Assignment 2 - Machine Learning

Topic: Federated Learning Simulation

Members:

1) Shantanu Ambekar (2021A7PS2540P)
2) Dhruv Shrimali (2021A7PS0008P)
3) Rudra Jewalikar (2021A7PS0450P)

Table of Contents

1. Introduction to Federated Learning

Federated Learning (FL) stands at the forefront of modern machine learning, offering a transformative solution to two critical challenges prevalent in centralized models: communication overhead and data privacy. Traditional approaches necessitate centralizing data onto servers, posing privacy risks and communication bottlenecks, particularly in the era of Big Data. FL revolutionizes this landscape by enabling collaborative model training across decentralized devices, preserving data privacy by keeping it local.

2. Problem Statement and Objective

This report undertakes a comparative study between Centralized Machine Learning and Federated Learning methodologies in the context of animal image classification. FL's unique approach involves distributing a base model to individual devices (representing clients), allowing them to improve it using their local data without sharing sensitive information. Only model updates, not raw data, are communicated to a central server for aggregation.

The objective is to analyze and contrast these methodologies in terms of communication costs and model accuracy. FL's promise lies in its ability to refine models while respecting data privacy—a critical factor in domains such as healthcare, finance, and e-business. This comparative study aims to elucidate the trade-offs and implications of adopting FL over traditional centralization in machine learning.

3. Methodology

3.1 Data Details

The dataset utilized for this study was derived from the ANIMAL10 dataset, focusing on animal image classification tasks. The derived dataset comprises images featuring three distinct animal classes: dogs, chickens, and spiders. The distribution of images across classes is as follows:

- Chicken Class: This class contains 3098 high-resolution photos showcasing various chicken breeds and poses.
- Dog Class: Comprising 4863 images capturing diverse breeds and contexts, the dog class encapsulates a wide spectrum of dog-related images.
- Spider Class: This class includes 4821 images showcasing different types of spiders in various settings.

In total, approximately 1500 images have been systematically segregated from the dataset to form a distinct testing subset, ensuring an independent evaluation of model performance.

The images within the dataset are in full color, offering rich visual information for classification tasks. Each class represents a varied collection of images capturing different angles, lighting conditions, and backgrounds, presenting a comprehensive and diverse dataset for animal image classification experimentation.

3.2 Experimental Setup

To simulate numerous clients, we first divide the initial dataset into numerous smaller datasets and hence each smaller dataset acts as a kind of node which can provide data to the model. The data is not sent to the model instead the model acts on the data and only saves the parameters generated from the model.

3.3 Implementation Details

The code defines a simple neural network using PyTorch. It consists of two fully connected layers with a ReLU activation function in between. The network takes input data, flattens it, passes it through a hidden layer of 128 neurons with ReLU

activation, and produces the final output with the specified number of classes (3 in our case).
This is the base deep learning model (or the central server model). The federated learning model uses this as the main model and the results of the federated learning are compared to this model.

To simulate the federated learning approach, the dataset is first split into smaller dataset folders. We have simulated 100 clients considering the small size of the data. To simulate 1000 clients, we would need a very large dataset. Out of the 100 clients, 10 were selected at random in each iteration (simulating selection based on battery percentage, connectivity, etc). Nevertheless, the approach will remain the same, only the parameter needs to be changed.

After the dataset is split, the following steps are used to simulate federated learning-

1) The model is taken to the first client.
2) The model is trained on that client for the required number of epochs
3) The model parameters are saved in a list
4) This continues for all the clients with the parameters being saved in the list
5) At the end of the first iteration, the model parameters are averaged and the new model parameter is set to this value.
6) The parameter list is cleared and the process continues for the required number of iterations.
7) At the end of this process, the final model parameters are saved in the saved_model.pth file.

Testing steps

1) The model parameters are saved in the saved_model.pth file are now utilized to test the accuracy of the code.
2) The dataset was initially separated into the test and train datasets.
3) The model with these parameters is then used to calculate the accuracy of the model

Other points

1) In order to increase the effectiveness of the model, we are modifying the learning rate in every iteration. At the beginning, the learning rate is high and then progressively reduces at every iteration. This is a proven method to increase the efficiency of models.
2) In federated learning, the clients are selected based on their battery percentage and other parameters. However, in this project we are simulating the clients using different dataset folders. Hence these parameters cannot be used. Instead we can randomly choose a certain number of clients which makes it seem as if some screening of clients is taking place.

Comparison of centralized and federated machine learning model

Parameters of the centralized model

Epochs	Learning rate	Clients	Accuracy (train dataset)	Accuracy(test dataset)
5	0.01	10	55.085%	53.96%

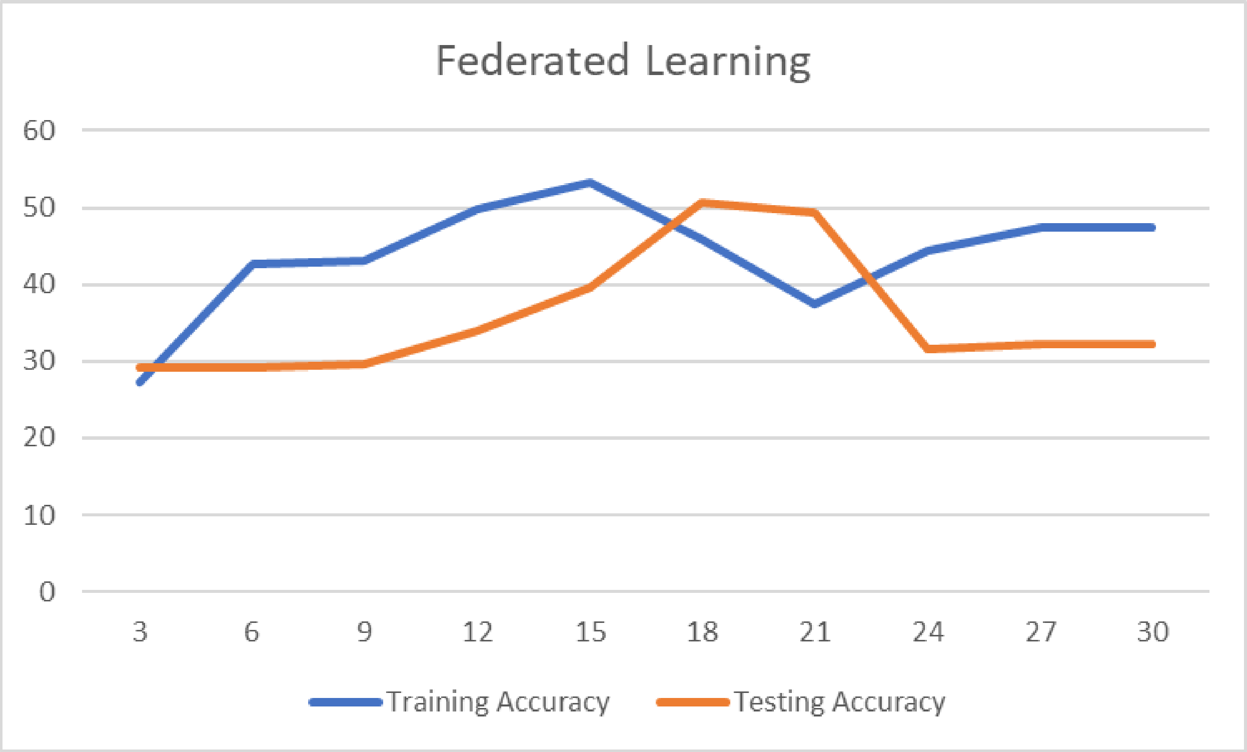
Parameters of the federated model

Epochs	Learning rate	Iterations	Clients	Accuracy (train dataset)	Accuracy (test dataset)
5	0.001	18	10	45.82%	42.96%

As can be observed from the above tables, the test accuracy of the centralized model is higher than the federated model by nearly 10 percentage points.

The lower accuracy of federated models compared to traditional machine learning models may be attributed to several factors inherent to the federated learning paradigm:

1) The limited local data on individual datasets may result in models that are less representative of the entire dataset.
2) The way models are aggregated at the central model impacts performance too. In this case, we are averaging the model weights after each iteration. This may not represent the data in the right way which can reduce the accuracy.



Comparing Communication Costs: Federated Learning vs. Centralized Learning

Data Transmission per FL Iteration

In Federated Learning (FL), each device communicates model updates, amounting to a mere 88 bytes in and out per iteration. This minimal data transfer involves sharing updated model parameters without transmitting raw data.

Local Dataset per Device

FL devices hold approximately 810 images for training, contributing to local model updates during FL rounds.

Centralized Learning Data Import

Conversely, Centralized Learning requires importing the entire dataset, estimated at 1.13GB, to a central server. This centralized approach demands substantial data transmission for processing.

Comparative Efficiency

FL's communication strategy minimizes data transfer by sharing lightweight model updates. In contrast, Centralized Learning faces substantial data import challenges, necessitating the transmission of the entire dataset to a central server.

This difference highlights FL's efficiency in reducing communication overhead by transmitting minimal data per iteration, addressing the traditional bottleneck associated with centralized learning approaches.