Federated Learning Pipeline for Image Classification on MNIST and CIFAR10

Nader Nemati
Turku Applied Science
Brief Report of the Assignment
Email: naderr.nemati@gmail.com

I. Introduction

Federated Learning (FL) is a distributed machine learning approach that enables multiple clients, such as mobile devices or institutions, to collaboratively train a shared model without sharing raw data. This is especially useful in scenarios where data privacy is crucial, such as healthcare or finance, where sharing sensitive information is not permissible.

In this project, we implement a Federated Learning pipeline using the Flower framework and PyTorch, enabling image classification on both the MNIST and CIFAR10 datasets. The pipeline ensures privacy by allowing clients to train models locally on their data partitions and only share model updates (weights) with a central server. The server aggregates these updates using the Federated Averaging (FedAvg) algorithm, forming a global model that is shared back with the clients for further local training.

II. Project Overview

This project builds a Federal Learning (FL) pipeline for collaborative training on the MNIST and CIFAR10 datasets. The key features of the project include client-side privacy, dynamic model selection, and centralized aggregation with Federated Average.

A. Client-Side Training with Privacy

Clients train models locally, sending only model updates (weights) to the server. To ensure data privacy, Differential Privacy (DP) adds noise to the model updates, and Secure Aggregation ensures that individual client updates remain private during the aggregation process.

B. Training and Validation Strategy

Data is split IID (Independent and Identically Distributed) between clients, making each client's data representative of the entire data set. 10% of the data is used for local validation and the global model is evaluated on the complete test data set after each training round.

C. Dynamic Model Selection

The pipeline supports both the MNIST and CIFAR10 datasets, with models that adapt to the complexity of each dataset:

MNISTNet: A simple model for MNIST's grayscale images with two convolutional layers.

CIFAR10Net: A more complex model with deeper layers, batch normalization, and dropout for RGB images in CIFAR10. It uses gradient clipping to prevent exploding gradients.

TABLE I
MNISTNet Architecture

Layer	Input Shape	Output Shape
Conv2D	(1, 28, 28)	(32, 24, 24)
MaxPool2D	(32, 24, 24)	(32, 12, 12)
Conv2D	(32, 12, 12)	(64, 8, 8)
MaxPool2D	(64, 8, 8)	(64, 4, 4)
Flatten	(64, 4, 4)	(1024)
Fully Connected	(1024)	(120)
Fully Connected	(120)	(84)
Fully Connected	(84)	(10)

TABLE II
CIFAR10Net Architecture

Layer	Input Shape	Output Shape
Conv2D	(3, 32, 32)	(64, 32, 32)
BatchNorm2D	(64, 32, 32)	(64, 32, 32)
MaxPool2D	(64, 32, 32)	(64, 16, 16)
Conv2D	(64, 16, 16)	(128, 16, 16)
BatchNorm2D	(128, 16, 16)	(128, 16, 16)
MaxPool2D	(128, 16, 16)	(128, 8, 8)
Conv2D	(128, 8, 8)	(256, 8, 8)
BatchNorm2D	(256, 8, 8)	(256, 8, 8)
MaxPool2D	(256, 8, 8)	(256, 4, 4)
Flatten	(256, 4, 4)	(4096)
Fully Connected	(4096)	(512)
Dropout (0.5)	(512)	(512)
Fully Connected	(512)	(256)
Fully Connected	(256)	(10)

III. Modular Design

The project is organized into modular components, including:

- **client.py:** Defines the client-side logic for federated learning.
- **dataset.py:** Handles data loading, partitioning, and preprocessing.
- **model.py:** Contains model definitions for both MNIST and CIFAR10.
- **server.py:** Manages server-side aggregation and communication.

- **main.py:** Orchestrates the federated learning simulation using the Flower framework.

IV. Results and Analysis

Table III shows the performance of the federated learning model for MNIST and CIFAR10 after multiple communication rounds.

TABLE III
Performance Summary for MNIST and CIFAR10

Metric	MNIST		CIFAR10	
Round	1	2	1	2
Loss (Centralized)	28.25	2.72	180.88	125.42
Accuracy (Centralized)	96.69%	98.96%	12.40%	45.48%
Distributed Loss	14.07	1.52	73.41	52.62
Distributed Accuracy	96.20%	98.58%	11.34%	43.70%

A. *MNIST Analysis*

The MNIST dataset performs exceptionally well with the federated learning model, achieving near-perfect accuracy (98.96%) after just two rounds. This is due to the simplicity of the dataset and the model's ability to quickly converge.

B. *CIFAR10 Analysis*

The CIFAR10 dataset presents more challenges. After two rounds, the accuracy is only 45.48%, which is significantly lower than MNIST. This reflects the complexity of CIFAR10's RGB images and the need for more training rounds or model improvements.

V. Conclusion

The Federated Learning Pipeline demonstrates the effectiveness of federated learning on simpler datasets like MNIST while highlighting the challenges of more complex datasets like CIFAR10. The modular design and privacy-preserving techniques make this a robust solution for real-world applications where data privacy is critical.