FEDWAGG: Federated Learning based on Weighted Aggregation

Sumin Song[*] Junsoo Jang[†] Chanyoung Maeng[‡]

2023.06.08

Abstract

When aggregating updated parameters from clients in federated learning, arithmetic average has been mainly used, which is called as FEDAVG. However, if the data contained in the client is biased to one side or maliciously manipulated, it can lead the model update in the wrong direction, causing performance degradation. In addition, in order to show high performance in federated learning, a prerequisite is required that the data in each client must be distributed in an IID However, this is difficult in the real world. We suggest solution called FEDWAGG, and it tried to solve the non-IID problem through intermediate clusters which gather similar model's weight trained with each different non-IID dataset. We conduct MNIST classification experiments with FEDWAGG and FEDAVG. These experiments demonstrate FEDWAGG stabilizes the learning of the global model by preventing unstable loss and accuracy peak.

1 Introduction

1.1 Background

With the recent development of deep learning model technology, various deep learning models paradigms have begun to emerge. We noted a method called federated learning among several paradigms. This technology was recently announced by GoogleAI in 2017 to solve data privacy issue.(2) Existing centralized model training can invoke data privacy issues, it is because data related to privacy can be transmitted to models located on a central server, causing privacy leakage issues such as, hospitals records and auto-complete keywords. Federated learning is a new way to learn deep learning models, a technology that builds centralized model via decentralized data on many clients. The server stores only the parameters of the model, the data is stored on the device, the clients update the parameters of the model, and sends the updated parameters to the server. Hence, it can ensure privacy while using big data collected from several clients for model training. Federated learning has previously proven its effectiveness in a variety of applications, solving data privacy issues and it is expected to become more advanced technologies in the future.

[*] Department of Computer Science and Engineering, POSTECH, South Korea; e-mail: songsm921@postech.ac.kr

[†] Department of Electrical Engineering, POSTECH, South Korea; e-mail: junsoo37@postech.ac.kr

[‡] Department of Convergence IT Engineering, POSTECH, South Korea; e-mail: mcy5712@postech.ac.kr

1.2 Motivation/Problem

When aggregating updated parameters from clients in federated learning, arithmetic average has been mainly used. However, if the data contained in the client is biased to one side or maliciously manipulated, it can lead the model update in the wrong direction, causing performance degradation. In addition, in order to show high performance in federated learning, a prerequisite is required that the data in each client must be distributed in an IID However, this is difficult in the real world. Noting these points, we would like to try to improve performance by applying weighted average aggregation.

1.3 Related Work

A. non-IID Circumstance
As a study related to federated learning, there is a study that attempted to improve performance in an environment with a distribution of data that is non-IID(3) According to this study, non-IID is known to degrade performance by 55%, and a method of using global data is proposed to solve this problem. The study says that 5% of the CIFAR-10 dataset is used as global data, resulting in up to 30% performance improvements.

B. Aggregation Algorithms in Federated Learning
In generic federated learning, an aggregation algorithm is used for updating a global model. Federated Averaging, FEDAVG is a widely used aggregation algorithm which updates the global model with arithmetic averages of each client's model updates. Since the algorithm just takes an arithmetic average, it is exposed to the imbalanced training data issue between clients. FEDPROX is an aggregation algorithm to enable personalized federated learning.(1) After averaging the model updates received from the devices, the server additionally calculates the penalty term for the learning results of the individual devices.

2 Main Idea

Three main approaches have been proposed to solve the non-IID problem in federated learning, such as data approach, algorithm approach, and system approach. FEDWAGG presents a problem-solving solution that approaches from the system side of the above three methods. FEDWAGG tried to solve the non-IID problem through intermediate clustering and to explain this, we defines a total of three layers: top layer which has global model, middle layer which is located cluster, and bottom layer which has clients.

Phase 1: Client Layer → Clustering Layer
Before running the first FEDWAGG, we assume that each client has one identical global model. In each client, the learning is conducted with each dataset. At this time, dataset is irrelevant to any distribution, but we proceeded by generating non-IID dataset. After the learning, clustering with similar values is carried out through k-clustering without directly delivering each different weight to the global model.

Phase 2: Clustering Layer → Client Layer
A new cluster model is created through FEDAVG, an existing aggregation method for weights gathered in each cluster. This cluster model is re-deployed to the client, client model will be deployed randomly which candidate includes all clients. In other word, if model 1 is collected in cluster 1, cluster 1 model can be re-deployed to model 1.

Phase 3: Client Layer → Clustering Layer

The randomly selected client executes a test with its own data and measures accuracy. Accuracy is sent back to the cluster, and the average value of acuity is ranked between clusters. Based on accuracy, we calculate so that the sum of accuracy of each cluster model is 1 in order to apply weight to the subsequent cluster model weight.

Phase 4: Clustering Layer → Global Model Layer

Using the weights calculated in Phase 3, each cluster model weight is multiplied by the weights to create a new weight. Finally, aggregation is performed on the global model. This process is defined as round, and N round is executed.

We tried to prevent parameters learned by non-IID dataset from being directly updated to the global model by clustering a client model representing a similar value among the newly learned models through clustering in Phase 1 to Phase 2. In addition, through Phase 3 to Phase 4, the cluster model was sent down to the client layer to measure accuracy in other non-IID dataset, and among them, aggregation was performed by reflecting high weight on the cluster model with high accuracy to improve a kind of generalization performance.

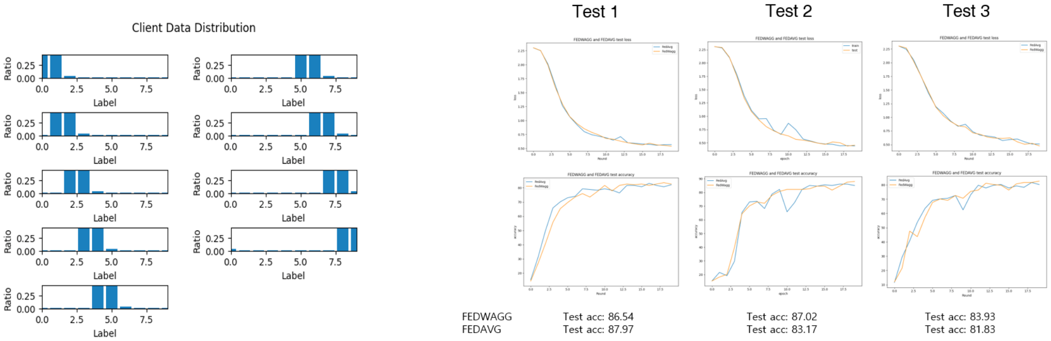
In summary, the weight of the model learned in the state in non-IID was directly reflected as a global model to prevent the model's learning tendency from being sharply biased, and to obtain generalization through validation in the client.

3 Results

We proceeded federated learning by using FEDWAGG and FEDAVG method to the MNIST Classification task. For both, we use the same 2-layer CNN architecture model with same initial parameter. The number of clients is 9, the number of clusters is 3, the number of clients per cluster is 3. We did federated learning with 20 round, and two set of client data distribution is non-IID We evaluate Loss and accuracy of the global model per round, and its average value. We trained the global model of FEDWAGG and FEDAVG algorithm with 1500 non-IID MNIST dataset.(166 training data per clients) The number of client epoch is 5, batch size is 16, learning rate is 0.01, and round is 20.

For first client training data distribution setting displayed in Figure 1 (a), Figure 1 (b) shows the loss and accuracy of global model per every rounds. We can see that there was no significant performance difference between FEDWAGG and FEDAVG. However, the graph in the case of FEDWAGG shows the stabilizing effect to learning global model. On the other hand, the graph of FEDAVG shows unstable learning process.

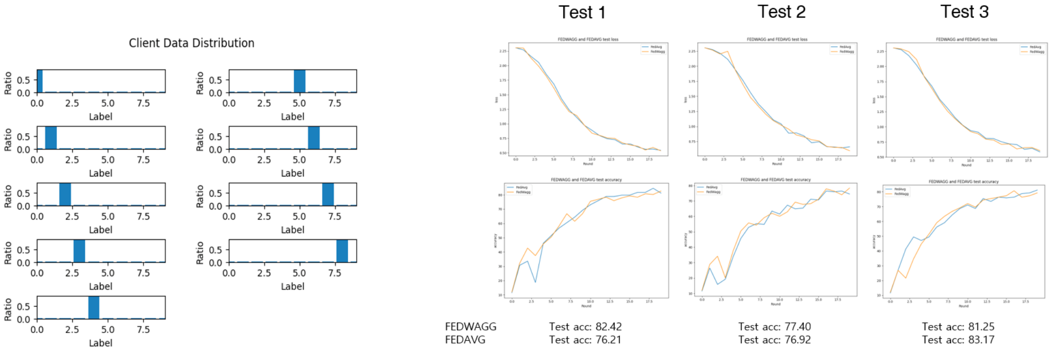
For second client training data distribution setting displayed Figure 2 (a), Figure 2 (b) shows the loss and accuracy of global model per every rounds. We can see that there was no significant performance difference between FEDWAGG and FEDAVG like the previous experiment. However, The stabilization effect of FEDWAGG becomes weaker than in the first experiment.



(a) This is the first non-IID client distribution setting for 9 clients. For each client, two of labels have 0.45 ratio and the other has 0.03, and others have 0.01.

(b) This is outcome of 3 tests for FEDWAGG and FEDAVG test loss and test accuracy, which are trained with the first client data distribution setting. The graph of orange color is for FEDWAGG and the blue color one is for FEDAVG.

Figure 1: Test result on the first training data distribution setting.



(a) This is the second non-IID client distribution setting for 9 clients. For each client, one of labels have 0.9 ratio and the other has 0.02, and others have 0.01.

(b) This is outcome of 3 tests for FEDWAGG and FEDAVG test loss and test accuracy, which are trained with the second client data distribution setting. The graph of orange color is for FEDWAGG and the blue color one is for FEDAVG.

Figure 2: Test result on the second training data distribution setting.

4 Discussion

Summarizing our experimental results, FEDWAGG had the effect of stabilizing global model learning. However, there was no significant performance improvement in FEDWAGG compared to FEDAVG for non-IID dataset.

First, considering the stabilizing effect occured in FEDWAGG, it is because we can control bad parameters to be reflected with less weight to a global model. Hence, the weight of the unique parameter that has an bad effect is reduced. This leads to a stabilizing the accuracy and loss. Second, the reason why stabilizing effect occured weakly in FEDWAGG test result for the second non-IID dataset is because clustering does not work meaningfully for the second case. Unlike the first case, the second client data distribution had little similarity between each client and was completely different. Hence, FEDWAGG algorithm might not construct clusters with similar clients. Third, the reason for the lack of performance improvement in FEDWAGG is because the client models rapidly converged as round repeated. Each client model's parameter converged rapidly as round repeated, After round 5, similarities between each client model's parameter are all 0.99. In other words, after the initial few rounds clustering may not be meaningful. We think this is why FEDWAGG does not outperform FEDAVG meaningfully overall view of federated learning.

A Code Submission

We verify that we have included all the resources relevant to our project, including demo files written in Jupyter Notebook snippets and experimental results. Link: https://drive.google.com/drive/folders/1q3d_VWcge8TOxACSbCuozdtG6NWRTv76?usp=sharing

References

[1] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems 2*, 429–450.

[2] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. 1273–1282.

[3] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.