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.

0.1 Bộ dữ liệu thực nghiệm

Đồ án này được thúc đẩy bởi các tác vụ phân loại hình ảnh nói riêng và thậm trí 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. Do đó hai tập dữ liệu ảnh đa dạng được sử dụng, bao gồm EMNIST [1] và CIFAR-10 [2]. Mỗi tập dữ liệu phục vụ một mục đích riêng biệt và đặt ra các thách thức độc đáo, góp phần vào một thực nghiệm 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 [3]. Tập dữ liệu EMNIST cung cấp 6 bộ khác nhau khác nhau, và trong thực nghiệm này, bộ 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.

0.2 Giả lập cấu hình non-IID trong FL

Các thực nghiệm của nghiên cứu được kế thừa và phát triển từ hai bài báo [4] và [5].

0.2.1 Cách chia dữ liệu

Để mô phỏng một tập hợp đa dạng giữa các nút, phương pháp tương tự như trong [6] 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 1: 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 1 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.

0.2.2 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:

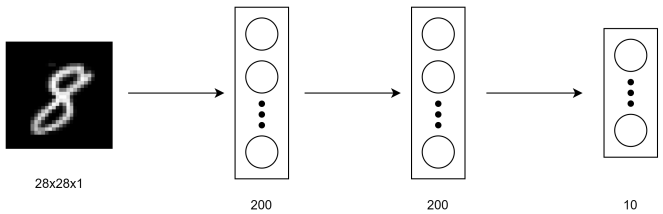
- 7 nút dữ liệu cân bằng + 3 nút dữ liệu mất cân bằng
- 5 nút dữ liệu cân bằng + 5 nút dữ liệu mất cân bằng
- 3 nút dữ liệu cân bằng + 7 nút dữ liệu mất cân bằng
- 1 nút dữ liệu cân bằng + 9 nút dữ liệu mất cân bằng

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 [5] và [4]. 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.

0.3 Các mô hình và tham số thực nghiệm

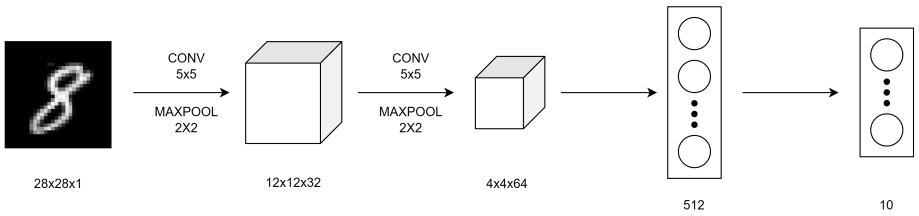
0.3.1 Kiến trúc mô hình

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 0.1 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 0.10 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.



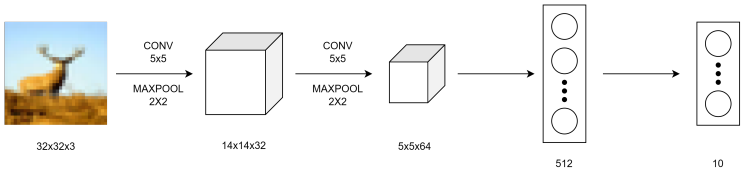
Hình 0.1: Kiến trúc MLP sử dụng với bộ dữ liệu EMNIST

Đối với tập dữ liệu CIFAR-10, có hai mô hình khác nhau được triển khai, bao gồm: (i) Một kiến trúc CNN với 2 lớp tích chập như hình 0.11 với hai lớp convolution 5x5 (lớp đầu tiên có 32 kênh, lớp thứ hai có 64 kênh, mỗi lớp theo sau

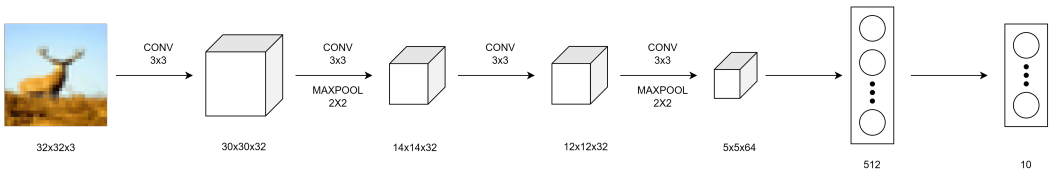


Hình 0.2: Kiến trúc CNN sử dụng với bộ dữ liệu EMNIST

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 0.12 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 0.3: Kiến trúc 2-layer CNN trên bộ dữ liệu CIFAR-10



Hình 0.4: 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.

0.3.2 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[7], FedOpt(FedAdam, FedYogi, FedAdagrad)[6], FedAvgM[8] nhằm cung cấp một cái nhìn tổng quan về mô hình đề xuất.

0.3.3 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.

0.4 Kết quả thực nghiệm và các đánh giá

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

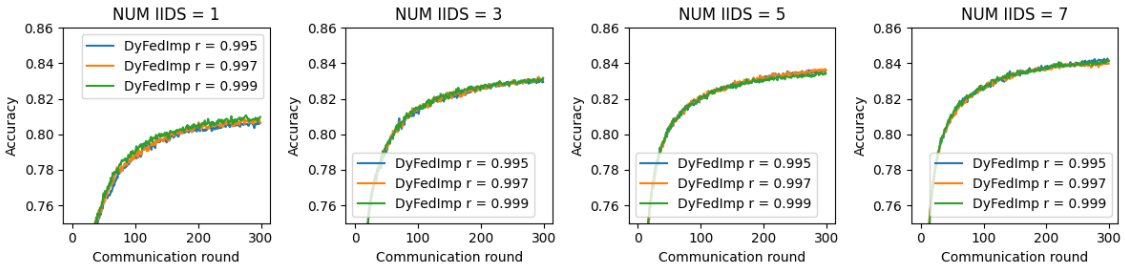
Các hình 0.5, 0.6, 0.7 và 0.8 dưới đây lần lượt biểu diễn hiệu suất của thuật toán DyFedImp sử dụng tham số giảm khoảng trọng số r trên các mô hình và bộ dữ liệu khác nhau và trên các kịch bản non-IID khác nhau. Có 3 giá trị được thực nghiệm nhằm đánh giá bao gồm: 0.995, 0.997 và 0.999.

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

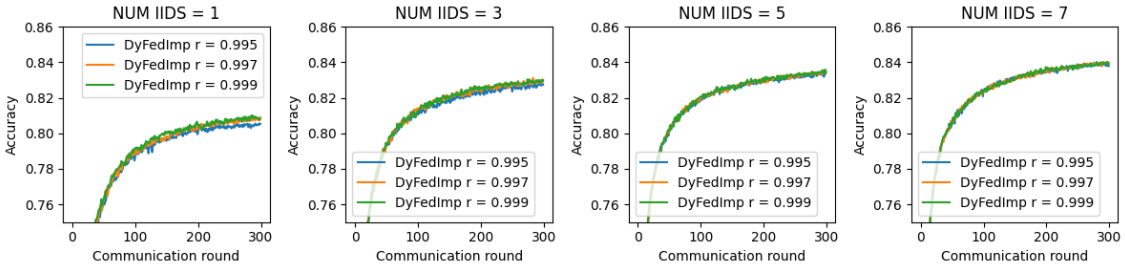
- Đối với bộ dữ liệu EMNIST, có thể thấy rằng sự khác biệt mà các giá trị r ảnh hưởng đến hiệu suất của mô hình là không đáng kể. Điều này có thể được giải thích phần nào là nhờ sự đơn giản về dữ liệu khi EMNIST là tập các ảnh đen trắng.
- Mặt khác đối với bộ dữ liệu CIFAR-10, có thể thấy giá trị r ảnh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

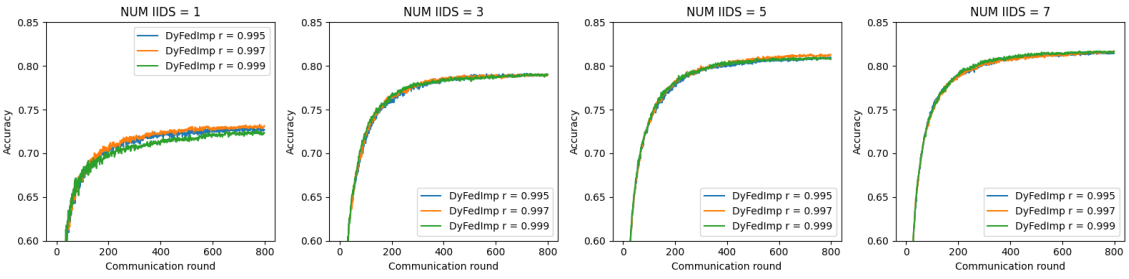
- Tổng kết lại các giá trị r lần lượt là 0.999, 0.999, 0.997 và 0,999 sẽ được áp dụng tương ứng với các mô hình MLP, CNN, 2-layer CNN và 4-layer CNN trong các thực nghiệm sau này.



Hình 0.5: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu EMNIST và mô hình MLP



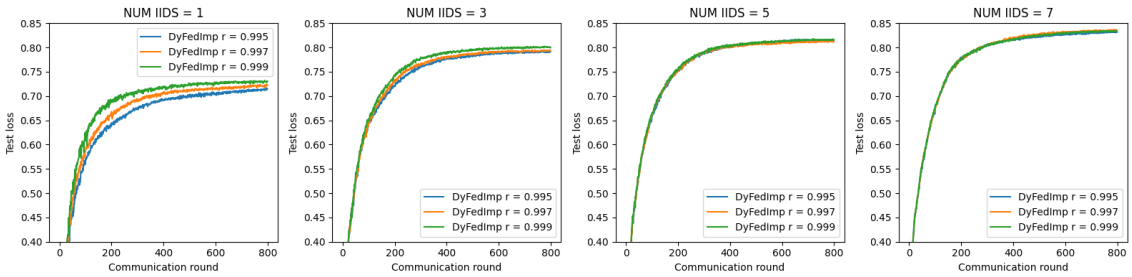
Hình 0.6: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu EMNIST và mô hình CNN



Hình 0.7: 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

0.4.2 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.

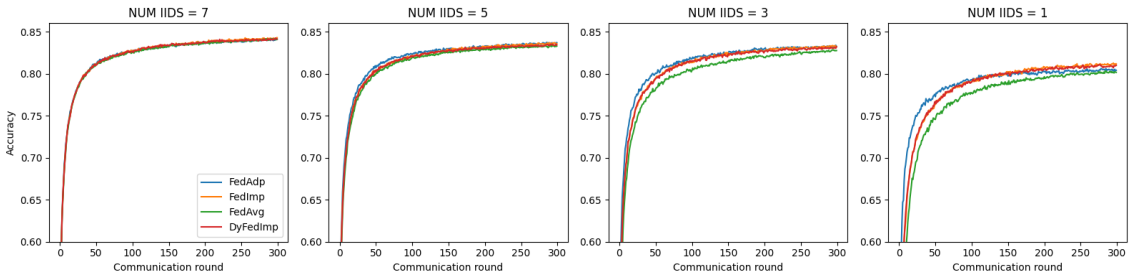


Hình 0.8: Đánh giá hiệu suất của DyFedImp với các tham số r khác nhau trên bộ dữ liệu CIFAR-10 và mô hình 4-layer CNN

a, Thực nghiệm trên bộ dữ liệu EMNIST

Hình 0.9 và bảng 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 0.9: 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

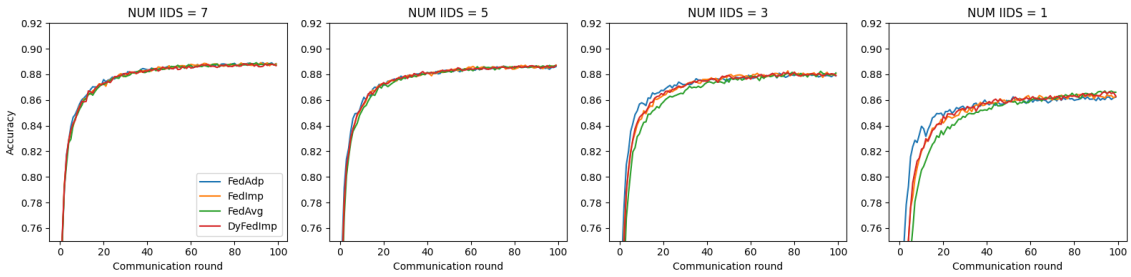
Thuật toán	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	257	204	181	237
FedImp	225	157	125	140
FedAdp	290	**145**	**102**	141
DyFedImp	**221**	174	126	**139**
FedProx	271	206	175	234
FedAdam	N/A	N/A	N/A	N/A
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	N/A	728	N/A	N/A
FedAvgM	280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

Bảng 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 0.10 và bảng 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 cân 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 đầu 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 hơn 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 lệch 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.
- 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 đó 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 hứa 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 0.10: 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

Thuật toán	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 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

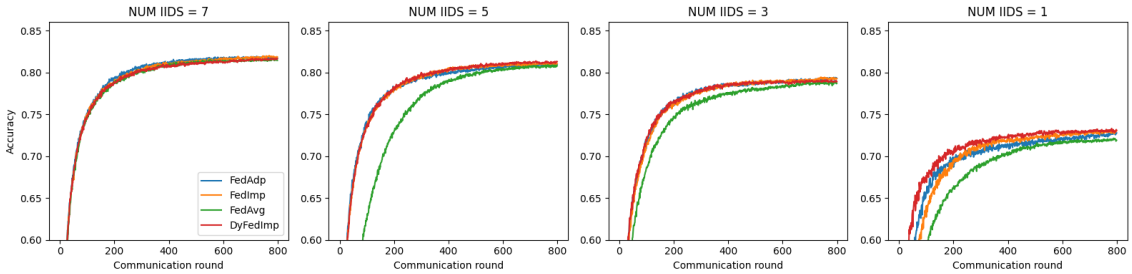
b, Thực nghiệm trên bộ dữ liệu CIFAR-10

Hình 0.11 và bảng 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 0.11: 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

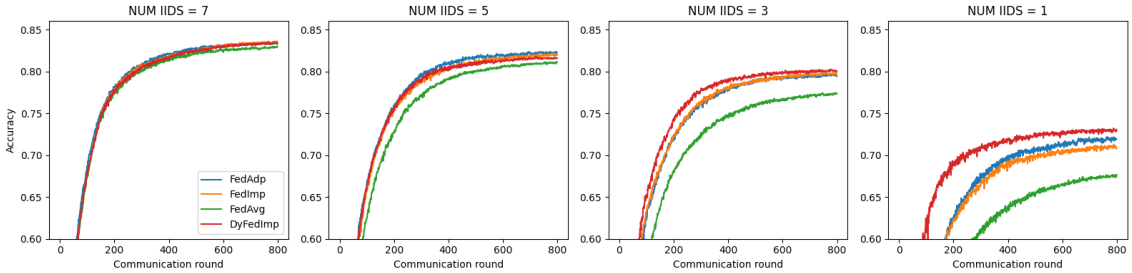
Thuật toán	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	402	502	462	672
FedImp	360	334	317	358
FedAdp	**319**	384	**283**	513
DyFedImp	421	**334**	314	**308**
FedProx	522	418	530	723
FedAdam	N/A	N/A	N/A	779
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	728	770	659	571
FedAvgM	376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

Bảng 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 0.12 và bảng 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 thuật 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 đầu 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 0.12: 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

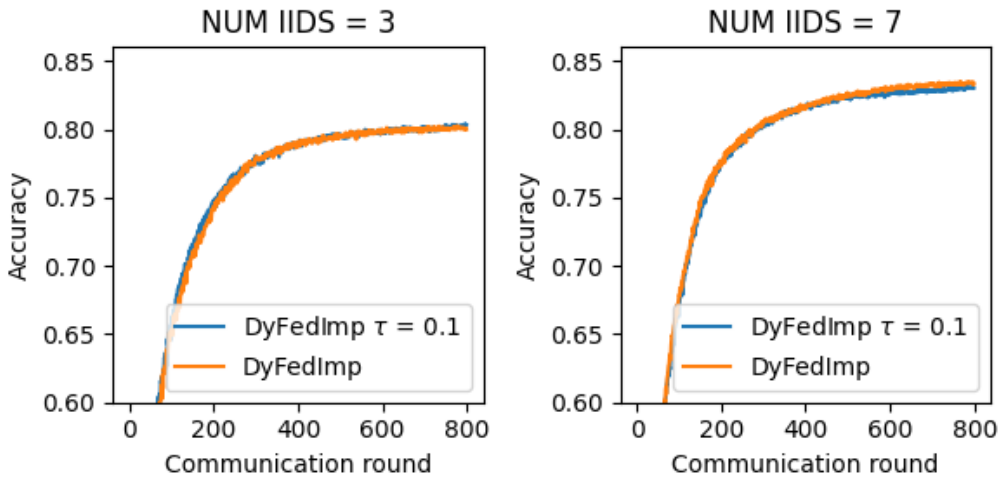
Thuật toán	7 IID + 3 non-IID	5 IID + 5 non-IID	3 IID + 7 non-IID	1 IID + 9 non-IID
FedAvg	800	724	660	623
FedImp	566	459	313	306
FedAdp	**527**	**387**	338	280
DyFedImp	588	482	**270**	**152**
FedProx	604	745	575	640
FedAdam	787	786	N/A	434
FedAdagrad	N/A	N/A	N/A	N/A
FedYogi	737	720	760	465
FedAvgM	719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

Bảng 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

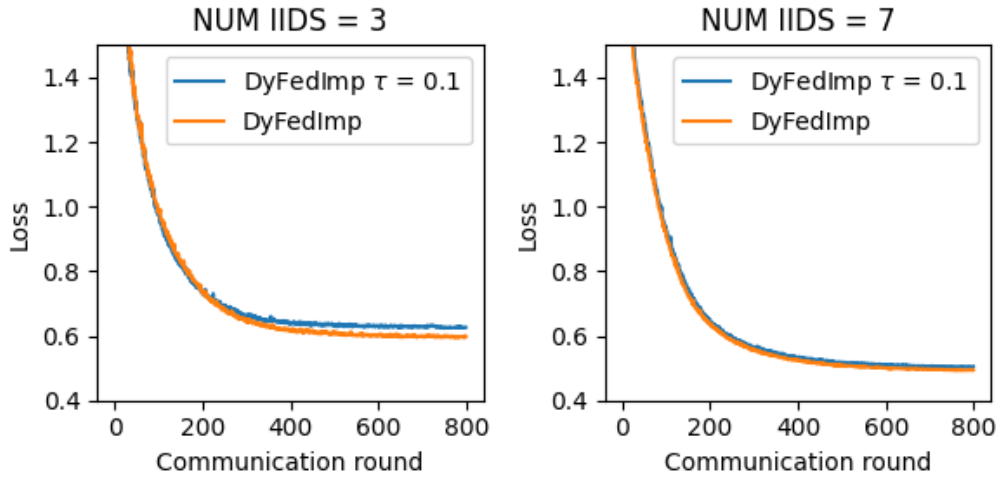
c, Thực nghiệm DyFedImp với giá trị τ nhỏ

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 τ = 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 0.13.

Từ hình 0.13 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 0.13: 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 như 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 trường hợp này.