Research on Federated Learning Algorithms in Non-Independent Identically Distributed Scenarios

Jikun Wu[1,*], JiaHao Yu[2] and YuJun Zheng[3]

[1] Chengdu University of Technology, Computer and Network Security, 610059, Chengdu, SiChuan, China

[2] Zhejiang University of Media and Communication, School of media Engineering, 310018, China

[3]Beijjng Union University, College of Urban Rail Transit and Logistics, 100101, China

* Corresponding Author Email: 3212945384@qq.com

Abstract. Federal learning is distributed learning and is mainly training locally by using multiple distributed devices. After receiving a local parameter, a server performs aggregation and performs multiple iterations until convergence to a final stable model. However, in actual application, due to different preferences of clients and differences in local data of different clients, data in federal learning may be not independently and identically distributed. (Non-Independent Identically Distribution). The main research work of this article is as follows: 1)Analyze and summarize the methods and techniques for solving the non-IID data problem in past experiments.2) Perform in-depth research on the basic methods of federal learning on non-IID data, such as FedAvg and FedProx.3) By using the FedAvg algorithm, using the CIFAR-10 data set, the simulation method is used to simulate the number of types contained in each client, and the distribution of the data set divided according to the distribution of Dirichlet to simulate the non-independent identical distribution of data. The detailed data analysis is made on the influence of the data on the accuracy and loss of model training.

Keywords: Federal Learning, Non Independently Distributed, FedAvg, Model Training.

1. Introduction

As a large quantity of intelligent devices are widely used, user data is continuously expanded. Therefore, it becomes a hot technology to process massive data by using artificial intelligence technology to mine data value. On the premise of ensuring the security and privacy of user data, how to mine the value of data in massive data also becomes a challenging task. Federal learning is one of the methods that can solve these problems. The Federal learning method may be used to combine a client-side and a server-side, and jointly train a global model without allowing the client side to upload local data. This ensures security and privacy of user data. As a result, federal learning has become a hot topic of research.

Federal learning is a mode of distributed learning in which local training is performed by using multiple distributed devices, aggregation is performed after a server receives a local parameter, and multiple iterations are performed until convergence is performed to a finally stable model. The core is to choose a certain amount of clients to use local data for model training, and then upload encrypted model parameters for model aggregation to update the global model. However, in actual application, due to different preferences of clients and differences in local data of different clients, data in federal learning may be not independently and identically distributed. (Non-Independent Identically Distribution). Non-independent data synchronization causes a parameter difference between a local model and a parameter difference between a local model and a global model. Consequently, 1) a data deviation is caused, and finally, the model may overpay for the data of some parties during training. Consequently, the performance of predicting data of other parties decreases. 2) It is difficult to update a model collaboratively. In federal learning, a client performs model training by sharing a model parameter and improves model performance by performing coordinated updates. However, if data is not independently distributed, the parameter update of a model is limited, thereby reducing a model collaboration training effect. 3) In federal learning, models must be trained on data supplied by several

parties in order to model global data because model generalization performance is compromised. However, the model might not be able to properly capture a feature of the global data during training if the difference in data distribution between participants is particularly substantial, meaning that data is not independently distributed in the same way. As a result, the model's generalization performance suffers. Federal learning tasks perform less well than anticipated as a result of non-iid data's impact on the global model's forecast accuracy and even its inability to converge.

Because the data partition strategy of the past research among all parties is very fixed, that is, according to the difference between the number of user data and the distribution of data, the model aggregates the dynamic adjustment of the parameter weight, which lacks the experimental research of systematically understanding its advantages and disadvantages. In this paper, a comprehensive data partitioning strategy is proposed to cover the typical non-independent and co-distributed data cases, and extensive experiments are conducted to evaluate the most advanced algorithms.

In this paper, the methods and techniques for non-iid data are introduced, and the advantages and disadvantages of different methods are discussed. Then, a basic manner of understanding the non-iid data is specifically described, for example, FedAvg and FedProx. An improvement manner for the non-iid data is discussed according to a definition and a generation manner of the non-iid data. We then demonstrated our experimental setup, specifically illustrating the experimental learning algorithm and parameter setup using federal learning, and then evaluated the metrics. Demonstrate and analyze experimental results on non-iid data sets, compare the performance of different methods, and explore the benefits and applicability of different methods. Finally, this paper highlights how federal learning improves today's federal learning algorithms by addressing the potential and limitations of non-iid data sets and looking to the future.

2. Related Research

In previous research, researchers proposed algorithms to address the fact that data is not independently and identically distributed.

1) A FedAvg algorithm in which each client locally uses its local data to make further gradients of the current model, and the server then weighted averages the resulting model. However, a disadvantage of the algorithm is also obvious. The brute-force aggregation of a model is not a linear summation. A model parameter of each user applies only to the data of the user, and a data distribution and a quality difference between different user data are relatively large.

2) Compared with the FedAvg algorithm, the FedProx algorithm adds the proximal term to ensure that the model parameters of the client after the local training will not be much different from the initial server parameters, which reduces the influence of Non-iid on the premise of tolerating the isomorphism of the system. By inaccurate solving of the local model, the local iteration times are dynamically adjusted to ensure tolerance to the heterogeneous system.

3) Personalization. Because of the high Non-iid of the user data and the inconsistency of the performance requirements of the user on the model, it is difficult for a single global model to guarantee the requirements of all clients. Therefore, a personalized method is needed to optimize the global model for each user. Current technology approach. 1) The Multi-task Learning model learns by leveraging commonalities and differences between tasks. The generalization performance of the original task can be improved by sharing the representation of the associated task. 2) The Meta-Learning method is a Learn to Learn learning method, which makes the model results universal to various tasks. 3) Model-Agnostic Meta-Learning. The original random initialization strategy is taken as the learning optimization target. The tasks suitable for this task are obtained by sampling a series of tasks, and the backward propagation optimization is lost in the calculation.4) The FedDCS algorithm quantifies the data heterogeneity degree of the client through the cosine similarity between the client model and the global model and selects the more potential client model by using the similarity information and the client model performance index. Based on the optimization of the aggregation policy of the FedAvg algorithm, a dynamic weighted aggregation policy is designed

according to the similarity information to reduce the influence of the Non-iid data and improve the accuracy and convergence efficiency of the global model.

In the existing research, algorithms are proposed from different perspectives to optimize a case in which Non-iid data appears in federal learning. However, most of the research work is to treat all devices equally, and differences in data feature distribution of the devices are not considered. This brings an extra round of communication to devices with limited resources and degrades model performance. Existing challenges 1) Heterogeneous system, the difference in storage, computing, and communication performance of different devices. 2) Heterogeneous statistics, inconsistency in data distribution. 3) Heterogeneous models, inconsistent models for different environments and tasks.

3. Method

3.1. FedAvg

Using local input from each client and a step gradient of the current model, Federated Averaging produces a model that is weighted by the server. By updating locally many times before the average step once the algorithm has been built in this fashion, we can add more calculations to each client. So that the user can perform multiple gradient downgrades before uploading an update and upload a trained model parameter instead of a gradient to the server. This approach significantly improves communication efficiency and is considered a typical representation in federal learning algorithms. The Google team was the first to propose and introduce the basic algorithm FedAvg for federal learning. Its general form is: $\min_{\omega \in R^d} f(\omega)$where $f(\omega) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(\omega)$, In mechanical learning, it is usually assumed that$f_i(\omega) = l(x_i, y_i; \omega)$,The loss of the i th sample, $\min_{\omega \in R^d} f(\omega)$means minimize the average loss of all samples.Suppose there are K clients partitioning the data, where is the index set of the dataset on client K, $n_k = |P_k|$.Therefore, the general form above can be rewritten as $f(\omega) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\omega)$where $F_k(\omega) = \frac{1}{n_k}\sum_{i \epsilon P_k} f_i(\omega), F_k(\omega)$is the average loss of all data within a client, $f(\omega)$ is the weighted average loss of all clients under the current parameter. If all P_kare by distributing the training samples evenly across the client, so every $F_k(\omega)$ expectation is for $f(\omega)$.This is usually an IID assumption made by distributed optimization computing, where each client's data is distributed independently of the other [1].

3.2. FedProx

Federal learning is a type of distributed learning in which local training is performed by using multiple distributed devices, aggregation is performed after a server receives a local parameter, and multiple iterations are performed until convergence is performed to a finally stable model. Algorithms such as FedAvg and FedProx are introduced into federal learning. However, the FedAvg algorithm has no good solution to data heterogeneity and device heterogeneity (non-iid). Therefore, the FedProx algorithm is obtained by improving the FedAvg algorithm. The FedProx algorithm can be regarded as a generalization of the FedAvg algorithm. The former allows to use a more real joint data set and is more convergent than the latter. Therefore, this method is used to solve the non-iid. FedProx algorithm is a simple but powerful distributed endpoint optimization method. It is mainly used in the federally learned data processing of heterogeneous data to process and solve the heterogeneity in federated networks. The convergence behavior of the algorithm is characterized by some non-standard and impractical hypotheses of differences and similarities of local functions [2].

Li and others proposed a new federal learning framework FedProx: If γ is an imprecise solution, $h(\omega; w_0) = F(\omega) + \frac{\mu}{2}\|\omega - \omega_0\|^2$,if $\gamma \in [0, 1]$, it means that ω^* is a γ - imprecise solution of $min_\omega h(\omega; \omega_0)$. If $\|\nabla h(\omega^*; \omega_0)\| \leq \gamma\|\nabla h(\omega_0; \omega)\|$, where $\nabla h(\omega; w_0) = \nabla F(\omega) + \mu(\omega - \omega_0)$, it says that γ is smaller, ω^* is more accurate. That is, a smaller gradient is more accurate. Because the bigger the gradient, it takes more time to converge [3]. If data between devices is heterogeneous,

a model obtained after each client is optimized differs greatly from an initial global model allocated by the server. Therefore, a local model deviates from the initial global model, and the convergence of the global model is reduced. To limit this bias, Li and others introduced a proximal term (proximal term). After a near-end item is introduced, model parameters obtained by the client after local training do not differ from initial server parameters by much [3].

3.3. Definition and Generation of Non-Independent Same Distribution Data

In conventional machine learning, data is distributed on the same machine, and it is assumed that the data is independently sampled from the same distribution, that is, data is independently distributed (iid). Subsequently, with a large quantity of Internet of Things devices being used, tens of thousands of devices perform model training of federal learning. In this case, the distribution of various data is usually greatly different and is referred to as data non-independent distribution (non-iid). Non-iid: The use of the local device greatly influences the level of training for each client in federal learning, thus the distribution of data for connected clients may differ from one another, leading to significant model divergence, especially for parametric models [4]. The accuracy of federal learning has substantial obstacles as a result of non-iid data settings, which can diminish the effectiveness of machine learning services despite the fact that many algorithms for federal learning can help solve the learning impacts of non-iid data settings. 1) Data is generated and collected in various ways, and data from many customers is easily heterogeneous. As a result, data becomes more difficult to handle, and modeling and assessment in federal learning become more complex and time-consuming [5]. 2) There is significant heterogeneity in the data collected from many clients, and when data is heterogeneous, "client-drift"—the introduction of drift in each client update—has an impact on federal learning and causes instability and slow convergence [6]. 3) Local models with the same initial parameters converge to different models, non-iid creates the heterogeneity of local data distribution, and the learning performance of federal learning deteriorates. 4) As convergence slows and learning performance declines, global model variations in federal learning continue to grow.

3.4. Improved Methods

Data Sharing. Data sharing is a way to effectively process non-iid data in horizontal federal learning. Store global datasets with even distribution on the server, making it easy for client models to get data from local training and global shares [4]. To improve data with non-iid, a tiny piece of data that is globally shared across all devices can be created, with the size and random distribution score of the global data-sharing set being changed for questions and practical applications.Experiments have shown that a willingness to centralize and distribute 5% of the shared data can improve accuracy by 30%. Data sharing regulations advance Since data sharing only occurs once when federal learning is initiated and FedAvg for non-iid data simply introduces a trade-off between accuracy and centralization, communication costs are not a major concern. Since the worldwide shared dataset is unrelated to client information, privacy concerns are irrelevant [7].

Federated Ensemble Distillation with Imbalance Calibration. Unbalanced Calibration Joint Distillation is a server-side method that uses a set of models to leverage the diversity of models trained on the non-iid data set. First, the bias set model is calibrated by using a calibration gating network and logic adjustment of a client and a class, to counter a new distilled method of long-tail distribution (researchers propose various methods to solve non-iid data, but a precondition is that the general-class distribution is balanced, and in a real application program, the general-class distribution is long-tail, which causes serious model deviation). The calibrated collection is then used as an instructor model to further optimize the client by shifting knowledge to a global model [8].

FedNS. FedNS(Federated Node Selection), to set a possibly better model by combining the best client components, this strategy, which is a new approach offered in aggregating the server global model in a FL configuration, filters and reweights the client model at the server node/kernel level. Concentrate on how modules aggregate on the server while keeping the federal learning paradigm's straightforwardness. Without a predefined hyperparameter, this algorithm can handle iid or non-iid

distribution. FedNS is a new model aggregation technique for FLs that doesn't significantly alter the FL architecture and can be used for applications like mobile age classification.It is more applicable to real-world situations because, unlike classic FL settings, each client has a defined percentage of data, allowing for flexibility and different data availability [9].

4. Experiment

4.1. Dataset Introduction

A common image data set CIFAR-10 is used to verify the proposed method. The CIFAR-10 data set is an image data set that contains 60000 RGB color images, and the data set is often used as a baseline experiment for image classification or recognition tasks. 50000 images in the CIFAR-10 data set are training sets, and the other 10000 images are test sets. Each image in the CIFAR-10 data set is 32*32 pixels in size, and each image has three channels of RGBs: Airplanes, automobiles, birds, cats, dogs, deer, frogs, horses, ships, and trains.

4.2. Data Partition Mode

Many studies of federally learned data non-IID distributions generate non-IID experimental data sets by dividing the data set into distinct distributions. Mc Mahan made an experiment of the Fed Avg algorithm under the non-IID data set, the non-IID experiment data set is generated by simply dividing the non-identically distributed user data of the MNIST data set into blocks [10]. The experiment data on each worker is divided into at most two sample data of a label type by using the block method, to generate the Non-IID experiment data set whose label data distribution is oblique. There are also data set generation methods under study that take into account real-world scenarios. The Non-IID partition of the EMNIST data set by Caldas is to partition the data by alphabetic writer, not simply by alphabetic label category [11].

In this paper, the data set is divided mainly by using a data label category (class) and by using a Dirichlet process simulation non-IID data (dir.) method.

By using a data label category (class), because a common parameter server used in federal learning exists below a plurality of parties client. A worker in each client has its own data set, and a class of the data set can adjust a quantity by using a parameter. In this experiment, the CIFAR-10 data set used in the data set is classified into ten data sets. Therefore, the quantity of classes for each client is adjusted from one to ten. When the data set is adjusted to one, the data set becomes severely unbalanced. On the contrary, when the data set is adjusted to ten, even if the data set is not independently distributed in the same manner, the data set is simply distributed in the same manner. When we adopt partition mode split type: class, we simulate the effect of the experiment on the experiment result by changing the number parameter of categories contained in each client.

The data set is divided by using a method of simulating non-IID data (dir.) by using a Dirichlet process. Dirichlet distribution is an important multi-dimensional continuous distribution in probability statistics and is usually denoted as Dir, and the distribution is controlled by a parameter of a positive real number vector a. It generates a non-identically distributed data set by a continuous Dirichlet distribution controlled by parameters to produce a Non-IID data set in different cases [12-13].In the federal learning experiment, the Non-IID data set can be obtained by sampling the data set of IIDs according to the distribution of Dirichlets. It is assumed that the source data set is of the class M label category, and the training sample data on each worker is obtained by sampling the probability of the source data set with the probability as the parameter vector: q ($q_i \geq 0, i \in [1, M]$).To generate worker data samples whose data is not identically distributed, sampling may be performed by using a probability of a distribution of Dirichlets corresponding to the formula 4-1, where a probability vector p represents a prior distribution of the class M label category and a parameter alpha controls the consistency of generated worker sample data

$$q \sim Dir(\alpha p) \tag{4-1}$$

(4-1) Probability distribution q Sample data distribution varies with the value of a parameter, alpha [10]. When $\alpha \to \infty$, the resulting distribution of data on the worker is consistent with the prior distribution of the original data set. On the other hand, in the case of $\alpha \to 0$, the sample data for each worker is randomly sampled from only one class of samples. When we adopt partition mode split type: dir., we simulate the effect of alpha on the experiment result by changing its parameters.

4.3. Algorithm and Parameter Settings

As a comparison experiment, this paper implements the FedAvg algorithm and compares its performance under different data distributions. In the Fed Avg algorithm, first, the federated learning server broadcasts the model parameters to all workers. Then, each worker updates the local model parameter to the received model parameter, executes the gradient descending algorithm based on the received model parameter, and optimizes the model parameter by using the worker's local data. Finally, the federal learning server aggregates the local model parameters of each worker to optimize the global model parameters. When other conditions are kept the same, in this experiment, a partition manner split type: class is first used, and a quantity parameter 1,5,10 of categories and types included in each client is changed to control a training round fifty times. Precision Test Accuracy Comparison of a model training result is compared and recorded, and Test Loss Comparison comparison is lost. Then, when other conditions are the same, a division manner split type: dir is used. Parameters 0.2, 0.5, and 0.8 of the alpha are changed to control the training round fifty times, so as to compare and record the accuracy of a model training result and a loss comparison.

4.4. Data Label Category (class)

By using a data label category (class), because a common parameter server used in federal learning exists below a plurality of parties client. A worker in each client has its own data set, and a class of the data set can adjust a quantity by using a parameter. In this experiment, the CIFAR-10 data set used in the data set is classified into ten data sets. Therefore, the quantity of classes in each client is adjusted from one to ten. When the data set is adjusted to one, the data set becomes severely unbalanced. On the contrary, when the data set is adjusted to ten, even if the data set is not independently distributed in the same manner, the data set is simply distributed in the same manner.

The Fed Avg algorithm performance for the CIFAR-10 dataset is shown in Figure 1 and Figure 2. In Figure 1, as the quantity of categories included in each client is gradually increased from one to five and then to ten in this experiment, accuracy Test Accuracy Comparison of the model is gradually increased from about ten percent to about seventy percent. When the quantity of categories included in each client is one, accuracy is always maintained at ten percent due to extremely imbalanced data sets. However, when the quantity of categories included in each client is ten, accuracy is always increased with training. In Figure 2, the loss Test Loss Comparison gradually decreases from about 0.2 to 0.05, and a loss of 0.05 is always kept relatively low when the quantity of categories included in each client is ten.

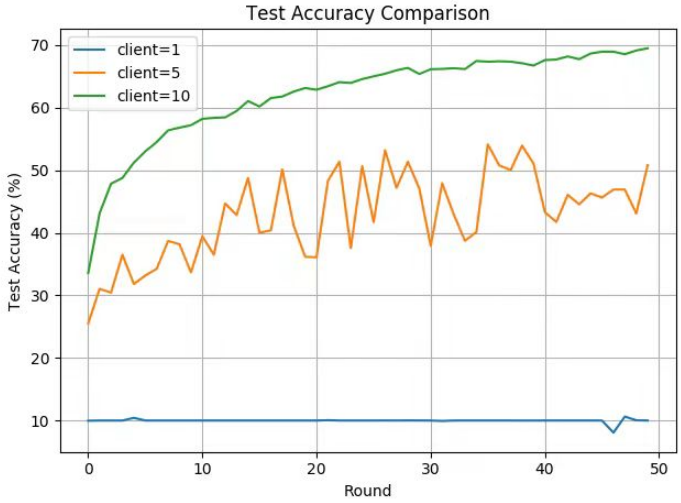


Fig. 1. Effect of inter-class imbalance on accuracy(Photo/Picture credit :Original)

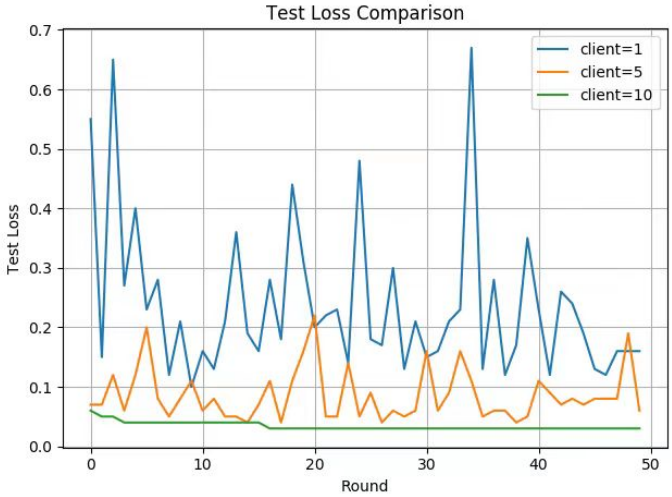


Fig. 2. Effect of inter-class imbalance on loss(Photo/Picture credit :Original)

4.5. Simulate Non-IID Data with Dirichlet Process(dir.)

The distribution of the data set divided according to the Dirichlet distribution is controlled by the parameter α of the Dirichlet. The larger the alpha is, the closer the probability distribution is to the uniform distribution, and the data set generated by sampling is to the same distribution. The smaller alpha is, the more likely the resulting probability distribution is to focus on some point, the more skewed the resulting data set is, and the more approximate the resulting data set is to the non-IID data set.

Figures 3 and 4 display the Fed Avg algorithm's performance for the CIFAR-10 dataset. In Figure 3, as alpha gradually increases from 0.2 to 0.5 and then to 0.8, the final accuracy Test Accuracy Comparison of the model is gradually increased from about 45% to about 60%. In Figure 4, a loss Test Loss Comparison gradually decreases from about 0.08 to 0.04, a smaller oscillation occurs in a model training process, and the convergence speed of the algorithm is also higher.

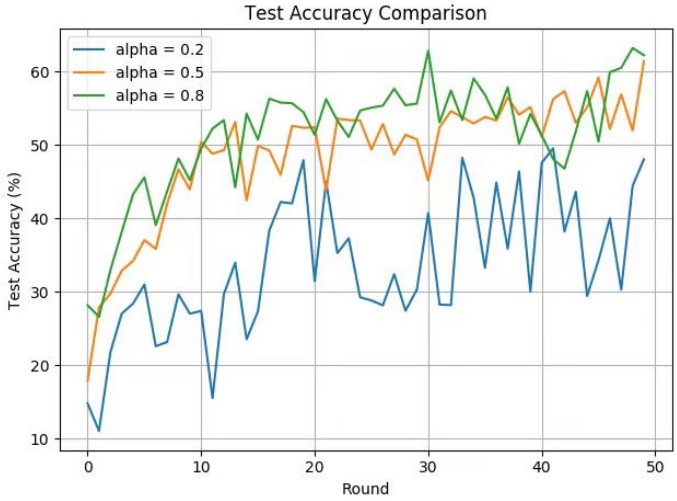


Fig. 3. Non-IID impact of the dataset on accuracy(Photo/Picture credit :Original)

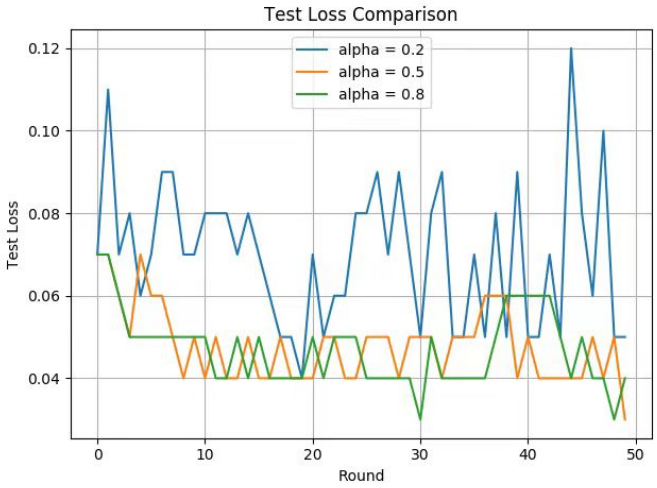


Fig. 4. Non-IID impact of datasets on losses(Photo/Picture credit :Original)

5. Conclusion and Prospect

In distributed machine learning, where data privacy is crucial, federal learning is essential. However, the quality of model training suffers if each edge device sees a different distribution of data. In our experiments, a neural network trained on non-iid data under different tilt degrees is simulated by simulating the effect of the experiment on the experimental results by changing the number of parameters of categories and types contained in each client and by generating non-identically distributed data sets by using continuous Dirichlets distribution controlled by parameters. According to the experiment, when the number of classes in the client is adjusted to one, the data set is severely out of balance, and the accuracy is maintained at ten percent, while when the α is 0.2, the accuracy is also minimized, and the accuracy of federal learning is significantly reduced. We optimize the model parameters with worker local data, and finally, the federated learning server aggregates the local model parameters of each worker to optimize the global model parameters.

In federal learning, each device trains the model and sends parameters to the server side for aggregation. Data is saved locally on the device only, improving performance with a global model. Federal learning can ensure that data does not leave the local, and at the same time make full use of the computing power of the terminal equipment, which reduces the computing cost and data storage on the server side, but it also introduces new problems. Because a mechanical learning model is trained in a case in which local data is not exchanged in federal learning, heterogeneous distribution of data of all parties may occur, which causes impact such as a data deviation and a model

collaborative update difficulty and reduces the accuracy of training the federal learning model. A major problem of the heterogeneity in federal learning is heterogeneity between many client devices, including differences in a plurality of aspects such as storage, CPU computing capability, and network transmission. The heterogeneity causes different computing times of devices and even causes a case in which some devices are directly dropped. In addition, data in federal learning is also heterogeneous to some extent. Therefore, in future research, how to ensure data privacy and model accuracy while reducing the negative impact on federal learning is a major challenge.

References

[1] McMahan, H. B., Moore, E., Ramage, D., Hampson, S.,and Arcas, B. A. y. Communication-efficient learning of deep networks from decentralized data. In International Conference on Artificial Intelligence and Statistics, (2017).

[2] Yuan, X., and Li, P. On Convergence of FedProx: Local Dissimilarity Invariant Bounds, Non smoothness and Beyond. NeurIPS, 1-40(2022).

[3] Li, T., Sahu, A, Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated Optimization in Heterogeneous Networks. Computer Science, 1-22(2020).

[4] Zhu, H., Xu, J., Liu, S., and Jin, Y. Federated Learning on Non-IID Data: A Survey. Computer Science 465(20), 371-390(2021).

[5] Zhang, K., Song, X., Zhang, C., and Yu, S. Challenges and future directions of secure federated learning: a survey. Frontiers of Computer Science16(5), 165817(2022).

[6] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T., SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In: Daumé, H., and Aarti, S. PMLR vol. 119, pp. 5132-5143. Proceedings of Machine Learning Research (2020).

[7] Zhao, Y., Li, M. Lai, L., Suda, N., Civin, D., and Chandra, V. Federated Learning with Non-IID Data. AriXiv, pp.1-12(2018).

[8] Shang, X., Lu, Y., Cheung, Y., Wang, H. Fedic: Federated Learning on Non-iid and Long-Tailed Data Via Calibrated Distillation. 2022 IEEE International Conference on Multimedia and Expo (ICME), pp.1-6(2022).

[9] Zhuo, Y., and Li, B., Fedns: Improving Federated Learning for Collaborative Image Classification on Mobile Clients 2021 IEEE International Conference on Multimedia and Expo (ICME), pp.1-6(2021).

[10] B. Mc Mahan, E. Moore, D. Ramage, et al: Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intel-ligence and Statistics (AISTATS), Fort Lauderdale, FL,USA,1273-1282(2017).

[11] S. Caldas, S. M. K. Duddu, P. Wu, et al: Leaf: A benchmark for federated settings. ar Xiv preprint ar Xiv:1812.01097(2018).

[12] LI Jian: Research on federated learning algorithm under non-independent homogeneous distribution data. University of Electronic Science and Technology of China (2021).

[13] M. Yurochkin, M. Agarwal, S. Ghosh, et al: Bayesian nonparametric federated learning of neural networks. Proceedings of the 36th International Conference on Machine Learning (ICML),Long Beach, CA, USA, 7252-7261(2019).