Federated GNN-to-MLP Using Contrastive Learning

Fan Yang, Kyungho Yoon, Won-Yong Shin
Yonsei University

vanyeung@yonsei.ac.kr, yoonkh@yonsei.ac.kr, wy.shin@yonsei.ac.kr

Abstract

This paper presents an efficient and effective federated learning method when graph data are used. Specifically, we propose federated GNN-to-MLP, a method that employs a graph neural network (GNN) model at the global server and a multilayer perceptron (MLP) model at each client so that the local MLP model at each client is more precisely trained via bi-directional knowledge distillation. The local model is trained based on our own loss using contrastive learning, which enables the model to be robust to the case where there is a lack of training data. Experimental results on the Planetoid dataset demonstrate that the proposed federated GNN-to-MLP outperforms the benchmark graph federated learning method in terms of the classification accuracy, while greatly reducing the inference time as well as the communication cost incurred by parameter exchanges.

Ⅰ. Introduction

Traditional graph federated learning such as FedSage [1] using graph neural networks (GNNs) at each client faces several technical challenges. First, the inference time is known to increase exponentially with the number of GNN layers, which disables us to apply graph federated learning to real-time services. Second, the communication cost is expensive due to frequent parameter exchanges between the local clients and the global server. In this study, to tackle these challenges, we propose federated GNN-to-MLP that employs a GNN model at the global server and a multilayer perceptron (MLP) model at each client. Unlike the prior study in [1], the MLP model at each client is trained via bi-directional knowledge distillation[1] by exchanging output logits between the local clients and the global server. The MLP model distilled from the GNN is trained based on our own loss using contrastive learning, which enables the model to be robust to the case where there is a lack of training data.

Ⅱ. Methodology

We consider a whole graph G that is divided into multiple subgraphs, each of which is assigned to each client. We aim to more precisely train the local MLP model via bi-directional knowledge distillation. To this end, we deploy a reference graph, shared by the global server and all the clients, whose classes are the same as those of the underlying graph G. The proposed federated GNN-to-MLP performs the following steps iteratively until convergence:

· **Step 1**: A GNN at the global server is pre-trained through the reference graph.

· **Step 2**: The pre-trained GNN generates the output logits for nodes in the reference graph, and the server sends them to the clients.

· **Step 3**: Each client trains its local MLP on the reference graph. The loss function at Client i is expressed as $\mathcal{L}_{i,1} = \alpha_1\mathcal{L}_{\mathrm{CE},i} + \beta_1\mathcal{L}_{\mathrm{NC},i} + \gamma_1\mathcal{L}_{\mathrm{KL},i}$, where $\mathcal{L}_{\mathrm{CE},i}$ is the cross-entropy loss, $\mathcal{L}_{\mathrm{NC},i}$ is the neighboring contrastive loss in [3], and $\mathcal{L}_{\mathrm{KL},i}$ is the Kullback-Leibler (KL) divergence using two logits from the server and the client i.

· **Step 4**: Each client trains its local MLP again on the dedicated subgraph with the following loss function: $\mathcal{L}_{i,2} = \alpha_2\mathcal{L}_{\mathrm{CE},i} + \beta_2\mathcal{L}_{\mathrm{NC},i}$.

· **Step 5**: The trained MLP at each client re-generates the output logits for nodes in the shared graph and sends them to the server.

· **Step 6**: The server aggregates these logits to train its GNN again with the following loss: $\mathcal{L}_3 = \alpha_3\mathcal{L}_{\mathrm{CE,\,sv}} + \gamma_2\mathcal{L}_{\mathrm{KL,\,sv}}$. Then, the server sends the logits to the clients.

Repeating Steps 3—6 constitutes one communication round.

Ⅲ. Experimental Results

We assess the performance of node classification according to different proportions of labeled nodes as training data on the Planetoid dataset. In our experiment, the number of clients is set to 5. Each dataset is split into training/validation/test sets, where $\{1,10,20\}\%$ of the given dataset is used for training. Table 1 summarizes the performance of our method in comparison with the state-of-the-art graph federated learning method, FedSage [1]. Notably, our method outperforms FedSage for all cases. It is observed that our method substantially outperforms FedSage; in particular, the gain over FedSage is significant when there is a lack of training data.

Table 1. Classification accuracy.

Dataset	Method	Portion of training		
		1%	10%	20%
Citeseer	Proposed	**0.67**	**0.68**	**0.68**
	FedSage	0.21	0.58	0.66
Cora	Proposed	**0.63**	**0.71**	**0.66**
	FedSage	0.30	0.45	0.54
Pubmed	Proposed	**0.82**	**0.83**	**0.84**
	FedSage	0.61	0.81	**0.84**

[1] The notion of knowledge distillation from GNNs to MLPs was originally introduced in [2].

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grants 2021R1A2C3004345 and RS-2023-00220762 and by the Institute of Information and Communications Technology Planning and Evaluation (IITP), Republic of Korea Grant by the Korean Government through MSIT (6G Post-MAC—POsitioning and Spectrum-Aware intelligenT MAC for Computing and Communication Convergence) under Grant 2021-0-00347.

REFERENCES

[1] Zhang *et al.*, "Subgraph federated learning with missing neighbor generation," in *Proc. NeurIPS* (2021).
[2] Zhang *et al.*, "Graph-less neural networks: Teaching old MLPs new tricks via distillation," in *Proc. ICLR* (2022).
[3] Hu *et al.*, "Graph-MLP: Node classification without message passing in graph," arXiv preprint *arXiv:2106.04051* (2021).