Author: Aidan Cowan
Date: 03/19/24
Github Repo: https://github.com/Adn02/Fed-Learning/ MNIST-FedAvg_and_Scheduling.ipynb

Replicating Results from *"Communication-Efficient Learning of Deep Networks from Decentralized Data"*

Introduction

This report documents the results from replicating the MNIST CNN model and Federated Averaging algorithm as discussed in '*Communication-Efficient Learning of Deep Networks from Decentralized Data'*. It also mentions an analysis of IID versus Non-IID Data Distrubtions as well as Random versus Age-Based scheduling techniques.

Background Information

IID (Independent and Identically Distributed) data refers to each sample in a dataset having no relation to another sample and each sample having the same probability distribution hence independent and identically distributed.

Non-IID (Non-Independent and Identically Distributed) data refers to a dataset where samples might have correlation or dependence on other samples thus the presence of one sample may be related to or may influence the presence of another sample. This results in an unbalance or uneven distribution in the dataset as mentioned in *Communication-Efficient Learning of Deep Networks from Decentralized Data*. Clients' local dataset will depend on mobile device usage and thus not be an accurate representative of the population distribution (McMahan et al. 2).

Federated Averaging is a Federated Learning technique that allows for better generalization when compared to FedSGD (Federated Stochastic Gradient Descent). It accomplishes this by allowing clients to perform more than one local gradient descent update and when the server aggregates the clients, only the clients' weights are shared and averaged together. The average of the clients' weights will then update the global modals parameters and proceed to the next round.

The paper provides the psuedocode for the FederatedAveraging algorithm as follows,

Server executes:
  initialize w_0
  for each round t = 1,2, … **do**
    $m \leftarrow max(C \cdot K, 1)$
    $S_t \leftarrow (random\ set\ of\ m\ clients$
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow ClientUpdate(k, w_t)$
    $m_t \leftarrow \sum_{k \in S_t} n_k$ (1)
    $m_{t+1} \sum_{k \in S_t} n_k\, w_{t+1}^k$

ClientUpdate(k,w):
  $\beta \leftarrow$ $(split\ P_k\ into\ batches\ of\ size\ B)$
  for each local epoch i from 1 to E **do**
    for batch $b \in \beta$ **do**
      $w \leftarrow w - \eta \nabla \ell(w: b)$
  return w to server

Author: Aidan Cowan
Date: 03/19/24
Github Repo: https://github.com/Adn02/Fed-Learning/ MNIST-FedAvg_and_Scheduling.ipynb

Clients were chosen each round of model aggregation based on a schedule. Two types of scheduling was considered, random scheduling and age-based scheduling. Random scheduling refers to clients being randomly selected to participate in each round without considering the availability, the performance, or other characteristics of the client. One problem with random scheduling prsents itself if a client has not been randomly chosen to participate in aggregation, then its local modal will be out of date compared to the most recently updated clients thus clients not chosen will eventually become stale. Age-based scheduling aims to do more by keeping track of each clients' age (better known as Age of Update) to ensure all clients partcipate, however, it may not prioritize clients with the freshest data.

Four global models were created and trained based on the two types of data distrubtions and scheduling methods, as previously mentioned. Each model had the same training experience with one hundred total clients and each round ten clients were selected for a total of 500 rounds, a learning rate of 0.1, a local batch size of ten, and five local epochs. For both IID and Non-IID the MNIST dataset was chosen to train on. The IID data was shuffled and then divided up across 100 clients each receiving 600 examples, and the Non-IID data was sorted by digit label, divided up into 200 'shards' of 300 examples, and then each client receieves 2 'shards' (McMahan et al. 5).

Results Found

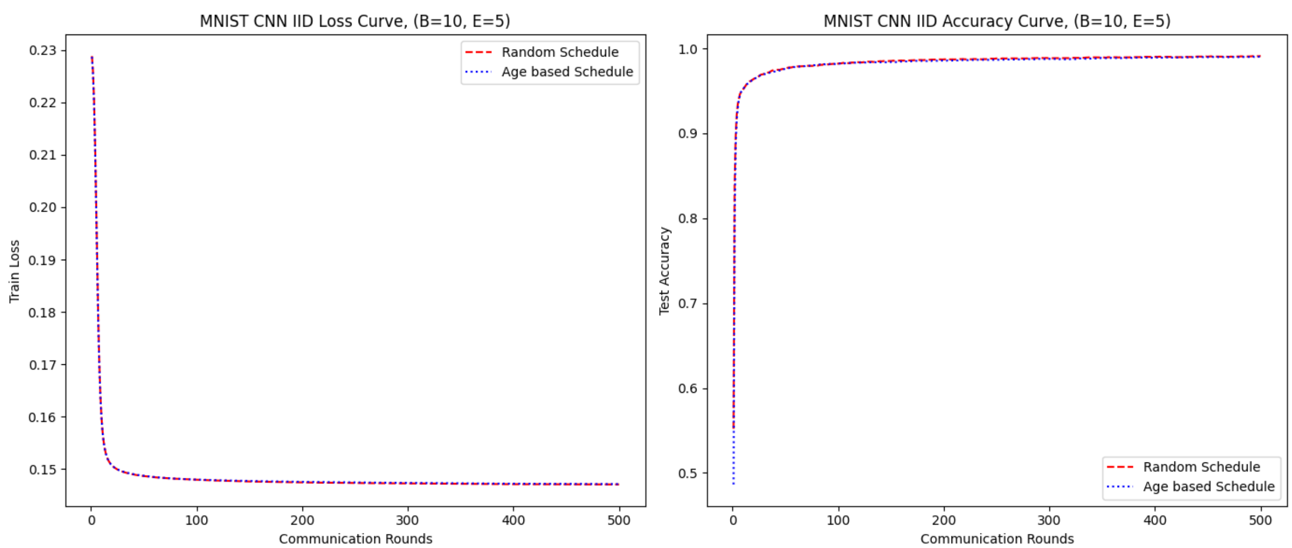

Figure 1: Training Loss & Test Accuracy for IID Models

Author: Aidan Cowan
Date: 03/19/24
Github Repo: https://github.com/Adn02/Fed-Learning/ MNIST-FedAvg_and_Scheduling.ipynb

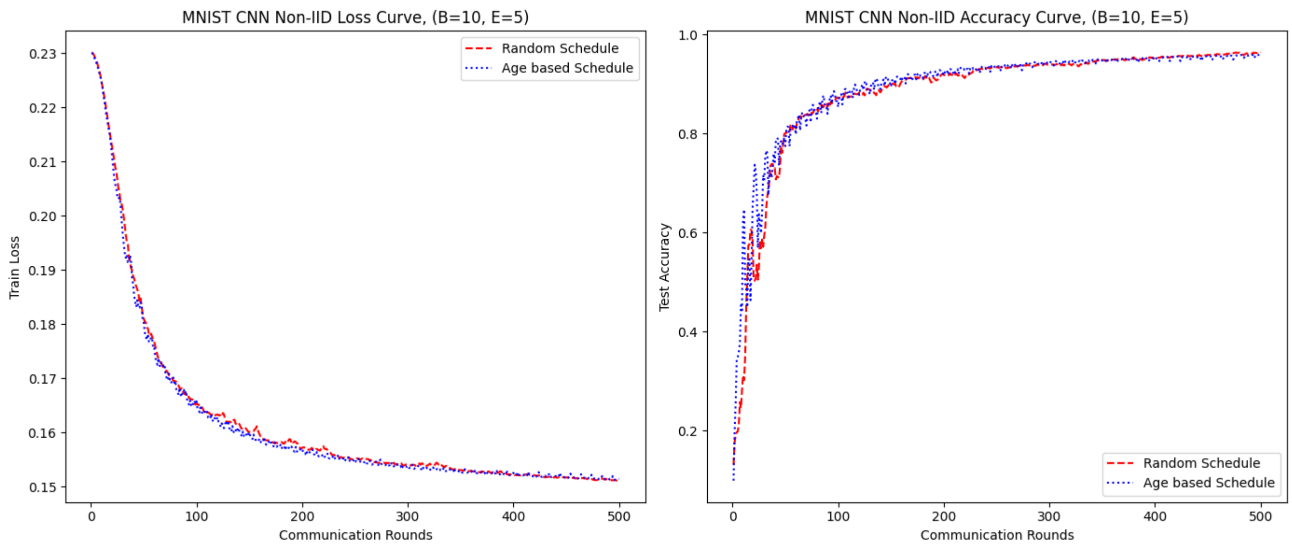

Figure 2: Training Loss & Test Accuracy for Non-IID Models

Model Type	**Final Test Accuracy**
IID Random Scheduling	99.09%
IID Age Based Scheduling	99.01%
Non-IID Random Scheduling	96.45%
Non-IID Age Based Scheduling	95.85%

After training, It was found that the models that trained on IID data did as expected. However, the models that trained on Non-IID data did not reach the expected accuracies as described in *Communication-Efficient Learning of Deep Networks from Decentralized Data.* Originally planned for 1000 communication rounds, the training was limited to 500 rounds due to hardware constraints and time limitations. From the graphs above, training on Non-IID data displayed a lower overall test accuracy compared to the IID models. However, more communication rounds could be considered to let the loss curve fully converge similar to the IID models above it. In regards to scheduling, Age-based scheduling showed a higher overall accuracy per communication round and lower loss in comparison to random scheduling however the Non-IID model's accuracy curve does not seem to be as stable as the model that utilized the random scheduling.

References

(1) McMahan, H. B., et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data." ArXiv, 2016, /abs/1602.05629. Accessed 19 Mar. 2024.