2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

FedLoop: A P2P Personalized Federated Learning Method on Heterogeneous Data

Li Fei
Faculty of Computer Science and Information Technology
Universiti Malaya
Kuala Lumpur, Malaysia
&
Faculty of Artificial Intelligence
Neijiang Normal University
Neijiang, China
S2021084@siswa.um.edu.my,
21021312@zju.edu.cn

Chu Kiong Loo
Faculty of Computer Science and Information Technology
Universiti Malaya
Kuala Lumpur, Malaysia
ckloo.um@um.edu.my

Liew Wei Shiung
Faculty of Computer Science and Information Technology
Universiti Malaya
Kuala Lumpur, Malaysia
liew.wei.shiung@um.edu.my

Liu xiaofeng
Department of Telecommunication
HoHai University
Changzhou, China
xfliu@hhu.edu.cn

***Abstract*—In federated learning scenarios, data heterogeneity can significantly impact performance. Personalized federated learning seeks to provide individualized models for each client to enhance convergence on heterogeneous data. We discovered that initially training the personalized layers, also known as the head, of the model first can alleviate the effects of data heterogeneity. As a result, we propose a simple method named FedLoop. This method uses a loop topology structure, eliminating the need for a central server or data exchanges between participants, thereby safeguarding privacy. Within FedLoop, clients act as nodes in a loop. The training process for each node consists of two phases: an initial phase solely for the personalized layers and a subsequent phase dedicated to the training of all layers. This looping process continues until a set round limit is achieved. Experimental findings reveal that FedLoop outperforms the existing state-of-the-art algorithm, FedALA. FedLoop effectively addresses challenges posed by data heterogeneity and its rapid convergence significantly cuts down communication overheads in federated learning.**

Keywords—personalized federated learning, loop, heterogeneous data

I. Introduction

Federated Learning (FL)[1] prioritizes data privacy while facilitating knowledge sharing. However, when FL is confronted with non-IID (non-Independently Identically Distributed, a form of data heterogeneity that we will not differentiate in this paper) data local optimization goals may not align with global ones, potentially leading to degraded performance [2]. Researchers have introduced Personalized Federated Learning (PFL) to provide personalized models for each client and improve convergence on highly heterogeneous data. Notably, studies from Ramasesh [3] and Luo, M. [4] indicate that the 'model head' - the final layers of a neural network where client-specific learning occurs - plays a crucial role in the occurrence of catastrophic forgetting[12]. Their findings served as a key inspiration for our research.

Inspired by recent studies, we propose a simple method called FedLoop under the domain of PFL. This method introduces a loop topology where each client possesses its own unique personalized layers. Additionally, all nodes share a common part consisting of some layers (the 'backbone'). The training protocol for each client is divided into two distinct steps: the initial training of personalized layers, followed by the training of all layers. In the initial step, the 'head' is trained while the 'backbone' is kept frozen. This allows for an initial understanding of the unique characteristics of each node, and importantly, mitigates the issue of catastrophic forgetting by reducing the discrepancy caused by varied supervisory signals from the top layers towards the shared layers.

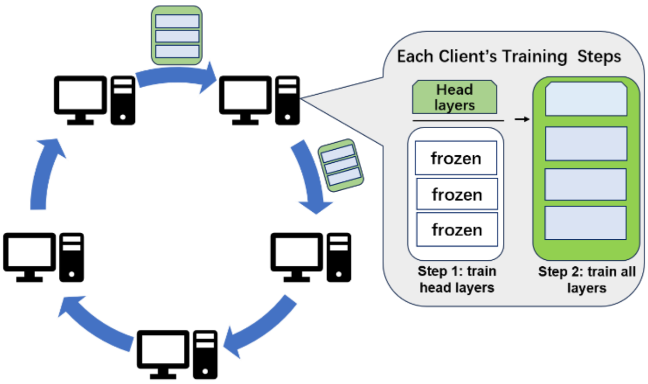

Fig. 1. Schematic representation of the FedLoop method: On the left is the loop architecture with shared layers being transferred among clients; on the right are the training steps of one client.

The mainstream architecture of FL relies on a central server to aggregate parameters from each client. However, there are also decentralized Federated Learning methods that enable direct communication between clients[5]. The FedLoop method falls under the category of decentralized Federated Learning methods, operating in a peer-to-peer manner, thus eliminating the need for a central server. This algorithm shares model parameters instead of raw data. As such, it meets the basic privacy protection requirements in FL scenarios.

The FedLoop method, adopting a loop structure, is better suited for cross-silo federated learning situations [5] (clients are data centers distributed across different organizations or geographical locations) demanding stable networks and clients across silos, rather than a more general cross-device federated learning scenario. For unstable network scenarios, we also proposed a solution using a dual-loop redundant structure, similar to FDDI[6]. Fig. 2 shows its principle.

At first glance, the training process for each node may seem sequential. However, in reality, FedLoop method facilitates a loop pipeline structure amongst nodes, enabling parallel training.

Our contributions can be summarized as follows:

- We propose the FedLoop method, which eliminates the need for data sharing or a centralized server, enhancing the performance of personalized federated learning. Our experiments have confirmed the superiority of this approach over existing methods.
- We address the efficiency concern associated with the sequential training of the FedLoop method, offering specific solutions for client and network failures within the loop structure.

II. RELATED WORK

Federated Learning (FL) was introduced by Mcmahan et al. in 2017 [1]. The initial FedAvg method aggregates a global model from individual client models and is highly reliant on IID data distribution [1]. Nevertheless, many real-world applications encounter non-IID data distributions. To address this, Personalized Federated Learning (PFL) emerged, aimed at developing personalized models for individual clients, considering their unique data distributions. One effective strategy adopted in PFL is parameter decoupling, which separates local private model parameters from the global FL model parameters, allowing for the learning of specific task representations and enhancing personalization [2]. The current state-of-the-art in PFL, however, is the FedALA method [7].

As for the application of the loop structure in FL, the CWT method applies this structure to FL with the goal of enhancing the performance of deep learning algorithms in medical image diagnosis. It enables the periodic transfer of the model's weights among different institutions, facilitating the process of knowledge sharing and transfer [8]. According to a study conducted by L. Qu et al., the Vision Transformer (ViT)[9] structure is found to be particularly compatible with heterogeneous data. By merely replacing CNNs with ViTs, it was demonstrated that both CWT and FedAvg could maintain model accuracy even in highly heterogeneous non-IID settings [10].

In the realm of head-first training, studies conducted by Ramasesh et al. showed that tasks of medium similarity suffer the most from forgetting, a phenomenon primarily driven by higher layers in the model. Interestingly, they discovered that pre-training the new task's head for a few epochs before training the whole network can alleviate the performance drop on the original task [3]. Echoing this, Luo et al. observed a larger bias in the classifier compared to other layers. Consequently, they proposed the CCVR algorithm, which mitigates this bias by sampling virtual representations from an approximate Gaussian mixture model [4].

In their work, Legate, G. et al. proposed a step-wise training approach. The first step involves FL to obtain a classification head (Head-Tuning stage), followed by an extensive fine-tuning process (Fine-Tune stage) to generate the global model. Their findings indicated that in some instances, head-first training can be as effective as updating the entire model [11].

III. METHED

A. *Detailed Steps of FedLoop method Training*

As illustrated in Figure 1, we segment the network into 'head' (personalized layers) and 'backbone' (shared layers). In this study using the cifar100 dataset[14] as an example, the last fully connected layer alone is adequate to serve as the personalized layer. Each client has its own personalized layer.

Subsequently, starting from the initial node, we train each node through the following two steps:

- Step 1 (Personalized Layers Training): At this stage, we freeze the parameters of the shared layers and train only the parameters of the node's personalized layers.
- Step 2 (All Layers Training): Next, we unfreeze the parameters of the shared layers and continue training all the layers of the model.

After training the current node, we transfer the shared layer parameters to the next node. Then, we train the next node following the two aforementioned steps. The loop repeats until the maximum number of rounds is reached.

It is worth noting that the FedLoop method does not replace other methods but serves as a new training strategy that can be combined with other methods, providing more training choices and possibilities.

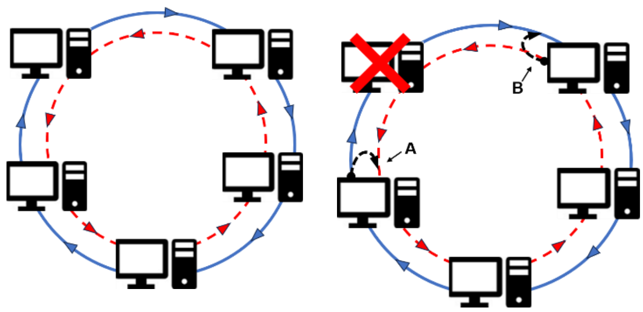

Fig. 2. Left: Standard transmission with outer loop transferring data and inner loop as redundancy. Right: With a network host issue, inner loop activates. At points A and B, loops converge to form a new loop structure.

B. *Parallel Processing and Data Transmission*

Despite the seemingly sequential training process, the actual operation is such that when the next client begins training, the current client can also initiate its own training. This forms a loop pipeline-like structure where each client can train in parallel.

Compared to the conventional FedAvg, the FedLoop method requires transmitting less than half the data volume in the same round. This is because, in FedAvg, each node has to send and receive a complete model in every round, whereas FedLoop only needs to send the shared layers to the next node. However, when there's a significant discrepancy in training speed or data volume across clients, the bottleneck of the entire network might manifest at the slowest client or the one with the most extensive data set, since all clients have to wait for the slowest client to finish. In such scenarios, coordination becomes essential. It's advisable to train faster layers more frequently and slower layers less, aiming to synchronize training speeds across the network, and thereby maximizing the throughput of the entire loop.

In situations with unstable networks, we propose a potential solution—adopting a dual-loop structure similar to the Fiber Distributed Data Interface (FDDI), as depicted in Figure 2. Under normal circumstances, the outer loop route is responsible for data transmission. When there's a client or network failure as shown in the top left corner of Fig. 2, the inner loop route activates, especially at points A and B in Fig. 2, and together with the outer loop route, forms a new circuit. If the network breaks into two parts, the network will split into two subnetworks. However, once the network connection is restored, the loop training can resume.

In a real-world peer-to-peer network, as shown in Fig. 3, each node constantly exchanges models with its neighbors. For a particular node within this p2p network, it may form multiple loops with other nodes. In this way, through the exchange of models, the node can benefit from other nodes across the entire network.

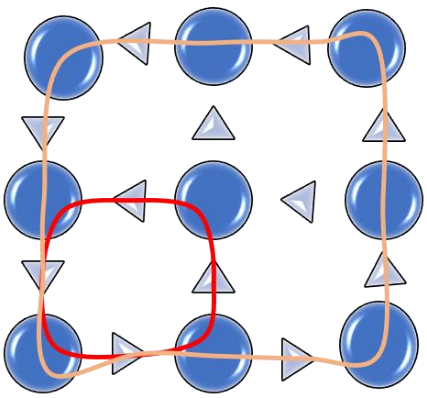

Fig. 3. In a complex P2P network with multiple loops, the bottom-left node exemplifies this: red forms a smaller loop, while orange constructs a larger one.

IV. EXPERIMENTS

We have adopted the environment provided by FedALA(https://github.com/TsingZ0/PFL-Non-IID), and made modifications based on their Fedavg code to implement FedLoop. First, we started with a simple four-layer CNN network, as provided in their code. Based on their experimental settings, we divided the CIFAR100 data among 20 clients, setting the 'dir' value to 0.1 for all. The 'dir' value controls the Dirichlet distribution - the smaller it is, the more heterogeneous the setting. We set the minimum category count for each client to 1. For the optimization method, the FedALA algorithm chose SGD and AdamW, while FedLoop adopted AdamW. The results are shown in Fig 4. As we can see, regardless of whether FedALA uses SGD or AdamW, FedLoop outperforms FedALA, and notably, FedLoop achieves the maximum accuracy with fewer rounds.

Next, we conducted the ResNet18 experiment, also adopting their experimental settings, and found that, whether using AdamW or SGD, FedLoop surpassed FedALA. Next, we tested the CoAtNet[13] model – a hybrid of CNN and transformer. Consistently, as shown in Fig. 5 and Fig. 6, FedLoop outperformed FedALA.

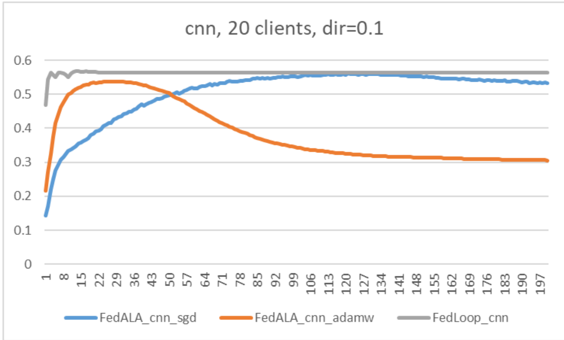

Fig. 4. Comparison between FedLoop and FedALA with 20 clients, a 'dir' value of 0.1, and utilizing the CNN model

We also tested FedLoop in environments with highly skewed data distributions, termed 'pathological' in FedALA. In these settings, all label categories for clients are disjoint, meaning no overlap. That is, each client's data has distinct labels. For instance, if Client 1 has categories 0 and 1, then none of the other clients have data in these two categories. In both the 10-client and 20-client scenarios, FedLoop outperformed FedALA. This is illustrated in Fig. 7 and Fig. 8.

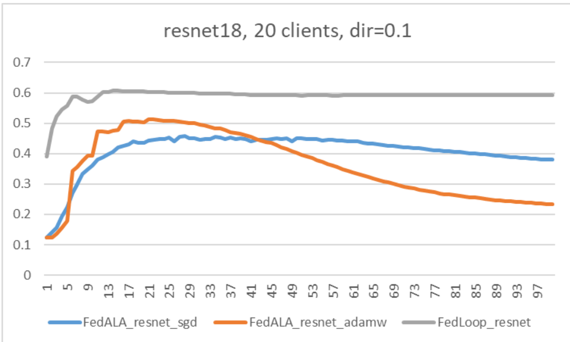

Fig. 5. Comparison between FedLoop and FedALA with 20 clients, a 'dir' value of 0.1, and utilizing the ResNet18 model

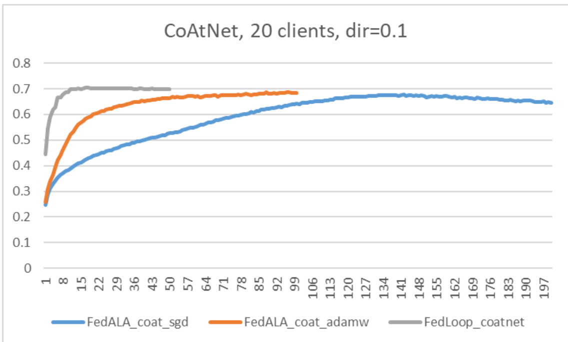

Fig. 6. Comparison between FedLoop and FedALA with 20 clients, a 'dir' value of 0.1, and utilizing the CoAtNet model

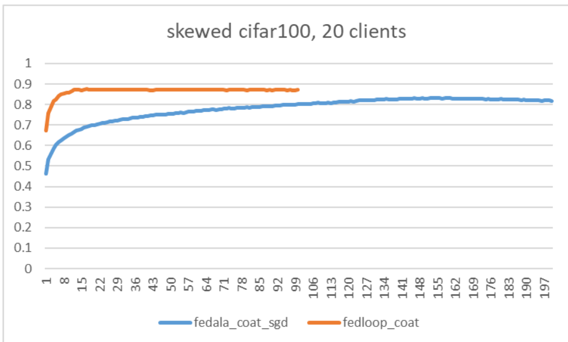

Fig. 7. Comparison between FedLoop and FedALA with 20 clients in skewed dataset, utilizing the CoAtNet model

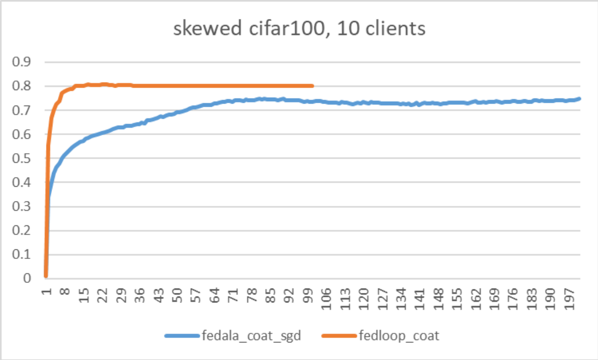

Fig. 8. Comparison between FedLoop and FedALA with 20 clients in skewed dataset, utilizing the CoAtNet model

We compared the performance of FedLoop and FedALA, varying the 'dir' value (set to 1 and 0.01) and the number of clients (20 and 10). Table I shows that in all scenarios, FedLoop consistently outperforms FedALA with fewer rounds.

TABLE I. FEEDLOOP VS. FEDALA

Client Count	dir	FedALA_coat_sgd		FedLoop_coat	
		best round	*best accuracy*	*best round*	*best accuracy*
20	1	127	0.5471	32	0.5898
20	0.01	183	0.7913	38	0.8472
10	0.1	72	0.6375	38	0.7037
10	1	73	0.5391	39	0.5880
10	0.01	57	0.7125	30	0.8138

V. CONCLUSION AND PROSPECT

We propose the FedLoop algorithm, which combines a loop structure with the initial training of the model's head, eliminating the need for a central server or data exchange between nodes. Through experiments, we have validated the effectiveness of the FedLoop algorithm. It surpasses the current state-of-the-art, FedALA, and requires fewer rounds. This significantly reduces communication costs in federated learning. The experimental results indicate that prioritizing the training of the model's personalized layers is a promising approach.

REFERENCES

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[2] Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2022). Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems.

[3] Ramasesh, V. V., Dyer, E., & Raghu, M. Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics. In International Conference on Learning Representations.

[4] Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., & Feng, J. (2021). No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. Advances in Neural Information Processing Systems, 34, 5972-5984.

[5] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2), 1-210.

[6] Katz, D., "Proposed Standard for the Transmission of IP Datagrams over FDDI Networks", RFC 1188, DOI 10.17487/RFC1188, October 1990, <https://www.rfc-editor.org/info/rfc1188>.

[7] Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., & Guan, H. (2023, June). Fedala: Adaptive local aggregation for personalized federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 9, pp. 11237-11244).

[8] Chang, K., Balachandar, N., Lam, C., Yi, D., Brown, J., Beers, A., ... & Kalpathy-Cramer, J. (2018). Distributed deep learning networks among institutions for medical imaging. Journal of the American Medical Informatics Association, 25(8), 945-954.

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[10] L. Qu, et al., "Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022 pp. 10051-10061.

[11] Legate, G., Bernier, N., Caccia, L., Oyallon, E., & Belilovsky, E. (2023). Guiding The Last Layer in Federated Learning with Pre-Trained Models. arXiv preprint arXiv:2306.03937.

[12] French, R. M. (1999). Catastrophic forgetting in connectionist networks. Trends in cognitive sciences, 3(4), 128-135.

[13] Dai, Z., Liu, H., Le, Q. V., & Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. Advances in neural information processing systems, 34, 3965-3977.

[14] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.