Personalized Federated Graph Learning

Karthik Pansetty[1], Yuhang Yao[1], Mohammad Mahdi Kamani[2], Carlee Joe-Wong[1]
[1]Carnegie Mellon University, [2]Wyze Labs

Background

Graph neural networks has been widely used for applications from behavior classification in social networks to anomaly detection in the Internet of Things. The data, however, is highly sensitive with private information. Federated learning is proposed as a distributed learning approach that reduces privacy risks and communication costs for training machine learning models on data located at multiple clients, which has been widely adopted for privacy-preserved training of non-graph data.

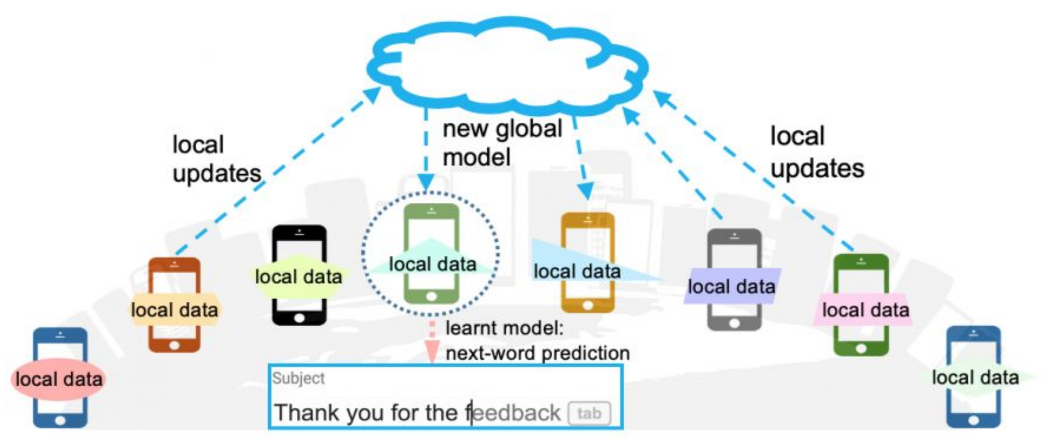

Figure-1: Example of the Federated learning flow for the task of next-word prediction [1]

However, one major challenge of federated training on graphs is that many clients have little local data, which makes statistical heterogeneity - clients' data is not identically and independently distributed (IID) [2] a challenge. Prior works have addressed this challenge by training a shared global model and a local model at every client. **Adaptive Personalized Federated Learning (APFL)**, for example, tunes a personalization parameter that interpolates the global and local models [3] for personalization. Learning personalized models as in APFL, however, leads to overfitting when there is only a small amount of data at each client.

- We propose a Federated Learning algorithm APFLGate that leverages both personalization and variance reduction techniques.
- We understand the tradeoff between personalization and variance reduction and how they affect convergence and the performance on Graph Neural Networks.
- We show that our algorithm APFLGate has an improvement of 10.08% over FedGate in the Graph Classification task.

Personalized Federated Learning with Variance Reduction

Our approach to overcome this overfitting method is by using a variance reduction method. Using only local gradient information in updating the local models might lead to poor performance, since the directions of the local gradient may be different from the directions of the global gradient. This is the motivation behind FedGate [4], which uses a local gradient tracking scheme for variance reduction. Our approach is to use APFL for personalization and FedGate for variance reduction. We present our results and analysis for this approach in the next section.

At global round t and local step r with R steps per round, the output of the personalized model for the i-th client is

$$h_i^{t,r} = \alpha_i h_{loc,i}^{t,r} + (1-\alpha_i) h_{glob}^t$$

where h_i is the output of personalized model, α_i is the personalization parameter and $h_{loc,i}$ is the output of the local model at the i-th client and h_{glob}^t is the output of global model at round t. α_i is associated with the diversity of the local model and the global model. Higher α_i means more personalization on the i-th client. The variance reduction updates of client i are given by

$$\text{Device update: } d_i^r = g_i^r - \delta_i^t, \quad g_i^r \triangleq \nabla f_i(w_i^r)$$
$$w_i^{r+1} = w_i^r - \eta d_i^r$$
$$\text{After } R \text{ local steps}$$
$$u_i^t = w^t - w_i^{t,R} \text{ (to server, get } u^t\text{)}$$
$$\bar{w}^t = w^t - u^t$$
$$\delta_i^{t+1} = \delta_i^t + \frac{1}{\eta\tau}(\bar{w}^t - w_i^{t,R})$$
$$\text{Server update: } u^t = \frac{1}{m}\sum_{i=1}^{m} u_j^t$$
$$w^{t+1} = w^t - \gamma u^t,$$

where d_i^r is the local gradient of client i at step r, w^t is the weights of the global model and w_j^t is the model of client i at step t, δ is the gradient tracking parameter, τ is the variance reduction parameter, η and γ are learning rates.

The local gradient tracking term $\boldsymbol{\delta}_i$ ensures that each client i uses an estimate of the global gradient direction to locally update its model, which reduces the variance among clients. The variance reduction parameter **τ** controls the amount of gradient tracking. Lower **τ** means more gradient tracking, which leads to lower variance among clients.

Results and Analysis

- In a high client setting, in both the tasks of Graph classification and Link Prediction in Graph Neural Networks using a Graph Convolutional Network (GCN) model, a combination of APFL + FedGate algorithms has higher test accuracy than either algorithm on its own.
- We do not see similar results when we have fewer clients, because there is enough data at each client for APFL as it does not overfit as easily as a higher client setting and FedGate has enough data to do well without personalization.
- From Figure-2 and Figure-3, we can see that APFL overfits very easily and adding a variance reduction technique to it helps prevent overfitting and improves performance.

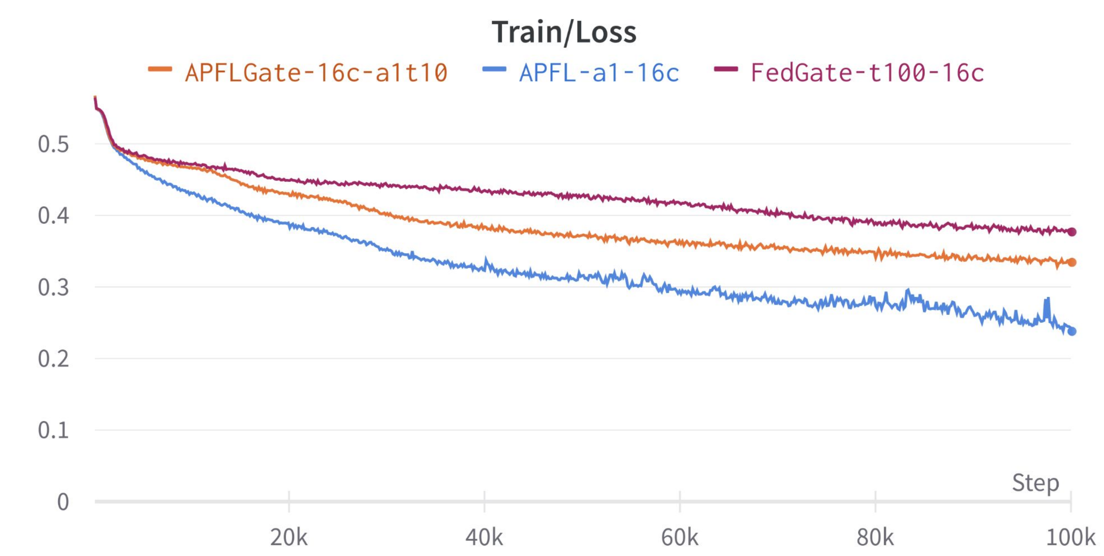

Figure 2: Train Loss on the Graph classification task for 16 clients

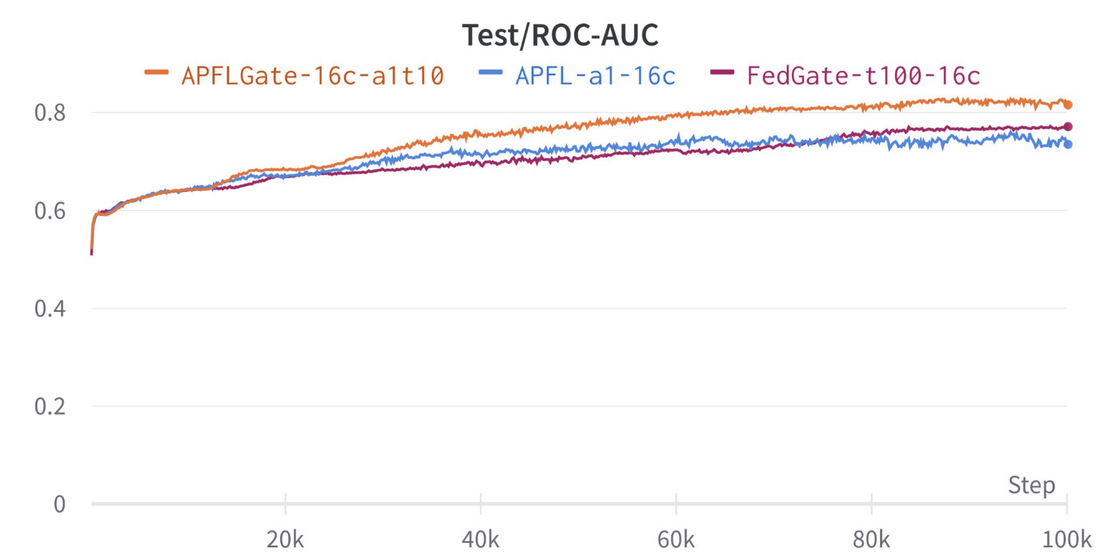

Figure 3: Test ROC-AUC on the Graph classification task for 16 clients

- Our experiments indicate that the variance reduction parameter τ controls the convergence of the algorithm and the personalization parameter α controls the performance of the model.
- The tradeoff between personalization and variance reduction has a large impact on when the model converges and its performance and this can be seen from Table-1 and Table-2.
- We see that experiments with α=1 converge faster but the experiments with τ = 10 have higher performance. In conclusion, by balancing the personalization and variance reduction, APFLGate performs better than both APFL and FedGate as seen in Table-2 and Table-3 on Graph classification and Link prediction respectively.

APPROACH	ROC-AUC
APFL ($\alpha = 0.1$)	0.7492
FEDGATE ($\tau = 100$)	0.7703
APFLGATE ($\alpha = 0.1, \tau = 0.1$)	0.8342
APFLGATE ($\alpha = 0.1, \tau = 100$)	0.8347
APFLGATE ($\alpha = 0.1, \tau = 1$)	0.8349
APFLGATE ($\alpha = 0.9, \tau = 10$)	0.8429
APFLGATE ($\alpha = 0.75, \tau = 10$)	0.844
APFLGATE ($\alpha = 0.5, \tau = 10$)	0.8458
APFLGATE ($\alpha = 0.25, \tau = 10$)	0.8472
APFLGATE ($\alpha = 0.1, \tau = 10$)	**0.848**

Table 2. Test performance for different approaches on the Graph Classification task for 16 clients (Higher is better for ROC-AUC).

APPROACH	MAE
APFL ($\alpha = 0.25$)	0.8054
FEDGATE ($\tau = 10$)	0.8055
APFLGATE ($\alpha = 0.1, \tau = 10$)	0.7921
APFLGATE ($\alpha = 0.25, \tau = 10$)	0.7895
APFLGATE ($\alpha = 0.5, 0.75, 0.9, \tau = 10$)	**0.7891**

Table 3. Test performance for different approaches on the Link Prediction task for 28 clients (Lower is better for MAE).

References:
[1] Li, T. (2019, November 12). Federated Learning: Challenges, Methods, and Future Directions. https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions.
[2] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1–2):1–210, 2021.
[3] Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461, 2020.
[4] Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In International Conference on Artificial Intelligence and Statistics, pp. 2350–2358. PMLR, 2021.