Efficient Federated Learning via Clients-to-Server Knowledge Distillation (Student Abstract)

Huifang Sun[1], **Jiaming Pei**[2], **Lukun Wang**[1]

[1]Shandong University of Science and Technology, China
[2]School of Computer Science, The University of Sydney, Australia
huifang.sun@sdust.edu.cn, jpei0906@uni.sydney.edu.au, wanglukun@sdust.edu.cn

Abstract

To diminish the substantial communication costs incurred by federated learning during the training of the global model and enhance the model update efficiency across both clients and server domains, we have integrated knowledge distillation into the federated learning framework. This integration has led to the development of a novel approach termed ClientsToServerKDFL, which streamlines the distillation process by directly transferring model insights from clients to the server for computational learning without the need for extensive computations across numerous clients. This iterative process ensures model accuracy and curtails communication expenses. Experimental data analysis has validated the efficacy of this algorithm.

Introduction

Federated learning (FL) enables collective model training with shared global knowledge, avoiding the need to upload individual client data. Since FL requires ongoing broadcasting and model exchanges between the server and clients during training (McMahan et al. 2017), efficient communication is crucial. With large-scale models, extensive datasets, and millions of devices involved, concerns about communication efficiency and bandwidth constraints can emerge. These challenges often lead to significant communication costs as the global FL model interacts and consolidates information among participants to facilitate updates.

Knowledge distillation (KD), as referenced in (Pei et al. 2024), refines and boosts model accuracy by transferring insights from complex to simpler models, enhancing generalization. This is especially beneficial in scenarios with limited computational resources or high deployment demands. Multi-teacher knowledge distillation is superior to single-teacher methods and aligns naturally with the FL paradigm, leading to numerous innovative FL methods, as described in (Shen et al. 2020) and (Wu et al. 2022).

Integrating knowledge distillation into federated learning has enhanced model performance, but existing techniques demand heavy client computations. Considering millions of clients, this incurs high computational and communication costs. To overcome these issues, we present ClientsToServerKDFL, a multi-teacher FL-KD algorithm for streamlined communication. Here, clients act as teachers and the server as the student. By applying multi-teacher KD to assign varying weights based on data quality, we prioritize key knowledge. Directly transferring knowledge from clients to the server, our method cuts down on computation, simplifies personalized model integration, boosts communication efficiency, and ensures high accuracy.

Methodology

Figure 1 shows the flowchart for the ClientsToServerKDFL. In the ClientsToServerKDFL algorithm, the server initializes an original model and distributes it to participating clients. Each client then trains the model using their private data. Post-training, clients act as teachers, scaling model logits based on data quality-adjusted learning weights before distilling the model to the server via knowledge distillation. The server, acting as a student, integrates this knowledge and broadcasts it back to clients. This cycle repeats until model convergence.

On the server side, after launching the base model and sharing it with the involved clients, the server shifts to a "learner" mode. It then continuously assimilates and propagates models from different clients, all while respecting their data privacy. This cycle fortifies the original model, bolstering its resilience and dependability by incorporating a wide array of data from each client throughout the training phase.

In the knowledge distillation training phase, the teacher model's predicted outcomes (soft targets) act as extra training data, helping the student model grasp a broader knowledge base. By assimilating the teacher's insights, the student model taps into more nuances and features, thus boosting its performance. For the specific formula, see Section A.1 Server in the supplemental file.

On the client side, the original model from the server is received for the first training round, after which training starts with the client's private data. The resulting client models act as teachers, with weights assigned based on cross-entropy loss between their predictions and true labels. This setup enables multi-teacher knowledge distillation, allowing the server to learn from various teacher models.

Clients regularly update the global model from the server, train it, and use multi-teacher knowledge distillation to synthesize client logits. This synthesis is distilled back to the server for learning. This cycle repeats until model conver-

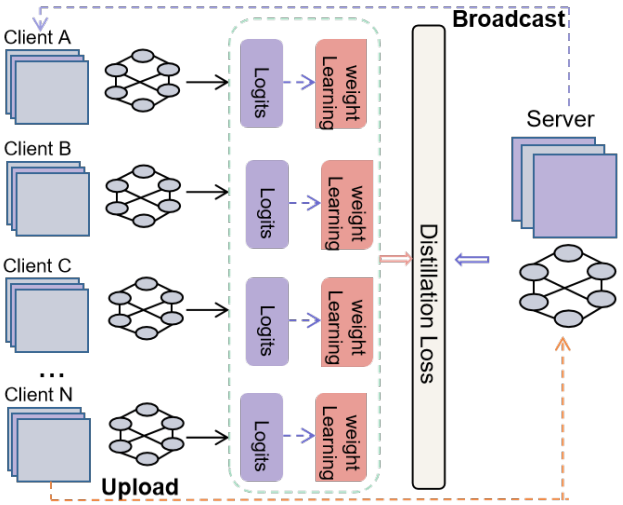

Figure 1: ClientsToServerKDFL Flowchart. This process involves the transfer of knowledge between the client and the server.

gence, marking the end of training. For the specific formula, see Section A.2 Clients in the supplemental file.

This approach reduces server computational costs compared to direct distillation on multiple clients and further lightens the server load compared to traditional federated learning by refining post-distillation instead of immediate retraining. Its theoretical convergence guarantees and privacy analysis are provided in Section A.3 and Section A.4 in the supplementary file.

Experiment

We evaluated our framework on the MNIST, FMNIST, CIFAR10, and CIFAR100 datasets using CNN. Post-local training, the server model syncs and updates parameters. We benchmarked against FedAvg, FedGen (Zhu, Hong, and Zhou 2021), FedKD (Wu et al. 2022), FedProto (Tan et al. 2022), and FedNTD (Lee et al. 2022) on accuracy, training time, parameter quantity, and communication load.

In this study, we train models on MNIST, FMNIST, CIFAR10, and CIFAR100 datasets individually. MNIST and FMNIST training include 100 epochs at a 0.01 learning rate. For CIFAR-10 and CIFAR100, it's 200 epochs at a 0.005 learning rate with 10 clients. The performance comparisons of the algorithms are detailed in Tables 1 and 2.

Tables 1 and 2 show that ClientsToServerKDFL has shorter average run times, ranging from seconds to minutes, compared to other methods. It also cuts parameter count by about 50% and reduces memory usage. These results confirm our algorithm's efficiency, compact parameter size, low memory demand, and lightweight nature without sacrificing accuracy. The accuracy and loss comparison line graph of different federated learning algorithms on the four datasets can be found in the Section B Experimental Diagram in the supplemental file. All these prove the effectiveness of the ClientsToServerKDFL algorithm.

Algorithm	Time(s)	Tensors	Overhead(M)
FedGen	338	7.9E+06	23.67
FedKD	249	9.7E+06	26.22
FedProto	194	5.2E+06	15.60
FedNTD	168	5.8E+06	17.77
Ours	146	5.2E+06	15.57

Table 1: Performance comprison on MNIST dataset.

Algorithm	Time(s)	Tensors	Overhead(M)
FedGen	4286	3.2E+07	85.22
FedKD	5289	6.2E+07	160.81
FedProto	4386	2.9E+07	77.40
FedNTD	2472	3.0E+07	80.46
Ours	2167	2.8E+07	77.19

Table 2: Performance comprison on CIFAR-10 dataset.

Conclusion

ClientsToServerKDFL seamlessly merges federated learning with multi-teacher knowledge distillation. It's designed to efficiently transfer and distill knowledge in a distributed setting, leveraging each participant's data to boost model performance. This approach simplifies creating personalized models in federated distillation, cuts down on communication costs, and ensures model effectiveness. The introduction of ClientsToServerKDFL has great potential to advance efficient communication in federated learning.

References

Lee, G.; Jeong, M.; Shin, Y.; Bae, S.; and Yun, S.-Y. 2022. Preservation of the global knowledge by not-true distillation in federated learning. *Advances in Neural Information Processing Systems*, 35: 38461–38474.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Pei, J.; Liu, W.; Li, J.; Wang, L.; and Liu, C. 2024. A Review of Federated Learning Methods in Heterogeneous scenarios. *IEEE Transactions on Consumer Electronics*.

Shen, T.; Zhang, J.; Jia, X.; Zhang, F.; Huang, G.; Zhou, P.; Kuang, K.; Wu, F.; and Wu, C. 2020. Federated mutual learning. *arXiv preprint arXiv:2006.16765*.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Wu, C.; Wu, F.; Lyu, L.; Huang, Y.; and Xie, X. 2022. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1): 2032.

Zhu, Z.; Hong, J.; and Zhou, J. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, 12878–12889. PMLR.