# Báo cáo thực nghiệm về Fed-iMAML

Bao-Long Nguyen[1,2] and Bac Le[1,2]

[1] Faculty of Information Technology, University of Science, Ho Chi Minh City, VietNam
[2] VietNam National University, Ho Chi Minh City, VietNam

**Abstract.** Hệ thống FL-MAML hoạt động trên dữ liệu non-IID, dù cho kết quả cao và khả năng hội tụ nhanh nhưng tiêu tốn rất nhiều phần cứng. Báo cáo đề xuất kết hợp `iMAML` (thuật toán meta-learning) và hệ thống FL và chứng minh bằng thực nghiệm rằng: (1) Hệ thống `FL-iMAML` cho độ chính xác và khả năng hội tụ nhanh tương đương hệ thống `FL-MAML` trên tập dữ liệu CIFAR-10, tuy nhiên, trên tập dữ liệu MNIST, thuật toán đề xuất bị giảm hiệu suất so với FL-MAML; (2) Hệ thống `FL-iMAML` tiêu thụ ít phần cứng (CPU, GPU) hơn so với hệ thống `FL-MAML`.

**Keywords:** Federated learning · non-IID · Meta-learning

## 1 Thực nghiệm

### 1.1 Mục tiêu

- Thực hiện đánh giá 2 thuật toán `Fed-iMAML` (thuật toán đề xuất) và `Fed-MAML` về 2 phương diện: (1) - Độ chính xác; (2) - Chi phí phần cứng.
- Kỳ vọng: Thuật toán đề xuất cho độ chính xác tương đương với `Fed-MAML` nhưng tiêu tốn phần cứng ít hơn. Các tiêu chí đánh giá phần cứng là thời gian chạy 1 global epoch trên CPU và dung lượng VRAM tiêu tốn trên GPU.

### 1.2 Dữ liệu

- Báo cáo này sử dụng 50 thiết bị cho quá trình huấn luyện (chứa 75% dữ liệu) và 30 thiết bị cho quá trình kiểm thử (chứa 25% dữ liệu). Các thiết bị chứa dữ liệu lần lượt là ảnh của 2 tập dữ liệu MNIST [1] và CIFAR-10 [2].
- Để mô phỏng dữ liệu non-IID, mỗi thiết bị sẽ chứa 2/10 nhãn dữ liệu. Phân phối dữ liệu của các thiết bị là không giống nhau. Tại mỗi thiết bị, support set chứa 20% dữ liệu và query set chứa 80% dữ liệu.
- Bảng 1 trình bày thống kê mô tả cho dữ liệu giữa các thiết bị.

### 1.3 Thang đánh giá

- $acc_{macro}$: Được tính bằng cách lấy trung bình độ chính xác trên các clients. Đơn vị: %.
- Thời gian chạy 1 global epoch của CPU. Đơn vị: second/iteration.
- Lượng VRAM mà GPU tiêu tốn trong quá trình chạy. Đơn vị: %.

Table 1: Thống kê trên dữ liệu MNIST and CIFAR-10

| Dataset | #clients | #samples | #classes | #samples/client | | | | #classes/client |
|---|---|---|---|---|---|---|---|---|
| | | | | min | mean | std | max | |
| MNIST | 50 | 69,909 | 10 | 135 | 1,398 | 1,424 | 5,201 | 2 |
| CIFAR-10 | 50 | 52,497 | 10 | 506 | 1,049 | 250 | 1,986 | 2 |

### 1.4 Tiến hành thí nghiệm

- Nhóm thực hiện các thí nghiệm để so sánh `Fed-MAML` và thuật toán đề xuất `Fed-iMAML`.
- Kiến trúc mạng cho dữ liệu MNIST nhận vào một tensor kích thước $(784 \times 1)$. Tensor này lần lượt đi qua hay tầng ẩn có đầu ra lần lượt là 100 và 10. Các hàm kích hoạt tương ứng với mỗi tầng là `ReLU` và `Softmax`.
- Kiến trúc mạng cho dữ liệu CIFAR-10 nhận vào một tensor kích thước $(3 \times 32 \times 32)$. Tensor này lần lượt đi qua hai lớp `Convolution` với kernel cùng kích thước $(5 \times 5)$, số channel lần lượt là 6 và 16. Theo sau các lớp `Convolution` là các lớp `MaxPooling` có kích thước $(2 \times 2)$. Các đặc trưng sau đó được làm phẳng và đưa qua ba lớp FC có đầu ra là $120, 84$ và 10. Các hàm kích hoạt lần lượt là hai hàm `ReLU` và một hàm `Softmax`.
- Tất cả các thí nghiệm được thực hiện trên một máy tính có cấu hình:
  - CPU: Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz
  - RAM: 32GB
  - GPU: NVIDIA GeForce RTX 3070 - 8GB VRAM
- Các tham số sau được giữ nguyên trong quá trình huấn luyện:
  - Số global epochs: 200.
  - Số clients tham gia trong một lần huấn luyện toàn cục: 5.
  - Batch-size: 32.
- Sau mỗi 20 bước huấn luyện toàn cục, thực hiện đánh giá trên toàn bộ testing client.
- Các tham số sau được điều chỉnh để thu được mô hình tốt nhất:
  - Siêu tham số toàn cục: $\beta$.
  - Siểu tham số cục bộ: $\alpha$.
- Các tham số sau được điều chỉnh để đánh giá quá trình tiêu thụ phần cứng của các thuật toán:
  - Số bước huấn luyện cục bộ: $local\_epoch \in \{1, 5, 10, 15\}$.
  - Số bước tối ưu của `iMAML`: $cg\_step \in \{1, 5, 10\}$.

## 2 Results & Discussion

### 2.1 Accuracy

- Biểu đồ 1 và 2 thể hiện quá trình hội tụ của 2 thuật toán `Per-FedAvg` và `Fed-iMAML` trên các tập dữ liệu CIFAR-10 và MNIST. Cả hai thuật toán đều được fine-tune và chọn ra bộ tham số tốt nhất. Trên dữ liệu CIFAR-10, các

thuật toán đạt mức hội tụ tương đương nhau (xấp xỉ 68%). Về phần tập MNIST, 2 thuật toán cũng hội tụ nhưng kết quả tốt nhất của thuật toán đề xuất thấp hơn 12% so với `Fed-MAML`.

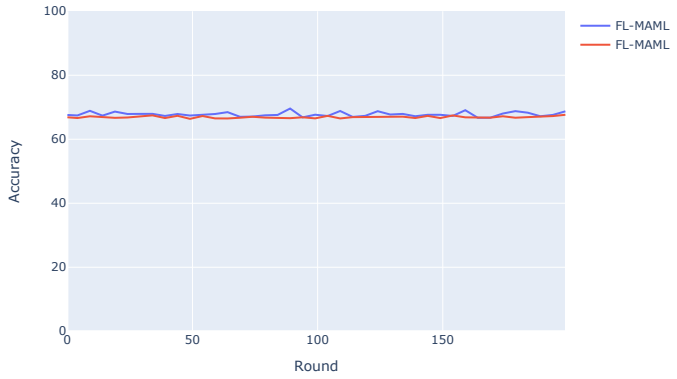


Fig. 1: Độ chính xác của Per-FedAvg và Fed-iMAML trên dữ liệu CIFAR-10

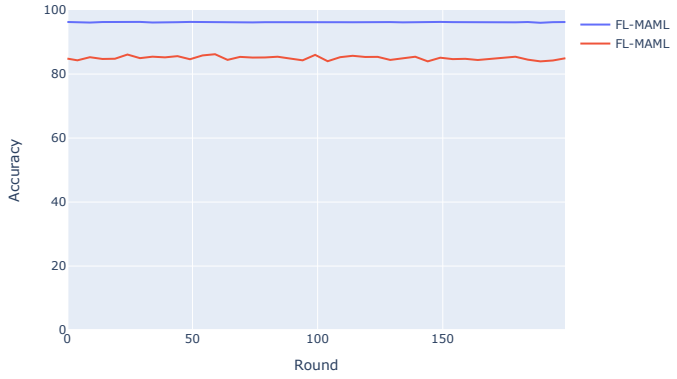


Fig. 2: Độ chính xác của Per-FedAvg và Fed-iMAML trên dữ liệu MNIST

### 2.2 Running time

– Biểu đồ 3 và 4 thể hiện thời gian chạy một global epoch của các thuật toán `Per-FedAvg` và `Fed-iMAML`. Trong đó, `Fed-iMAML` được cho chạy với số bước CG khác nhau. Có thể quan sát thấy:
  - Thời gian tăng dần đều đối với các thuật toán khi inner step tăng lên.
  - Đối với `Fed-iMAML`, số bước CG tăng lên thì thời gian cũng tăng theo.
  - Thời gian chạy của `Fed-iMAML` phần lớn là nhỉnh hơn so với `Per-FedAvg`. Phần nhỉnh hơn này thực sự không quá đáng kể nhưng vẫn có thể coi đây là một điểm trừ của `Fed-iMAML`.

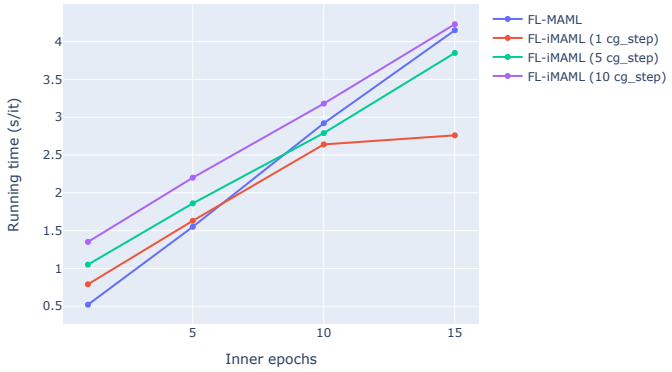


Fig. 3: Thời gian CPU chạy một global step của Fed-iMAML và Per-FedAvg trên dữ liệu CIFAR-10

### 2.3 Hardware

– Quan sát biểu đồ 5 và 6, về khả năng tiêu thụ bộ nhớ trên GPU, thuật toán `Fed-iMAML` với số bước CG khác nhau chiếm dụng cùng một lượng RAM và không đổi khi số lượng inner step tăng lên (dẫn đến các đường biểu diễn của `Fed-iMAML` chồng khít lên nhau). Trái lại, `Fed-MAML` tiêu thụ lượng RAM tăng tuyến tính khi số bước huấn luyện cục bộ tăng dần.

## 3 Conclusion

– Hiện tại em dự định thực nghiệm thêm trên một phần của tập dữ liệu FEMNIST. Đây là tập dữ liệu chữ và số viết tay bao gồm 62 phân lớp với

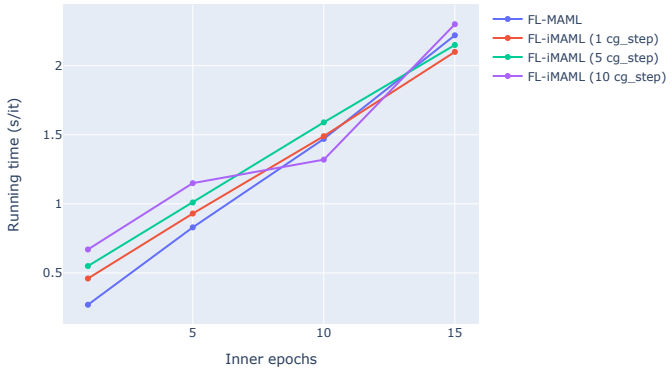


Fig. 4: Thời gian CPU chạy một global step của Fed-iMAML và Per-FedAvg trên dữ liệu MNIST

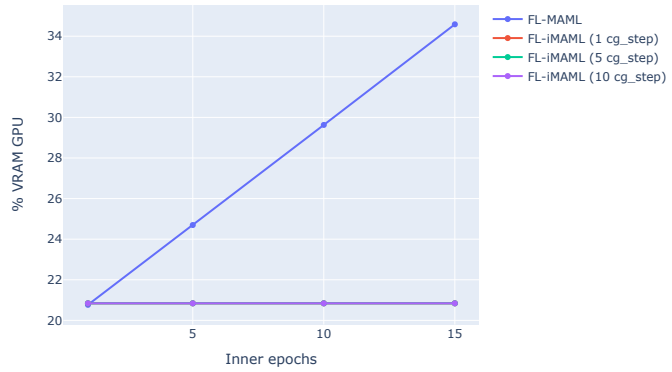


Fig. 5: Lượng VRAM của GPU của Fed-iMAML và Per-FedAvg trên dữ liệu MNIST

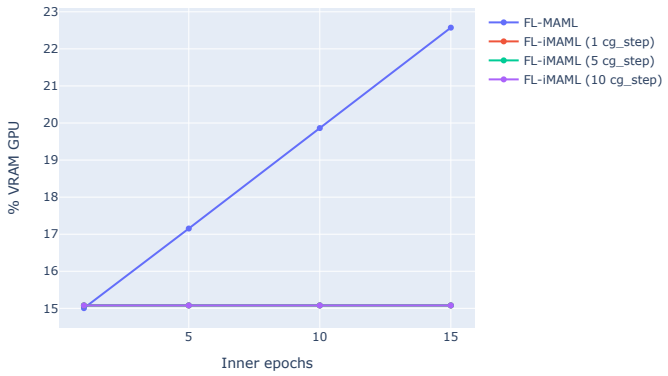


Fig. 6: Lượng VRAM của GPU của Fed-iMAML và Per-FedAvg trên dữ liệu MNIST

hơn 800.000 ảnh đen trắng cỡ $28 \times 28$. Trong tập dữ liệu này chứa hơn 3000 thiết bị biên nhưng em chỉ dự kiến sẽ dùng khoảng 500 thiết bị biên.

– Các ý tưởng tối ưu khác như tối ưu quá trình tổng hợp trọng số, sử dụng lớp cá nhân hóa và tối ưu cho phần cứng của từng thiết bị cũng đang được xem xét để tích hợp vào thuật toán.

## References

1. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
2. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)