2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE 2021)

2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE) | 978-1-6654-1540-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICBAIE52039.2021.9389820

An Adaptive Accuracy Threshold Aggregation Strategy Based on Federated Learning

1st Daoqu Geng*
School of Automation, CQUPT, Chongqing, China
gengdq@cqupt.edu.cn

2nd Hanwen He
School of Automation, CQUPT, Chongqing, China
494855668@qq.com

3rd Xingchuan Lan
School of Automation, CQUPT, Chongqing, China
1159812162@163.com

4th Chang Liu
School of Automation, CQUPT, Chongqing, China
1013648658@qq.com

***Abstract*—Fault diagnosis can be used to identify the type and severity of fault accurately and automatically. If the data is transmitted to the cloud, it will increase the risk of data security and computing. Therefore, the demand for artificial intelligence operations on edge network devices is increasing day by day. Because federated learning mechanism does not need to transmit information data, it is more suitable for edge network machine learning scenarios with limited data. Therefore, this paper proposes a bearing fault diagnosis method based on Federated learning. In the model aggregation, according to the accuracy threshold adaptive algorithm, high-quality local models are selected to participate in the aggregation, so as to reduce the number of communication. The experimental results show that when the communication times are reduced to 77.6%, the accuracy is only reduced by 2.8%.**

Keywords—federated learning, fault diagnosis, Convolutional neural network(CNN)

I. INTRODUCTION

Fault diagnosis technology can help technicians find problems in time, understand the operation status of equipment, and effectively help researchers locate problems. Common fault diagnosis algorithms are divide into data-driven [1][2], model-driven [3], and signal-driven [4]. Due to the rapid development of Internet of things devices, edge devices have certain computing power and storage capacity, which makes the combination of edge devices and machine learning is no longer just a hypothesis. Federal learning provides a valuable way for this research. Therefore, this paper also hopes to improve the intelligence of motor fault diagnosis through federated learning.

Federated learning is to train the prediction model with multiple devices in the form of cooperation. Federated learning can be built on edge devices (smart phones, video surveillance devices, etc.). Each edge node independently trains the machine learning model locally, and optimizes and merges the local model through the aggregation server. McMahan et al. [5] described federal learning in detail. Chen et al. [6] and Bonawitz et al. [7] also continued to explore the theory. In the whole federated learning process, private data does not leave the data owner, and does not need to share data with other nodes, which solves the problems of privacy and data security

The contributions of this paper are as follows.

1. The AAFedAvg algorithm is proposed. In the process of model aggregation, the algorithm uses the adaptive precision threshold to select the appropriate local model, which increases the quality of the aggregation model and reduces the number of communication.

The structure of this paper is as follows: after the introduction, the second section reviews the related work. The third section introduces our method. The fourth section introduces the test and test results. Finally, the fifth section summarizes the whole paper.

II. BACKGROUND

Federated learning is a new basic technology of artificial intelligence. Its design goal is to carry out machine learning between multiple devices on the premise of ensuring the data security of local nodes. In federated learning, local machine training algorithms and aggregation algorithms are adopted by local nodes and aggregation nodes respectively. Among them, local training algorithms include neural networks, logic regression, and other machine learning algorithms. FedAvg algorithm is an aggregation algorithm proposed by [1]. It has been applied to mobile phones for federated learning on the premise of protecting users' privacy.

The Federation mechanism consists of one aggregation node and several local nodes. The aggregation node is responsible for collecting the model parameters uploaded by each participating node, updating the local model parameters, and maintaining the global model according to the aggregation algorithm. In the whole learning process, the local node only communicates with the aggregation node to ensure the confidentiality of private data

The principles of FedAvg are detailed below.

Problem formulation. In this work, we consider the following distributed optimization model:

$$M(w)^{init} = \sum_{i=1}^{P} \frac{n_i}{N} M(w)_i \tag{1}$$

978-1-6654-1540-8/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:23:45 UTC from IEEE Xplore. Restrictions apply.

Where P is the number of devices, N represents the total number of samples of all local nodes, and n_i represents the number of samples of the i-th local node. The weight of the i-th device is composed of n_i and N. $M(w)_i$ is the local objective function of the i-th device on its local data set. Assume that the i-th device saves n_i pieces of training samples. The aggregation node calculates the weighted average of the model results according to formula (1) and sends the aggregated model parameter M^{init} to each local node.

The local objective function is defined by formula (2).

$$M(w)^{iter} = M(w)^{iter} - \eta \nabla l(w;b) \tag{2}$$

Where $l(;)$ represents the local loss function during a local training task. $M(w)^{iter}$ is the local model. η is a constant of the step size, and $\nabla l(w;b)$ is a one-step stochastic gradient of the objective function evaluated on the data set of the i-th device. The locally computed model parameter M^{iter} is sent to the aggregation node by the local node.

The following describes the specific role of pseudocode in the FedAvg algorithm.

ClientUpdate: The initial model M^{init} sent by aggregation node, is accepted by local node i and trained with local data set. M^{init} is trained by the local nodes in batches, that is, the local nodes use the local data set to train their miters in parallel. After training, the trained M_i^{iter} and n_i are sent to the aggregation node by the local node for model aggregation and convergence checking.

ServerAggregation: The core task of the aggregation node is model collection and aggregation. Firstly, all models trained by local nodes are collected, and then weighted average based on the number of local samples. Finally, the aggregated model $M^{init}$is sent to the local node.

III. METHOD

In the federal average algorithm, fixed threshold or random selection is used as the judgment condition of local training check, which has many shortcomings. Fixed thresholds overtrain local models, causing them to fluctuate and not converge easily at the end of training; however, because the degree of model change is different in different learning processes, simply randomly selecting local nodes is easy to ignore The model with a large amount of information has an impact on the global model training.

In AAFedAvg, the local node automatically adapts to the change of model accuracy in each round of the training process and calculates the appropriate threshold as the judgment condition. Only local nodes that meet the condition can communicate with the aggregation server on the appropriate rounds. Otherwise, the local node starts the next learning iteration. Finally, the accuracy of the local model reaches the threshold and is uploaded to the aggregation node. Whether the local node is qualified or not, it should be evaluated after the next round of learning. That is, accuracy assessment runs through the learning process of local nodes.

The main principle of the algorithm is that the aggregate node ignores the "lazy" local node in a certain round, and only communicates with the "hard" local node. These ignored local nodes are represented by $\{P\}_L$, while the set of nodes communicating with the server is represented by $\{P\}_H$. $ACC^{iter}{}_p$ represents the accuracy of the p-th local node in the *iter* round. The formula (3), (4) can be obtained

$$\{P\} = \{P\}_L + \{P\}_H \tag{3}$$

$$\sum_{\{P\}}^{p=0} ACC_p^{iter} = \sum_{\{P\}_L}^{p=0} ACC_p^{iter} + \sum_{\{P\}_H}^{p=0} ACC_p^{iter} \tag{4}$$

Definition of "lazy" node can be said to be the key part of precision threshold adaptation. Different definitions will not only lead to a different amount of ignored information and convergence speed of the model, but also affect the accuracy of the model. In this paper, the lazy node-set $\{P\}_L$ satisfies:

$$\frac{\sum_{\{P\}_L}^{p=0} ACC_p^{iter}}{T_L} \le \frac{\sum_{\{P\}}^{p=0} ACC_p^{iter}}{T} \tag{3}$$

In formula (3), T_L and T are the numbers of lazy nodes and the total number of all local nodes respectively. Since the number of lazy sets T_L can not be obtained in advance, we introduce the proportional coefficient β to simplify the problem.

$$T_L = \beta T \tag{4}$$

According to formula (3), the following results can be obtained.

$$\sum_{\{P\}_L}^{p=0} ACC_p^{iter} \le \beta \sum_{\{P\}}^{p=0} ACC_p^{iter} \tag{5}$$

Assuming that the accuracy of the k-th iteration is related to k-1 and k-2, based on the change of accuracy in the experiment, we define the formula of accuracy threshold as follows:

$$\frac{1}{ACC^{k-2}} - \frac{1}{ACC^{k-1}} = \frac{1}{ACC^{k-1}} - \frac{1}{ACC^{k}} \tag{6}$$

Because the accuracy of lazy nodes is lower than the accuracy threshold, so:

$$ACC_L^k \le ACC^k \tag{7}$$

$$ACC_L^{iter} \le \frac{ACC_{base}^{k-1} \times ACC_{base}^{k-2}}{2ACC_{base}^{k-2} - ACC_{base}^{k-1}} \tag{8}$$

Formula (8) is the lazy node check expression, that is, the aggregation node selects the lazy node according to formula (8) after a round of learning. If the result is true, the local node is added with $\{P\}_L$. On the contrary, the node is added with $\{P\}_H$. then, select the set $\{P\}_L$, and bring it into (5) for checking. If the result is true, the $\{P\}_H$ node enters the model aggregation, and the $\{P\}_L$ node abandons the aggregation. On the contrary, the lazy node will start the next round of learning and be degraded according to formula (9).

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:23:45 UTC from IEEE Xplore. Restrictions apply.

$$ACC_i^{iter} \le \frac{ACC^{iter-1} \times ACC^{iter-2}}{2ACC^{iter-2} - ACC^{iter-1}} * 0.8^j \quad (9)$$

In formula (9), j is the number of demotions performed by the aggregation node. The principle of the demotions strategy is to reduce the threshold value of judging lazy nodes so that local nodes have a greater chance to be selected as hard nodes to participate in aggregation.

Compared with traditional federated learning, this method is less dependent on the network bandwidth and other device configurations. Because the bandwidth affects the transmission time of the node, the lower bandwidth will lead to the increase of each round of learning time of the node, thus prolonging the overall learning time. Therefore, to control the overall learning time, it is necessary to limit the bandwidth configuration. Therefore, in theory, the requirement of this method for bandwidth configuration is smaller than that of the traditional uncompressed method, which can better adapt to the network with limited bandwidth

IV. EXPERIMENTAL SETUP

A. EXPERIMENT SETTING

At present, the algorithm of bearing fault diagnosis is being updated rapidly. Therefore, the experimental data in this paper is based on an open data set [8] from the Rolling Bearing Data Center of Casey West Storage University. The experimental object is the active end bearing shown in Fig. 5. The bearing type to be diagnosed is deep groove ball bearing skf6205. Faulty bearings are processed by EDM. The sampling frequency of the system is 12 kHz. There are three kinds of defects in bearings, namely, damaged rollers, damaged outer rings, and damaged inner rings. Damage diameters were 0.007 inches, 0.014 inches, and 0.021 inches, respectively. In addition, there are nine injury states. This paper uses overlapping sampling for feature processing.

Overlap sampling is performed on the original signal to obtain training data. The number of samples in each class can be obtained from formula(5).

$$D = \frac{N - Step}{Offset} + 1 \quad (5)$$

N is the single fault type data point. The *Step* is the signal length. The *Offset* is the data sampling interval. And, By processing the original data set, 800 pieces of data can be produced for each fault type.

To facilitate the training of the convolution neural network, each signal is normalized. Where z is the current sample data value, z_{max} is the maximum value of the sample data, and z_{min} is the minimum value of the sample data. The normalization processing is based on the formula (6):

$$\widetilde{z} = \frac{z - z_{\min}}{z_{\max} - z_{\min}} \quad (6)$$

The experiment includes 5 nodes, 4 local nodes and one aggregate node. The local node holds the data set. The original data can be divided into 10 classes, of which 1-9 corresponds to 9 fault classes and 0 is no fault class. The total number of data sets is 8000. To verify the effectiveness of the experiment, the whole bearing fault diagnosis data set is divided into 7000 original training samples and 1000 prediction samples. To ensure that the datasets of each local node are not exactly the same, but the feature space is similar. Therefore, 400 samples are randomly selected from each class in the original training samples as the training samples of each local node.

CNN is chosen as the local training algorithm in this paper. Because the CNN structure will affect the aggregation model, the quality of the local model, and the training time. Therefore, this article selects a suitable CNN structure as the training model after many times optimizations. It has two 1D-convolutional layers (the first has a 64×1 convolution kernel, the second has a 3×1 convolution kernel) and a 2×1 maximum pooling layer. Then, follow a fully connected layer with 64 units and activate ReLu. An output layer is a softmax unit. The optimizer is Adam [9].

B. EXPERIMENT RESULT AND ANALYSE

TABLE I ACCURACY UNDER DIFFERENT β

β	Accuracy of Train Set	Accuracy of Test Set	Communication times
0.1	0.9930244	0.995	680
0.2	0.99352264	0.991	651
0.3	0.99778773	0.993	632
0.4	0.9925262	0.989	610
0.5	0.9920279	0.984	588
0.6	0.99152964	0.981	560
0.7	0.98006976	0.98	536
0.8	0.978575	0.971	534
0.9	0.9795715	0.968	529

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:23:45 UTC from IEEE Xplore. Restrictions apply.

Table I shows the influence of different β values on training time and accuracy. On the whole, the accuracy of the training set is higher, which may be because each node has all the class samples, and the feature space of the data set is relatively simple, so the model is easier to learn the data features. However, with the decrease of β value, the accuracy on the test set decreases slightly. This may be because, with the decrease of β value, the number of local nodes participating in aggregation is limited. Especially in the case of $\beta \in [0.7,0.8]$, the decline of accuracy is higher than that of other intervals. However, even when $\beta = 0.1$, our algorithm still has high accuracy in the test set. This shows that the AAFedAvg algorithm can effectively select high-quality models for model aggregation while reducing the number of communication. Besides, we find that with the increase of βvalue, the number of model exchanges between the aggregation node and the local node gradually decreases, that is, the bandwidth traffic during training is lower. It may be that the value of β affects the number of lazy nodes, and a larger value of β increases the threshold value for the number of lazy nodes, thus reducing the number of aggregation nodes using the degradation strategy. Finally, it makes the diligent nodes participate in the model aggregation faster and speeds up the training speed.

V. CONCLUSION

In this paper, an efficient federated learning mechanism for edge devices AAFedAvg is proposed, which can meet the practical needs of multi-party data learning without sharing private data. The local node learns from the data set it owns locally. After local training, the aggregation node checks whether the aggregation conditions are met in this round. If so, the local model data is obtained from the local node and the model aggregation is performed. We propose an adaptive algorithm of accuracy threshold, which can adaptively change the accuracy of each round in the process of model training, and calculate the appropriate threshold to reduce the number of communication between local nodes and aggregation nodes. Because the local model data indirectly reflects the node sample information, the attacker can deduce the sample data from the effective model information, reduce the number of communication, and effectively reduce the possibility of privacy leakage. On the test set, when the gradient communication is compressed to 77.6% of the original communication times, the accuracy is only reduced by 2.8%

ACKNOWLEDGMENT

This research was supported by the National Key R&D Program of China under Grant 2018YFB1700200

REFERENCES

[1] J. Li, X. Li, and D. He, "A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction," *IEEE Access*, vol. 7, pp. 75464–75475, 2019, doi: 10.1109/ACCESS.2019.2919566

[2] H. Shi, L. Guo, S. Tan, and X. Bai, "Rolling bearing initial fault detection using long short-term memory recurrent network," *IEEE Access*, vol. 7, pp. 171559–171569, 2019, doi: 10.1109/ACCESS.2019.2954091.

[3] Y. Wu, B. Jiang, and N. Lu, "A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices," IEEE Trans. Syst. Man, Cybern. Syst., vol. 49, no. 10, pp. 2108–2118, 2019, doi: 10.1109/TSMC.2017.2757264.

[4] Y. Niu, J. Fei, Y. Li, and D. Wu, "A novel fault diagnosis method based on EMD, cyclostationary, SK and TPTSR," *J. Mech. Sci. Technol.*, vol. 34, no. 5, pp. 1925–1935, 2020, doi: 10.1007/s12206-020-0414-y.

[5] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017*, vol. 54, 2017.

[6] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation," *IEEE Trans. Neural Networks Learn. Syst.*, no. Iid, pp. 1–10, 2019, doi: 10.1109/tnnls.2019.2953131..

[7] K. Bonawitz, F. Salehi, J. Konecny, B. McMahan, and M. Gruteser, "Federated Learning with Autotuned Communication-Efficient Secure Aggregation," Conf. Rec. - Asilomar Conf. Signals, Syst. Comput., vol. 2019–Novem, pp. 1222–1226, 2019, doi: 10.1109/IEEECONF44664.2019.9049066.

[8] http://csegroups.case.edu/bearingdatacenter/home.

[9] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:23:45 UTC from IEEE Xplore. Restrictions apply.