CS 726 Course Project
Survey of Client Selection Methods in Federated Learning

Team: Honey, I Shrunk the Model

Neelkamal Bhuyan (180020058)
Neilabh Banzal (170010014)
Shaun Zacharia (180110073)
Shreya Laddha (180070054)

July 4, 2022

Code Repository

Problem Statement

Federated Learning is a form of distributed learning with the key challenge being the non-identically distributed nature of the data in the participating clients. The popular federated averaging algorithm involves randomly selecting K clients out of a total of N available clients and then averaging the weight updates sent by the selected clients after a round's training. We plan to explore different kinds of unbiased and biased client selection algorithms and compare their performance with FedAvg. By biased (unbiased) we mean that the server will (not) use any local training information of the clients. We plan to use the LEAF Federated Learning Benchmark for the datasets as well as codebase. We plan to start with Synthetic(α, β) dataset and CelabA dataset as they are less computationally expensive and then we will move on to the NLP datasets (shakespeare and reddit). We will compare client selection policies based on the evolution of training/validation accuracy during the training process.

Introduction & Motivation

Federated Learning (FL) is a distributed optimization paradigm that enables a large number of resource-limited client nodes to cooperatively train a model without data sharing. It allows continual learning on end-user devices while ensuring that end user data does not leave end-user devices. There are a variety of motivations behind, e.g., maintaining privacy by avoiding individuals revealing their personal data to others, or latency by allowing data processing to take place closer to where and when the data is generated.

In FL, typical participating clients are energy-restricted mobile devices, and thus energy efficiency is a key challenge. Unlike traditional distributed systems, the data on clients are most likely heterogeneous, i.e. non-identical independent distributed (non-IID), and the data are often not equally split among clients. Different FL clients thus have an unequal effect on training. By only selecting the clients that can contribute most significantly to participate in training, the system can reduce energy cost in communication while maintaining the model accuracy. Several works have analyzed the convergence of federated learning by accounting of data heterogeneity, communication and computation limitations, and partial client participation. However, currently the most widely used client selection method is still random selection, and the research on client selection strategy is not well explored yet.

Literature and Code Survey

A typical Federated Learning process using the averaging algorithm, known as FedAvg, mentioned in the paper by Brendan McMahan et al. [1] has a server that coordinates the training process, by first selecting a set of clients, followed by having the clients download the training procedure and model weights. After this, each client performs an individual update to the model by executing the training procedure and then the server collects an aggregate of the device updates. Here, it may even drop stragglers once enough devices have performed updates and finally, the server updates the shared model based on the aggregate obtained from the clients.

One common bias that may occur is that minority groups are under-represented in training datasets, which leads to poor performance of our models on inputs belonging to these groups as demonstrated by Joy Buolamwini et al. [2]. We intend to compare performance of different client selection strategies to see the performance we can obtain on baised and unbaised datasets.

For our literature survey, we went through the papers mentioned below. The paper Yae Jee Cho et al. [3] talks about convergence analysis of federated optimization for biased client selection strategies, and quantifies how the selection bias affects convergence speed. It demonstrates that biasing client selection towards clients with higher local loss achieves faster error convergence. Fair resource allocation is discussed in the paper by Tian Li et al. [4]. The paper by Takayuki Nishio et al. [5] discusses a strategy which leverages distributed client data and computation resources for training high-performance ML models while preserving client privacy and the paper by Zhao et al. [6] talks about a client selection framework that finds the clients that provide significant information in each round of FL training with low energy cost.

Datasets

We plan to use LEAF [7] benchmark datasets in our project. Currently, LEAF consists of six datasets:

- *Federated Extended MNIST (FEMNIST)*, which is built by partitioning the data in Extended MNIST [8, 9] based on the writer of the digit/character.
- *Sentiment140* [10], an automatically generated sentiment analysis dataset that annotates tweets based on the emoticons present in them. Each device is a different twitter user.
- *Shakespeare*, a dataset built from The Complete Works of William Shakespeare. Here, each speaking role in each play is considered a different device.
- *CelebA*, which partitions the Large-scale CelebFaces Attributes Dataset [11] by the celebrity on the picture.
- *Reddit*, where preprocessed comments are posted on the social network on December 2017.
- A *Synthetic* dataset [4], which modifies the synthetic dataset presented in [4] to make it more challenging for current meta-learning methods. It involves labelled feature vectors that can be non-identically or identically distributed across the clients. The variation in models is decided by α and β introduces non-IID nature in the local data of the devices.

FedAvg Training Algorithm

Algorithm 1 Pseudo-code for FedAvg

Input: $\mathcal{S}$, K
procedure FEDAVG($\mathcal{S}$, K)
    $\mathbf{w}_0^k = \overline{\mathbf{w}}_0 \ \forall \ k$
    for $t \in \{1, 2, .., T\}$ **do**
        $\mathcal{S}^{(t)} \leftarrow$ subset of size K from $\mathcal{S}$, based on the client selection policy used
        for $\tau \in \{0, 1, .., E-1\}$ **do**
            $\mathbf{w}_{t+\tau}^k \leftarrow \mathbf{w}_{t+\tau-1}^k - \eta_{t+\tau-1} \nabla F_k(\mathbf{w}_{t+\tau-1}) \ \forall \ k \in \mathcal{S}^{(t)}$
        end for
        $t \leftarrow t + E - 1$
        $\overline{\mathbf{w}}_t \leftarrow \frac{1}{K} \sum_{k \in \mathcal{S}^{(t)}} p_k \mathbf{w}_t^k$
        $\mathbf{w}_t^k \leftarrow \overline{\mathbf{w}}_t \ \forall \ k$
    end for
end procedure

Client Selection Policies

Our main work is to provide a client selection function which takes in the set of available clients and the number of required clients. It outputs a subset of the set of available clients. Up till now, we have come up with the following client selection strategies,

1. **Biased selection**: Using a scoring mechanism to rank clients and then selecting them

   - **Policy 1** - Score = Average Training loss
     Selecting the K topmost clients with highest training loss out of N available clients. Below is the formula for the score,
     $$A_k(t) = p_k \times \frac{L_k(t)}{N_k(t)} \tag{1}$$
     where, $L_k(t)$ is cumulative loss of client k, $N_k(t)$ is number of times client k is selected upto round t and p_k is the proportion of total samples stored in client k.
   - **Policy 2** - Score = Average Training Loss + exploration term
     This is inspired by the UCB algorithm in the Reinforcement Learning Community. Here we use the concept of Upper Confidence Bound to design the exploration term so that there is a good balance between exploitation (purely based on ranking) and exploration (purely random). Below is the formula for the score,
     $$A_k(t) = p_k \times \left(\frac{L_k(t)}{N_k(t)} + \sqrt{\frac{2\log(t)}{N_k(t)}} \right) \tag{2}$$
     where, $L_k(t)$ is cumulative loss of client k, $N_k(t)$ is number of times client k is selected upto round t and p_k is the proportion of total samples stored in client k.

2. **Unbiased selection**: selecting clients randomly but not the same as FedAvg

   - **Policy 3** - We select subset S_{t+1} of round $t+1$ such that $S_{t+1} \cap S_t = \phi$. We strictly do not select the clients sampled in the previous round. This allows for greater exploration as it removes the possibility of a client being selected in back-to-back rounds

Implementation

Trying with $N = 100$ clients and choosing 5 clients per round. The model we are training is a multi-class logistic regression for 60 dimensional vectors and 5 classes. We generate it through the Synthetic(α, β) dataset with $\alpha = 1$ and $\beta = 1$. Note that the policy 1 and policy 2 graphs coincide.

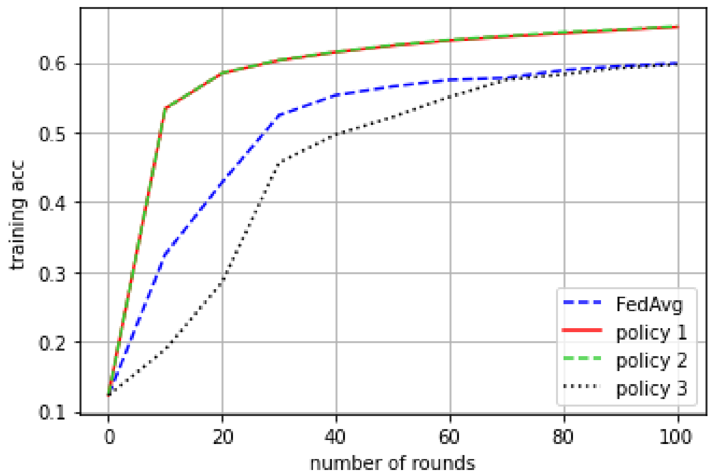

Figure 1: Comparing training accuracy of policies over time

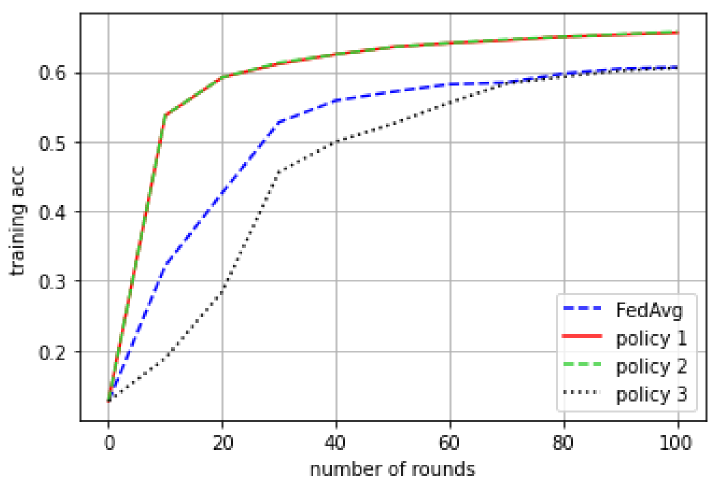

Figure 2: Comparing testing accuracy of policies over time

Observations

Based on Fig. 1 and Fig. 2, we make the following observations:

1. Score-based ranking policies perform better than unbiased selection policies
2. Adding the exploration term did not make a significant difference in this problem
3. Among unbiased selection policies, sampling without replacement is worse than FedAvg, which is sampling with replacement

References

[1] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[2] Joy Buolamwini and Timnit Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91.

[3] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies". In: *arXiv preprint arXiv:2010.01243* (2020).

[4] Tian Li et al. "Fair resource allocation in federated learning". In: *arXiv preprint arXiv:1905.10497* (2019).

[5] Takayuki Nishio and Ryo Yonetani. "Client selection for federated learning with heterogeneous resources in mobile edge". In: *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE. 2019, pp. 1–7.

[6] Jianxin Zhao et al. "Energy-efficient client selection in federated learning with heterogeneous data on edge". In: *Peer-to-Peer Networking and Applications* (2022), pp. 1–13.

[7] Sebastian Caldas et al. "Leaf: A benchmark for federated settings". In: *arXiv preprint arXiv:1812.01097* (2018).

[8] Gregory Cohen et al. "EMNIST: Extending MNIST to handwritten letters". In: *2017 international joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[9] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998).

[10] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report, Stanford* 1.12 (2009), p. 2009.

[11] Ziwei Liu et al. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.