Reducing Model Cost Based on the Weights of Each Layer for Federated Learning Clustering

Hyungbin Kim, Yongho Kim and Hyunhee Park*
dept. Information and Communication Engineering
Myongji University
Yongin, South Korea
hbkim@mju.ac.kr, yhkim@mju.ac.kr, hhpark@mju.ac.kr*

2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN) | 978-1-7281-6476-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICUFN49451.2021.9528575

***Abstract*— Federated Learning (FL) has a different learning framework from existing machine learning, which had to centralize training data. Federated learning has the advantage of protecting privacy because learning is performed on each client device rather than the central server, and only the weight parameter values, which are the learning results, are sent to the central server. However, the performance of federated learning shows relatively low performance compared to cloud computing, and in reality, it is difficult to build a federated learning environment due to the high communication cost between the server and multiple clients. In this paper, we propose Federated Learning with Clustering algorithms (FLC). The proposed FLC is a method of clustering clients with similar characteristics by analyzing the weights of each layer of a machine learning model, and performing federated learning among the clustered clients. The proposed FLC can reduce the communication cost for each model by reducing the number of clients corresponding to each model. As a result of extensive simulation, it is confirmed that the accuracy is improved by 2.4% and the loss by 47% through the proposed FLC compared to the standard federated learning.**

Keywords—Federated learning, Distributed machine learning, Distributed databases, Distributed processing, Clustering algorithms, Computational modeling

I. INTRODUCTION

Machine learning is showing achievements beyond human cognitive ability in various fields through various methods such as deep neural network (DNN) and convolutional neural network (CNN). Recently, a decentralized training method in which a large number of local clients train a global model in cooperation with a central server after local training is attracting attention. An example of that, Federated Learning (FL) proposed by Google shows high performance while protecting privacy due to distributed model design [1].

FL consists of multiple clients and one FL server, and is a training model in which the weights of models trained from clients are centrally collected. Unlike traditional models, it has the characteristic of privacy protection because raw data is not collected by the FL server. However, if the scale of federated learning is huge, a large amount of uplink and downlink traffic occurs when communicating with the server because the number of participating clients is large [2]. In addition, due to the characteristic of FL, which utilizes each client's data in the training process, it is common to achieve biased training by the small amount of data each client has. In particular, there may be a case in which the data distribution is non-independent and identically distributed (Non-IID). Due to the Non-IID problem, also the Cloud Computing [3] performance cannot be reached.

Therefore, research has been conducted to improve communication efficiency and improve the Non-IID problem in data distribution [4, 5, 6]. A. Huang *et al.* proposes the residual pooling network (RPN) method to improve communication efficiency, through reducing the number of parameters of the FL model [4]. However, it cannot be said that effective improvement is derived because the accuracy performance is reduced compared to the standard FL. In [5], the FL server improves the Non-IID problem by distributing less common data to each client. However, the performance claimed in the paper cannot be generalized because it may show overfitted training results due to the small number of common data given to each client. F. Chen *et al.* proposes federated meta-learning algorithm (FedMeta) using meta-learning, which has recently been attracting attention [6]. Meta-learning is a model suitable for processing small amounts of data for the purpose of training how to learn itself. FedMeta combined with FL and meta-learning is proposed, and it is shown that a model with effective performance can be created with only a small amount of data. Although FedMeta performs better than the standard federated learning in terms of accuracy, the exact figure shows 86.23%, which is far from cloud computing performance, so performance improvement is necessary.

In this paper, we propose FLC to improve the performance in terms of communication efficiency and accuracy of federated learning. As we will see below, our approach can significantly reduce the number of clients participating in the model, thereby increasing the communication efficiency for each federated learning model. In addition, we propose an end-to-end procedure that improves performance in terms of accuracy by performing federated learning among similar clients in the clustering result.

Contributions: Our main contributions in this paper are as follows:

- In order to increase communication efficiency by reducing the uplink and downlink traffic generated in communication between one server and multiple clients, clients are divided into two categories.
- We propose a method to proceed with each of the two clusters as a federated learning model.
- By clustering clients by the weights of each layer of the client learning model, federated learning between similar clients is made to improve the performance in terms of federated learning accuracy.

The rest of this paper is organized as follows: In Section II, we review studies related to federated learning and global average pooling (GAP) and we introduce the proposed method FLC. In Section III, we introduce our approach in detail. After that, the simulation environment and simulation results of the method are presented, followed by conclusions and future research plans.

II. PRELIMINARIES

A. Federated Learning

The traditional machine learning method centralizes the data of all clients to learn. However, as data privacy and data security become issues, centralizing clients' original data is regarded as insecure. In order to solve these problems, the FL which is a new machine learning method for protecting client data, has been proposed.

The standard federated learning protocol goes through 3 steps as follows, and the number of times 3 steps are repeated is expressed as a communication round. An illustration of the federated learning concept is shown in Fig. 1.

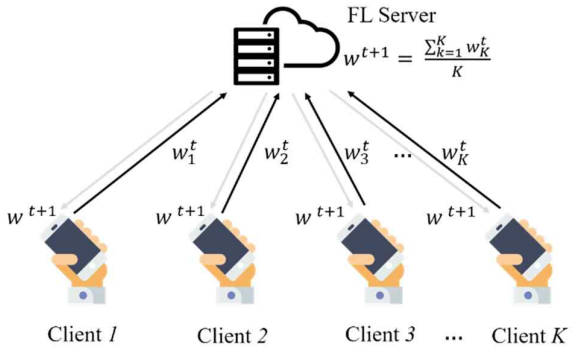

Fig. 1. An illustration of the concept of federated learning.

- ***Local Model Training:*** Clients synchronize with the server. Local training is carried out with the data each client has. When the number of clients participating in federated learning is k, the weight values derived by learning from k clients are transmitted to the central server.
- ***Global Aggregation:*** After local training, the weights of each client are aggregated to a central server. In a large-scale federated learning model, in this process, federated learning environments may constitute of millions of participants by a large number of clients, which can generate huge uplink and downlink traffic [7]. On the central server there are choices of aggregation operator, including FedAvg, and weights are usually handled in the same way as in FedAvg [1].

  The FedAvg method takes the average of the weights collected in the central server and uses it as the initial weight of each client in the next round. When k is the number of participating clients and t is the communication round, can be expressed as a (1).

$$w^{t+1} = \frac{\sum_{k=1}^{K} w_k^t}{K} \tag{1}$$

- ***Update Local Model:*** When the weight comes out through the aggregation operator from the central server, the global model weight is sent back to the client. This value is used as an initial weight when the client learns in the next communication round, and this cycle is repeated until the communication round is finished.

B. A Study to improve communication efficiency

Among clients participating in federated learning, a participant selection method to reduce the training bottleneck by preventing clients with outliers from participating in learning was studied, and the authors in [8] proposed FedCS. Through the FedCS simulation results, it was proved that the more clients participating in the learning, the better the performance. However, FedCS has a limitation that it can be applied only to a simple DNN model because when extended to the training of more complex models, it may be difficult to estimate how many participants should be selected.

III. PROPOSED FLC FOR REDUCING MODEL COST

This section introduces the FLC proposed in this paper. The core of FLC is to cluster clients participating in federated learning and reconfigure them into different federated learning models. An ideal FLC is that the clients are correctly clustered and the federated learning model is reconstructed.

In FLC, the criteria for clustering clients are set by using the weights for each layer of the local training model. After communication round 1, spatial pooling is applied to the weights for each layer of each client collected in the FLC central server, and the average value for each layer of all clients participating in the FLC is derived.

Algorithm 1 represents how to derive using spatial pooling the weight for each layer on the client and the average of the weight parameters for each layer on the server.

Algorithm 1 Spatial Pooling: Each Client and Server

Input: number of client k

Input: number of model layer n

Input: $I \times J$ size feature map

Output: average weight for n layer of the client k $w_{c\_avg}^{kn}$

Output: average weight for n layer of all clients $w_{sv\_avg}^{n}$

for $k = 1, \ldots, K$ **do**

  for $n = 1, \ldots, N$ **do**

    $w_{c\_avg}^{kn} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} w_{s(i,j)}^{kn}}{I+J}$

  end for

end for

for $n = 1, \ldots, N$ **do**

$w_{sv\_avg}^{n} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{j=1}^{J} w_{s(i,j)}^{kn}}{K(I+J)}$

end for

$w_{sv\_avg}^{n}$ for each layer of all clients collected in the FLC central server obtained through Algorithm 1. is compared with $w_{c\_avg}^{kn}$ for each layer of each client, and a model to be clustered is determined. This process is shown in Algorithm 2.

Clients with $w_{c\_avg}^{kn}$ larger than $w_{sv\_avg}^{n}$ are designated as FLC_high, otherwise, as FLC_low, and from the next communication round, federated learning is performed between clients corresponding to each model based on the clustered results.

Algorithm 2 Model clustering

Input: $high = 0$; $low = 0$

Input: average weight for n layer of the client k $w_{c\_avg}^{kn}$

Input: average weight for n layer of all clients $w_{sv\_avg}^{n}$

for $k = 1, \ldots, K$ **do**
  for $n = 1, \ldots, N$ **do**
    if ($w_{c\_avg}^{kn} \leq w_{sv\_avg}^{n}$) **then**
      $low \mathrel{+}= 1$
    else
      $high \mathrel{+}= 1$
  end for
  if ($low \leq high$) **then**
    cluster the kth client as FLC_high model
  else
    cluster the kth client as FLC_low model
end for

An illustration of FLC concept with Algorithm 1 and Algorithm 2 applied is shown in Fig. 2. Through Algorithm 1, spatial pooling is performed, and w_{Avg} is generated. During Algorithm 2, w_{Avg} of each model is analyzed and clusters each model into FLC_high and FLC_low models.

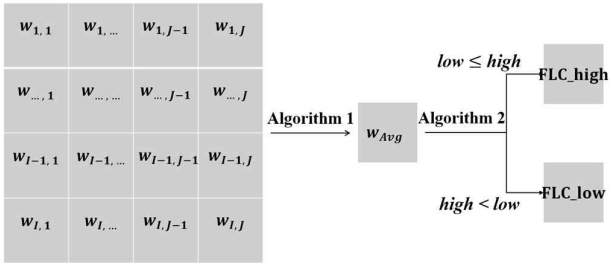

Fig. 2. An illustration of the concept of FLC.

IV. SIMULATION

To evaluate the performance of the proposed method, the LeNet-5 model is used [9]. The weight parameters of each layer of the model we used are shown in Table I. Cluster clients into FLC_high and FLC_low models by applying Algorithm 1 and Algorithm 2 proposed in Section III to the weights of each layer of clients.

The MNIST dataset is used as the data, and the data are distributed to become Non-IID for 10 clients. The simulation environment is shown in Table II.

TABLE II. SIMULATION ENVIRONMENT

Simulation Environment	
Edge device	10
Local epoch	1
Batch size	100
Optimizer	SGD
Train/Test ratio	4:1

Since the proposed FLC method finally creates two models, the average accuracy and loss values of the two models are used for performance comparison. The average value obtained by simulating each of the standard federated learning method and the proposed FLC method 100 times was compared. The performance evaluation results are shown in Fig. 3, and our simulation performance summarize in Table III.

TABLE III. Simulation performance at 100^{th} communication round

	Accuracy	Loss (cross-entropy)
FL_standard	0.9298	0.2529
FLC_high	0.9601	0.1423
FLC_low	0.9440	0.2015
FLC_avg	0.9521	0.1719

FLC_high and FLC_low are the names of models clustered using the method proposed in this simulation. As shown in Table III, compared to the standard federated learning, the accuracy was improved by 2.4% and the loss by

TABLE I. THE WEIGHT PARAMETERS OF EACH LAYER OF THE LENET-5

Layer	Layer type	Feature Maps	Kernel (Filter)	Input size	Trainable parameters	Activation
Input	image	-	-	32x32	-	-
C1	Conv	6	5x5	28x28	156	Sigmoid
S2	Pool	6	2x2	14x14	12	Tanh
C3	Conv	16	5x5	10x10	1516	Tanh
S4	Pool	16	2x2	5x5	32	Sigmoid
C5	Conv	120	5x5	1x1	48120	Tanh
F6	Dense	-	-	84	10164	Tanh
Output	Dense	-	-	10	-	Softmax

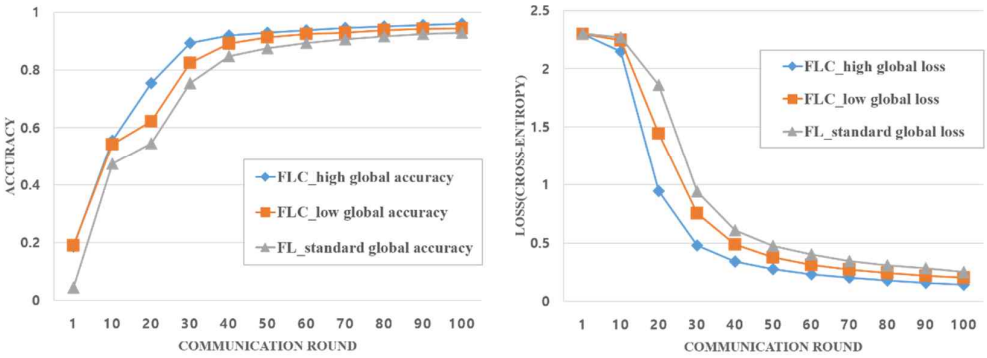

Fig. 3. Simulation on MNIST dataset.

47%, confirming that our standard of FLC used to cluster clients was appropriate.

In addition, the clients were divided into two classes by FLC, and training was carried out in each model of the two classes. As a result, FLC can reduce the communication cost for each model by reducing the number of clients corresponding to each model.

Contributions: Our simulation results in this paper are as follows:

- We use the average performance of FLC_high and FLC_low to confirm the performance of the proposed FLC.
- In terms of accuracy, it shows improved performance in all communication rounds, and in terms of loss, it shows a clear performance improvement after 10^{th} communication round.

V. CONCLUSION AND FUTURE WORK

In this paper, the clients participating in federated learning are clustered using the parameters of the federated learning model. By using the clustered clients to be trained in different models, communication cost was reduced and performance was improved for each model. This is because the number of clients participating in one federated learning model could be reduced by classifying the federated learning model through proposed FLC. In addition, it is confirmed that proposed FLC improves the performance of the standard method in terms of accuracy and loss through iterative simulations. Specifically, the average performance of the models generated through proposed FLC is used in the simulation, and it is confirmed that not only the average performance but the performance of each model is improved compared to the performance of standard federated learning.

Future work may include performance measurement according to the proportion of clients clustered in each model and a study on how many clustered models we need to set for ideal performance.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2019R1F1A1060742, Big Data Analysis and Development of Security Protocol for Massive IoT Security from Cloud to Edge Computing) and the Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00990, Research on Advanced Core Technologies for WLAN based on eXplainable AI).

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data." *Artificial intelligence and statistics*. PMLR, 2017.

[2] Kairouz, Peter, et al. "Advances and open problems in federated learning." *arXiv preprint arXiv:1912.04977* (2019).

[3] Armbrust, Michael, et al. "A view of cloud computing." *Communications of the ACM* 53.4 (2010): 50-58.

[4] A. Huang, Y. Chen, Y. Liu, T. Chen, and Q. Yang, "RPN: A Residual Pooling Network for Efficient Federated Learning." *arXiv preprint arXiv:2001.08600* (2020).

[5] Zhao, Yue, et al. "Federated learning with non-iid data." *arXiv preprint arXiv:1806.00582* (2018).

[6] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication." *arXiv preprint arXiv:1802.07876* (2018).

[7] K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, arXiv:1902.01046. [Online]. Available: https://arxiv.org/abs/1902.01046.

[8] T. Nishio, and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge." *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.