A Study on Blockchain-Based Asynchronous Federated Learning Framework

Zhuohao Qian[1], Cho Nwe Zin Latt[1], Sung-Won Kang[2], Kyung-Hyune Rhee[3]

[1]Dept. of Information Security, Pukyong National University
[2] Dept. of Artificial Intelligence Convergence, Pukyong National University
[3]Division of Computer Engineering, Pukyong National University

zhuohaoq@gmail.com, chocho1612@pukyong.ac.kr, jsm2371@hanmail.net, khrhee@pknu.ac.kr

Abstract

The federated learning can be utilized in conjunction with the blockchain technology to provide good privacy protection and reward distribution mechanism in the field of intelligent IOT in edge computing scenarios. Nonetheless, the synchronous federated learning ignores the waiting delay due to the heterogeneity of edge devices (different computing power, communication bandwidth, and dataset size). Moreover, the potential of smart contracts was not fully explored to do some flexible design. This paper investigates the fusion application based on the FLchain, which is the combination of asynchronous federated learning and blockchain, discusses the communication optimization, and explores the feasible design of smart contract to solve some problems.

1. Introduction

Nowadays, the machine learning has made great breakthroughs in image recognition, natural language processing, recommendation system and other fields, and such developments and breakthroughs are based on the demand for a large amount of data. In the meantime, with the help of edge computing, the scale of the IOT continues to expand, and the sharp increase in the scale of devices has produced a large amount of data for machine learning. Nevertheless, the previous methods of using data are confronted with the test of privacy. Therefore, the federated learning under edge computing optimization has become its solution.

The federated learning attracted attention after Google proposed Fedavg algorithm in 2016 [1]. Initially, in a bid to solve the problem that Android mobile terminals do not disclose any local data for joint training, each device trained the model and uploaded it to the server. All the models were aggregated in the server. Fedavg is a client-Server structure composed of terminals and aggregation servers. However, in cross-device edge joint scenario, the conventional client-Server structure will face risks such as single-point failure, malicious attacks from nodes and provide low system stability and robustness. Consequently, the FLchain architecture integrating blockchain and federated learning is proposed. In FLchain record the updated model on-chain and design a reward distribution mechanism to incentivize participants. Most of the existing literature on FLchain focuses on federated training with a synchronization mode. It waits for all nodes to aggregate in each loop. However, in the IoT scenario with different computing power, bandwidth, and data size, it will result in meaningless waiting for delayed nodes in the same round of high-performance devices, harming convergence. so the asynchronous federated learning method applicable to blockchain distributed system is worthy of discussion. Also, non-IID(independent and identically distributed) has always been a hot topic in the federated learning world. However, current FLchain architecture research lacks the design to cope with non-IID and does not fully utilize the benefits of smart contract compilation and consensus.

This paper investigates the fusion application based on asynchronous federated learning and blockchain, discusses the communication optimization scheme from the aspects of delay model, communication frequency and communication information size, and explores the feasible design of smart contract for reward allocation, malicious detection and Non-IID

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 explains the asynchronous FLchain architecture. Section 4 discusses the challenges and directions. Section 5 concludes this paper.

2. Related work

The classic representative of synchronous aggregation is Fedavg[1]. After the aggregation server collects all local model updates from the working node. The total training time in a round depends on the slowest node. The difference therefrom is the asynchronous combination method, which executes the global model update immediately after receiving the local update, allowing each node to be in different iteration rounds, thus eliminating the lag caused by waiting for each other during the synchronization.

The blockchain maintains a tamper proof distributed ledger in a cryptographic way, and leverages the smart contract composed of automated script codes to provide some protocol designs such as identity authentication, audit, traceability, and anti-counterfeiting. Chen et al. [9] proposed using a decentralized framework to store global models and local updates in the blockchain, and weng et al. [10] leveraged the blockchain's value-driven mechanism to motivate all parties to participate in federated training, with the goal of enhancing federated learning. FLchain architectures have been extensively researched for their robustness and training quality. For the research of FLchain architecture, most of the existing work gives consideration to the synchronization method, which means that all local updates need to be collected before the block is listed. Still, in the IOT scenario with heterogeneous devices, there will be problems, not only causing unnecessary waiting between nodes, but also the excessive size of a single block in the blockchain will prolong the propagation time, which will raise the bifurcation rate [2]; consequently, the FLchain architecture will be a more suitable asynchronous federated learning mode. Liu et al. [3] designed the federated learning with asynchronous convergence (FedAC) considering obsolescence coefficient in the IOT scenario of edge computing plus. Feng et al. [4] puts forward an asynchronous federated learning (BAFL) framework based on blockchain, calculates the optimal block generation rate of blockchain to achieve the purpose of efficient transaction, and dynamically adjusts the local training time to balance the communication delay and local computing energy consumption.

3. Asynchronous FLchain architecture

Asynchronous FLchain can be divided into eight communication and computing processes.

<In the local device>

Step 1: Download the initialization model

Step 2: Local model training

Step 3: Upload the model update

<In the miner>

Step 4: Generate the candidate block

Step 5: Cross validation

Step 6: Generate the block from the selected miner

Step 7: Aggregate the global model & cross validation

<In the local device>

Step 8: Download the global model and enter the next iteration

The step 1 ~ 3 are carried out by local devices and the step 4 ~ 7 are processed by miner nodes(edge server) in P2P network. In the step 8, local devices download the global model from the blockchain network and then carry out the next round of iteration.

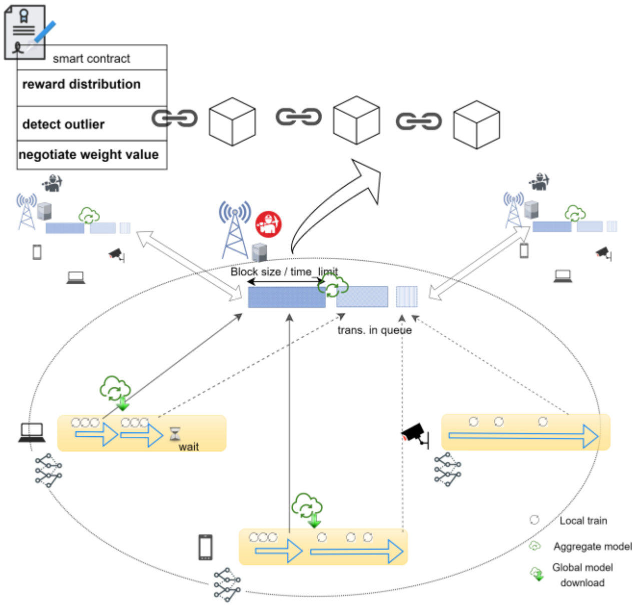

Figure 1: Asynchronous FLchain architecture.

The figure 1 reveals training process of asynchronous FLchain and the way to interact with blockchain queues. It indicates that the local device can upload the updated local model and download the latest global model at any time after completing this round of training. Each node is allowed to be in a different update round. In the meantime, the principle in the blockchain queue is to fill in a candidate block according to the maximum time limit τ or block size. After cross verification, such block is mined by the miner screened by PoW. Also, the updated models are still queuing or arriving to the miner. As a result, in the asynchronous mode of FLchain, the miner updates the global model when a time limit or block size requirement is reached. Ma et al. [5] demonstrated the update approach has a better convergent effect in the aggregate quantitative *m* models per a round. Therefore, the method overcomes the long duration of a single round of synchronous FL and solves the adverse effects of frequent communication and model obsolescence in the pure asynchronous mode.

4. Challenges and directions

This section discusses the challenges and countermeasures of designing asynchronous FLchain from the perspectives of communication optimization and smart contract application.

Asynchronous FLchain provides an exploration of the integration of two cutting-edge technologies. Nevertheless, they, whether asynchronous federated learning or blockchain technology, are faced with many unfinished challenges. For example, there are some problems with asynchronous federated learning, such as high communication frequency, data offset, Non-IID and privacy security. Blockchain is a high-delay system based on the complex consensus mechanism. The unreasonable combination of the two will magnify each other's shortcomings. As a result, this paper discusses the communication optimization from the aspects of delay model, communication frequency and the size of communication information, and explores the potential of more smart contracts to solve the problems of compensation allocation, model poisoning, data distribution offset and so on.

3.1 Communication optimization

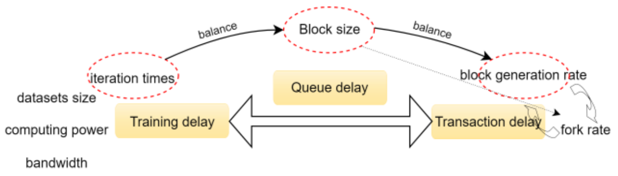

Figure 2. FLchain delay model

A global model update based on block size or maximum time limit is performed in the asynchronous FLchian system. The fig2 shows the system's delay model, it is divided into three parts: training delay, queue delay and transaction delay. Wherein, the local iteration number, block size and block generation rate may be used for balancing and optimization.

3.1.1 Parameter

Dynamically Adjusting the Local Iteration**:** Reasonably mobilize the computing resources of the device and reduce communication congestion [4].

Calculating the Optimal Block Size**:** The block size is a complex parameter. First, the design of block size with reference to the training update rate and block generation rate can optimize the queue delay. Nonetheless, what is more complicated is that a large block can carry more single-round information and improve the convergence performance, which however will affect the transaction confirmation and block propagation delay and improve the bifurcation rate [2].

Optimal Block Generation Rate**:** While speeding up the search of random numbers, it will raise the bifurcation rate, but affect the transaction processing. Hence, it is necessary to find an optimal value [4].

3.1.2 Communication frequency

Communication frequency refers to the number of communication loops between worker and server (on P2P network). In the FL scenario, the optimal relationship between local computing and communication frequency is usually studied and measured. For example, the classic Fedavg communicates after multiple rounds of local computing, which is intended to optimize the communication at the cost of compromising the local computing energy consumption. Further, to avoid the communication congestion of asynchronous FLchian. BAFL [4] proposes to limit the learning times of each device according to the transaction processing rate and allocate more proportions to high-performance devices. In addition, they built mathematical models to balance the size between time delay and device energy consumption.

3.1.3 The size of communication information

The size of communication information can refer to transmission model or gradient size. For an example, the size of CNN model is 100~500MB, and the number of parameters is up to 20~140 million. Because of the large-size, the redundant and the confidential data, it is not suitable for blockchain. However, the model parameters are required the collaborative processing and verification in the federated learning; thus, the full-model compression or the partial-model saving strategy is generally utilized on the IPFS.

3.2 Design of smart contract

Smart contract is a piece of code written on the blockchain, which features self-verification, automatic execution and tamper-proofing. Recently, the federated learning has been adopted to design incentive strategies to gain better model quality. Nonetheless, it was not given full play to its potential of federated learning.

3.2.1 Design for reward distribution

Smart contracts are often used in the federated learning to design reward distribution. There are different distribution mechanisms can be adopted, such as based on the reputation retention mechanism, the bidding strategy, and the voting mechanism, etc.

As usual, the worker contribution is evaluated by referring to the parameters such as dataset size, model quality (accuracy, loss, cost etc.) or reputation score. However, it is not appropriate to evaluate the model update with accuracy due to the Non-IID data between devices. In the proposal of Feng et al. [4], worker is comprehensively evaluated according to data size, model correlation, number of errors and other parameters based on entropy weighting algorithm. Zhang et al. [6] designed an incentive mechanism based on smart contract and rewards the token to the worker according to the data size and centroid distance obtained in the model training.

In the asynchronous federated learning, as each node allows to update the local model at any time in different iteration rounds, the final participation of each node will be different. Accordingly, weighting the data size, the model relevance and the iteration participation to evaluate the contribution becomes a solution in the asynchronous FLchain. Especially, our attention should be paid to the high computational cost in the evaluation process.

3.2.2 Design for Malicious detection

Although blockchain can indeed retain and trace malicious acts, it is still difficult to detect malicious transmission models. It is usually proposed to calculate the accuracy on the common test data set for verification; yet it is not applicable in the scenario of data distribution shift (non-IID).

Liu et al. [3] and Qu et al. [7] both bring forward to cross verify the correctness of the uploaded model by whether the data size is proportional to the local training time and strengthen trust in combination with the time proofing under SGX technology. Nevertheless, this design is not aimed at poisoning attacks or cannot solve the practical problems.

Xu et. al [8] proposes to design a smart contract for defense, aggregate a temporary global model, and generate a list of the difference between each local weight and the temporary global weight. In this list, the Box-plot is utilized to rule out the abnormal values for verification, and the dynamic threshold is set with the number of training rounds. So far, there has been no particularly good solution to the Byzantine general problem in federated learning. Whether a reliable solution can be designed in conjunction with smart contract has become the research direction.

3.2.3 Design to against Non-IID

Data distribution shift（Non-IID）is a difficult problem in the federated learning. Generally, the strategies such as dynamically adjusting training parameters and training personalized model are employed to deal with it. The smart contract is based on the characteristics of underlying compilation and consensus, which can help design flexible strategies.

For example, in the asynchronous federated learning, the delayed nodes (large data sets/low computing power) may be given low weight, resulting in sustained low participation, which will aggravate the damage to the ability to resist the Non-IID. Ma et. al [5] proposes that dynamically setting the learning rate η based on the computing power of nodes can reduce the jitter when converging Non-IID data. Xu et. al [8] balances the weights of local model and global model to generate personalized model training, and relies on the smart contract to design the optimal weight of each round in the synchronous federation and reach a consensus. The methods are feasible directions to explore the use of smart contract to design a personalized model training.

5. Conclusion

In the IOT scenario of edge computing, the FLchain, which is federated learning method with the help of blockchain, is better than the conventional FL architecture in combating single-point failure and malicious tracing. Meanwhile, smart contracts are often employed to design compensation allocation to motivate the participants to provide high-quality model updates. However, the existing FLchain research focuses on the synchronous mode, which is not applicable to the scenario of heterogeneous edge devices in communication and computing. Hence, this paper discusses an asynchronous FLchain architecture, and explores the communication optimization from three aspects including parameter, communication frequency and communication information size. Meanwhile, it takes into account the challenges in benefit distribution, malware detection, Non-IID and other problems as well as the available space of smart contract. This paper summarizes several powerful development directions of FLchain in the fusion field. We hope to design FLchian system with low communication delay and resistance to Non-IID by referring to the above aspects in the future works. Also, we adds consideration of privacy computing to improve the security part.

Acknowledgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) (IITP-2022-2020-0-01797) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2021R1I1A3046590)

Reference

[1] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

[2] Wilhelmi, Francesc, Lorenza Giupponi, and Paolo Dini. "Blockchain-enabled Server-less Federated Learning." arXiv preprint arXiv:2112.07938 (2021).

[3] Liu, Yinghui, et al. "Blockchain-enabled asynchronous federated learning in edge computing." Sensors 21.10 (2021): 3335.

[4] Feng, Lei, et al. "BAFL: A Blockchain-Based Asynchronous Federated Learning Framework." IEEE Transactions on Computers 71.05 (2022): 1092-1103.

[5] Ma, Qianpiao, et al. "FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing." IEEE Journal on Selected Areas in Communications 39.12 (2021): 3654-3672.

[6] Zhang, Weishan, et al. "Blockchain-based federated learning for device failure detection in industrial IoT." IEEE Internet of Things Journal 8.7 (2020): 5926-5937.

[7] Qu, Youyang, et al. "Decentralized privacy using blockchain-enabled federated learning in fog computing." IEEE Internet of Things Journal 7.6 (2020): 5171-5183.

[8] Xu, Chenhao, et al. "Scei: A smart-contract driven edge intelligence framework for iot systems." arXiv preprint arXiv:2103.07050 (2021).

[9] Chen, Xuhui, et al. "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design." 2018 IEEE international conference on big data (big data). IEEE, 2018.

[10] Weng, Jiasi, et al. "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive." IEEE Transactions on Dependable and Secure Computing 18.5 (2019): 2438-2455.