FedDM: Data and Model Heterogeneity-Aware Federated Learning via Dynamic Weight Sharing

Leming Shen
Department of Computing
The Hong Kong Polytechnic University
Hong Kong SAR, China
leming.shen@connect.polyu.hk

Yuanqing Zheng
Department of Computing
The Hong Kong Polytechnic University
Hong Kong SAR, China
yqzheng@polyu.edu.hk

2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-3986-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDCS57875.2023.00093

***Abstract*—Federated Learning (FL) plays an indispensable role in edge computing systems. Prevalent FL methods mainly address challenges involved in heterogeneous data distribution across devices. Model heterogeneity, however, has seldom been put under scrutiny. In practice, different devices (*e.g.*, PCs and smartphones) generally have disparate computation and communication resources, necessitating neural network models with varying parameter sizes. Therefore, we propose FedDM, a novel data and model heterogeneity-aware FL system, which improves the FL system's accuracy while reducing edge devices' computation and communication costs for heterogeneous model training. FedDM features: 1) *dynamic weight sharing scheme* that handles model heterogeneity by dynamically selecting parts of the large model to share with smaller ones; 2) *tree-structured layer-wise client cooperation scheme* that handles data heterogeneity by allowing clients with similar data distribution to share some network layers. We implement FedDM and evaluate it using five public datasets with different tasks.**

***Index Terms*—Federated Learning, Data Heterogeneity, Model Heterogeneity, Parameter Sharing**

I. Introduction

Federated Learning (FL) is an emerging technology to collaboratively train a shared model while protecting user privacy. During model training process, each device only uploads its local model parameters to the server for aggregation. In real-world applications, FL faces two challenges: data heterogeneity and model heterogeneity. Data heterogeneity refers to the non-independent and identically distributed (non-IID) data across devices, resulting in significant performance degradation when applying classic FL scheme [1]. Model heterogeneity refers to the fact that edge devices usually have dissimilar computation and communication resources, which calls for heterogeneous and adaptable models. For example, PCs can afford models with more weight parameters, while smartphones or smartwatches may only store simpler models. Assigning the smallest model to each client will result in sub-optimal issues [2] as the size of the model is limited by the device with the least system resources.

To handle data heterogeneity, recent works propose 1) clustering-based methods [3]–[5] that enhance the cooperation among clients with similar data distributions; 2) personalization algorithms [6]–[9] that train a sub-model or add some task-specific layers to steer local model toward fitting its specific data distribution.

For model heterogeneity, existing solutions mainly include knowledge distillation, parameter pruning, and model weight sharing. For example, FedDF [10] generates ensemble of heterogeneous models and generates different smaller models by knowledge distillation to meet each client's resource budget. TailorFL [11] proposes a model pruning-based approach to support heterogeneous models deployed on resource-constrained devices. However, replacing aggregation with ensemble learning or pruning out filters can lead to essential weight information missing [12], causing unexpected feature loss and performance degradation. HeteroFL [13] is a weight-sharing scheme that selects subsets from global model parameters for devices. However, a critical issue arises because the unshared part (shown in Fig. 1) of the large models can only be trained on a small portion of the dataset. These extra parameters cannot be well combined with other aggregated parameters, causing the weight unbalancing problem [2].

We propose FedDM, a data and model heterogeneity-aware FL system. To handle model heterogeneity, we propose *dynamic weight sharing scheme* that leverages a varying shrinkage ratio to generate different sizes of sub-models to fit each device's resource budget. To handle data heterogeneity, the *tree-structured layer-wise client cooperation scheme* is proposed to make clients with similar data or model weight distribution selectively share some network layers.

We face three key technical challenges: 1) How to design weight sharing scheme without causing the unbalancing problem. 2) How to orchestrate sub-model aggregation with different shapes of parameters. 3) How to make clients cooperate with each other to selectively share some network layers.

The key contributions of our work can be summarized as follows: 1) We propose FedDM that uncovers dynamic weight sharing and client cooperation schemes, handling both data and model heterogeneity to meet each device's resource budget without causing performance degradation and the unbalancing problem. 2) We exploit *dynamic weight sharing scheme* to generate sub-models that dynamically share weights with the global model accordingly. 3) We devise *tree-structured layer-wise client cooperation scheme* that enhances cooperation among clients based on a refined metric to measure similarity between clients.

2575-8411/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDCS57875.2023.00093

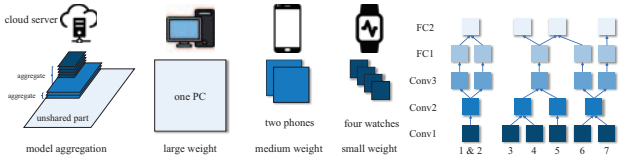
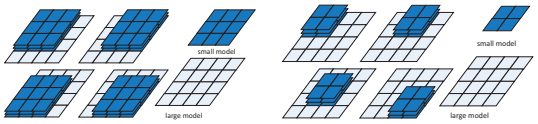


Fig. 1. The weight sharing scheme of HeteroFL. Fig. 2. Client cooperation.



(a) Four-corner sharing scheme. (b) Dynamic sharing scheme.

Fig. 3. Different sharing schemes with different shrinkage ratios.

II. Methodology

A. Dynamic Weight Sharing

The key idea of HeteroFL is that resource-constrained devices can select subsets of global model parameters based on a pre-defined shrinkage ratio. For instance, when the shrinkage ratio is 0.75 and one layer's weight of the large model has a shape of $(64, 32, 3, 3)$, the counterpart of a sub-model will then be $(48, 24, 3, 3)$. In HeteroFL, the shared part is fixed. As illustrated in Fig. 1, the unshared part of the global model is only aggregated by PCs, thus cannot see data on other devices, causing the unbalancing problem.

Thus, to design a more balanced sharing scheme, one natural solution is to allow smaller weight parameters to iteratively switch among the four corners as shown in Fig. 3 (a). Therefore, each corner in the global model has the opportunity to be shared with smaller models. Hence it mitigates the problem of parameter imbalance to some extent. However, we observe that when the shrinkage ratio is high, the central part of the large model's parameter will still be shared multiple times, causing unbalanced aggregation similar to that in HeteroFL *i.e.*, the first challenge).

Therefore, we adopt a different weight-sharing scheme, *i.e.*, dynamic parameter sharing based on the server's configuration in each global communication round. Specifically, the server's configuration contains three key hyper-parameters, *i.e.*, shrinkage ratio, row index, column index. The shrinkage ratio for each client depends on the client's system resource budget and thus is fixed (*e.g.*, 0.5 for smartwatches, 0.75 for smartphones). The row index and the column index refer to the position of the upper-left value of the smaller weight matrix within the large matrix. In HeteroFL, both the row index and the column index are 0, referring to the upper left value of the large weight parameter. We will design a specific rule to assist the server in determining these hyper-parameters, considering both client performance and weight aggregation balancing. To address the second challenge, only the intersected parameters across the sub-models are aggregated while keeping the non-intersected parameters unchanged.

B. Tree-Structured Layer-Wise Client Cooperation

In order to address the third challenge, we decide to enhance the cooperation among clients that share higher similarities in data and model weight distribution. As mentioned in [8], the Kullback-Leibler Divergence (KLD) is used to measure the similarity between two distributions. Hence, we define the similarity between the i-th and j-th client as: $S_{i,j} = \mathrm{KLD}(x_i, x_j) + \lambda \cdot \mathrm{KLD}(\Phi(w_i, x_i), \Phi(w_j, x_j))$ where $\mathrm{KLD}(x, y)$ is the KLD value of two distributions, and $\Phi(w_i, x_i)$ is the pre-softmax output of the model with parameter matrix w_i and input local data x_i. λ is a hyper-parameter that balances the impact between data and model weight distribution on the client cooperation scheme. Clients with higher similarities can share some layers (as illustrated in Fig. 2) to collaboratively train the layers while reducing communication costs. We will leave the details of sharing scheme implementation as our future works.

III. Future Works

Based on the aforementioned methodology, our future works contain three steps: 1) In order to steer the client training process, we will study how to meticulously refine the server's configuration. 2) To better illustrate the similarity among clients, we plan provide a detailed client cooperation scheme. 3) We will also carry out extensive experiments (*e.g.*, case study, ablation study) to evaluate of our proposed system compared with the state-of-the-art baselines.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[2] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, "Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training," in *ICML*, 2022.

[3] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *ACM MobiCom*, 2021.

[4] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," *World Wide Web (WWW)*, vol. 26, no. 1, pp. 481–500, 2023.

[5] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *NeurIPS*, 2020.

[6] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *ACM SenSys*, 2021.

[7] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *ACM MobiCom*, 2021.

[8] L. Tu, X. Ouyang, J. Zhou, Y. He, and G. Xing, "Feddl: Federated learning via dynamic layer sharing for human activity recognition," in *ACM SenSys*, 2021.

[9] C. Li, X. Zeng, M. Zhang, and Z. Cao, "Pyramidfl: a fine-grained client selection framework for efficient federated learning," in *ACM MobiCom*, 2022.

[10] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *NeurIPS*, 2020.

[11] Y. Deng, W. Chen, J. Ren, F. Lyu, Y. Liu, Y. Liu, and Y. Zhang, "Tailorfl: Dual-personalized federated learning under system and data heterogeneity," in *ACM SenSys*, 2022.

[12] Y. J. Cho, J. Wang, T. Chirvolu, and G. Joshi, "Communication-efficient and model-heterogeneous personalized federated learning via clustered knowledge transfer," *IEEE J. Sel. Top. Signal Process.*, 2023.

[13] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *ICLR*, 2021.