sửa đổi hội tụ đến hàm mất mát toàn cục thực tế. Khi các hàm mất mát cục bộ được căn chỉnh, FedDyn đạt hiệu quả với tốc độ hội tụ O(1/T) với T là số vòng huấn luyện, trong cả các thiết lập lồi và không lồi, và tốc độ tuyến tính trong thiết lập lồi mạnh, mà không phụ thuộc vào tính không đồng nhất của thiết bị.

Nhận ra sự cần thiết của việc cá nhân hóa mô hình toàn cục trong việc xử lý các thách thức phát sinh với dữ liệu không đồng nhất, Saeed Vahidian và cộng sự [19] đã giới thiệu Personalized Federated Learning by Pruning (Sub-FedAvg). Sub-FedAvg áp dụng tìm kiếm một mạng con nhỏ cho mỗi nút bằng cách áp dụng phương pháp cắt tỉa hỗn hợp (kết hợp cắt tỉa có cấu trúc và không có cấu trúc). Thuật toán này đã chứng minh khả năng vượt trội so với các thuật toán hiện đại trên các tập dữ liệu CIFAR-10/100, MNIST, và EMNIST.

Hong và cộng sự [20] đã giới thiệu thuật toán tổng hợp có trọng số trong Học Liên kết (Federated Weighted Averaging - FedWAvg), nhằm giải quyết vấn đề quên ví dụ trong học liên kết. Trong các kịch bản mà dữ liệu giữa các nút không đồng nhất (non-IID) và mất cân bằng nhãn, các mô hình có xu hướng "quên" một số điểm dữ liệu sau khi tổng hợp toàn cục, đặc biệt là từ các nhãn ít được đại diện.

FedWAvg khắc phục điều này bằng cách xác định trọng số cho đóng góp của các nút dựa trên số lượng ví dụ dễ quên (các điểm dữ liệu bị quên sau quá trình tổng hợp) ở mỗi nút. Điều này đảm bảo rằng các nút có nhiều ví dụ dễ quên hơn sẽ được đại diện tốt hơn trong quá trình tổng hợp, giúp tạo ra một mô hình toàn cục giữ được nhiều kiến thức hơn từ các nút thường bị bỏ qua. Phương pháp này đã được thử nghiệm và cho thấy cải thiện hiệu suất trong các kịch bản không đồng nhất.

Trong các kịch bản có dữ liệu không đồng nhất, các đóng góp của các nút tham gia vào quá trình huấn luyện thường không đồng đều. Để giải quyết vấn đề này, Hongda Wu và cộng sự đã giới thiệu thuật toán Federated Adaptive Weighting (FedAdp) trong công trình của họ [14]. Thuật toán này động gán các trọng số khác nhau cho mỗi nút, dựa trên đóng góp của chúng trong mỗi vòng huấn luyện, để cập nhật mô hình toàn cục. Các tác giả chỉ ra rằng, khi sử dụng FedAdp, số vòng giao tiếp cần thiết có thể giảm tới 54,1% trên bộ dữ liệu MNIST và 45,4% trên bộ dữ liệu FashionMNIST so với thuật toán FedAvg.

Một nghiên cứu gần đây đề xuất một thuật toán trọng số mô hình mới, FedImp [15]. Mặc dù FedAdp mang lại những tiến bộ đáng kể, nó vẫn gặp phải một số hạn chế, đặc biệt là sự không nhất quán trong việc căn chỉnh gradient toàn cục với hướng mong đợi. Sự khác biệt này thường xuất hiện khi một số nút tham gia gặp phải dữ liệu thiếu nhãn đại diện, dẫn đến gradient cục bộ lệch hướng. FedImp được đề xuất nhằm khắc phục vấn đề này bằng cách áp dụng giá trị entropy cho mỗi máy huấn luyện, để đo lường mức độ non-IID của dữ liệu. Các thử nghiệm thực nghiệm của FedImp cho thấy hiệu suất vượt trội so với FedAdp và FedAvg trong phần lớn các trường hợp, với tốc độ hội tụ nhanh hơn tới 60% trên bộ dữ liệu EMNIST và 40% trên bộ dữ liệu CIFAR-10. Đặc biệt, trong các tình huống mất cân bằng dữ liệu cao, chẳng hạn như 3 máy IID và 7 máy non-IID, FedImp hoàn toàn vượt trội so với các thuật toán khác.

CHƯƠNG 3. SƠ SỞ LÝ THUYẾT

Trong bối cảnh chương này, các khái niệm chính về dữ liệu non-IID cũng như Học kết hợp được trình bày với mục tiêu cung cấp cho người đọc nền tảng vững chắc để hiểu các phần tiếp theo. Ngoài ra, các thành phần quan trọng của các thuật toán FedAvg cơ bản và FedImp cũng được giải thích, làm sáng tỏ cấu trúc cơ bản và cơ chế hoạt động của chúng.

3.1 Vấn đề mất cân bằng phân phối dữ liệu giữa các thiết bị

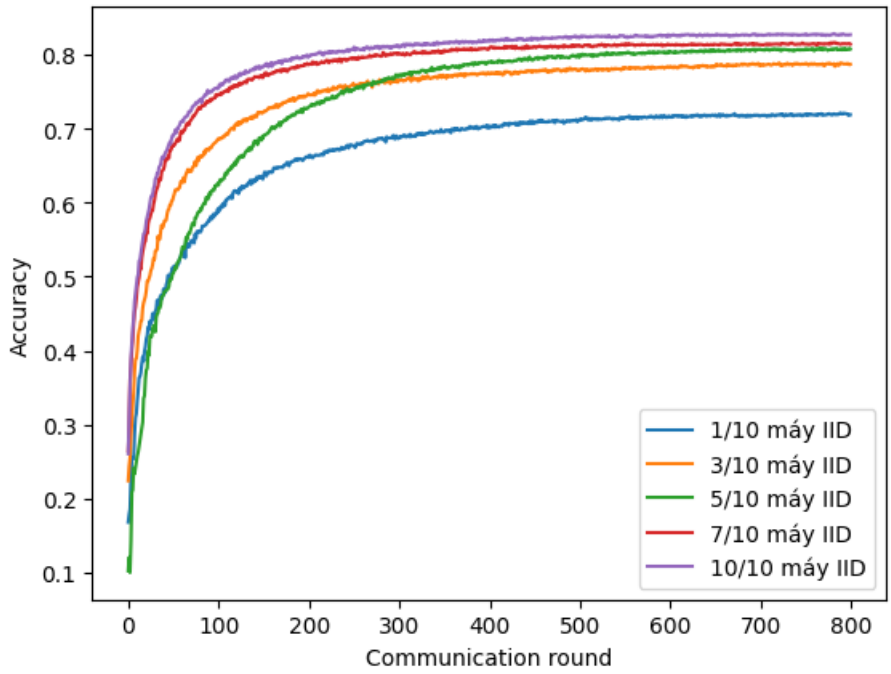
Mặc dù có tiềm năng cực kì hứa hẹn tuy nhiên mô hình Học kết hợp vẫn tồn tại một số các hạn chế. Trong đó một trong những vấn đề được quan tâm nhất là sự khác biệt về phân phối dữ liệu (Statistical Heterogeneity), hay non-IID (Non independent and identically distributed) giữa các nút. Sự khác biệt về phân phối dữ liệu thường đến từ khác biệt về mục đích sử dụng, cường độ sử dụng của người dùng cho đến những khác biệt về khoảng cách địa lý. Đây được coi là vấn đề cốt lõi trong các nghiên cứu về Học kết hợp vì nó làm ảnh hưởng trực tiếp đến hiệu suất của mô hình học máy trong quá trình huấn luyện, với việc các mô hình học máy thường mặc định rằng dữ liệu là IID. Đặc biệt trong thời đại mà các mô hình trí tuệ nhân tạo bị ảnh hưởng bởi việc ra quyết định dựa trên dữ liệu (data-driven decision-making), đây trở thành một "nút thắt cổ chai" của việc áp dụng Học kết hợp trong các bài toán thực tế.

Sự khác biệt về phân phối dữ liệu có thể là khác biệt thuộc tính, phân phối nhãn, khác nhau giữa các dữ liệu cùng nhãn, số lượng dữ liệu trên mỗi thiết bị, etc. Trong đó vấn đề thường hay xảy ra nhất là sự khác biệt về phân phối nhãn giữa các thiết bị với nhau hay còn gọi là dữ liệu non-IID (non independent and identically distributed). Sự khác biệt đến từ việc mỗi thiết bị có một mức độ sử dụng khác nhau, dẫn đến dữ liệu trên một thiết bị tồn tại số lượng dữ liệu của nhãn này nhiều hơn các nhãn khác hay thậm trí có nhãn hoàn toàn không có dữ liệu. Sự khác biệt này là không thể tránh khỏi trong thực tế.

Vấn đề non-IID đã dẫn đến việc độ chính xác của mô hình bị giảm cũng như tăng thời gian hội tụ của mô hình dẫn đến việc mô hình cần nhiều round huấn luyện để đạt được độ chính xác mong muốn, thậm chí là không thể hội tụ được trong các trường hợp có mức độ mất cân bằng rất lớn. Sự hội tụ kém xuất phát từ sự khác biệt của các mô hình được huấn luyện cục bộ, dẫn đến khó khăn trong việc hợp nhất chúng thành một mô hình toàn cầu thống nhất. Hiệu suất tổng quát kém càng làm trầm trọng thêm vấn đề, vì mô hình toàn cầu kết quả có thể gặp khó khăn trong việc nắm bắt những hiểu biết từ các phân phối dữ liệu đa dạng, làm cho nó kém

hiệu quả khi áp dụng vào các trường hợp dữ liệu chưa xuất hiện.

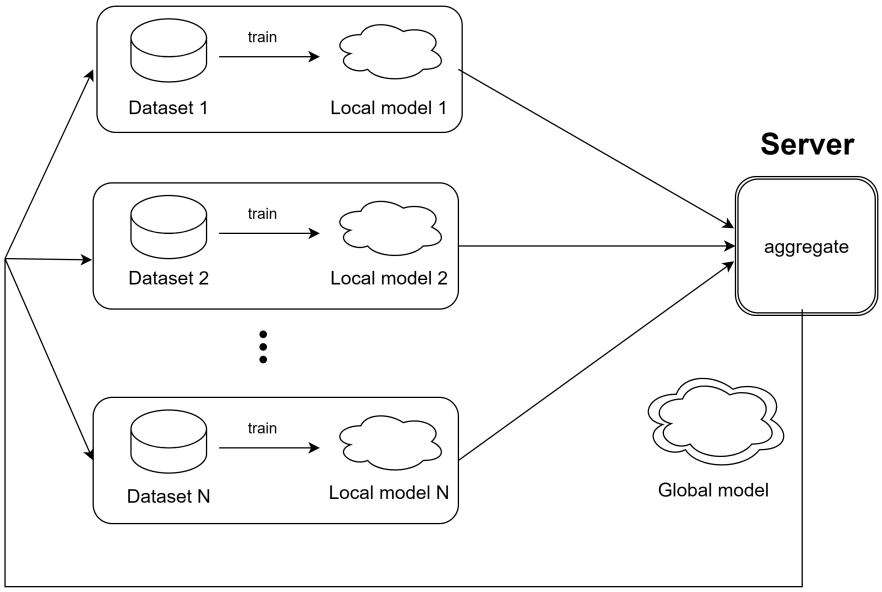
Trong bài báo gốc giới thiệu mô hình Học kết hợp [12], các tác giả đã thừa nhận ảnh hưởng của dữ liệu non-IID lên hiệu suất của mô hình Học kết hợp. Tuy nhiên họ vẫn tin rằng thuật toán FedAvg đơn giản vẫn có thể hoạt động đủ tốt trên dữ liệu non-IID. Tuy nhiên các kết quả thực nghiệm ở Hình 3.1 đã cho thấy rằng càng ít máy có phân phối đều thì độ chính xác cũng như mức độ hội tụ của mô hình càng kém, với chênh lệch của trường hợp có 1 máy IID và 10 máy IID có thể lên đến 10% độ chính xác.



Hình 3.1: So sánh hiệu suất của FedAvg với các mức độ non-IID khác nhau

3.2 Học kết hợp truyền thống

Quá trình Học kết hợp (FL) phát triển qua một chuỗi các bước cập nhật mô hình lặp đi lặp lại như mô tả trong hình 3.2. Ban đầu, máy chủ khởi tạo các tham số mô hình, thông qua khởi tạo ngẫu nhiên hoặc lấy từ một điểm lưu trữ đã có sẵn. Sau đó, máy chủ gửi các tham số mô hình toàn cục đến các nút nút được kết nối, đảm bảo sự đồng nhất trong điểm khởi đầu cho quá trình huấn luyện cục bộ trên tất cả các nút tham gia. Với các tham số mô hình toàn cục mới cập nhật, mỗi nút nút bắt đầu quá trình huấn luyện cục bộ bằng cách sử dụng dữ liệu đặc thù của mình. Đáng chú ý, quá trình huấn luyện cục bộ không kéo dài đến khi hội tụ hoàn toàn, mà chỉ thực hiện trong một vòng epoch hoặc một số batch nhỏ. Sau khi hoàn thành huấn luyện cục bộ, mỗi nút nút sở hữu một phiên bản mô hình được điều chỉnh nhẹ so với các tham số ban đầu, phản ánh sự khác biệt trong dữ liệu cục bộ của mình. Các tham số mô hình được cá nhân hóa này sau đó được truyền lại cho máy chủ, và máy



Hình 3.2: Quá trình Học kết hợp

chủ sẽ tổng hợp các tham số cục bộ này thành các tham số toàn cục. Quá trình lặp đi lặp lại này tiếp tục, dần dần tinh chỉnh các tham số mô hình toàn cục, cho đến khi đạt được mô hình hoàn toàn được huấn luyện.

Mục tiêu chính của quá trình FL là giảm thiểu sự khác biệt giữa mô hình được tổng hợp toàn cục và các mô hình cục bộ được phân phối trên nhiều nút phi tập trung. Điều này được thực hiện thông qua việc xây dựng một hàm mục tiêu, F(w), thường được biểu diễn dưới dạng tổng (hoặc trung bình) của các mục tiêu cục bộ từ mỗi nút tham gia:

$$F(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\mathbf{w}) \tag{3.1}$$

Trong phương trình này, N biểu thị tổng số các nút tham gia, và w biểu thị hàm mục tiêu cục bộ tại nút i. Hàm mục tiêu cục bộ w phụ thuộc vào hiệu suất của mô hình đối với dữ liệu được lưu giữ cục bộ, thường bao gồm một hàm mất mát đo lường sự chênh lệch giữa giá trị dự đoán của mô hình và giá trị thực tế trong tập dữ liệu cục bộ. Mục tiêu tổng thể là xác định các tham số mô hình tối ưu w để tối thiểu hóa hàm mục tiêu toàn cục này.

Trong bối cảnh FL tiêu chuẩn, các nút tham gia tiến hành huấn luyện cục bộ với các cấu hình nhất quán, bao gồm cả trình tối ưu hóa và tốc độ học. Ở mỗi vòng giao tiếp t, một tập hợp con gồm K nút ($K \leq N$) được chọn và mô hình toàn cục từ vòng giao tiếp trước đó được truyền đến các nút được chọn. Sau đó, mỗi nút tham gia i thực hiện huấn luyện bằng phương pháp gradient descent ngẫu nhiên (SGD)

để tối thiểu hóa mục tiêu cục bộ:

$$\mathbf{w}_i(t) = \mathbf{w}(t-1) - \eta \nabla F_i(\mathbf{w}(t-1)) \tag{3.2}$$

Trong phương trình này, η đại diện cho tốc độ học, và w(t-1) biểu thị gradient tại nút i.

Các thuật toán FL sau đó nhằm mục tiêu cập nhật các tham số mô hình toàn cục w bằng cách tổng hợp các cập nhật mô hình cục bộ (cả tham số hoặc gradient) từ mỗi nút:

$$\mathbf{w}(t) = \sum_{i=1}^{K} \psi_i \mathbf{w}_i(t) \tag{3.3}$$

Trong phương trình này, biểu thị trọng số được gán cho nút i. Các thuật toán khác nhau sử dụng các chiến lược gán trọng số khác nhau, và phần tiếp theo cung cấp minh họa chi tiết về quá trình xác định trọng số của nút, đặc biệt là các thuật toán FedAvg và FedAdp.

3.3 Thuật toán Federated Averaging (FedAvg)

Phần này cung cấp một khám phá chi tiết về các nguyên lý nền tảng của FedAvg [12]. Thuật toán FedAvg hoạt động dựa trên giả định rằng đóng góp của tất cả các nút tham gia là ngang nhau, với trọng số được gán cho mỗi nút trong quá trình tổng hợp toàn cục chỉ phụ thuộc vào kích thước tập dữ liệu huấn luyện của nó. Về mặt toán học, điều này được biểu diễn như sau:

$$\psi_i(t) = \frac{D_i}{\sum_{i'=1}^{K} Di'} \tag{3.4}$$

trong đó D_i đại diện cho số lượng mẫu huấn luyện trên nút i.

Mặc dù FedAvg đã chứng tỏ hiệu quả trong các kịch bản học phân tán, nó tồn tại một hạn chế rõ ràng là tốc độ hội tụ chậm và độ chính xác giảm, đặc biệt là khi xử lý dữ liệu không đồng nhất (non-IID) trên các nút. Do không phải tất cả các nút tham gia đều đóng góp dữ liệu huấn luyện có giá trị tương đương hoặc có tài nguyên tính toán tương tự, việc trung bình các tham số từ các nút tham gia có thể không phải lúc nào cũng hợp lý hoặc công bằng. Do đó, các phương pháp thay thế như trung bình trọng số hoặc tổng hợp có chọn lọc đang được nghiên cứu trong lĩnh vực FL để đảm bảo một đại diện công bằng hơn về sự đóng góp của các nút tham gia.

3.4 Thuật toán Federated Impurity Weighting (FedImp)

Công trình gần đây [15] đã đề xuất FedImp, một chiến lược gán trọng số bằng cách sử dụng giá trị entropy được tính cho mỗi nút. Kết quả đánh giá cho thấy rằng FedImp có thể tăng tốc độ hội tụ từ 25% đến 65% trên nhiều trường hợp thử nghiệm khác nhau.

Để cải thiện sự hội tụ của mô hình toàn cục, FedImp tính toán trọng số cho mỗi nút tham gia trong một vòng truyền thông nhằm điều chỉnh mức độ ảnh hưởng của nó lên mô hình toàn cục. Mức độ mất cân bằng giữa các nút được tính bằng thống kê entropy, là một thước đo thống kê về sự rối loạn của nhãn trong tập dữ liệu. Giá trị entropy S_i cho nút thứ i được tính như sau:

$$S_i = -\sum_{j=1}^{C} p_j \log_C P_j \tag{3.5}$$

trong đó p_j đại diện cho tỷ lệ của các nhãn của lớp j trong tổng số nhãn L_i của nút i. Sau đó, trong mỗi vòng giao tiếp t, trọng số của nút được tạo ra như sau:

$$\psi_i(t) = \frac{D_i e^{\frac{S_i}{\tau}}}{\sum_{i=1}^{\mathcal{K}} D_i e^{\frac{S_i}{\tau}}} \tag{3.6}$$

Chúng ta có thể thấy rằng nút có mức độ non-IID thấp hơn sẽ có giá trị entropy cao hơn và do đó có trọng số cao hơn. Tham số $\tau > 0$ được thêm vào để kiểm soát ảnh hưởng của các nút non-IID lên mô hình toàn cục. Một giá trị τ nhỏ làm nổi bật sự đóng góp của các nút có nhiều thông tin hơn trong khi giảm thiểu ảnh hưởng của các nút có ít thông tin hơn. Tuy nhiên, cần lưu ý rằng việc sử dụng giá trị nhỏ của τ không phải lúc nào cũng hiệu quả. Vì trong nhiều trường hợp, việc duy trì một mức độ ảnh hưởng nhất định từ các nút có mức độ non-IID thấp vẫn cần thiết, vì chúng đóng góp thông tin độc đáo mà các nút có dữ liệu cân bằng hơn có thể không có. Cuối cùng, ở giai đoạn tổng hợp của vòng huấn luyện thứ t, mô hình sẽ được tổng hợp như sau:

$$w(t) = \sum_{i=1}^{\mathcal{K}} \frac{1}{\psi_i} w_i(t) \tag{3.7}$$

Mô hình tổng quan của thuật toán FedImp có thể được biểu diễn trong sơ đồ thuật toán sau:

Thuật toán 1 mô tả chi tiết từng bước của thuật toán FedImp. FedImp hoạt động trong một số vòng giao tiếp được định trước, ký hiệu là T. Trong mỗi vòng giao

Algorithm 1: Federated Impurity Weighting Algorithm

Input: T: the number of communication rounds, N: the number of nodes in the system, K: the number of participating nodes in each round, B: the local minibatch size, E: the number of local epochs, η: the learning rate, λ: the control parameter

1 **Server executes**:
2 Initialize $\mathbf{w}(0)$;
3 **for** $t = 1, 2, \ldots, T$ **do**
4 Choose random set of K nodes;
5 **for** *each node* $i \in K$ *nodes* ***in parallel*** **do**
6 $\mathbf{w_i}(\mathbf{t}) \leftarrow \texttt{LocalUpdate}(i, \mathbf{w}(t-1))$
7 Calculate ψ_i (equations (3.5), (3.6));
8 $\mathbf{w}(t) = \sum_{i=1}^{K} \psi_i \mathbf{w}_i(t)$;
9 **Function** `LocalUpdate` $(i, \mathbf{w})$ **:**
10 // Run on node i;
11 **for** *e = 1, 2, ..., E* **do**
12 **for** *b = 1, 2, ...,* $\lceil \frac{D_i}{B} \rceil$ **do**
13 $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla F(\mathbf{w})$
14 **return** w;

tiếp, K nút được chọn ngẫu nhiên để tham gia. Thuật toán còn bao gồm các tham số như kích thước minibatch cục bộ B, số epoch cục bộ E, tốc độ học η, và tham số điều khiển τ. Mô tả chi tiết của thuật toán như sau:

- **Dòng 2:** Khởi tạo tham số $w(0)$ của mô hình toàn cục.
- **Dòng 3-10:** Trình bày vòng lặp chính cho các vòng giao tiếp t từ 1 đến T:
  - **Dòng 4:** Chọn ngẫu nhiên tập hợp K nút cho vòng hiện tại.
  - **Dòng 5, 6:** Duyệt qua từng nút i song song và cập nhật mô hình cục bộ của nó bằng hàm `LocalUpdate` (sẽ được mô tả chi tiết ở phần sau).
  - **Dòng 8:** Tính toán ψ_i theo các phương trình (6) và (7).
  - **Dòng 9:** Tổng hợp các mô hình cục bộ dựa trên trọng số ψ_i đã tính để cập nhật mô hình toàn cục $w(t)$.
- **Dòng 12-18:** Trình bày định nghĩa của hàm `LocalUpdate`, được chạy trên mỗi nút i:
  - **Dòng 13:** Vòng lặp qua các epoch cục bộ e.
  - **Dòng 14, 15:** Vòng lặp qua các minibatch cục bộ b, cập nhật tham số mô hình cục bộ w bằng phương pháp descent gradient ngẫu nhiên (SGD) với

kích thước bước η.

– **Dòng 18:** Trả về tham số mô hình cục bộ đã cập nhật.

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

Trong chương này, một phương pháp tổng hợp mô hình mới được giới thiệu để cải thiện tốc độ hội tụ của Học kết hợp. Đầu tiên, mục 4.1 phân tích các hạn chế của FedImp, từ đó thúc đẩy đề xuất một thuật toán cải tiến mới toàn diện hơn. Trong mục 4.2, các cải tiến của thuật toán được mô tả cùng với các nhận xét để minh chứng cho đề xuất.

4.1 Nhược điểm của thuật toán FedImp

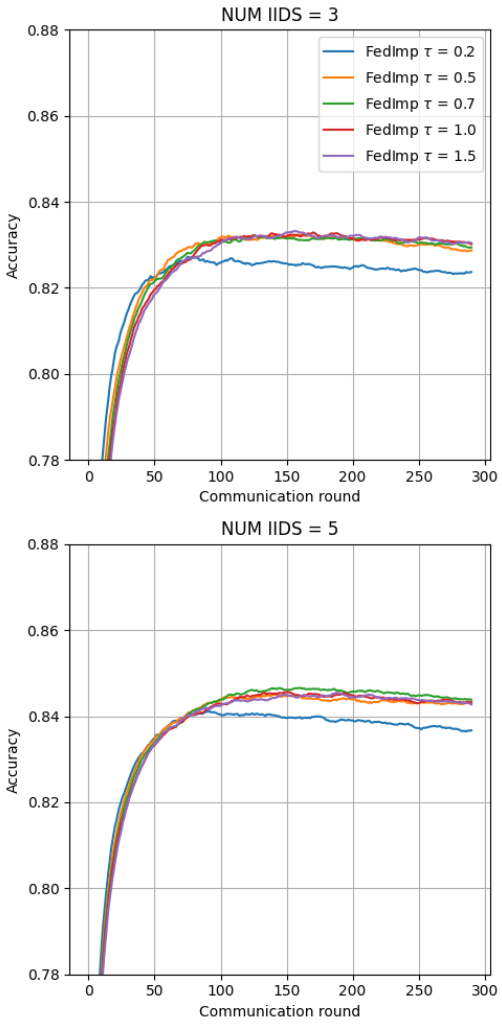
Mặc dù FedImp đã cải thiện đáng kể sự hội tụ của Học kết hợp, thuật toán này vẫn còn những điểm chưa hoàn thiện để đạt được kết quả tốt hơn. Hai yếu tố chính của FedImp có thể được cải tiến bao gồm:

- ***Phạm vi trọng số của các nút***: Trong thuật toán gốc, phạm vi của trọng số mà nút có thể nhận được được điều chỉnh thông qua tham số τ. Trong [15], giá trị này được cố định mà không quan tâm đến các đặc tính dữ liệu trên các nút hay giai đoạn huấn luyện. Điều này đặt ra một giả thuyết rằng thuật toán FedImp mặc dù có hiệu suất tốt nhưng chưa hoàn toàn tối ưu trên tất cả các kịch bản thực nghiệm.
- ***Ảnh hưởng của trọng số trong giai đoạn sau của quá trình huấn luyện***: Mức độ non-IID cao giữa các nút có thể dẫn đến việc một số nút có trọng số cao hơn nhiều so với những nút khác. Trong quá trình tổng hợp mô hình việc này có thể dẫn đến việc một hay vài mô hình cục bộ lấn át hoàn toàn các mô hình khác và khiến mô hình toàn cầu không có những tri thức học được từ các mô hình có trọng số bé hơn. Không những thế, trong các giai đoạn sau của quá trình huấn luyện, điều này có thể gây ra tác động tiêu cực đến mô hình toàn cầu khi khiến mô hình toàn cầu bị quá khớp với các mô hình cục bộ và mất đi tính tổng quát.

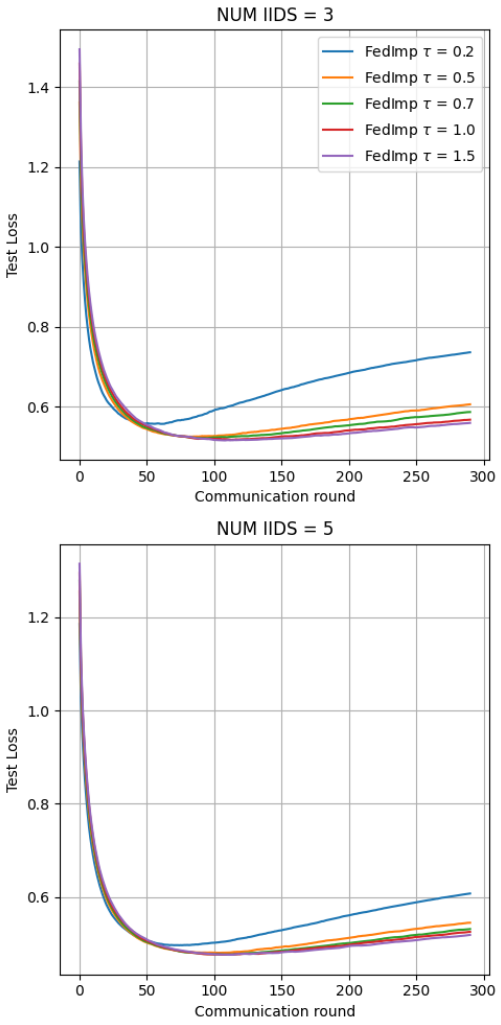
Hai yếu tố được nêu trên đều xoay quanh tham số τ được sử dụng trong thuật toán FedImp gốc. Trong công thức tính trọng số của FedImp gốc, như đã trình bày trong chương 3, tham số τ được sử dụng để điều chỉnh trọng số của các nút. Giá trị τ càng nhỏ, khoảng cách giữa các trọng số trong quá trình tổng hợp càng lớn. Giá trị τ quá lớn có thể khiến mô hình toàn cục bị lệch về dữ liệu của một số nút và làm mất tính tổng quát của mô hình. Ngược lại, giá trị τ quá nhỏ sẽ làm chậm sự hội tụ của mô hình toàn cục. Mặc dù vậy nhưng trong bài báo giá trị τ này được lựa chọn chỉ dựa trên thực nghiệm của một trường hợp non-IID duy nhất. Điều này tạo nên nghi vấn rằng liệu nó có phải là tối ưu nhất trong tất cả các trường hợp non-IID hay

không. Ngoài ra mặc dù các kết quả thực nghiệm cho thấy kết quả rất tốt tuy nhiên liệu thuật toán FedImp có thể đạt được kết quả tốt hơn nữa hay không? Điều này tạo động lực cho một thuật toán cải tiến bằng cách áp dụng các yếu tố động nhằm tăng tốc độ hội tụ lên hơn nữa trong quá trình huấn luyện.

Có thể thấy rằng thay vì chỉ tính toán trọng số cho các nút một lần dựa trên một giá trị τ cố định trong mọi trường hợp, thay vào đó giá trị τ trong DyFedImp sẽ được điều chỉnh trước khi huấn luyện một cách phù hợp dựa trên mức độ non-IID giữa các máy tham gia vào quá trình huấn luyện. Do đó giá trị τ khởi đầu sẽ là khác nhau trong các kịch bản non-IID khác nhau.



(a) Tốc độ hội tụ của FedImp với các giá trị τ khác nhau trong 2 kịch bản non-IID khác nhau



(b) Giá trị loss của FedImp với các giá trị τ khác nhau trong 2 kịch bản non-IID khác nhau

Hình 4.1: Hiệu suất của FedImp với các giá trị τ khác nhau trong 2 kịch bản non-IID khác nhau

Trong [15], nhiều giá trị τ đã được thử nghiệm để tìm giá trị tốt nhất cho thuật toán. Thực nghiệm này đã được tái thực thi lại trong hình 4.1 tuy nhiên ở đây có hai trường hợp non-IID khác nhau được xem xét. Có thể thấy từ hai trường hợp này

là giá trị $\tau = 0.7$ chỉ tối ưu với trường hợp 5 nút IID chứ không tối ưu với trường hợp 3 nút IID. Mặt khác cũng từ biểu đồ này, chúng ta có thể thấy rằng τ nhỏ cho thấy hiệu suất tốt hơn một chút ở các vòng đầu tiên ($\tau = 0.1, 0.2$) được thể hiện rõ nhất với trường hợp số nút IID bằng 3, tuy nhiên sự hội tụ giảm đáng kể ở các giai đoạn sau. Mặt khác các giá trị τ lớn hơn có hiệu năng tương đối tốt ở giai đoạn sau của quá trình huấn luyện ($\tau = 1.5, 1.0, 0.7$). Vấn đề quá khớp như nêu trên cũng được thể hiện khi quan sát biểu đồ loss của thuật toán trong hai trường hợp. Có thể thấy rằng giá trị τ càng nhỏ thì thuật toán càng bị quá khớp. Các ý này gợi lên sự khả thi của một cải tiến động cho FedImp với τ với khoảng trọng số của thuật toán được thu hẹp dần trong quá trình huấn luyện nhằm đạt được hiệu năng tối ưu hơn.

4.2 Thuật toán Federated Dynamic Impurity

Từ các nhận xét trong phần trước, đồ án này giới thiệu một biến thể của thuật toán FedImp, gọi là Federated Dynamic Impurity Weighting (DyFedImp), cho phép điều chỉnh trọng số của nút cho từng trường hợp thử nghiệm bằng cách xem xét phân phối của các nút khác cũng nhau điều chỉnh các giá trọng số này một cách linh hoạt xuyên suốt quá trình huấn luyện.

Mục tiêu của thuật toán bao gồm các ý chính sau đây:

- Cải thiến tốc độ hội tụ của mô hình so với thuật toán FedImp gốc cũng như các thuật toán tổng hợp mô hình khác trong các trường hợp dữ liệu non-IID khác nhau.
- Đảm bảo được tính tổng quát của mô hình toàn cầu so với thuật toán FedImp gốc.

Phạm vi Trọng số của nút. Để khắc phục giới hạn đầu tiên, thay vì cố định giá trị τ ở đầu giai đoạn huấn luyện, trước hết thuật toán sẽ tính toán giá trị Δ trước khi thực hiện huấn luyện công thức sau:

$$\Delta = \frac{\sigma(S_{i,i\in k}) + 0.01}{\overline{S_{i,i\in k}} + 0.01} \tag{4.1}$$

trong đó $\sigma(S_{i,i\in k})$ là giá trị độ lệch chuẩn của entropy của tập k các nút tham gia huấn luyện và $\overline{S_{i,i\in k}}$ biểu thị giá trị trung bình entropy của các nút này. Độ lệch chuẩn là thước đo sự biến đổi hoặc phân tán trong một tập hợp giá trị, được tính bằng công thức:

$$\sigma(S_{i,i\in k}) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(S_i - \overline{S_i})} \tag{4.2}$$

Nó định lượng mức độ mà các điểm dữ liệu trong một tập dữ liệu khác với giá trị

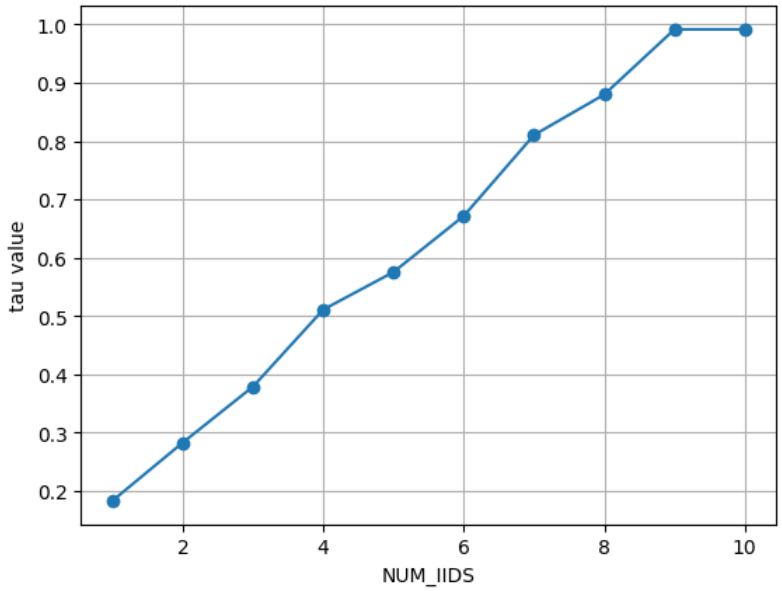
trung bình của tập hợp.

Phương trình 4.1 được lấy cảm hứng từ công thức hệ số tương quan của một tập dữ liệu. Nó đánh giá sự biến thiên tương đối của một tập dữ liệu so với giá trị trung bình của nó. Trong nghiên cứu, điều này đo lường mức độ non-IID tổng thể của nền tảng Học kết hợp bằng cách xem xét hai thuộc tính: mật độ phân phối của entropy và giá trị entropy trung bình trên tất cả các nút. Vì giá trị entropy mà mỗi nút có thể nhận có thể là 0, một kỹ thuật làm mượt được áp dụng bằng cách thêm giá trị hằng số 0.01 vào cả tử số và mẫu số của công thức.

Như vậy giá trị τ_0 được tính toán như sau:

$$\tau_0 = 1 - \Delta \tag{4.3}$$

Với cách tính này, mục tiêu của nghiên cứu là tìm giá trị τ_0 phù hợp trong phạm vi [0, 1]. Điều này có nghĩa là nếu mức độ biến thiên của giá trị entropy của các nút càng nhỏ, giá trị τ_0 sẽ càng nhỏ để mở rộng phạm vi giá trị của trọng số nút. Ngược lại, giá trị τ_0 lớn hơn sẽ được sử dụng để thu hẹp khoảng cách của các trọng số khi có sự biến động lớn giữa các entropy của nút. Sự thay đổi của giá trị τ_0 qua các trường hợp non-IID khác nhau có thể được biểu diễn như hình 4.2 dưới đây:



Hình 4.2: Sự thay đổi của giá trị τ_0 trong các trường hợp khác nhau

Ảnh hưởng của Trọng số trong Giai đoạn Sau. Để giải quyết vấn đề thứ hai, một cơ chế lấy cảm hứng từ giảm dần theo hàm mũ trong Học Máy. Giá trị τ sẽ không được giữ cố định xuyên suốt quá trình huấn luyện. Mục tiêu là giảm phạm vi của trọng số nút khi quá trình huấn luyện tiến triển. Điều này sẽ giảm nguy cơ mô hình bị quá khớp với một số nút và mất tính tổng quát. Đặc biệt là ở các vòng

sau khi mô hình gần với cực tiểu. Ngoài ra, điều này cũng giúp đồng bộ hóa quá trình học của mô hình khi sử dụng cơ chế giảm dần theo hàm mũ trong việc huấn luyện học máy.

Tuy nhiên, thay vì sử dụng một giá trị giảm dần cố định với mọi trường hợp như trong học máy, giá trị r cũng được thêm yếu tố động để điều chỉnh một cách phù hợp cho từng giai đoạn huấn luyện. Để đạt được điều này, giá trị r_i của vòng i được tính toán như sau:

$$r_i = r_0^{\tau_{i-1}} \tag{4.4}$$

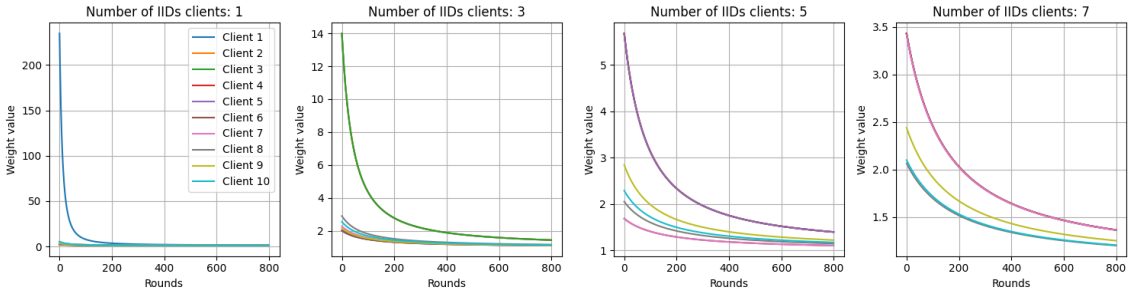
Ở đây r_0 là một giá trị cố định được xác định từ trước trong khoảng [0, 1], τ_{i-1} là giá trị sau khi đã được điều chỉnh ở vòng i-1. Sau mỗi vòng, sau khi tổng hợp mô hình thì giá trị τ_i sẽ được điều chỉnh như sau:

$$\tau_i = \frac{\tau_{i-1}}{r_i} \tag{4.5}$$

Như vậy nếu giá trị τ_{i-1} nhỏ thì giá trị r_i sẽ nhỏ và dẫn đến τ_i tăng nhanh và ngược lại. Cuối cùng công thức tổng hợp ở vòng thứ i được trình bày như sau:

$$\psi_i(t) = \frac{D_i e^{\frac{S_i}{\tau_i}}}{\sum_{i=1}^{\mathcal{K}} D_i e^{\frac{S_i}{\tau_i}}} \tag{4.6}$$

Với cơ chế điều chỉnh phạm vi trọng số mô hình như trên, độ tương quan giữa trọng số các mô hình trong các cấu hình khác nhau có thể được mô tả như Hình 4.5.

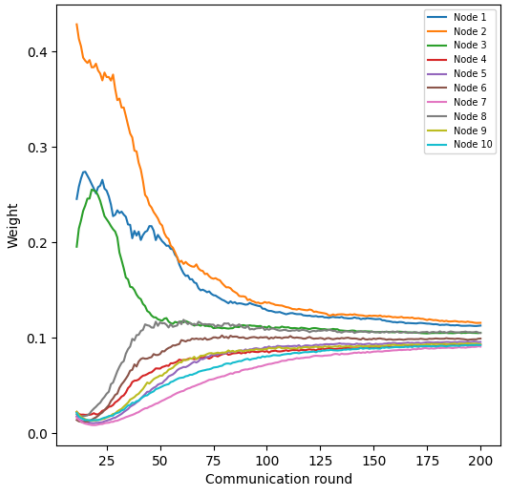


Hình 4.3: Sự thay đổi của các giá trị trọng số xuyên suốt quá trình huấn luyện trên các kịch bản mất cân bằng khác nhau

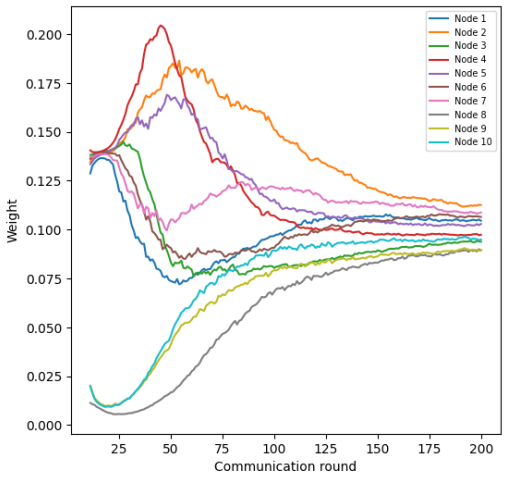
Nhìn vào hình trên, có thể thấy rằng mức độ mất cân bằng tổng thể các nút càng lớn (ít nút IID) thì trọng số có miền giá trị rất rộng ở thời điểm ban đầu, tuy nhiên miền giá trị này cũng sẽ được thu hẹp rất nhanh trong quá trình huấn luyện để mô

hình toàn cầu có thể học được đa dạng tri thức từ các máy nhất có thể và không bị mất tính tổng quát. Mặt khác khi nhiều nút IID thì khoảng trọng số ở thời điểm ban đầu sẽ nhỏ hơn và tốc độ giảm cũng chậm hơn xuyên suốt quá trình huấn luyện. Với cơ chế này, một giả định có thể được đạt ra là tại sao không đặt giá trị τ cố định rất nhỏ ở thời điểm ban đầu (ví dụ $\tau = 0.1$) cho tất cả các trường hợp. Mặc dù điều này trên lý thuyết có thể giúp tốc độ hội tụ được cải thiện tuy nhiên nó sẽ gần như loại bỏ hoàn toàn dóng góp của các nút non-IID. Đã có rất nhiều các nghiên cứu trước đây chứng minh rằng việc này có thể không tối ưu do nó có thể dẫn đến giảm sự đa dạng của dữ liệu tham gia vào quá trình huấn luyện, làm mô hình toàn cục trở nên kém khả năng tổng quát hóa với các phân phối dữ liệu thực tế hơn [21], [22], từ đó dẫn đến việc không đạt được hiệu suất mong muốn. Do đó giá trị τ sẽ chỉ nhỏ nếu mức độ cân bằng rất lớn khiến nó phải nhỏ để giữ cho mô hình toàn cầu không quá xa tối ưu hướng tối ưu.

Tính đúng đắn các cải tiến trên trong quá trình huấn luyện cũng được thể hiện ở việc nó khá tương đồng với trọng số của các máy tính được trong thuật toán FedAdp, khi khoảng trọng số ban đầu của các mô hình cục bộ cũng khác nhau trong quá trình huấn luyện và cũng giảm dần trong quá trình huấn luyện, như mô tả với 2 trường hợp khác nhau trong hình dưới đây:



(a) Trọng số của 10 nút trong trường hợp 3 nút dữ liệu cân bằng và 7 nút dữ liệu mất cân bằng



(b) Trọng số của 10 nút trong trường hợp 7 nút dữ liệu cân bằng và 3 nút dữ liệu mất cân bằng

Hình 4.4: Ví dụ về trọng số của các client được tính toán trong 2 kịch bản khác nhau

Tổng kết lại, quá trình hoạt động của thuật toán Federated Dynamic Impurity Weighting có thể được được mô tả như thuật toán 2 và hình **??** dưới đây:

Tương tự như thuật toán FedImp, DyFedImp hoạt động trong một số vòng giao tiếp được định trước, ký hiệu là T. Trong mỗi vòng giao tiếp, K nút được chọn ngẫu

Algorithm 2: Federated Dynamic Impurity Weighting Algorithm

Input: T: the number of communication rounds, N: the number of nodes in the system, K: the number of participating nodes in each round, B: the local minibatch size, E: the number of local epochs, η: the learning rate

1 Each client calculate the entropy value on its dataset and send the value to the server.
2 **Server executes**:
3 Calculate the intialize τ value using equations 4.1 and 4.3
4 Initialize $\mathbf{w}(0)$;
5 **for** $t = 1, 2, \ldots, T$ **do**
6 Choose random set of K nodes;
7 **for** *each node* $i \in K$ *nodes* ***in parallel*** **do**
8 $\mathbf{w_i(t)} \leftarrow \texttt{LocalUpdate}(i, \mathbf{w}(t-1))$
9 Update the τ value for next round using equations 4.4 and 4.5
10 Calculate ψ_i as in equation 4.6;
11 $\mathbf{w}(t) = \sum_{i=1}^{K} \psi_i \mathbf{w}_i(t)$;
12 **Function** `LocalUpdate`$(i, \mathbf{w})$ **:**
13 // Run on node i;
14 **for** *e = 1, 2, ..., E* **do**
15 **for** *b = 1, 2, ...,* $\lceil \frac{D_i}{B} \rceil$ **do**
16 $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla F(\mathbf{w})$
17 **return** $\mathbf{w}$;

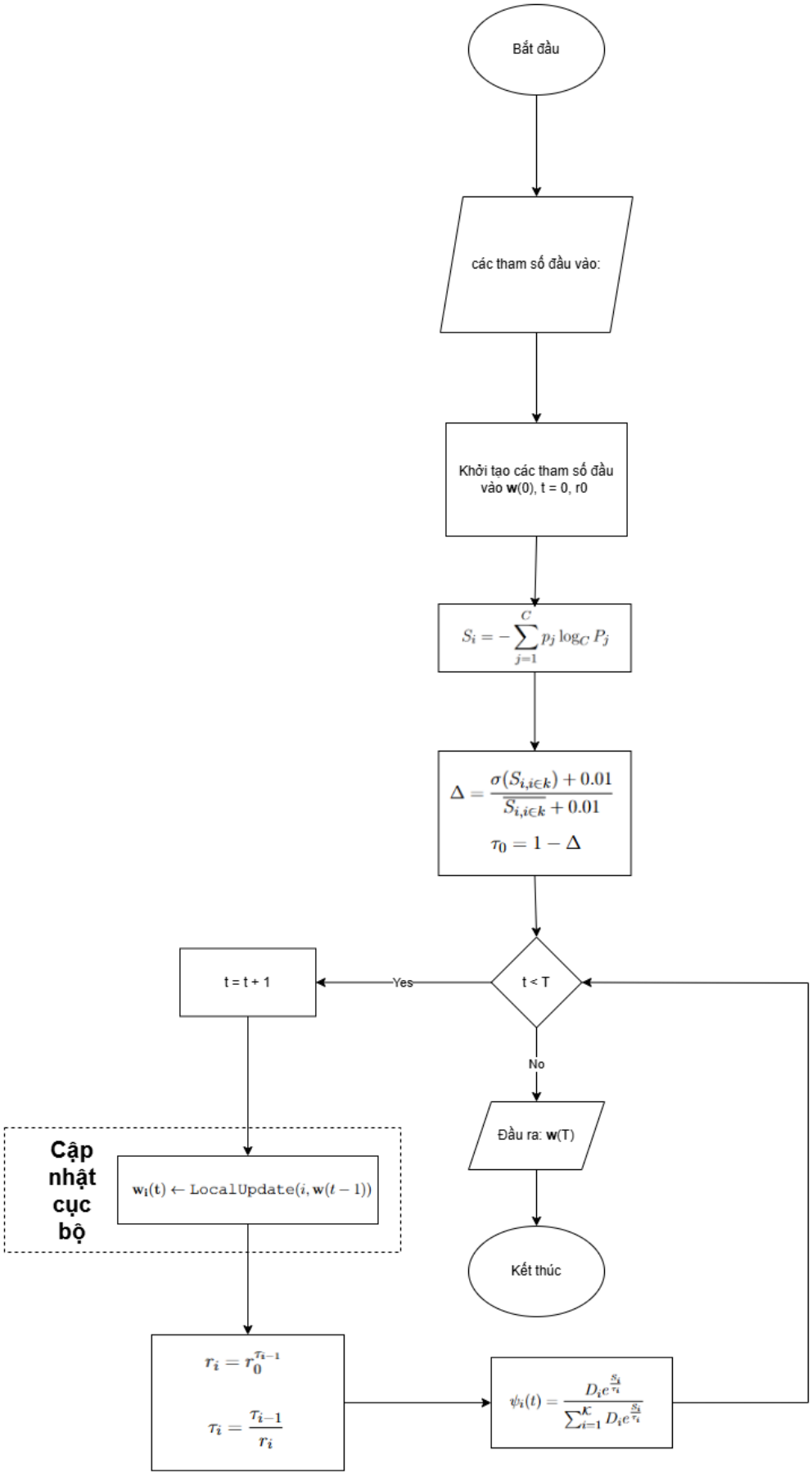
nhiên để tham gia. Thuật toán còn bao gồm các tham số như kích thước minibatch cục bộ B, số epoch cục bộ E, tốc độ học η, và tham số điều khiển τ. Mô tả chi tiết của thuật toán như sau:

- **Dòng 1:** Các nút trong quá trình khai báo bản thân với môi trường Học kết hợp cần phải tính toán giá trị entropy trên bộ dữ liệu nó sở hữu và gửi lại giá trị này về server
- **Dòng 3:** Dựa trên các giá entropy được gửi về, server sẽ tính toán giá trị τ khởi đầu cho quá trình huấn luyện.
- **Dòng 4:** Khởi tạo tham số $w(0)$ của mô hình toàn cục.
- **Dòng 5-11:** Trình bày vòng lặp chính cho các vòng giao tiếp t từ 1 đến T:
  - **Dòng 6:** Chọn ngẫu nhiên tập hợp K nút cho vòng hiện tại.
  - **Dòng 7, 8:** Duyệt qua từng nút i song song và cập nhật mô hình cục bộ của nó bằng hàm `LocalUpdate` (sẽ được mô tả chi tiết ở phần sau).

  - **Dòng 9:** Cập nhật giá trị τ cho vòng huấn luyện tiếp theo Tính toán trọng số cho từng nút ψ_i.
  - **Dòng 10, 11:** Tổng hợp các mô hình cục bộ dựa trên trọng số ψ_i đã tính để cập nhật mô hình toàn cục $w(t)$.

- **Dòng 12-17:** Trình bày định nghĩa của hàm `LocalUpdate`, được chạy trên mỗi nút i:
  - **Dòng 14:** Vòng lặp qua các epoch cục bộ e.
  - **Dòng 15, 16:** Vòng lặp qua các minibatch cục bộ b, cập nhật tham số mô hình cục bộ w bằng phương pháp descent gradient ngẫu nhiên (SGD) với kích thước bước η.
  - **Dòng 17:** Trả về tham số mô hình cục bộ đã cập nhật.

Lưu ý rằng trong nghiên cứu này giả định rằng các nút sẽ không xuất hiện thêm dữ liệu mới trong quá trình huấn luyện.



Hình 4.5: Sơ đồ khối quy trình hoạt động của thuật toán DyFedImp

CHƯƠNG 5. ĐÁNH GIÁ THỰC NGHIỆM

Chương này nhằm khám phá các kết quả số liệu thu được từ việc thực nghiệm phương pháp được đề xuất. Chương bắt đầu với phần phân tích chi tiết về các tập dữ liệu được sử dụng trong thực nghiệm, các tham số sử dụng để đánh giá, và phương pháp dùng để so sánh và đánh giá phương pháp đề xuất. Tiếp theo đó là các biểu đồ và bảng mô tả kết quả thực nghiệm đi kèm với các phân tích số liệu, cung cấp những quan sát có giá trị về hiệu suất và hiệu quả của phương pháp được đề xuất.

5.1 Bộ dữ liệu thực nghiệm

Đồ án này được thúc đẩy bởi các tác vụ phân loại hình ảnh nói riêng và thậm trí và các tác vụ phân loại rời rạc nói chung, với mục tiêu tổng quát là cải thiện đáng kể tính khả dụng của các thiết bị di động. Hai tập dữ liệu ảnh đa dạng được sử dụng, bao gồm EMNIST [23] và CIFAR-10 [24]. Mỗi tập dữ liệu phục vụ một mục đích riêng biệt và đặt ra các thách thức độc đáo, góp phần vào một cuộc điều tra toàn diện về hiệu quả và khả năng thích ứng của thuật toán đề xuất trên các lĩnh vực khác nhau.

- **EMNIST**: Tập dữ liệu EMNIST bao gồm các ký tự số viết tay được chuyển đổi sang định dạng ảnh tiêu chuẩn 28x28 pixel, phù hợp với cấu trúc của tập dữ liệu MNIST [25]. Tập dữ liệu EMNIST cung cấp sáu phân chia khác nhau, và trong thí nghiệm này, phân chia EMNIST Balanced được chọn, chứa tập hợp 47 ký tự với số lượng mẫu bằng nhau cho mỗi lớp. Tập huấn luyện bao gồm tổng cộng 112,800 mẫu, trong khi tập kiểm tra bao gồm 18,800 mẫu.
- **CIFAR-10**: Tập dữ liệu CIFAR-10 bao gồm 60,000 hình ảnh màu, mỗi hình ảnh có kích thước 32x32 pixel, được phân bố trên 10 lớp khác nhau với 6,000 hình ảnh cho mỗi lớp. Tập dữ liệu này được chia thành 50,000 hình ảnh huấn luyện và 10,000 hình ảnh kiểm tra.

5.2 Giả lập cấu hình non-IID trong FL

Các thực nghiệm của nghiên cứu được kế thừa và phát triển từ hai bài báo [15] và [14].

5.2.1 Cách chia dữ liệu

Để mô phỏng một tập hợp đa dạng giữa các nút nút, phương pháp tương tự như trong [16] được sử dụng với một số sửa đổi. Mỗi nút có một phân phối đa thức liên kết với các lớp, được lấy mẫu từ phân phối Dirichlet đối xứng, $q \sim \text{Dir}(\theta)$. $\theta > 0$ đóng vai trò là tham số tập trung, kiểm soát mức độ cân bằng giữa các lớp. Giá trị θ đủ lớn sẽ tạo ra một tập dữ liệu cân bằng, trong khi giá trị θ đủ nhỏ sẽ dẫn đến

tập dữ liệu mất cân bằng. Khi θ tăng đến vô cực, tất cả các nút thể hiện các phân phối giống nhau; ngược lại, khi θ tiến về 0, mỗi nút chỉ có các ví dụ từ một lớp duy nhất được chọn ngẫu nhiên.

Algorithm 3: Data partition

Input: X, Y, θ_1, θ_2, M

for $i = 1, 2, \ldots, X + Y$ **do**
  if $i \leq X$ **then**
    Sample $q \sim \text{Dir}(\theta_1, C)$;
  else
    Sample $q \sim \text{Dir}(\theta_2, C)$;
  $D_i = \emptyset$;
  for $j = 1, 2, \ldots, M$ **do**
    Sample $y \in C$ with probability q;
    Sample randomly $x \in S_y$;
    $D_i = D_i \cup (x, y)$;
    $S_y = S_y \setminus (x, y)$;
    if $|S_y| = 0$ **then**
      $C \setminus y$;
      $q \leftarrow \texttt{ReNormalize}(q, y)$;

Function `ReNormalize` $(q = (p_1, p_2, ..., p_C), y)$ **:**
  $p_y = 0$;
  $a = \sum_{i=1}^{C} p_i$;
  $q = q/a$
  return q;

Thuật toán 3 mô tả quy trình phân chia dữ liệu cho các tập dữ liệu với số lượng nút dữ liệu cân bằng (X) và không cân bằng (Y) được chỉ định. Số lượng mẫu trên mỗi nút là (M). Việc tạo dữ liệu bao gồm việc lấy mẫu xác suất lớp từ phân phối Dirichlet với các tham số tập trung θ_1 và θ_2 cho các nút cân bằng và không cân bằng, tương ứng. Đối với mỗi nút i từ 1 đến $X + Y$, các xác suất lớp q được lấy mẫu. Nếu i nhỏ hơn hoặc bằng X, q được lấy mẫu từ phân phối Dirichlet với tham số tập trung θ_1 và C loại lớp; nếu không, q được lấy mẫu với tham số tập trung θ_2. Đối với mỗi nút i, lặp qua M mẫu với các bước sau: lấy mẫu nhãn lớp y dựa trên phân phối xác suất q, chọn ngẫu nhiên một mẫu (x, y) từ các mẫu còn lại của lớp y trong tập dữ liệu S, thêm mẫu (x, y) vào tập dữ liệu D_i, và loại bỏ mẫu đã chọn khỏi tập dữ liệu S_y của lớp y. Nếu không còn mẫu nào cho lớp y, loại bỏ y khỏi tập các lớp C và chuẩn hóa lại xác suất lớp q. Hàm `ReNormalize` đảm bảo rằng xác suất lớp q được chuẩn hóa sau khi loại bỏ một lớp. Phương pháp phân chia này đảm bảo rằng tất cả các mẫu từ tập dữ liệu gốc được lấy mẫu mà không có sự trùng lặp

giữa các nút. Đối với các thực nghiệm trong phạm vi đồ án này, các giá trị được sử dụng là $\theta_1 = 100$, $\theta_2 = 0.01$, toàn bộ tập dữ liệu được sử dụng, đảm bảo rằng mỗi nút được phân bổ $M = \frac{S}{X+Y}$ mẫu.

5.2.2 Trường hợp thực nghiệm

Để so sánh thuật toán DyFedImp với các thuật toán khác, các kịch bản với các độ phức tạp khác nhau trong phân phối dữ liệu được thiết kế. Các biến thể bao gồm sự thay đổi trong số lượng nút với dữ liệu cân bằng và nút với dữ liệu mất cân bằng. Đối với mỗi tập dữ liệu, bốn kịch bản thử nghiệm được tạo ra để đánh giá hiệu suất, bao gồm:

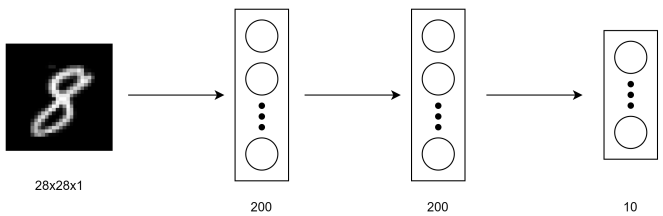
- 7 nút dữ liệu cân bằng + 3 nút dữ liệu mất cân bằng
- 5 nút dữ liệu cân bằng + 5 nút dữ liệu mất cân bằng
- 3 nút dữ liệu cân bằng + 7 nút dữ liệu mất cân bằng
- 1 nút dữ liệu cân bằng + 9 nút dữ liệu mất cân bằng

Các kịch bản được tạo ra sử dụng thuật toán như đã nêu ở phần trên, với nút dữ liệu cân bằng có giá trị $\theta_1 = 100$ và $\theta_2 = 0.01$. Các cấu hình thực nghiệm này được kế thừa từ bài báo [14] và [15]. Các thực nghiệm này giúp đánh giá một cách hiệu quả mô hình bằng cách loại bỏ các yếu tố khách quan như cách chọn tập nút để huấn luyện.

5.3 Các mô hình và tham số thực nghiệm

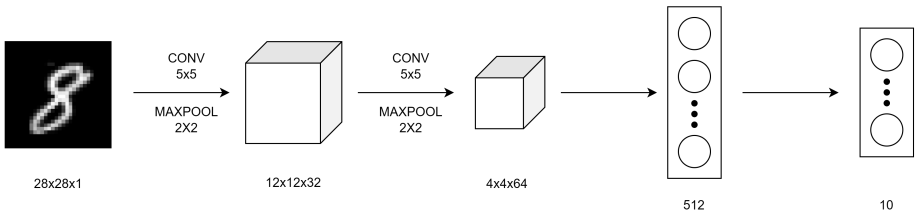
5.3.1 Kiến trúc mô hình

Đối với tập dữ liệu EMNIST, mỗi bộ dữ liệu sử dụng hai kiến trúc mô hình khác nhau, bao gồm: (i) Một mô hình MLP như hình 5.1 với hai lớp ẩn fully connected, mỗi lớp có 200 đơn vị, và một lớp đầu ra softmax cuối cùng, cùng với hàm kích hoạt ReLU áp dụng giữa mỗi lớp. (ii) Một mô hình CNN như hình 5.10 với hai lớp convolutional 5x5 (lớp đầu tiên có 32 kênh, lớp thứ hai có 64 kênh, mỗi lớp theo sau là lớp max pooling 2x2), một lớp fully connected với 512 đơn vị và hàm kích hoạt ReLU, và một lớp đầu ra softmax cuối cùng.



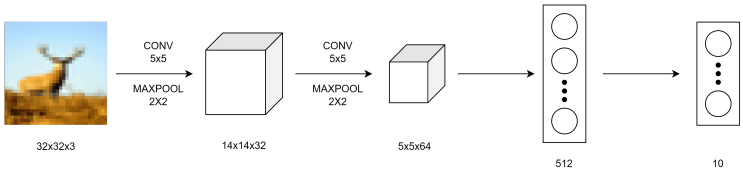
Hình 5.1: Kiến trúc MLP sử dụng với bộ dữ liệu EMNIST

Đối với tập dữ liệu CIFAR-10, có hai mô hình khác nhau được triển khai,

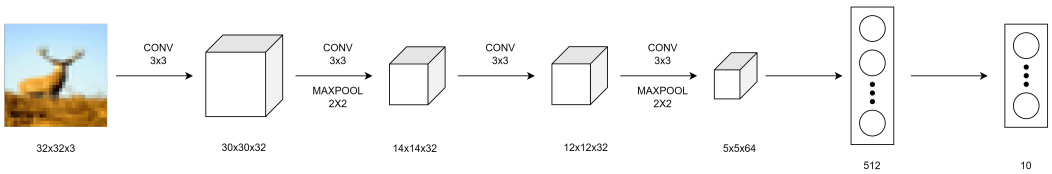


Hình 5.2: Kiến trúc CNN sử dụng với bộ dữ liệu EMNIST

bao gồm: (i) Một kiến trúc CNN với 2 lớp tích chập như hình 5.11 với hai lớp convolution 5x5 (lớp đầu tiên có 32 kênh, lớp thứ hai có 64 kênh, mỗi lớp theo sau là lớp max pooling 2x2), một lớp fully connected với 512 đơn vị và hàm kích hoạt ReLU, và một lớp đầu ra softmax cuối cùng. (ii) Một kiến trúc CNN với 4 lớp tích chập như hình 5.12 với bốn lớp convolution 3x3 với 32, 32, 64, 64 kênh tương ứng và hàm kích hoạt ReLU. Lớp thứ hai và lớp cuối cùng của convolution đi kèm với max pooling 2x2. Hai lớp fully connected tiếp theo với 512, 128 đơn vị và một lớp đầu ra softmax cuối cùng.



Hình 5.3: Kiến trúc 2-layer CNN trên bộ dữ liệu CIFAR-10



Hình 5.4: Kiến trúc 4-layer CNN trên bộ dữ liệu CIFAR-10

Tất cả các lớp fully connected trong 4 mô hình nêu trên đều được sử dụng các lớp dropout với tỷ lệ dropout 40% nhằm giảm sự quá khớp và hàm kích hoạt ReLU.

5.3.2 Các tham số mô hình và thuật toán thực nghiệm

Tiền xử lý ảnh: Trước khi huấn luyện mô hình, một số phép biến đổi ảnh được thực hiện: (i) EMNIST: chuẩn hóa các mảng ảnh để có giá trị trung bình là 0.5 và độ lệch chuẩn là 0.5, (ii) CIFAR-10: bao gồm việc cắt ngẫu nhiên kích thước 32 với phần đệm 4 pixel, lật ngang ngẫu nhiên và chuẩn hóa các mảng ảnh để có giá trị trung bình là 0.5 và độ lệch chuẩn là 0.5.

Chiến lược huấn luyện: Các thí nghiệm sử dụng tất cả các nút có sẵn để huấn

luyện ở mỗi vòng (tức là tỉ lệ tham gia huấn luyện là 1.0). Trong mỗi vòng, các nút huấn luyện các mô hình cục bộ của mình trong một epoch duy nhất, sử dụng kích thước batch bằng 100. Quá trình huấn luyện này sử dụng thuật toán tối ưu hóa Stochastic Gradient Descent (SGD) để giảm thiểu mất mát cross-entropy. Tốc độ học ban đầu được cấu hình khác nhau cho các tập dữ liệu trong các thí nghiệm được đặt là 0.1 cho cả hai tập dữ liệu EMNIST và CIFAR-10. Trong tất cả các trường hợp, tỷ lệ giảm tốc độ học là 0.995 được áp dụng sau mỗi vòng truyền thông. Đối với các thuật toán tổng hợp mô hình, các tham số đầu vào cho mỗi thuật toán sẽ được lấy dựa trên giá trị được các tác giả gốc xem là tối ưu nhất cho mỗi thuật toán. Sau mỗi vòng, hiệu suất của mô hình toàn cục được đánh giá bằng cách sử dụng bộ dữ liệu kiểm tra có sẵn.

Thuật toán so sánh: Bên cạnh các thuật toán là cảm hứng trực tiếp cho thuật toán đề xuất như FedAvg, FedAdp, FedImp, các thực nghiệm bổ sung cũng được thực hiện trên một số thuật toán tổng hợp mô hình phổ biến như FedProx[26], FedOpt(FedAdam, FedYogi, FedAdagrad)[16], FedAvgM[13] nhằm cung cấp một cái nhìn tổng quan về mô hình đề xuất.

5.3.3 Các độ đo

Để so sánh hiệu năng của các thuật toán, trong bài toán cụ thể ở đây là tốc độ hội tụ, các độ đo được sử dụng sẽ là số vòng huấn luyện truyền thống để đạt được độ chính xác mục tiêu trên tập test của mỗi bộ dữ liệu. Trong đó độ chính xác mục tiêu trong mỗi kịch bản sẽ được xác định dựa trên độ chính xác mà thuật toán FedAvg đạt được do đây là thuật toán cơ sở của Học kết hợp.

5.4 Kết quả thực nghiệm và các đánh giá

5.4.1 Đánh giá hiệu suất của DyFedImp với tham số r khác nhau

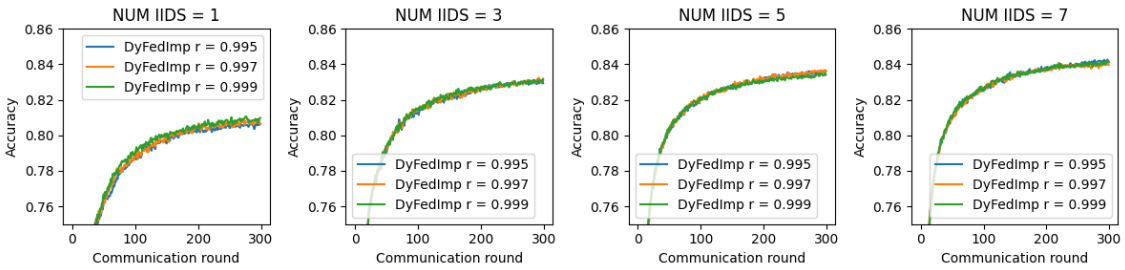
Các hình 5.5, 5.6, 5.7 và 5.8 dưới đây lần lượt biểu diễn hiệu suất của thuật toán DyFedImp sử dụng tham số giảm khoảng trọng số r trên các mô hình và bộ dữ liệu khác nhau và trên các kịch bản non-IID khác nhau. Có 3 giá trị được thực nghiệm nhằm đánh giá bao gồm: 0.995, 0.997 và 0.999.

Từ các kết quả thực nghiệm, ta có một số nhận xét sau đây:

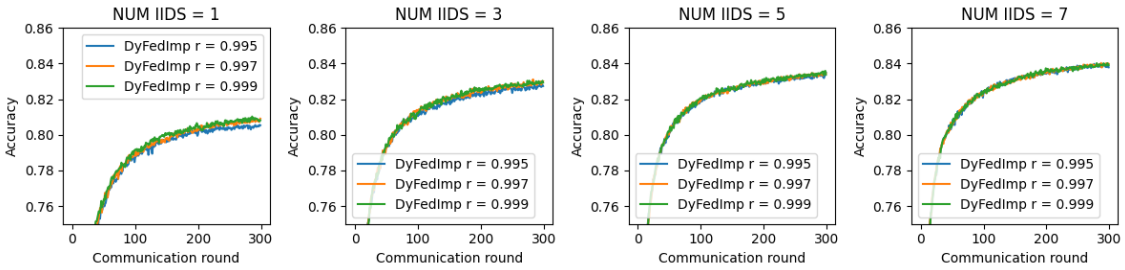
- Đối với bộ dữ liệu EMNIST, có thể thấy rằng sự khác biệt mà các giá trị r ảnh hưởng đến hiệu suất của mô hình là không đáng kể. Điều này có thể được giải thích phần nào là nhờ sự đơn giản về dữ liệu khi EMNIST là tập các ảnh đen trắng.
- Mặt khác đối với bộ dữ liệu CIFAR-10, có thể thấy giá trị r ảnh hưởng tương đối rõ ràng lên hiệu suất của mô hình. Đối với mô hình 2-layer CNN, có thể

thấy rằng r = 0.997 có hiệu suất tốt trong phần lớn các trường hợp. Tuy nhiên r = 0.999 lại có hiệu suất tốt hơn trên mô hình 4-layer CNN. Điều này gợi ý rằng có mối quan hệ giữa giá trị này với kích thước và số lớp của mô hình.

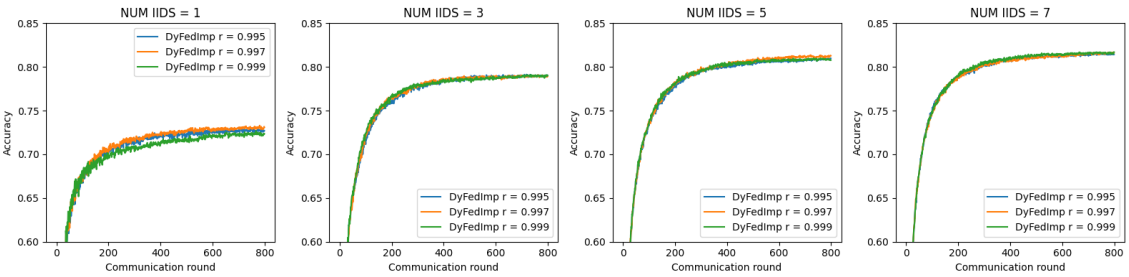
- Tổng kết lại các giá trị r lần lượt là 0.999, 0.999, 0.997 và 0,999 sẽ được áp dụng tương ứng với các mô hình MLP, CNN, 2-layer CNN và 4-layer CNN trong các thực nghiệm sau này.



Hình 5.5: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu EMNIST và mô hình MLP



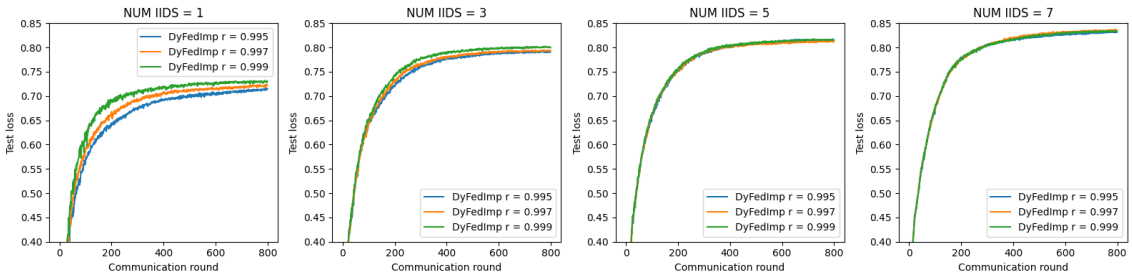
Hình 5.6: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu EMNIST và mô hình CNN



Hình 5.7: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu CIFAR-10 và mô hình 2-layer CNN

5.4.2 Tốc độ hội tụ của thuật toán đề xuất so với các thuật toán khác

Phần này sẽ so sánh và đánh giá tốc độ hội tụ của thuật toán hội tụ với các thuật toán tổng hợp mô hình khác trên các kịch bản non-IID khác nhau. Các thuật toán được so sánh như đã nêu ở phần trên bao gồm FedImp, FedAdp, FedAvg, FedOpt(FedAdam, FedYogi, FedAdagrad), FedAvgM, FedProx.

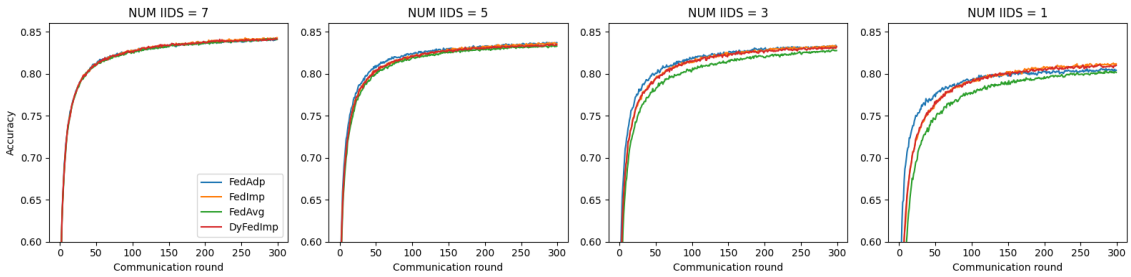


Hình 5.8: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu CIFAR-10 và mô hình 4-layer CNN

a, Thực nghiệm trên bộ dữ liệu EMNIST

Hình 5.9 và bảng 5.1 mô tả hiệu suất của các thuật toán đã liệt kê ở trên trong 300 vòng huấn luyện toàn cầu trên mô hình MLP. Từ các kết quả có thể rút ra một số nhận xét như sau:

- Nhìn chung có thể thấy rằng không có nhiều sự chênh lệch về số vòng để đạt được ngưỡng hội tụ cũng như độ chính xác giữa ba thuật toán FedImp, FedAdp và DyFedImp trong hầu hết các kịch bản thực nghiệm
- Đối với kịch bản đầu tiên là 7 IID + 3 non-IID, hai thuật toán là DyFedImp và FedImp có tốc độ hội tụ gần như tương đồng nhau (221 và 225) và nhanh hơn các thuật toán khác từ 25 đến 70 vòng huấn luyện với độ chính xác mục tiêu là 84%.
- Trong kịch bản thứ 2 là 5 IID + 5 non-IID, thuật toán FedAdp vượt trội các thuật toán khác khi đạt được độ chính xác mục tiêu là 83% sau 145 vòng, nhanh hơn các thuật toán khác từ 10 đến 50 vòng huấn luyện.
- Đối với kịch bản là 3 IID + 7 non-IID, hai thuật toán DyFedImp và FedImp vẫn thể hiện sự tương đồng với nhau và kém thuật toán FedAdp để đạt được độ chính xác mục tiêu là 82%.
- Kịch bản cuối cùng là 1 IID + 9 non-IID, ba thuật toán là DyFedImp, FedImp và FedAdp có thời gian đạt được độ chính xác mục tiêu 80% là tương đồng, lần lượt là 139, 140, 141 vòng huấn luyện. Trong khi đó các thuật toán còn lại mất đến hơn 200 vòng hoặc thậm trí không thể đạt được độ chính xác này.
- Nhìn chung có thể thấy rằng trên bộ dữ liệu EMNIST và mô hình MLP, ba thuật toán đánh trọng số mô hình đã thể hiện được sự vượt trội của mình so với các thuật toán còn lại. Tuy nhiên ở đây ngoài thuật toán FedAdp vượt trội trong 2/4 trường hợp thì hai thuật toán còn lại gần như tương đồng với nhau.



Hình 5.9: So sánh hiệu suất của các thuật toán với các kịch bản non-IID khác nhau trên bộ dữ liệu EMNIST và mô hình MLP

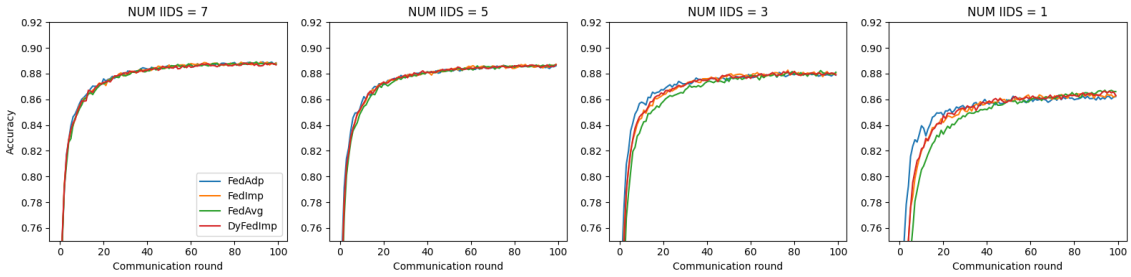
Algorithm	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	257	204	181	237
FedImp	225	157	125	140
FedAdp	290	**145**	**102**	141
DyFedImp	**221**	174	126	**139**
FedProx	271	206	175	234
FedAdam	N/A	N/A	N/A	N/A
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	N/A	728	N/A	N/A
FedAvgM	280	N/A	182	264

Bảng 5.1: Số vòng huấn luyện để đạt được độ chính xác mục tiêu của các thuật toán trên bộ dữ liệu EMNIST và mô hình MLP

Tương tự như trên, hình 5.10 và bảng 5.2 mô tả hiệu suất của các thuật toán trong 100 vòng huấn luyện toàn cầu đối với mô hình CNN. Từ các kết quả có thể rút ra một số nhận xét như sau:

- Nhìn chung có thể thấy rằng không có nhiều sự chênh lệch về số vòng để đạt được ngưỡng hội tụ cũng như độ chính xác của các thuật toán trong các kịch bản mất cần bằng dữ liệu. Điều này có thể giải thích là do sự đơn điệu của dữ liệu và sự phức tạp của mô hình kết hợp lại. Ngoại lệ duy nhất có lẽ là hai thuật toán FedAdam và FedYogi có hiệu suất khá tệ.
- Đối với kịch bản đâu tiên là 7 IID + 3 non-IID, hầu hết các thuật toán đều hội tụ trong khoảng từ vòng 30 đến vòng 35. Chỉ số FedImp hội tu sớm hon ở vòng 29, về cơ bản là không đáng kể. Độ chính xác mà các thuật toán đạt được nằm ở ngưỡng 88 - 89%. Trong trường hợp này thuật toán đề xuất chỉ hội tụ chậm hơn 1 vòng tuy nhiên độ chính xác đạt được lại kém hơn.
- Trong kịch bản thứ 2 là 5 IID + 5 non-IID, các thuật toán có ngưỡng hội tụ trong khoảng từ 35 đến 40 vòng huấn luyện, với thuật toán đề xuất DyFedImp và FedAvgM đều hội tụ sớm nhất ở ngưỡng 35 vòng Đối với độ chính xác thì hầu như không có chênh lệnh gì đáng kể, nằm trong khoảng 88%.

- Đối với kịch bản là 3 IID + 7 non-IID, đã có một số chênh lệch rõ rệt về hiệu năng của các thuật toán. Trong trường hợp này FedImp hội tụ khi mất 64 vòng để đạt được 88% độ chính xác, thứ hai là DyFedImp với 71 vòng. Một lần nữa không có điều gì đáng chú ý về độ chính xác đạt được của các thuật toán, chủ yếu nằm trong khoảng 88%.
- Kịch bản cuối cùng là 1 IID + 9 non-IID, có thể thấy trong kịch bản này DyFedImp đã đạt được hiệu suất tương đối vượt trội khi hội tụ ở 86% sau 45 vòng. trong khi phần lớn thuật toán khác mất đến hơn 50 vòng để hội tụ.
- Nhìn chung có thể thấy rằng trên bộ dữ liệu EMNIST và mô hình CNN, DyFedImp đã đạt được một số kết quả đáng hữa hẹn khi hội tụ nhanh nhất trong 2 trên 4 kịch bản và hội tụ nhanh thứ 2 trong 2 kịch bản còn lại. Mặc dù độ chính xác đạt được chưa hoàn toàn tốt tuy nhiên sự chênh lệch có thể coi là không đáng kể.



Hình 5.10: So sánh hiệu suất của các thuật toán với các kịch bản non-IID khác nhau trên bộ dữ liệu EMNIST và mô hình CNN

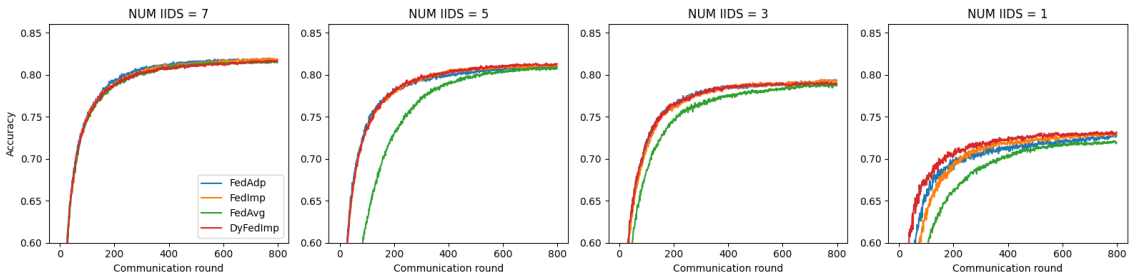
Algorithm	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	34	39	77	62
FedImp	**29**	39	**64**	47
FedAdp	30	36	74	53
DyFedImp	30	**35**	71	**45**
FedProx	31	39	87	59
FedAdam	N/A	N/A	N/A	N/A
FedAdagrad	35	39	100	50
FedYogi	N/A	N/A	N/A	59
FedAvgM	34	**35**	89	64

Bảng 5.2: Số vòng huấn luyện để đạt được độ chính xác mục tiêu của các thuật toán trên bộ dữ liệu EMNIST và mô hình CNN với các kịch bản khác nhau

b, Thực nghiệm trên bộ dữ liệu CIFAR-10

Hình 5.11 và bảng 5.3 mô tả hiệu suất của các thuật toán đã liệt kê ở trên trong 800 vòng huấn luyện toàn cầu trên mô hình 2-layer CNN. Từ các kết quả có thể rút ra một số nhận xét như sau:

- Ở bộ dữ liệu CNN, khi mà dữ liệu đã có sự đa dạng và phức tạp hơn, có thể thấy rõ sự phân hóa trong hiệu suất của các thuật toán. Số vòng để đạt được hội tụ đã có chênh lệch lớn. Ngoài ra có thể thấy từ các bảng rằng FedAdam vẫn có hiệu suất rất tệ ngoài ra còn có FedAdagrad nữa.
- Đối với kịch bản đầu tiên là 7 IID + 3 non-IID, thuật toán FedAdp hội tụ nhanh nhất khi đạt 81% độ chính xác trong chỉ 319 vòng huấn luyện, đúng thứ hai là FedImp (360 vòng). Trong trường hợp này thuật toán đề xuất DyFedImp có hiệu suất khá tệ khi mất tới 421 round. Tuy nhiên DyFedImp vẫn hội tụ nhanh hơn so với 4 thuật toán khác (FedProx, FedAdam, FedAdagrad va FeYogi).
- Trong kịch bản thứ 2 là 5 IID + 5 non-IID, đã có sự cải thiện khi thuật toán đề xuất DyFedImp và thuật toán FedImp đều hội tụ sớm nhất ở ngưỡng 334 vòng với độ chính xác 81%. Trong khi hầu hết các thuật toán còn lại mất trên 400 vòng để có thể hội tụ.
- Đối với kịch bản là 3 IID + 7 non-IID, FedAdp có tốc độ hội tụ vượt trội khi đạt được 78% sau 283 vòng, ngay sau là DyFedImp (314 vòng) và FedImp (317 vòng). Các thuật toán còn lại hầu hết mất 400+ vòng huấn luyện để đạt được độ chính xác tương tự.
- Kịch bản cuối cùng là 1 IID + 9 non-IID, có thể thấy trong kịch bản này DyFedImp đã đạt được hiệu suất tương đôi vượt trội khi hội tụ sau 308 vòng và độ chính xác đạt được là 73.3%. trong khi phần lớn thuật toán khác mất đến hơn 500 vòng huấn luyện.
- Nhìn chung có thể thấy rằng trên bộ dữ liệu CIFAR-10 và mô hình 2-layer CNN, DyFedImp đã đạt được các kết quả tốt trong 3/4 trường hợp, chỉ thua FedAdp trong 1 trường hợp duy nhất. IID + 3 non-IID).



Hình 5.11: So sánh hiệu suất của các thuật toán với các kịch bản non-IID khác nhau trên bộ dữ liệu CIFAR-10 và mô hình 2-layer CNN

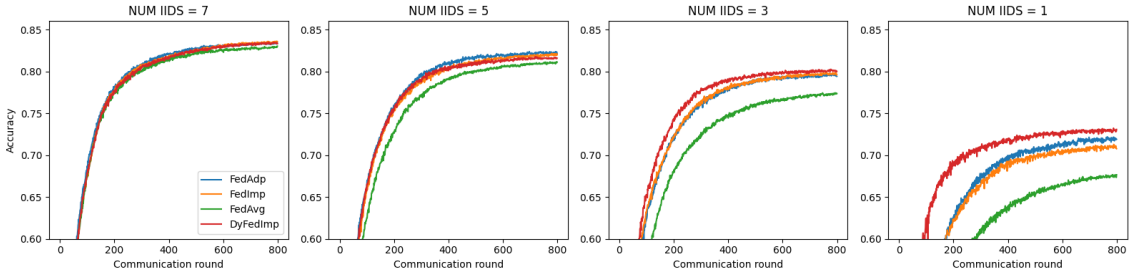
Thuật toán	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	402	502	462	672
FedImp	360	334	317	358
FedAdp	**319**	384	**283**	513
DyFedImp	421	**334**	314	**308**
FedProx	522	418	530	723
FedAdam	N/A	N/A	N/A	779
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	728	770	659	571
FedAvgM	376	432	402	N/A

Bảng 5.3: Số vòng huấn luyện để đạt được độ chính xác mục tiêu của các thuật toán trên bộ dữ liệu CIFAR-10 và mô hình 2-layer CNN với các kịch bản khác nhau

Hình 5.12 và bảng 5.4 mô tả hiệu suất của các thuật toán đã liệt kê ở trên trong 800 vòng huấn luyện toàn cầu trên mô hình 4-layer CNN. Từ các kết quả có thể rút ra một số nhận xét như sau:

- So với mô hình 2-layer CNN, mô hình 4-layer CNN đã thể hiện được rõ hơn sự chênh lệch giữa hiệu suất của các thuậ toán. Có thể lý giải cho việc này là mô hình càng lớn thì số lượng tính toán cũng nhiều và phức tạp hơn. Điều này dẫn đến việc ảnh hưởng của dữ liệu non-IID lên mô hình cũng rõ ràng hơn.
- Đối với kịch bản đâu tiên là 7 IID + 3 non-IID, thuật toán FedAdp hội tụ nhanh nhất khi đạt 83% độ chính xác trong chỉ 527 vòng huấn luyện, sau đó là FedImp (566 vòng), DyFedImp (588 vòng) và FedProx (604) vòng. Các thuật toán còn lại mất đến hơn 700 vòng để đạt được độ chính xác tương tự
- Trong kịch bản thứ 2 là 5 IID + 5 non-IID, tương tự như trường hợp trên khi FedAdp hội tụ nhanh nhất khi mất 387 vòng để đạt được 81% độ chính xác. DyFedImp trong trường hợp này mất thêm gần 100 vòng so với FedAdp để đạt được hội tụ nhưng vẫn vượt trội hơn phần lớn các thuật toán còn lại.
- Đối với kịch bản là 3 IID + 7 non-IID, DyFedImp đã vượt qua FedProx để hội tụ nhanh nhất với chỉ 270 vòng huấn luyện để đạt được 77% độ chính xác. Trong khi đó FedImp cần 313 vòng còn FedAdp cần 338 vòng. Các thuật toán còn lại cần tới hơn 550 vòng để đạt được độ chính xác tương tự.
- Kịch bản cuối cùng là 1 IID + 9 non-IID, có thể thấy trong kịch bản này DyFedImp đã hoàn toàn vượt trội khi hội tụ sau 152 vòng và độ chính xác đạt được là 73.22%. Trong khi phần lớn thuật toán khác mất từ 300 đến gần 700 vòng huấn luyện để hội tụ.
- Nhìn chung có thể thấy rằng trên bộ dữ liệu CIFAR-10 và mô hình 4-layer CNN, DyFedImp đã đạt được các số kết quả tốt nhất trong 2/4 trường hợp

non-IID và bám sát FedAdp trong 2 trường hợp còn lại.



Hình 5.12: So sánh hiệu suất của các thuật toán với các kịch bản non-IID khác nhau trên bộ dữ liệu CIFAR-10 và mô hình 4-layer CNN

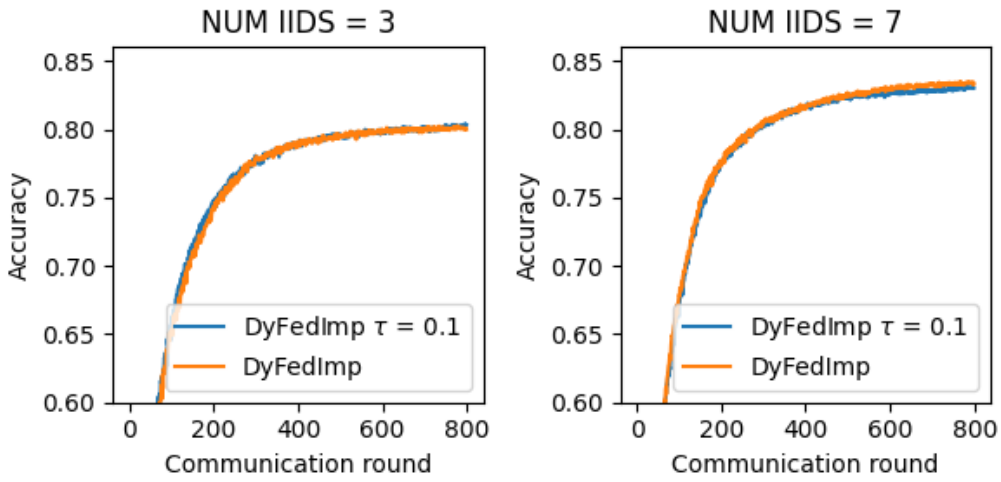
Thuật toán	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	800	724	660	623
FedImp	566	459	313	306
FedAdp	**527**	**387**	338	280
DyFedImp	588	482	**270**	**152**
FedProx	604	745	575	640
FedAdam	787	786	N/A	434
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	737	720	760	465
FedAvgM	719	588	570	682

Bảng 5.4: Số vòng huấn luyện để đạt được độ chính xác mục tiêu của các thuật toán trên bộ dữ liệu CIFAR-10 và mô hình 4-layer CNN với các kịch bản khác nhau

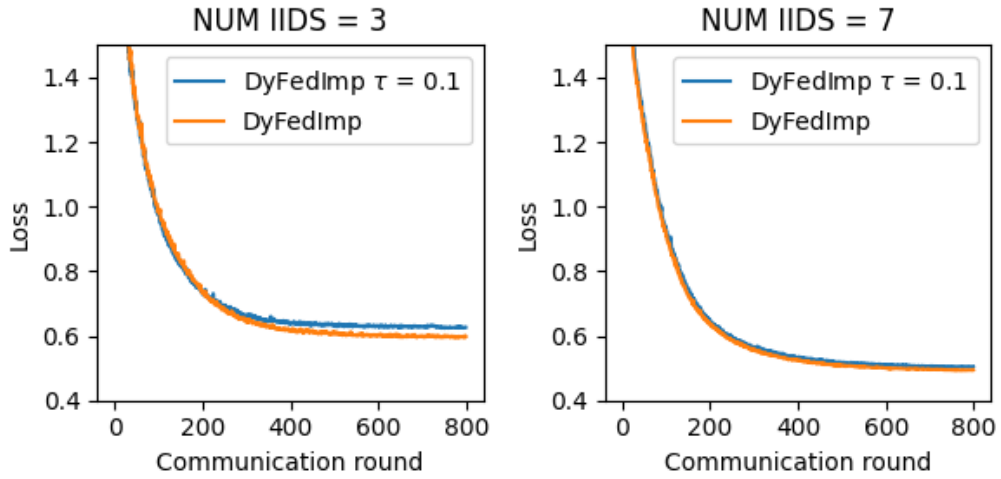
c, Thực nghiệm DyFedImp với giá trị τ nhỏ

Trong phần này, một số thực nghiệm bổ sung sẽ được thực hiện để chứng minh luận điểm đã được thảo luận trong chương trước. Một giả định có thể được đạt ra là tại sao không đặt giá trị τ cố định rất nhỏ ở thời điểm ban đầu (ví dụ $\tau = 0.1$) cho tất cả các trường hợp. Do đó các thực nghiệm với DyFedImp sử dụng giá trị $\tau = 0.1$ ở thời điểm ban đầu được thực nghiệm và so sánh với thuật toán DyFedImp đề xuất. Thực nghiệm được thực hiện sử dụng 2 kịch bản là 3 IID + 7 non-IID và 7 IID + 3 non-IID trên bộ dữ liệu. Các chỉ số được xem xét bao gồm độ chính xác và giá trị loss trên bộ test, biểu diễn trên hình 5.13.

Từ hình 5.13 có thể thấy rằng tốc độ hội tụ của hai mô hình trong kịch bản đầu tiên 3 IID + 7 non-IID là khá tương đồng nhau. Tuy nhiên cũng trong kịch bản này chỉ số loss của DyFedImp với $\tau = 0.1$ lại cao hơn với DyFedImp đề xuất. Điều này là do mô hình bị khớp với các nút IID nên mặc dù có độ chính xác ổn nhưng lại không giữ được tính đa dạng của mình. Trong kịch bản tiếp theo thì có thể thấy là độ chính xác của DyFedImp đề xuất là tốt hơn trong những vòng cuối vòng loss của nó cũng thấp hơn mặc dù không quá đáng kể.



(a) Độ chính xác trên tập test



(b) Loss trên tập test

Hình 5.13: So sánh hiệu suất của thuật toán DyFedImp trong 2 kịch bản khác nhau

d, Nhận xét tổng quan

Tổng quan lại, có thể thấy rằng thuật toán DyFedImp đã đạt được các kết quả rất tốt khi hội tụ nhanh nhất trong nhiều kịch bản khác nhau trên cả hai bộ dữ liệu. Đặc biệt thuật toán luôn hội tụ vượt trội so với các thuật toán khác trong trường hợp có mức độ non-IID cao (1 IID + 9 non-IID). Các kết quả thực nghiệm cho thấy rằng DyFedImp có thể cải thiện tốc độ hội tụ lên đến 30% tùy tình huống trên bộ dữ liệu EMNIST. Đối với bộ dữ liệu CIFAR-10, DyFedImp cải thiện từ 10 đến 50% tốc độ hội tụ so với FedAdp và FedImp và cải thiện lên đến 75% tốc độ hội tụ so với các thuật toán khác.

Tuy vậy cũng có thể thấy rằng thuật toán còn nhiều điểm chưa tối ưu. Bằng chứng là tốc độ hội tụ của thuật toán trên một số kịch bản thực nghiệm hay trên một số mô hình cũng chưa thực sự quá tốt hay vượt trội nhu kì vọng. Điều này chỉ ra rằng tồn tại mối quan hệ giữa mức độ non-IID, mô hình cũng như bộ dữ liệu và

công thức đề xuất chưa có khả năng thích ứng tối ưu trong một vài các trường hợp này.

CHƯƠNG 6. KẾT LUẬN

6.1 Kết luận

Đồ án này đánh giá một cách chi tiết các hạn chế của thuật toán FedImp trong các bối cảnh dữ liệu không đồng nhất (non-IID) khác nhau, từ đó đề xuất một thuật toán Học kết hợp mới, DyFedImp, như một giải pháp toàn diện để giải quyết những thiếu sót đã được xác định. Thuật toán được đề xuất cải thiện tính linh động của các công thức tính trọng số bằng cách điều chỉnh khoảng trọng số trong quá trình huấn luyện một cách phù hợp. Thông qua các đánh giá thực nghiệm sâu rộng, DyFedImp đã chứng minh tốc độ hội tụ cải thiện so với cả FedImp và các thuật toán khác trong nhiều kịch bản với dữ liệu không đồng nhất. Đáng chú ý, DyFedImp giảm số vòng truyền thông lần lượt lên đến 10% và 50% so với FedImp và FedAdp trên các tập dữ liệu EMNIST và CIFAR-10. Ngoài ra DyFedImp cũng có tốc độ hội tụ cải thiện từ 30% và 75% so với các thuật toán khác trên các tập dữ liệu tương tự.

6.2 Hướng phát triển trong tương lai

Mặc dù đạt được một số kết quả tốt với với FedImp nói riêng và các thuật toán tổng hợp mô hình nói chung, DyFedImp vẫn tồn tại một số điểm chưa tối ưu và hoàn toàn có thể phát triển trong tương lai. Đầu tiên là cải tiến các công thức tính trọng số sao cho có thể linh hoạt và phù hợp hơn trên từng bộ dữ liệu cũng như các mô hình khác nhau. Ngoài ra công thức thay đổi miền giá trị trong đồ án cũng có thể được cải tiến sao cho có thể thích nghi một cách phù hợp với trạng thái hiện tại của các mô hình cục bộ nhằm đạt được độ ổn định tốt hơn.

Mặt khác trọng tâm của đồ án mới chỉ tập chung vào việc cải thiện tốc độ hội tụ của mô hình toàn cầu chứ chưa thực sự cải thiện về độ chính xác khi hội tụ giữa trong các trường hợp non-IID khác nhau. Do đó trong tương lai có thể phát triển một khung thuật toán Học kết hợp hoàn thiện hơn bằng cách áp dụng chỉ số entropy vào việc điều chỉnh các tham số học của mô hình nhằm đạt được độ chính xác cao hơn.

TÀI LIỆU THAM KHẢO

[1] T. Li, A. K. Sahu, A. Talwalkar **and** V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, **journvol** 37, **number** 3, **pages** 50–60, 2020.

[2] P. Kairouz, H. B. McMahan, B. Avent **andothers**, "Advances and open problems in federated learning," *CoRR*, **journvol** abs/1912.04977, 2019. arXiv: 1912.04977. **url**: http://arxiv.org/abs/1912.04977.

[3] S. Pichai, "Google's sundar pichai: Privacy should not be a luxury good," *New York Times*, 2019.

[4] ai.google, *Under the hood of the pixel 2: How ai is supercharging hardware*, https://ai.google/stories/ai-in-hardware/, Retrieved Nov 2018, 2018.

[5] support.google, *Your chats stay private while messages improves suggestions*, https://support.google.com/messages/answer/9327902, Retrieved Aug 2019, 2019.

[6] Apple, *Private federated learning (neurips 2019 expo talk abstract)*, https://nips.cc/ExpoConferences/2019/schedule?talk_id=40, 2019.

[7] W. de Brouwer, *The federated future is ready for shipping*, https://doc.ai/blog/federated-future-ready-shipping/, 2019.

[8] K. Daly, H. Eichner, P. Kairouz, H. B. McMahan, D. Ramage **and** Z. Xu, *Federated learning in practice: Reflections and projections*, 2024. arXiv: 2410.08892 [cs.LG]. **url**: https://arxiv.org/abs/2410.08892.

[9] P. Kairouz, H. B. McMahan, B. Avent **andothers**, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, **journvol** 14, **number** 1–2, **pages** 1–210, 2021.

[10] H. Zhu, J. Xu, S. Liu **and** Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, **journvol** 465, **pages** 371–390, 2021.

[11] Y. Liu, L. Zhang, N. Ge **and** G. Li, "A systematic literature review on federated learning: From a model quality perspective," *arXiv preprint arXiv:2012.01973*, 2020.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson **and** B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," **in***Artificial intelligence and statistics* PMLR, 2017, **pages** 1273–1282.

[13] T.-M. H. Hsu, H. Qi **and** M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[14] H. Wu **and** P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Transactions on Cognitive Communications and Networking*, **jourvol** 7, **number** 4, **pages** 1078–1088, 2021.

[15] H. A. Tran, C. Ta **and** T. Tran, "Fedimp: The federated impurity weighting algorithm for improving convergence in federated learning," Manuscript submitted for review and publication.

[16] S. Reddi, Z. Charles, M. Zaheer **andothers**, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[17] Y. Yeganeh, A. Farshad, N. Navab **and** S. Albarqouni, "Inverse distance aggregation for federated learning with non-iid data," **in***Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2* Springer, 2020, **pages** 150–159.

[18] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough **and** V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[19] S. Vahidian, M. Morafah **and** B. Lin, "Personalized federated learning by structured and unstructured pruning under data heterogeneity," **in***2021 IEEE 41st international conference on distributed computing systems workshops (ICDCSW)* IEEE, 2021, **pages** 27–34.

[20] M. Hong, S.-K. Kang **and** J.-H. Lee, "Weighted averaging federated learning based on example forgetting events in label imbalanced non-iid," *Applied Sciences*, **jourvol** 12, **number** 12, **page** 5806, 2022.

[21] X. Meng, Y. Li, J. Lu **and** X. Ren, "An optimization method for non-iid federated learning based on deep reinforcement learning," *Sensors*, **jourvol** 23, **number** 22, 2023, ISSN: 1424-8220. DOI: 10.3390/s23229226. **url**: https://www.mdpi.com/1424-8220/23/22/9226.

[22] Z. Lu, H. Pan, Y. Dai, X. Si **and** Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, **jourvol** 11, **number** 11, **pages** 19 188–19 209, 2024. DOI: 10.1109/JIOT.2024.3376548.

[23] G. Cohen, S. Afshar, J. Tapson **and** A. Van Schaik, "Emnist: Extending mnist to handwritten letters," **in***2017 international joint conference on neural networks (IJCNN)* IEEE, 2017, **pages** 2921–2926.

[24] A. Krizhevsky, "Learning multiple layers of features from tiny images," techreport, 2009.

[25] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, **jourvol** 29, **number** 6, **pages** 141–142, 2012.

[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar **and** V. Smith, *Federated optimization in heterogeneous networks*, 2020. arXiv: 1812.06127 [cs.LG]. **url**: https://arxiv.org/abs/1812.06127.

PHỤ LỤC

A. Cài đặt và giả lập môi trường học kết hợp

A.1 Cấu hình máy thực nghiệm

Các kết quả được trình bày trong phạm vi đồ án được cài đặt và thực nghiệm trên máy với cấu hình như sau:

- Hệ điều hành: Ubuntu 20.04
- CPU: Intel(R) Xeon(R) W-2255 CPU @ 3.70GHz
- RAM: 125 Gb
- GPU: 2 x NVIDIA RTX A5000
- Phiên bản Python 3.11.10

A.2 Các thư viện sử dụng

Code được cài đặt để thực nghiệm trong đồ án sử dụng ngôn ngữ Python và 2 thư viện là thư viện Pytorch để cài đặt kiến trúc học mô hình học máy và thư viện Flower để giả lập môi trường Học kết hợp và các thuật toán liên quan.

A.2.1 Pytorch

Hình A.1: Pytorch framework

PyTorch là một thư viện học sâu mã nguồn mở được phát triển bởi Facebook AI Research (FAIR). Được phát hành lần đầu vào năm 2016, PyTorch nhanh chóng trở thành một trong những công cụ phổ biến nhất trong lĩnh vực học máy và học sâu nhờ tính linh hoạt, dễ sử dụng và hiệu năng cao. Thư viện này cung cấp một nền tảng mạnh mẽ để xây dựng và huấn luyện các mô hình học sâu, từ các mô hình cơ bản đến những hệ thống phức tạp.

PyTorch nổi bật nhờ tính năng tính toán tự động đạo hàm (autograd) dựa trên đồ thị tính toán động (dynamic computation graph). Điều này cho phép các nhà nghiên cứu dễ dàng thử nghiệm và kiểm tra các mô hình phức tạp, đồng thời hỗ trợ việc debug hiệu quả hơn. Khả năng này giúp PyTorch đặc biệt phù hợp cho nghiên

cứu và phát triển các giải pháp sáng tạo.

Ngoài ra, PyTorch tích hợp nhiều công cụ và thư viện phụ trợ như torchvision (dành cho xử lý dữ liệu hình ảnh), torchaudio (xử lý âm thanh), và torchtext (xử lý ngôn ngữ tự nhiên), cung cấp một hệ sinh thái phong phú để làm việc với dữ liệu. PyTorch cũng hỗ trợ triển khai các mô hình trên các nền tảng khác nhau, bao gồm GPU và CPU, giúp tối ưu hóa hiệu năng khi huấn luyện trên dữ liệu lớn.

Trong bối cảnh ứng dụng thực tế, PyTorch được sử dụng rộng rãi trong nhiều lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, và học tăng cường. Sự hỗ trợ từ cộng đồng lớn mạnh và khả năng tích hợp với các công cụ như TorchServe và PyTorch Lightning đã giúp PyTorch trở thành một lựa chọn hàng đầu cho cả nghiên cứu và triển khai học máy trong sản xuất.

A.2.2 Flower

Hình A.2: Flower framework

Flower là một framework python cho phép cài đặt hoặc giả lập các mô hình học liên kết được giới thiệu vào năm 2020. Flower cung cấp cơ sở hạ tầng để thực hiện chính xác các tác vụ Học kết hợp, đánh giá hay phân tích một cách dễ dàng, mở rộng và an toàn. Nó cho phép người dùng phân tán bất kỳ khối công việc nào, bất kỳ framework học máy nào, và bất kỳ ngôn ngữ lập trình nào. Flower cung cấp các đặc điểm sau đây:

- Tính mở rộng: Flower được xây dựng để cho phép các hệ thống thực tế với một lượng lớn các máy khách. Các nhà nghiên cứu đã sử dụng Flower để chạy các khối công việc với hàng chục triệu máy khách.
- Không ràng buộc với Framework học máy: Flower tương thích với hầu hết các framework học máy hiện có như Tensorflow hay Pytorch.
- Đám mây, Di động, Edge và Hơn thế nữa: Flower cho phép nghiên cứu trên

tất cả các loại máy chủ và thiết bị, bao gồm cả di động. AWS, GCP, Azure, Android, iOS, Raspberry Pi và Nvidia Jetson đều tương thích với Flower.

- Từ Nghiên cứu đến thực tiễn: Flower cho phép các ý tưởng bắt đầu dưới dạng các dự án nghiên cứu và sau đó dần dần chuyển đến triển khai sản xuất với mức độ công việc kỹ thuật thấp và cơ sở hạ tầng đã được chứng minh.
- Độc lập với Nền tảng: Flower tương thích với các hệ điều hành và nền tảng phần cứng khác nhau để hoạt động tốt trong môi trường thiết bị Edge đa dạng.
- Dễ sử dụng: Bắt đầu là một việc đơn giản. Chỉ cần 20 dòng mã Python là đủ để xây dựng một hệ thống học tập phân tán đầy đủ..

A.3 Code cài đặt thuật toán

Code giả lập môi trường Học kết hợp và cài đặt các thuật toán được sử dụng có thể được tìm thấy trong Link Github sau:

https://github.com/HieuHuyNguyenzz/Dynamic-Federated-Impurity-Weighting-Algorithm

B. Các công bố có liên quan của sinh viên

1. **Huy-Hieu Nguyen**, Hai-Anh Tran, Truong X. Tran. "Efficient Federated Learning Convergence with Epoch Adaptation", The 23rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2025). (**Đã nộp**)
2. **Nguyễn Huy Hiệu**, Vũ Ngọc Minh, Đào Đức Huy "Nghiên cứu và đề xuất giải pháp học liên kết phân cụm hiệu quả về tốc độ tổng hợp trên bộ dữ liệu không đồng đều", Hội thảo khoa học thường niên khoa Công nghệ thông tin, Đại học Xây Dựng Hà Nội, Ngày 18/06/2024.