Blockchain-based Node-aware Dynamic Weighting Methods for Improving Federated Learning Performance

You Jun Kim, Choong Seon Hong
Department of Computer Science and Engineering
Kyung Hee University, 17104
Republic of Korea
{yj4889, cshong}@khu.ac.kr

***Abstract*—Federated learning (FL) is a decentralized learning method that deviated from the conventional centralized learning. The FL progresses learning locally on each device and gradually improves the learning model through interaction with the central server. However, it can cause network overload because of limited communication bandwidth and the participation of a huge number of users. One of the ways to minimize the network load is for the model to converge rapidly and stably with target learning accuracy. In this paper, we propose blockchain based federated learning scenario. Blockchain can efficiently induce users to participate in learning and can separate each participating user as a 'node'. In addition, it can be pursued the integrity, stability, and so on. We consider two types of weights to choose the subset of clients for updating the global model. First, we consider the weight based on local learning accuracy of each client. Second, we consider the weight based on participation frequency of each client. We choose two key performance indicators, learning speed and standard deviation, to compare the performance of our proposed scheme with existing schemes. The simulation results show that our proposed scheme achieves higher stability along with fast convergence time for targeted accuracy compared to others.**

Keywords—Federated Learning, Blockchain, Node Selection, Weighting scheme

I. INTRODUCTION

Recently, Federated Learning is very attractive to the research community because it improves the learning model accuracy as well as it preserves the user's privacy in distributed manner. Federated Learning method prevents data leakage by learning the model within each user device with locally collected data. Next, the learned models from each device are sent to the central server and aggregated at the server to improve the global learning model accuracy[1]. With the development of the edge computing, the computing power of edge side is gradually increasing. As edges, devices can be mobile equipment, smart gateways, base stations, smart sensor, unmanned aerial vehicles (UAV), etc. as well. Furthermore, advances in sensor networks and communication technologies have led to an explosion in the amount of data that can be communicated between objects and objects as well as people[2]. Cisco forecasts 50 billion devices will be connected to the Internet in 2020[3]. In this trend of times, federated learning plays a very important role because it can even learn privacy-sensitive data.

There are several limitations to federated learning. The first is the reliability of the learning model from each devices and the incentive for users to participate in learning process. Malicious users can have an adverse impact on the global model by adjusting the local model. Users also lack the motivation to participate in learning because they use their own computing resources and data to learn model. The second is the problem of network overload. The number of users participating in the learning could be thousands or more. Massive amounts of models are transmitted at the same time, which can cause network overload because of limited bandwidth.

Blockchain-based federated learning can solve the above two problems. Blockchain stores all the learning models that are transmitted in integrity. Therefore local models cannot be adjusted. In addition, if users receive cryptocurrency in exchange for their participation in learning, they can gain an incentive to participate. Network overload can be mitigated by accurate node recognition of Blockchain[4]. One of the best ways to reduce network overload is to quickly and stably converge the target accuracy of the learning model in federated learning. In addition, fast and stable target accuracy convergence is also important when considering the user's unexpected departure from learning participation.

We propose node recognition based local learning weighting method, node selection method according to the frequency of participation and amount of data, and weighting method according to the frequency of participation to converge fast and stable learning accuracy. We also compare and analyze the differences in learning speed and stability between the proposed method and conventional federated learning. As a result, the methods proposed perform better in terms of learning speed and stability compared to conventional federated learning.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea goverment(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing) *Dr. CS Hong is the corresponding author.

II. Related work

A. Blockchain

Blockchain technology was first proposed by [5] with bitcoin. All participant can be recognized uniquely in blockchain network[6]. We can select the right participants in federated learning because all nodes have unique blockchain addresses which is immutable and unalterable. Also, all transactions are stored in blocks and each block is linked together because it contains a hash value of the previous block. These features ensure the integrity and security of the transaction data, and cannot be falsified. Therefore, transactions can be made without third-party intervention.

B. Federated Learning

Machine learning and deep learning models in federated learning is transmitted to each user's device from the central server. The transmitted model is learned from its own data on each device, and the learned model is sent to the central server. Then, the server combines all models from each device using the Federated Averaging algorithm and sends them back to the user device. The aggregated model weights in Federated Averaging is presented in the following equation[1].

$$w_{t+1} \leftarrow \sum_{n=1}^{N} \frac{k_n}{k} w_{t+1}^n \qquad ①$$

k is the total number of datasets of all users who participated in the tth federated learning. k_n is the number of nth participated user datasets and the w_{t+1}^n is the model weights learned from the nth participated user.

There are federated learning features that need to be considered[1] :

- ***Non-IID***: The data of a particular user does not represent a population distribution because it contains the characteristics of that user.
- ***Massively distributed***: The number of users participating in the federated learning is much larger than the average number of examples per user.
- ***Unbalanced***: Local datasets are unbalanced because some users produce a lot of data for model learning and some users produce less.
- ***Limited communication***: Devices that participate in model learning are frequently disconnected or slow.

C. Temporally Weighted Aggregation

Temporally Weighted Aggregation algorithm is proposed by [6] to increase the communication efficiency of federated learning. It enables faster convergence of learning accuracy compared to conventional federated learning. The temporally weighted aggregation for model integration is presented in the following equation.

$$w_{t+1} \leftarrow \sum_{n=1}^{N} \frac{k_n}{k} * \left(\frac{e}{2}\right)^{-(t-timestamp^n)} * w_{t+1}^n \qquad ②$$

e is the natural logarithm used to present time effects, t is the current round and t$imestamp$ is the round in which w^n was updated most recently. But it considered only the latest learning round for each user.

III. Senarios and System Architecture

Fig 1 shows our overall system model. Users can be any edge sides, including mobile phones, smart gateways and unmanned aerial vehicles (UAV) which have computing power to participate in model learning. The model is sent to all learning participants by certain user who want to provide AI services. Models learned locally are sent back and integrated to the user. All locally learnt models are stored in the blockchain. This ensures the integrity of model data and prevents malicious user's learning decline. In addition, blockchain is the best technology that can provide users with

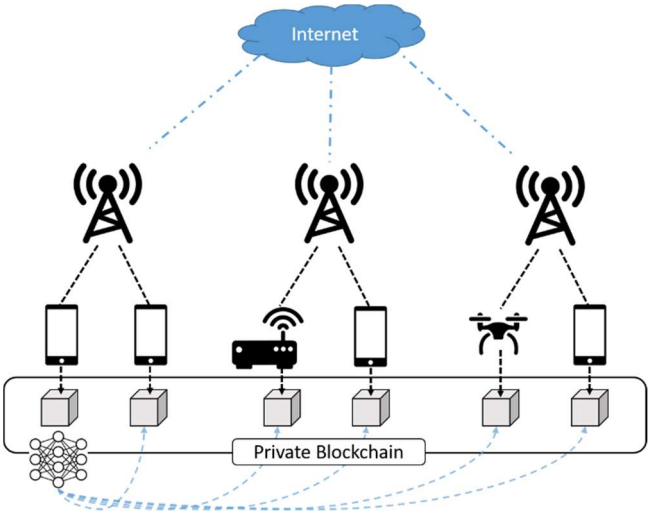

Figure 1. System Model

efficient incentives for motivation to participate. Users offer their computing resources for model learning. Users can receive cryptocurrency in exchange for providing computing resources, which can create an ecosystem of data trading without third-party intervention.

Another advantage of blockchain in federated learning is that users can be accurately recognized as blockchain local addresses. A number of strategies based on user awareness improve learning speed and stability. There are some methods we propose and it is different from the conventional federated learning equation ①, which only considers the number of datasets of each user.

- ***Weighted by local learning accuracy*** : If the certain user who wants to learn the model for the service provider has data including a label related to the service, it can be used in federated learning. We call the user as the service provider. The user sends the data which the user has to the participating users. Then, the accuracy of the models learned from each local user is measured by the data sent. The aggregated model weights are presented in equation ③. The sun of local accuracy for all participating users is a, and the accuracy of the model learned from the nth user is a_n.
- ***Select participants based on the frequency of learning participation and the number of local data set***: Depending on the characteristics of the users involved in learning, the learning accuracy

of the global model is very different. [1] selects the participating users randomly. However, Selectness the optimal user based on the frequency of users' learning participation and the number of datasets improves learning accuracy and speed. The method is presented in following Algorithm 1. f_i is the frequency of the ith user's participation in learning ($f = \|f_i\|$). d_i is the number of datasets of the ith user ($d = \|d_i\|$) and the total user set is U ($i \in U$). r is the rate of influence between the frequency of participation and number of datasets ($0 \leq h \leq 1$). n is the number of all users. C means the $r * k$ user candidates to participate in the learning as an output of Algorithm 1

- ***Weighted by frequency of participation in learning***: Equation ① considers only the number of datasets. However, we apply weights based on frequency of learning participation in the model aggregation process. The proposed weighting algorithm is shown in Algorithm 2. If weights based on frequency ratio are applied to the aggregation formula without modification, the difference in influence of the local model of users will be too large. Therefore, we compress the frequency difference by p(line 5) and u is the average frequency of all users.

$$w_{t+1} \leftarrow \sum_{n=1}^{N} \frac{a_n}{a} * w_{t+1}^{n} \qquad ②$$

Algorithm 1 : ***Select optimal participants***

1 : ***funtion*** $selectParticipants(f_i, d_i, U, r, n, i)$
2 : ***input*** : f_i, d_i, U, r, n, h
3 : ***Output*** : C
4: $a_i \leftarrow \frac{d}{d_i}$ ***for each*** $i \in U$
5: ***for*** $i \in U$
6: $s_i \leftarrow h * \frac{f_i}{f} + (1-h) * \frac{a_i}{a}$ s. t. $a = \|a_i\|$
7: ***Repeat*** $r * n$
8: $C \leftarrow i \quad s.t. \min s_i, s_i \in S$
9: ***romove*** $\min s_i$ $from$ S

Algorithm 2 : ***Weighted by frequency of participation in learning***

1 : ***funtion*** $WeightedFrequency(f_i, U, p, u, i)$
2 : ***input*** : f_i, U, p, u, i
3 : ***Output*** : e
4 : $e_i \leftarrow p(u - f_i)$ ***for each*** $i \in U$
5 : $f' \leftarrow \min e_i$
6 : ***for*** $i \in U$
7 : $e_i \leftarrow e_i + f' + 1$

Model aggregation algorithms with the proposed three methods are shown in Algorithm 3. The user execution(Line 11) is almost similar to the client update in [1]. The difference is that users calculate accuracy using the service provider's data. In addition, line 13 is a model aggregation equation proposed based on the number of user's data, the accuracy of the model from each user, and the frequency of the participation in learning. $alpha, beta, gamma$ is a ratio of the effect each method has on the aggregated model.

Algorithm 3 : ***Proposed methods for federated learning.***

1: ***Service provider***:
2: ***initialize*** $w_0, epochs, f_i, d_i, U, r$
3: $k, p, P_k, u, g, data$
4: $alpha, beta, gamma$
$s.t. alpha + beta + gamma = 1$
5: ***for*** $epochs \in \{0, 1, 2, 3, 4 \dots\}$
6: $C \leftarrow selectParticipants(f_i, d_i, U, r, n)$
7: $m \leftarrow max(n * g, 1)$
8: $P_k \leftarrow random\ set\ of\ m\ users\ in\ C$
9: $u \leftarrow avg\ f_i$
10: $e_i \leftarrow WeightedFrequency(f_i, U, p, P_k, u, i)$
11: ***for*** $n \in P_k$
12: $a_n, w_n \leftarrow UserExecution(t, w_{epochs}, data)$
13: $f_n \leftarrow f_n + 1$
14: $w \leftarrow \sum_{n=1}^{N} (alpha * \frac{k_n}{k} + beta * \frac{a_n}{a} + gamma * \frac{e_n}{e}) * w_n$

Before the performance analysis, we set the dataset for Non-iid, massively distributed, unbalanced. Dataset is MNIST handwritten. There are a total of 100 users and each user has one, two or three random labels. Also, users who own the same label randomly divide the data on that label. Examples of the types and number of data owned by each user are shown in Table 1 and only nine users are represented for convenience. There are 100 users, but only 9 users are shown in table 1 for convenience. User1 has a total of 525 data with 7 labels and 9 labels. User 4 has only one label with 5. User 5 has a total 1965 data.

IV. Performance and Analysis

In order to compare the performance of the proposed method and the conventional federated learning, we set up the MNIST data set as shown in Table 1. And we used multi-layer perceptron(MLP).

We want to find the optimum ratio of the values of alpha, beta and gamma in Algorithm 3 for rapid learning convergence. We found the optimal value of alpha, beta and gamma inefficiently. This method can be easily replaced by matching algorithm or reinforcement learning.

Table 1. Data setting

Labels / Users	0	1	2	3	4	5	6	7	8	9	Total
User 1	0	0	0	0	0	0	0	78	0	447	525
User 2	54	0	0	0	0	0	404	0	0	0	458
User 3	0	0	0	398	0	0	0	0	0	0	398
User 4	0	0	0	0	0	420	0	0	0	0	420
User 5	0	1672	0	0	0	0	0	0	293	0	1965
User 6	264	0	0	0	0	0	0	0	0	0	264
User 7	0	0	0	0	0	149	289	235	0	0	673
					⋮						
User 100	0	692	0	0	0	0	0	0	458	0	1150

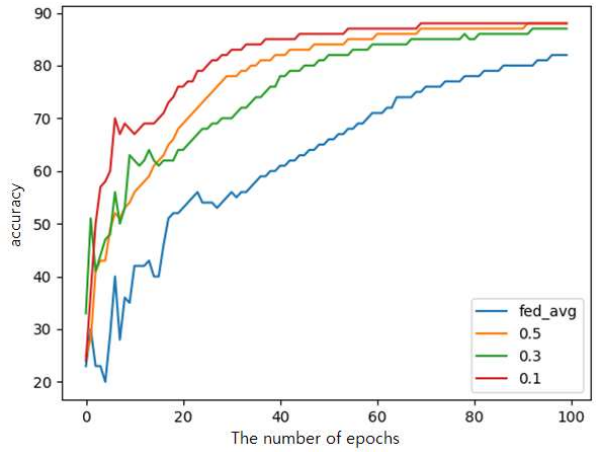

Figure 2. Comparison learning speed

Figure 2 shows the accuracy according to r in Algorithm 1 with parameters (Table 2). r decides which user candidate will participate in the learning. If r is 0.9 out of 100 users, there are 90 optimal candidates to participate in the learning. Conventional federated learning(fed_avg) and $r = 0.5$, 0.3 and 0.1 are compared in Figure 2. And the average and standard deviation are shown in Table 2. The average of proposed strategy($r = 0.1$) is 18.64 higher than fed_avg and standard deviation is 5.1376 lower.

Table 2. Parameter

Parameter	*Value*
r	0.1
k	100
h	0.5
p	0.5
g	0.1
$number\ of\ data$	1000

Table 3. The average and standard deviation

Strategy	Average	Standard deviation
Conventional Federated learning	62.69	16.0939
$r = 0.5$	77.45	14.0402
$r = 0.3$	75.59	12.2947
$r = 0.1$	81.33	10.9563

V. Conclusion and Future Work

Federated learning with blockchain is very slow in terms of learning rate in unexpectedly distributed, non-iid data and blockchain environments. The accuracy and frequency factors of learning of each model were considered for model aggregation. Therefore, this paper presented novel methods of blockchain-based node recognition for improving learning speed. The aggregated algorithm we propose is significantly faster and more stable than conventional federated learning algorithm.

References

[1] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP, volume. 54, 2017.

[2] Qingchen Zhang Laurence T.Yang Zhikui Chen Peng Li, "A survey on deep learning for big data",Information Fusion Volume 42, pages. 146-157, July 2018, M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[3] Seok Won Kang, Choong Seon Hong,, "Openflow based on Edge Cloud Structure for Efficient Packet Forwarding in Distributed Cloud Environment ",Korea Computer Congress(KCC 2018), page.1306-1308, June 2018

[4] Lifeng Liu, Chao Wu, Jun Xiao, "Blockchain-Based platform for Distribution AI", No. 764. EasyChair, 2019.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. [Online] Available: http://www.bitcoin.org/bitcoin.pdf, [downloaded. 01.May.2019]

[6] Salah, Khaled, et al. "Blockchain for AI: review and open research challenges." *IEEE Access* 7 (2019): 10127-10149.

[7] Chen, Yang, Xiaoyan Sun, and Yaochu Jin. "Communication-Efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation." *arXiv preprint arXiv:1903.07424* (2019)