Federated Learning System for CIFAR-10 Classification with Web Deployment

Ahlam Abu Mismar
Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois
Email: aabumismar@hawk.iit.edu

***Abstract*—This paper presents a complete Federated Learning (FL) system implementing the FedAvg algorithm for CIFAR-10 image classification. The retrained system improves accuracy and stability by increasing the Dirichlet parameter to $\alpha = 10.0$, selecting 12 clients per round, training each for 3 local epochs, and applying a learning rate of 0.1 with weighted FedAvg aggregation. These optimizations enhanced client balance and accelerated convergence, allowing the global model to reach 70% test accuracy by Round 25. The trained model is deployed as a FastAPI web application containerized with Docker on Chameleon Cloud.**

I. Introduction

Federated Learning enables distributed devices to collaboratively train a shared model while preserving data privacy. This project implements the FedAvg algorithm as part of CS 595 Assignment 2, focusing on decentralized learning under non-IID data distributions. The system follows the assignment requirements: simulating multiple clients, applying Dirichlet-based partitioning, performing concurrent training using `ThreadPoolExecutor`, and deploying the trained model through a containerized FastAPI application on Chameleon Cloud.

The retrained version expands the experiment to achieve higher accuracy and smoother training by adjusting key parameters and introducing weighted aggregation. This implementation demonstrates the practical integration of machine learning, parallel computing, and web deployment within a unified pipeline.

II. Methodology

A. Federated Learning Simulation

The FL simulation consists of 64 clients, with 12 selected randomly each communication round. Each client trains locally for three epochs using a batch size of 64. The local updates are combined using weighted FedAvg, where client contributions are proportional to their dataset sizes. This aggregation reduces bias from clients with fewer samples and produces a more stable global model update.

Non-IID client datasets are generated using Dirichlet partitioning with $\alpha = 10.0$, which produces more balanced label distributions across clients than the earlier $\alpha = 0.3$ setting. The optimizer is stochastic gradient descent (SGD) with a learning rate of 0.1, momentum 0.9, and weight decay $1\mathrm{e}{-4}$. The training continues until the global test accuracy exceeds 70%.

B. Web Application Deployment

A FastAPI web application is developed to serve the trained model for real-time CIFAR-10 predictions. The application includes REST API endpoints and an interactive interface where users can upload images for classification. It is containerized using Docker, exposing port 8080 for external access on Chameleon Cloud. This ensures reproducibility, accessibility, and compliance with the deployment requirement of the assignment.

III. Experimental Results

A. Non-IID Data Distribution

Figure 1 shows the Dirichlet-based non-IID data distribution across 64 clients with $\alpha = 10.0$. The smoother label spread results in fairer participation among clients, reducing extreme class imbalance and improving gradient consistency.

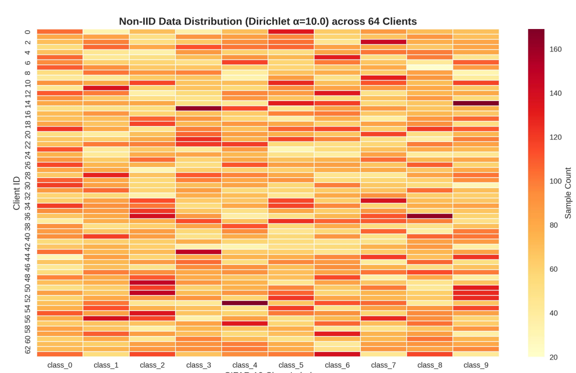

Fig. 1: Optimized non-IID data distribution across 64 clients ($\alpha = 10.0$).

B. Learning Performance

Figure 2 presents the test accuracy across communication rounds. The retrained configuration significantly improved performance, achieving 70% by Round 25. Weighted aggregation and higher α mitigated client divergence and produced smoother convergence compared to the original setup (10 rounds, $\alpha = 0.3$).

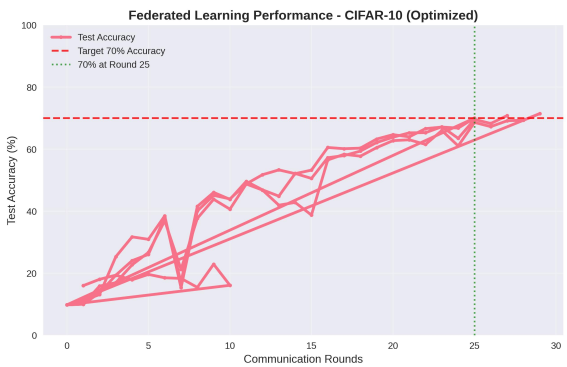

Fig. 2: Learning curve showing global test accuracy reaching 70% at Round 25.

C. Web Application Interface

Figure 3 displays the deployed FastAPI web interface, which enables users to upload CIFAR-10 images and visualize classification results. The interface was designed to be lightweight, responsive, and robust against invalid inputs, supporting multiple image formats.

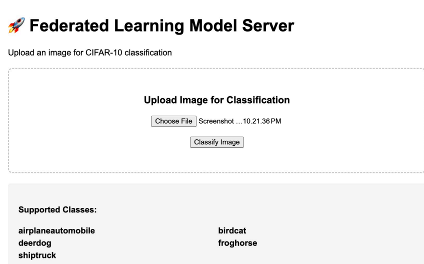

Fig. 3: Web application interface for image classification.

D. Classification Results

Figure 4 illustrates a sample prediction output. The model successfully classifies input images and displays probability distributions for all classes, validating both the FL training process and model deployment.

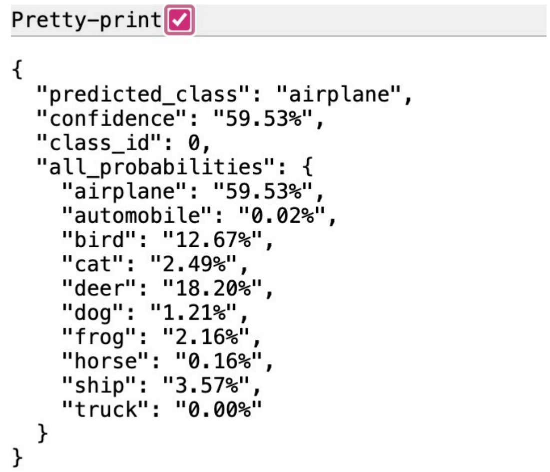

Fig. 4: Classification results with predicted labels and confidence scores.

IV. Conclusion

This project fully satisfies the assignment's requirements by implementing a complete Federated Learning system with end-to-end deployment. The retrained model achieved faster convergence and higher accuracy through improved parameter tuning and weighted FedAvg. The FastAPI-based web application demonstrates real-time inference, proving the system's usability beyond simulation.

The work highlights how adjusting data balance and aggregation strategies can significantly impact FL model performance. Future work may involve integrating differential privacy, dynamic client selection, or adaptive learning rate scheduling to further enhance efficiency and security.

References

[1] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.