FedFTHA: A Fine-Tuning and Head Aggregation Method in Federated Learning

Abstract

Personalized Federated Learning (PFL) is a subfield of federated learning focused on generating personalized models for each client that adapt to their local data distribution. Traditional PFL methods often emphasize enhancing client personalization while neglecting server-side generalization. To address this issue, the Fine-Tuning and Head Aggregation method in Federated Learning (FedFTHA) is proposed. FedFTHA allows each client to maintain a personalized model head and fine-tune it after local updates. During training, these personalized heads are aggregated to form a generalized head for the global model, balancing client personalization and server generalization. The method's convergence under convex and non-convex conditions is proven, and its personalization and generalization performance is validated on benchmark datasets.

Keywords: federated learning, fine-tune, head aggregation, personalized federated learning (PFL).

1 Introduction

In the context of the rapid development of deep learning technologies, the widespread use of modern mobile devices, and the maturation of technologies like the Internet of Things (IoT), there is a growing demand for performing model computation and prediction on edge devices. Client data on edge devices is crucial for enhancing the performance of client-side deep learning models. In traditional machine learning paradigms, each device uploads local data to a central server where model training is conducted. Although distributed machine learning can accelerate the training of large datasets by distributing the process across multiple nodes, these methods do not adequately protect user privacy on edge devices and fail to effectively address the issue of non-independent and identically distributed (non-i.i.d.) data across different clients. This training strategy, which allows the global server to access all data, not only risks user data leakage but also increases vulnerability to attacks.

The emergence of federated learning effectively addresses these challenges. Federated learning is designed to protect user privacy by defining a decentralized training process, allowing users to train high-performance models without uploading local data. However, it is important to note that traditional federated learning methods (such as FedAvg) [8] do not guarantee convergence or good performance under all conditions. As the number of client data samples changes and the degree of data distribution differences (i.e., non-i.i.d.) increases, the difficulty of achieving convergence for the global model in

federated learning significantly increases. Even if the model eventually converges, its performance may degrade substantially. From the client's perspective, when the non-i.i.d. degree of client data intensifies, the trained global model may deviate significantly from local data, greatly reducing the motivation for users to participate in federated learning. Therefore, generating personalized models that adapt to each client's data distribution is a crucial issue that needs to be addressed.

Personalized federated learning (PFL) aims to generate personalized models for each user under heterogeneous data conditions. From the client perspective, PFL seeks to enhance model performance on local data through federated learning and personalization methods. In recent years, numerous studies on federated learning personalization have emerged, covering approaches such as multi-task learning, meta-learning, hybrid models and model regularization, representation learning, and personalized layers. These methods strive to generate suitable local models for clients under data heterogeneity.

However, it is noteworthy that as the degree of personalization of local client models increases, their generalization performance tends to decrease. Most current PFL algorithms primarily focus on enhancing personalization and generalization capabilities on local data from the client's perspective, somewhat neglecting the provision of a global model with good generalization performance from the server's perspective. Based on this, we propose a federated learning method called FedFTHA, which enhances the personalization capabilities of client models while providing a globally generalized model on the server side, achieving a win-win situation for both clients and the server. Overall, our contributions can be summarized as below.

- We propose a novel federated learning framework, FedFTHA, which fully considers the personalized needs of clients and the generalization requirements of the server, achieving a balance and synergy between the two for a win-win scenario.
- In the FedFTHA framework, the FedFT component can be viewed independently as a new method for personalized federated learning (PFL). By fine-tuning the personalized heads, it effectively enhances the model's personalization capabilities, allowing it to better adapt to the local data distribution characteristics of clients.
- Based on FedFT, FedHA takes a server-centric approach by aggregating personalized heads to provide the global model with a global head that possesses generalization capabilities. This significantly improves the generalization performance of the global model, making it more adaptable and accurate when facing diverse data.
- Under various heterogeneous data conditions, we conduct comprehensive testing of the entire FedFTHA algorithm using multiple benchmark datasets. The experimental results strongly demonstrate that FedFTHA is an efficient algorithm capable of simultaneously meeting the dual requirements of client model personalization and server model generalization, providing solid theoretical support and practical evidence for the promotion of federated learning in real-world applications.

2 Related works

As the demands for data privacy and distributed learning are increasingly prominent, federated learning, as an emerging distributed machine learning paradigm, has attracted significant attention. Among them, the research on personalized methods has become a crucial direction for enhancing the effectiveness and adaptability of federated learning. The following will elaborate on three important personalized methods in federated learning and their research progress.

2.1 Meta-Learning and Multi-Task Learning

Meta-learning and multi-task learning are methods for learning from multiple related tasks and are significant for personalization in federated learning. In multi-task learning, the MOECHA method proposed by Smith et al [9]. addresses communication constraints, latency, and fault tolerance in federated multi-task learning but requires all clients to participate in each training round, making it unsuitable for real-world federated settings. The federated multi-task algorithm by Corinzia and Buhmann [2] can handle highly non-iid data and is applicable to non-convex models. In terms of meta-learning applications, Jiang et al [5]. demonstrated the similarity between the FedAvg [8] method and meta-learning approaches, while Fallah et al [3]. proposed a personalized federated learning variant based on the MAML framework [7], enhancing personalization performance by finding a better initial shared model.

2.2 Hybrid Global and Local Models

Since personalized models are in an intermediate state between local and global models, hybrid methods can enhance personalization capabilities. Hanzely [4] and Richtárik first introduced local models into the federated learning optimization objective, using a parameter to control the degree of mixing, reducing communication complexity. proposed a model interpolation method that obtains a hybrid interpolated model by weighting between local and global models, updating the global model each round by finding the best local model and interpolation weights for clients based on global model values. APFL method [?] uses the correlation between local and global models for adaptive model learning.

2.3 Representation Learning and Personalized Layers

The FedFT method in this paper belongs to this category. Arivazhagan et al. allowed clients to maintain personalized layers that do not participate in federated aggregation and only update locally. Liang et al. proposed the LG-FedAvg [6] method, keeping the representation part of the model local while the head participates in federated aggregation. Collins et al.'s FedRep [1] method is similar to FedFTHA, using a layered approach to learn global low-dimensional representations and train model heads with local data. However, it does not consider providing a globally generalized model for the server, which FedFTHA aims to achieve to better meet the practical needs of federated learning.

3 METHODOLOGY

3.1 problem formulation

The key objective function in federated learning is:

$$\min_{\theta \in \mathbb{R}^d} \left\{ f(\theta) := \frac{1}{N} \sum_{i=1}^{N} f_i(\theta) \right\} \tag{1}$$

where N represents the number of participating clients. For the i-th client, the input $x \in X$ is distributed according to D_i, with the true label $y \in Y$. The loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$f_i(\theta) := \mathbb{E}_{(x,y)\sim \mathcal{D}_i} \left[\hat{f}_i(\theta; x, y) \right] \tag{2}$$

where $\mathbb{E}$ denotes the expectation, and $\hat{f}_i(\theta; x, y)$ measures the distance between the global model's prediction and the true value.

Due to the issue of data heterogeneity in traditional federated learning, the PFL objective shifts to generating personalized models for each client. The objective function for the FedFTHA method becomes:

$$\min_{\mathcal{G},\mathcal{H}} \left\{ F(\mathcal{G}, \mathcal{H}) := \frac{1}{N} \sum_{i=1}^{N} f_i \left(\mathcal{G}, \mathcal{H}_i \right) \right\} \tag{3}$$

where $\mathcal{G} \in \mathbb{R}^{d_0}$ corresponds to the global shared representation parameters, and $\mathcal{H} = (\mathcal{H}_1, \ldots, \mathcal{H}_N), \mathcal{H}_i \in \mathbb{R}^{d_i} \forall i \in [N]$ corresponds to the personalized head parameters. The loss function $f_i : \mathbb{R}^{d_0+d_i} \to \mathbb{R}$ represents the loss of the local model composed of the global shared representation and the personalized head. In FedFTHA, f_i is also defined as

$$f_i \left(\mathcal{G}, \mathcal{H}_i \right) = \mathbb{E}_{(x,y)\in \mathcal{D}_i} \left[\hat{f}_i \left(\mathcal{G}, \mathcal{H}_i; x, y \right) \right] \tag{4}$$

3.2 FedFT and FedHA

The model is divided into a global shared representation part (such as in a typical convolutional neural network, from the first layer to the last hidden layer, denoted as $\mathcal{G}$) and a model head part (the final classification layer, denoted as $\mathcal{H}$). The choice of model head serves as a hyperparameter influencing model performance, denoted as $\mathcal{H}$ (where $\mathcal{H}_i$ is the personalized head parameter for the i-th client). In the FedFT method, clients receive model parameters sent by the server, first performing regular model updates. The gradient is calculated using the formula (assuming the local batch data is $\xi_{i,1}^k, \ldots, \xi_{i,B}^k \sim \mathcal{D}_i$, with B as the local batch size):

$$g_i^k = \frac{1}{B} \sum_{j=1}^{B} \nabla f_i \left(G_i^{k,t}, H_i^{k,t}; \xi_{i,j}^k \right) \tag{5}$$

To simplify the algorithm, use $\nabla f_i(G_i^{k,t}, H_i^{k,t}, D_i^t)$ to represent the gradient retrieval process (here, t denotes the training round). At each round, update the global shared representation $\mathcal{G}$ and the personalized

head $\mathcal{H}_i$ using the update formula:

$$\left(G_i^{k,t+1}, H_i^{k,t+1}\right) = \left(G_i^{k,t}, H_i^{k,t}\right) - \eta \cdot g_i^k \quad (\eta \text{ is the learning rate}) \tag{6}$$

During local freezing, only perform multiple rounds of gradient descent on the personalized head $\mathcal{H}_i$, with the update formula:

$$\left(G_i^{k,t+1}, H_i^{k,t+1}\right) = \left(G_i^{k,t}, H_i^{k,t}\right) - \eta \cdot g_i^k \quad (t = \tau_{\text{sync}} + 1 \text{ to } t = \tau_{\text{sync}} + \tau_{\text{head}}) \tag{7}$$

In the FedHA method, after each client completes local updates, the personalized head $\mathcal{H}_i$ is uploaded to the server. The server stores them in the head dictionary. During training, the personalized heads in the server dictionary are continuously updated until training ends or the model converges.

At the end of training, all personalized heads are aggregated to form the global head $\mathcal{H}_{\text{global}}$, with the aggregation formula:

$$\mathcal{H}_{\text{global}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}_i \tag{8}$$

Then, $\mathcal{H}_{\text{global}}$ is concatenated with the global shared representation $\mathcal{G}$ to obtain a global model with generalization capabilities for application to new clients.

When the server starts training each round, it selects $r \cdot N$ clients to participate in training. The current round's global representation G^k is sent to the selected clients. After receiving it, clients perform local updates using the FedFT method, then send the updated global representation G_i^{k+1} and personalized head $\mathcal{H}_i^{k+1}$ back to the server. After receiving them, the server aggregates the global representations G_i^k $(i \in N_k)$ to generate new global representation parameters G^{k+1}, with the formula:

$$G^k = \frac{1}{N_k} \sum_{i=1}^{N_k} G_i^k \tag{9}$$

The FedFTHA algorithm is shown in the following Figure 1:

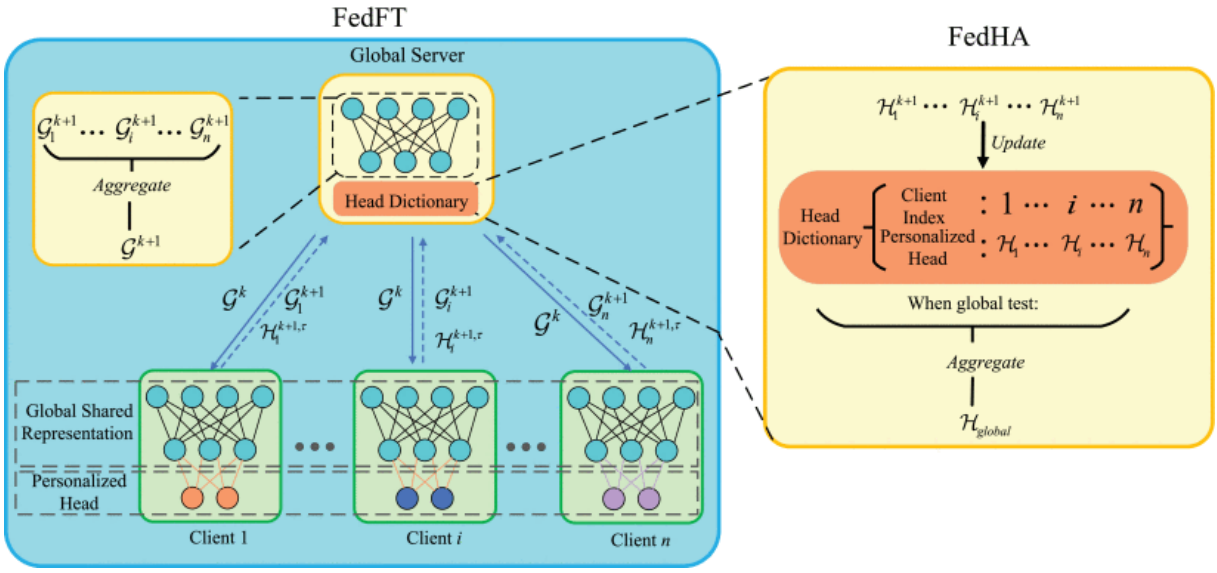

Figure 1. Overall structure diagram of the FedFTHA method.

4 Experiments

All experiments were conducted on a computing device equipped with an NVIDIA 4060ti GPU. The experimental code was written in Python and implemented using the PyTorch deep learning framework [4] to realize the FedFTHA method and comparative methods. PyTorch's efficient computing interface and rich features provided strong support for model construction, training, and optimization.

4.1 dataset and models

CIFAR datasets: The CIFAR10 and CIFAR100 datasets were used in our experiments. In our experiment, we tested the performance using ResNet and a modified LeNet-5 model.ResNet is well - known for its ability to effectively train deep neural networks. When dealing with image datasets like CIFAR - 10 and CIFAR - 100, deep networks can learn more complex and abstract features. CIFAR - 100 has 100 classes, and CIFAR - 10 has 10 classes. Models need sufficient capability to distinguish these different classes, and the depth of ResNet enables it to have such ability;LeNet - 5 can be used as a baseline model to evaluate the performance of more complex models. Understanding the performance of simple models on the CIFAR dataset can help us better understand the relationship between model complexity and performance.

4.2 Implementation Details

The FedFTHA method was evaluated through local and global testing. In local testing, each client used its personalized model to validate against local test data. The local test results were weighted and averaged based on the amount of local data to assess the model's personalization capabilities.In global testing, the final global model obtained by the server was validated against a test set containing all categories of data. By calculating accuracy and other metrics on this test set, the model's generalization performance was evaluated, particularly highlighting the effectiveness of the FedFTHA method in enhancing the global model's generalization ability.
In the experiment, the client participation ratio for communication rounds is set to $r = 0.2$, with a total of $K = 200$ communication rounds. For model updates, following existing research settings, stochastic gradient descent (SGD) is used as the optimizer, with a momentum of 0.5, a local batch size of $B = 10$, and a learning rate of $\eta = 0.01$. In experiments on three datasets, the total number of clients participating in federated training is typically set to 20.

4.3 Evaluation Metrics

Under various data heterogeneity conditions across different datasets, the generalization of FedFTHA is demonstrated by comparing the accuracy of FedFTHA with synchronous update rounds $\tau_{\text{sync}} = 5$ and head fine-tuning rounds $\tau_{\text{head}} = 5$ to that of FedAvg with local update rounds of 5. The personalization of FedFTHA is verified by comparing the accuracy with FedRep, where the feature extraction layer rounds are 5 and the head fine-tuning rounds are 10.

4.4 Results of FTHA compared to other methods

Baselines: In order to verify the effectiveness of our proposed FedFTHA method, the following methods were selected as benchmarks for comparison: FedAvg [8] and FedRep [1]

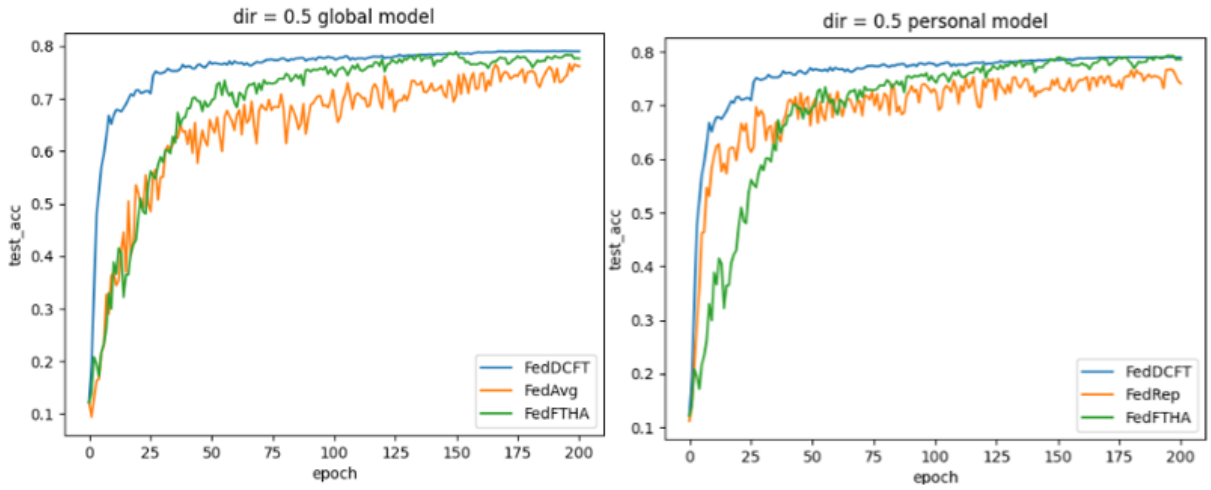

Figure 2. Interface

5 Conclusion and future work

The FedFTHA method proposed in this study effectively addresses the balance between client personalization and server generalization in federated learning. Experiments show that FedFT demonstrates strong personalization capabilities under various heterogeneous data conditions, performing well on different datasets without being affected by complex network structures. FedHA significantly enhances full-model generalization and performs well across different dataset settings, with the FedFT + FedHA combination achieving higher overall test accuracy than FedAvg. Theoretical analysis confirmed the adaptability of FedFT under different conditions. In summary, FedFTHA meets the needs for client personalization and server generalization, providing strong support for the application of federated learning.

In the future, we will focus on the issue of rapidly adapting models to new data, exploring new algorithm strategies and structures to reduce reliance on old data distribution and enhance adaptability to new data. At the same time, we will optimize FedFTHA's performance, including hyperparameter tuning and reducing communication overhead and data imbalance issues. Additionally, we plan to extend FedFTHA to fields such as healthcare, financial risk control, and intelligent transportation, customizing and optimizing according to the specific characteristics and needs of these domains to promote the development and application of federated learning technology.

References

[1] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai. Exploiting shared representations for personalized federated learning. Proc. 38th Int. Conf. Mach. Learn. (ICML), pages 2089 – 2099, Jul 2021.

[2] L. Corinzia and J. M. Buhmann. Variational federated multi - task learning. 2019.

[3] C. Finn, P. Abbeel, and S. Levine. Model - agnostic meta - learning for fast adaptation of deep networks. Proc. 34th Int. Conf. Mach. Learn. (ICML), pages 1126 – 1135, 2017.

[4] F. Hanzeley, B. Zhao, and M. Kolar. Personalized federated learning: A unified framework and universal optimization techniques. 2021.

[5] Y. Jiang, J. Konečný, K. Rush, and S. Kannan. Improving federated learning personalization via model agnostic meta - learning. arXiv:1909.12488, 2019.

[6] P. P. Liang, T. Liu, Z. Liu, R. Salakhutdinov, and L. Morency. Think locally, act globally: Federated learning with local and global representations. Proc. NeurIPS Workshop Federated Learn. Data Privacy Confidential., 2019.

[7] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal. Federated multi - task learning under a mixture of distributions. Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), pages 15434 – 15447, Dec 2021.

[8] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. JMLR: WCP, 54:1273–1282, 2017.

[9] V. Smith, C. K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi - task learning. Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst., pages 4424 – 4434, Dec 2017.