Personalized Federated Learning through Collaboration with Nearest Neighbors

Yuhong Luo
University of Massachusetts
yuhongluo@umass.edu

Kunjal Panchal
University of Massachusetts
kpanchal@umass.edu

Hui Guan
University of Massachusetts
huiguan@cs.umass.edu

ABSTRACT

Federated learning (FL) aims to collectively train a global model while preserving the data privacy for clients. However, due to data heterogeneity, the global model usually only works well for a subset of clients but not for the others. Personalization in FL has become an important research area. There is a stream of works inspired by clustering that tries to model various data distributions with multiple global models. However, since each cluster has a fixed center and each client needs to interact with all clusters during optimization these approaches are not flexible and not generalizable, especially if a client is not close to any centers. We propose FedAttn that encourages each client to find nearest neighbors, whose data distributions are similar, for collaborations. It collectively learns different personalized models with respect to different clients. We evaluate FedAttn and find that it achieves comparable performance with baseline, but it does not have a stable optimization and we propose some ideas for improvements.

KEYWORDS

Personalized federated learning, Self-attention

ACM Reference Format:
Yuhong Luo, Kunjal Panchal, and Hui Guan. 2022. Personalized Federated Learning through Collaboration with Nearest Neighbors. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Federated learning (FL) aims to improve machine learning models by encouraging collaborations among a set of clients while preserving their private data in their local devices. The initial methods start from aggregating the gradients from local models of each client to achieve a global model that performs well across clients. The effectiveness of the model is evaluated according to the overall performance of all clients. Suppose every client has a similar data distribution, then the global model is well fit for all clients naturally. In reality, however, it is common for clients to have different data distributions and the amount of data for each distribution differs widely. For example, consider the EMNIST [4] dataset that records the writing of letters and digits. Different clients may have different handwritings. The model needs to adjust accordingly so that each client can achieve equally high performances. However, traditional FL methods aggregating the gradients of different clients into one global model update usually favor distributions with more data and thus, the clients that provide those data.

This is a final report for Yuhong Luo's independent study with Professor Hui Guan during Fall 2022 at University of Massachusetts.

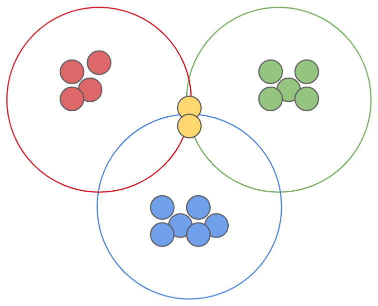

Figure 1: A toy example illustrating the clusters of clients in FL. Each dot is an individual client and different colors represents different data distributions. The circles are the clusters learned during FL training.

Some recent works propose to focus on personalization by learning multiple global models, each representing one of the data distributions. Specifically, some works group clients into clusters and encourage clients to collaborate for models that represent the clusters. Some other works assume each client has data under a mixture of distributions such that soft clustering is achieved. In most of these clustering approaches, the model performance is sensitive to the number of clusters.

We suspect that the clustering methods are still limited if we do not know how many cluster we need to expect. Furthermore, if we introduce a new client that is not close to any centers of the clusters, all models cannot represent the client well. For example, in Figure 1, suppose we learn three global models, one for each clusters. The yellow clients are not close to any clusters. In hard clustering methods, they are likely to be randomly assigned to one of the clusters. While in soft clustering, it is going to take the average of each model. They do not benefit from personalization in either approaches.

We propose FedAttn that aims to address these issues. It abandons the idea of using one or multiple global models to fit all clients. Instead, it places each client at the center and retrieve similar clients for collaboration. We calculate a personalized model for each client with the following steps: 1. Sample multiple nearest neighbors whose data distributions are similar. 2. Use self-attention to aggregate the personalized models of the sampled clients.

This idea makes personalization more flexible and generalizable, even for new clients with data distributions different from the majority. Lets use Figure 1 again as an example, when the yellow clients draw nearby clients for collaborations, they can actually learn directly from clients with the same distribution. Since we do not require each client to be involved in multiple global models, it is also much faster than clustering approaches.

We evaluate FedAttn and find that it achieves comparable performance with baseline FedAvg [15]. However, it does not have a stable optimization and may fail to converge sometimes. We suspect it is because using a machine learning model such as self-attention to directly find the parameters for the personalized models is unreliable. The parameters for the personalized models are too sensitive and they are difficult to be produced by another optimization model. We propose some ideas for improvements using regularization or Meta Learning approaches.

2 RELATED WORKS

Traditional Federated Learning (FL) methods such as FedAvg [15] focus on having all clients collaboratively train a machine learning model without sharing the private data that is stored in the local devices. However, such a ML model does not take into account the data heterogeneity such that the model may result in unbalanced performance across clients. In the recent years, a lot of researchers explore Personalized Federated Learning (PFL) [2, 3, 11, 13] to improve that. It assumes that different clients have private data under different distributions. Therefore, PFL needs to let clients not only collaboratively train one or multiple global models, but also be able to make adjustments according to the local data to achieve personalization.

Some methods including FedAMP, MOCHA, etc. [7, 18, 20] extend the idea of multi-task learning. They apply regularizations on client-client relationship such that the personalized models for similar clients have more limited differences.

Other methods try to directly model different distributions in different clients with the idea of clustering. [6, 10, 17] adopt hard clustering such that each client is assigned to one of the clusters whose global model generates the best performance. [12, 14, 16] are inspired by soft clustering, where each client has a data distribution under a mixture of multiple distributions with different probabilities.

The problem with clustering methods is that the performance of individual clients are highly dependent on not only what cluster they are assigned to but also on how close they are to the center of the clusters. We try to extend the clustering idea by placing each client at the center and search for similar clients for collaborations. Our motivation is similar to FedAMP [7] which also encourages clients with similar model parameters to have stronger collaborations. However, FedAMP uses regularization to guide the collaborations implicitly while we construct the personalized model based on similar clients more explicitly.

3 PRELIMINARIES

In this section, we formulate the personalized federated learning problem. We aim to train personalized models for a set of clients collectively while preserving the private data of the clients in the local devices.

Suppose there is a set of clients C, each client i has local data D_i. We aim to train personalized models $P = \{P_i | i \in C\}$ such that

$$P = \arg\min_P \sum_{i \in C} loss(P_i, D_i)$$

We define our personalized model P_i to consist of the global part G and personalized part M_i. In our experiment, we use FedAvg [15] to update the global part of the model. We keep all the biases in each layer of the model as our personalized part.

Definition 3.1 (Self-attention). Following the attention mechanism proposed in [19], given a query and keys of dimension d_k, and values of dimension d_v, it outputs the weighted average of values according to the relationship between the query and the keys. Formally, given a query vector q, a set of key-value pairs packed into matrices K and V,

$$Attention(q, K, V) = softmax\left(\frac{(W_K K)^T (W_q q)}{\sqrt{d_k}}\right) W_V V$$

where $W_q \in \mathbb{R}^{d_{model} \times d_k}$, $W_K \in \mathbb{R}^{d_{model} \times d_k}$, $W_q \in \mathbb{R}^{d_{model} \times d_v}$ are trainable parameters.

4 METHODOLOGY

In this section, we introduce our method FedAttn. Similar to other federated learning methods, to complete a round of training, the server first configures a new personalized model for each client on the next round. We then perform local update on the client side. Finally, the server collect the result from clients to perform updates. We explain each of the components in detail in the following paragraphs.

4.1 Constructing personalized models

We first discuss how to construct personalized models for each client we have chosen for the next round. It essentially constructs a new model with parameters calculated with self-attention (Def 3.1), aggregating the personalized models of similar clients.

Attention weighted biases. As mentioned in Section 3, our personalized model consists of a global part which is trained with FedAvg [15] and a personalized part, the biases. We calculate the biases with self-attention (Def 3.1) on the server side. For the queries and the keys of the attention module, we use client representations denoted as Z_i^t for the i-th client at the t-th round. Such representations aim to summarize the data distributions of individual clients and map to an embedding space. They can be calculated in many ways. For our experiment, we use the output of the second to last layer of the personalized model. For values, we keep the biases in personalized models, M_i^t. After the attention weighting, we get the updated model biases $\tilde{M_i^t}$. $\tilde{M_i^t}$ can be combined with the global model G^t by replacing the global biases to constitute the personalized model P_i^t. The global model can be trained with FedAvg [15] without any changes. Personalizing only the biases with attention can have several advantages: 1. It significantly reduces the cost of calculation and storage on server. 2. Using a global model as a

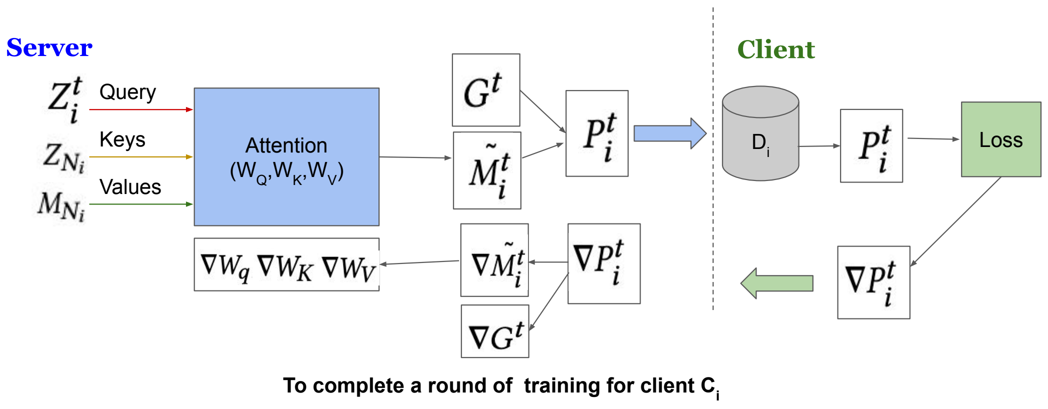

Figure 2: An illustration of completing a training round for a particular client C_i.

Algorithm 1: FedAttn

```
Input : Client set $C$, each with a set of private training data $\mathcal{D}_i$ for client $i$
1  Initialize $Z_1^0, ..., Z_{|C|}^0$ and $M_1^0, ..., M_{|C|}^0$ for each client;
2  for $t = 1, 2, ..., T$ do
3      Sample a set of clients $\mathcal{B}_t$ for round $t$;
4      for $i$ in $\mathcal{B}_t$ do
5          Find $K$ most similar clients according to distances with $Z_i^t$ and retrieve $Z_{N_i}$ and $M_{N_i}$;
6          Compute $\tilde{M}_i^t$ with $Attention(Z_i^t, Z_{N_i}, M_{N_i})$;
7          Construct $P_i^t$ with $G^t$ and $\tilde{M}_i^t$;
8          $Z_i^{t+1}, M_i^{t+1}, \nabla P_i^t \leftarrow localSolver(P_i^t)$;
9          Optimize $Attention$ with $\nabla \tilde{M}_i^t$ in $\nabla P_i^t$;
10         Optimize Global model $G$ with $\nabla G^t$ in $\nabla P_i^t$;
11 Function $localSolver(P_i^t)$ is
12     Optimize $P_i^t$ with local data $D_i$;
13     Return $Z_i^{t+1}, M_i^{t+1}, \nabla P_i^t$
```

backbone for all clients can stabilize the training and prevent the personalized models of clients to diverge.

Subsampling similar clients. We do not aggregate the personalized model biases of all clients. Instead, we only sample a small subset of clients that have similar data distributions. Since we do not know the true distributions of the data, we cannot calculate the KL-divergence. Hence, we simply compare the L2 distances of the client representations and get the nearest K clients. To implement this, we leverage the FAISS [9] library which is an optimized toolkit for approximate nearest neighbor search in embedding space. Formally, we denote $N_i = \{$the subsampled similar clients of client $i\}$. We pack the client representations of the similar clients into matrix Z_{N_i} and the model biases into M_{N_i}, which are used as keys and values for the attention respectively.

4.2 Gradient updates of the models

Although we perform the attention calculation on the server side while training the personalized models on the client side, they are actually chain together and can be seen as components of one model. Therefore, we can perform gradient update on both the personalized models and the attention module.

We explain the gradient descent mechanism with an example as shown in Figure 2. Suppose now we have configured the personalized model P_i^t for client i at round t which uses the weights $\tilde{M}_i^t$ as part of the model. Training the personalized model on client i locally introduces a loss L_i^t. Back-propagating based the loss, we get gradients ∇P_i^t which contains $\nabla \tilde{M}_i^t$ and $\nabla \tilde{G}^t$. $\nabla \tilde{M}_i^t$ can then be used as the gradients for further back-propagation on the attention module. Eventually we get ∇W_q, ∇W_K and ∇W_V, which can then be used to update the weights. $\nabla \tilde{G}^t$ is used to update the global model. Notice that we update the attention module on the server for all clients participated in the previous round in parallel after collecting the gradients from the clients. We show a summary of the training process in Alg 1.

5 EXPERIMENTS

In this section, we evaluate the performance of FedAttn and compare it with one baseline: FedAvg [15]. We show that FedAttn achieves comparable performance to the baseline. However, FedAttn shows some instability during the training. We discuss the improvements to be done toward the end of this section.

5.1 Datasets

We use one public benchmark dataset EMNIST for evaluation. the EMNIST dataset [4] partitions the data to 3400 clients. Since each partition is collected from a single device, the data is naturally heterogeneous.

5.2 Experiment Setup

We train all models including all clients for 1500 rounds. For each round, we sample 10 clients randomly and for each client, we train

Method	Dataset
FedAvg	88.27
FedAttn	88.46

Table 1: Performance in test accuracy. The mean of the test accuracy over the last 100 rounds.

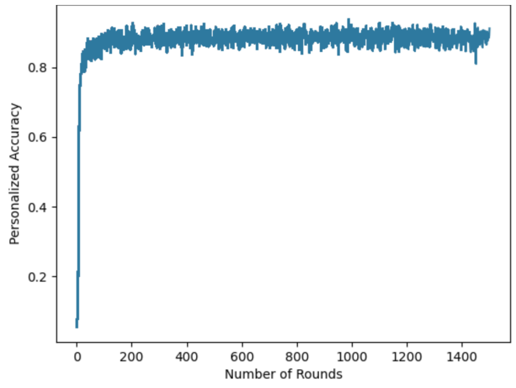

Figure 3: FedAvg training rounds v.s. personalized accuracy.

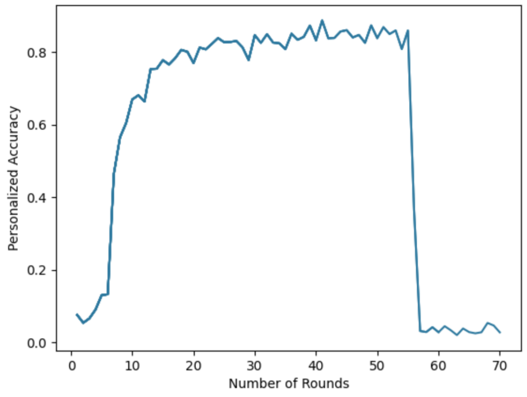

Figure 4: Performance of FedAttn that breaks at about round 58.

the model locally for 5 epochs. The test accuracy is calculated with the mean of the prediction accuracy over the last 100 rounds.

5.3 Results

We show the comparison between FedAttn and FedAvg in Table 1. The result shows that FedAttn has comparable performance with FedAvg. However, since FedAttn uses FedAvg as the backbone model for its global component, it is not clear whether the personalized biases is playing a role in helping the performance.

We further show that FedAttn is occasionally unstable. It sometimes produces some significant drops in performance during training. As seen in Figure 4 and 5, FedAttn training breaks at a random round suddenly. The performance also converges less smoothly compared to the traditional FedAvg (Figure 3).

We suspect that the model parameter such as the biases is very sensitive and if the attention module is producing biases that do not

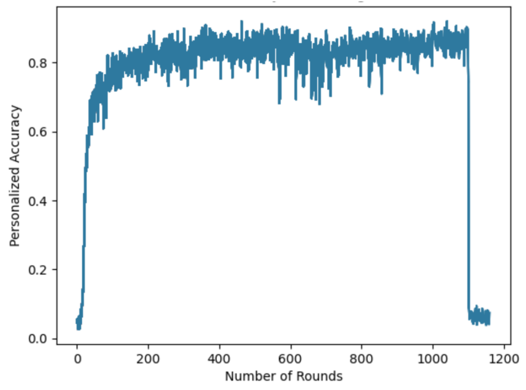

Figure 5: Performance of FedAttn that breaks at about round 1100.

fit with the result of the model, the personalization performance gets impacted significantly.

There might be several improvements we can make based on this observation. First, similar to MOCHA [18] and FedAMP [7], we can add regularizations to the personalized biases such that they do not diverge largely from each other and from the biases of the global model trained with FedAvg. Second, as inspired by Model Agnostic Meta Learning (MAML) [5], we can adopt a two-stage approach such that we only fine tune the biases for personalization when the global model is stabilized. For example, wait for the global model to converged with FedAvg training. There are some works [1, 8] that propose to use MAML for FL and achieve state-of-the-art.

6 CONCLUSION AND FUTURE WORKS

In this work, we proposed FedAttn, the first method that encourages collaborations with nearest neighbors. For each client, it samples clients with similar data distributions and uses self-attention to aggregate the personalized models of the sampled clients. In our experiment, we found that although FedAttn can achieve comparable performance with the baseline, the training is not stable and the performance cannot be guaranteed. For future work, we need to look into ways to stabilize the personalized training, such as using regularization or fine-tuning for personalization after a global model is trained.

REFERENCES

[1] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2022. FedAvg with Fine Tuning: Local Updates Lead to Representation Learning. *arXiv preprint 2205.13692* (2022).

[2] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive Personalized Federated Learning. *arXiv preprint arXiv:2003.13461* (2020).

[3] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).

[4] E. D. T. Federated. 2020a. TensorFlow Federated EMNIST Dataset (Online; accessed 11-Sept-2022). *https://www.tensorflow.org/federated/api docs/python/tff/simulation/datasets/emnist* (2020a).

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, Vol. 70. 1126–1135.

[6] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An Efficient Framework for Clustered Federated Learning. In *NeurIPS*.

[7] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data. In *AAAI*.

[8] Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. 2019. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *arXiv preprint 1909.12488* (2019).

[9] Jeff Johnson, Matthijs Douze, and Herve Jegou. 2017. Billion-scale similarity search with GPUs. In *IEEE Transactions on Big Data*.

[10] Yeongwoo Kim, Ezeddin Al Hakim, Johan Haraldson, Henrik Eriksson, José Mairton B. da Silva, and Carlo Fischione. 2020. Dynamic Clustering in Federated Learning. *arXiv preprint 2012.03788* (2020).

[11] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. 2020. Survey of personalization techniques for federated learning. *arXiv preprint arXiv:2003.08673* (2020).

[12] Chengxi Li, Gang Li, and Pramod K. Varshney. 2022. Federated Learning With Soft Clustering. In *IEEE INTERNET OF THINGS*.

[13] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* (2020).

[14] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated Multi-Task Learning under a Mixture of Distributions. In *NeurIPS*.

[15] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*. 1273–1282.

[16] Yichen Ruan and Carlee Joe-Wong. 2022. FedSoft: Soft Clustered Federated Learning with Proximal Local Updating. In *AAAI*.

[17] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2019. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. *arXiv preprint 1910.01991* (2019).

[18] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated Multi-Task Learning. In *Advances in Neural Information Processing Systems*. 4424–4434.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

[20] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. 2020. Salvaging Federated Learning by Local Adaptation. *arXiv preprint 2002.04758* (2020).