Federated Learning Applications in Deep Learning under FedScale Framework

DU Jiaxing 1155204280

December 2023

1 Introduction

Federated learning is a machine learning approach that allows clients and server train a model collaboratively, without transferring raw data to the central server. It has wide applications in many industries such bank systems and healthcare research with data privacy concerns. Graph Neural Network (GNN) is often used to capture these non-structural complex relationships and interactions via message passing between the nodes and graphs. We aims to explore the relationship between federated learning and GNN in real world applications. By combining federated learning and GNNs, organizations or entities can collectively train GNN models on diverse graph-structured data sources while preserving data privacy and security (Liu et al., 2022).

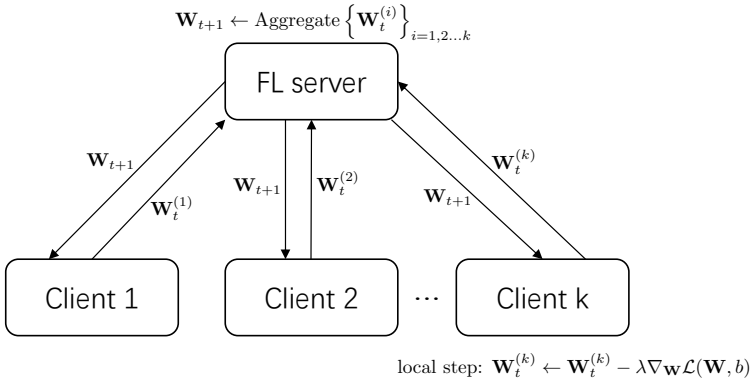
In this report, we begin with some basic concepts and popular available datasets, paving the way for further research on industrial applications. In section 2, we will briefly summarize the implementation of federated learning. Since graph neural network and convectional neural network share some similarity and massive amounts of easily accessible data, we will firstly experiment on federated CNN in section 3. In section **??**, we will introduce three different federated GNN scenarios and explore several graph classification datasets.

2 Background

The basic idea behind federated learning is to distribute a training task across multiple devices, such as smartphones, data centers, or local servers. Each device performs the training locally on its own data. and then send back parameters or gradients to central server. The key advantages of federated learning include preserving data privacy since the raw data remains on the local devices, reducing the need for data transfer, and enabling training on distributed, The parallel processing shorten the training time for large-scale data. It also allows for personalized models that can adapt to individual user preferences. McMahan et al. (2017) first proposed federated learning and introduce the FedAvg algorithm. As shown in Figure 1, the general federated learning process is realized as follows:

1. Initialize. $\mathbf{W}_0$ is initialized on a central server.
2. Clients selection. Randomly select k clients for training.
3. Local training. Each selected device receives parameters from the FL server, and performs training independently using its local data. This process can be done several epochs.

Figure 1: General federated learning process



4. Aggregation. The local model updates from the selected devices are sent back to the central server, where they are aggregated to create a new global model.

5. Model Distribution: The updated global model is then distributed back to the devices for the next round of training.

Repeat step 2-5 for N times, N is the global epoch.

While the FedAvg algorithm has gained popularity for its effectiveness in federated learning, there are several limitations needed to be concerned. During the iterative training process, the local model updates are shared with the central server, which can potentially leak sensitive information about the local data. It is possible to restore the original data based on the parameters passed by the server and clients. Besides, FedAvg aggregate received parameters by simple average, McMahan et al. (2017) mentioned that the aggregator can be arbitraily bad as it is based on IID assumption. There are more advanced aggregation techniques like Clipped Average Aggregation Wang et al. (2022), Differential Privacy Average Aggregation Wei et al. (2020), disturbing the model adding some noise. This provides a high level of privacy protection, and it may be computationally expensive, or even unable to converge.

3 Application in Convolutional Neural Network

We perform some experiments on popular computer vision (CV) and graph neural network (GNN) datasets, under the `FedScale` framework. It shows that federated learning is robust in both homogeneous and heterogeneous settings tab:CNN summary, converging to the performance of centralized training. We adopted Resnet18 for training due to its stability, which has around 11 million trainable parameters.

3.1 Homogeneous setting

We divide the dataset uniformly into 10 clients and select 5 clients for training in each session. There are 2 local epochs and 30 global epoch. We select several parameters and gradients of federated training on the CIFAR dataset for visualization. As shown in Figure 3, the gradients are collected after completing a global epoch. During the training process, the gradient tends to be stable, and there is no gradient disappearance or explosion.

Table 1: Performance of computer vision datasets

Datasets	Samples	Classes	Test Acc (%)		
			Centralized	IID FL	non-IID FL
CIFAR10	50000	10	79.43	75.65	
FEMNIST100	21345	62	71.47	72.84	71.07

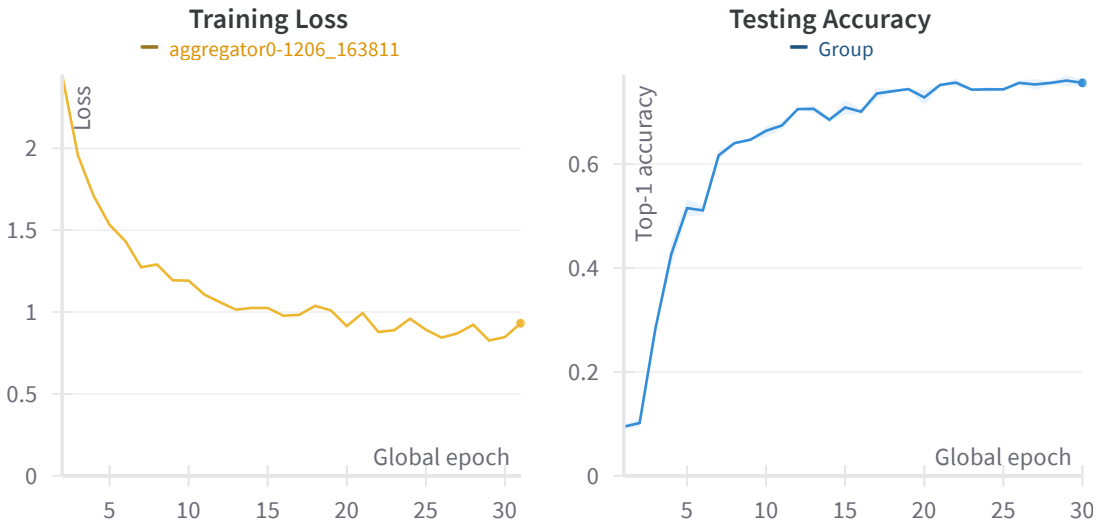


Figure 2: Federated learning ResNet18 on `CIFAR` dataset

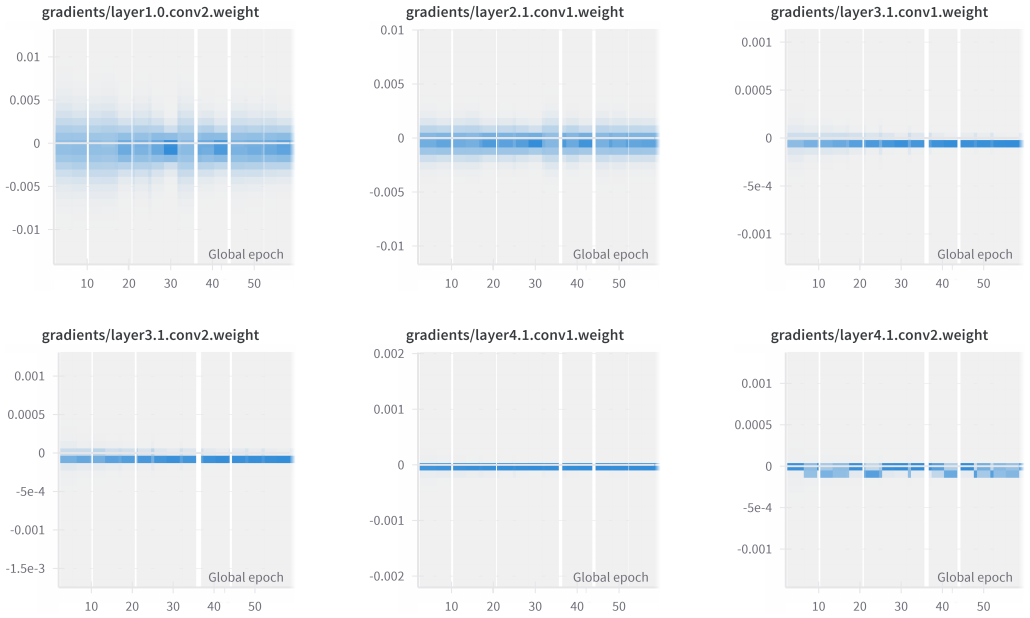


Figure 3: Some convolutional layer gradient distributions

3.2 Heterogeneous setting

The `FEMNIST` (federated extended MNIST) was specifically designed to simulate a realistic federated learning scenario by incorporating privacy concerns and the challenges of decentralized data. It contains images 62 classes handwritten digits (0-9, a-z, A,Z) from 3500 different clients, in total 637877 images are 28 by 28 pixels. The original dataset is both non-IID distributed and unbalanced. In Figure 4, we plot the how many samples and how many classes in each client.

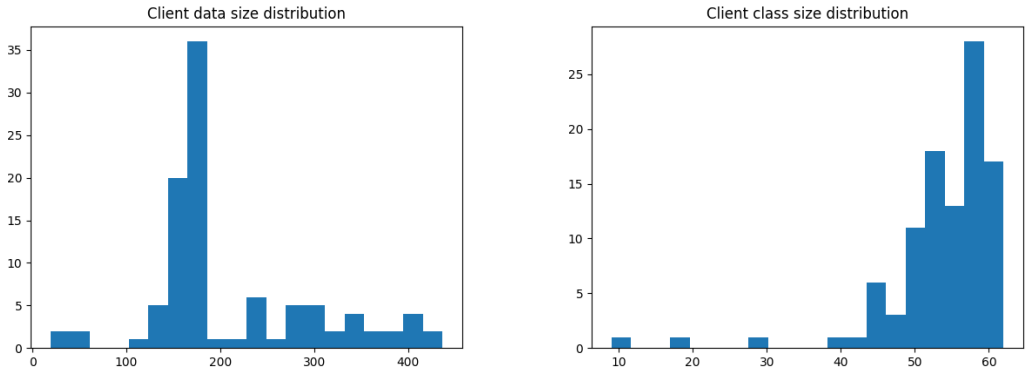


Figure 4: Data size and class size distribution of `FEMNIST` dataset

Due to the computation capacity of stand-alone deployment, we only select first 100 clients for training, only 21345 samples. We experiment on two settings, one is the original non-IID partitions, the other is uniform partion randomly by human. As shown is Figure 5, the dataset is both unbalanced and non-IID distributed. The performance in IID and non-settings are quite similar fig:size, indicating the robustness of the federated framework

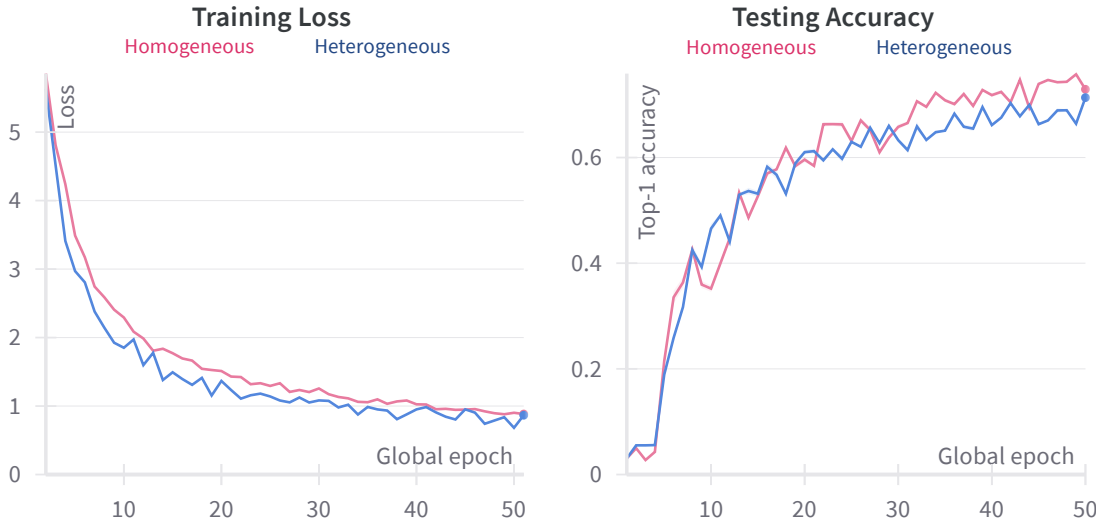


Figure 5: Comparison of IID and non-IID federated learning on `FEMNIST` dataset

References

Liu, R., Xing, P., Deng, Z., Li, A., Guan, C., & Yu, H. (2022). Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256*.

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

Wang, T., Zheng, Z., & Lin, F. (2022). Federated learning framew ork based on trimmed mean aggregation rules. *Available at SSRN 4181353*.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., ... Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, *15*, 3454–3469.