Image Processing, Electronics and Computers
L. Trajkovic et al. (Eds.)

doi:10.3233/ATDE240498

Optimizing Computing Efficiency and Performance Fairness in Federated Learning

Di LIAO[1]
School of Computer Science and Technology,WuHan University of Science and Technology, WuHan 430065, China

Abstract. Federated learning allows participants to complete joint training without having to share data directly, which is effective in solving the data silo problem and also protects clients' privacy. However, in practice, training time is limited and the variation in data quality per device can be significant, which will cause a significant impact on the performance of the global model. There are now a number of studies optimized for fairness and efficiency. However, these current methods fail to take into account data quality while considering fairness and efficiency. In this paper, a new aggregation method and local data point selection method are proposed. Experiments show that in both the FMNIST and MNIST datasets, in the ideal case our method performs closest to the baseline method FedAvg. Compared to fairness baselines like FedFa and q-Fl, the fairness of our method is in no way diminished. And our local data point selection algorithm makes the training of the same framework significantly faster in the first 15 rounds.

Keywords. Federated learning, fairness, data quality, efficiency

1. Introduction

The advent of Federated learning allows for privacy while training users' data. Federated learning was initially proposed in [1]. Federated learning also belongs to one of the distributed machine learning, which solves the problem without uploading all the client's data. Only the trained model is uploaded to the server, and then the weighted average is done on the server to achieve an outstanding training result. This allocates the tasks of cloud computing to the edge devices, thus improving the utilization of resources and reducing the considerable training overhead and computing overhead above the cloud service. However, this framework still has some problems. The limited training time can affect the fairness of FL as well as the efficiency of model training..

Nowadays, there are also some researches focusing on the problem of non-independent homogeneous distribution of client datasets, e.g.[2-4], and the k-asynchronous method proposed in [5], which can solve the non-independent homogeneous distribution problem efficiently. While these methods offer a great improvement in efficiency, they also have a certain loss of accuracy.

However, the following challenges make the optimization of aggregation weights unsatisfactory anyhow.

[1] Corresponding Author: Di LIAO, 373942368@qq.com

Data quality. Due to the large number of customers, there will be different quality of data, without distinguishing the blind aggregation, may lead to a large deviation from the global model.

Performance fairness. Because it is locally generated data from different clients, the high quality of the data cannot be guaranteed. For example, there may be a large amount of redundant or incomplete data in some user data. If these low-quality data are directly used for training, the training effect will likely be unsatisfactory. In the traditional federated learning framework, it is directly weighted average with the amount of data for model aggregation. However, this is highly unfair to those customers with high data quality but not a lot of data volume, while clients with high data volume and low quality are given high weights, which also affects the performance of the global model.Improving performance equity while maintaining data quality is a big challenge.

Limitations on training time. In general, one thing that makes federated learning utterly different from traditional distributed machine learning is that the clients are quite distributed, and the performance of each client's equipment is different, which leads to different training speeds. In synchronized federated learning, it is necessary to wait for the customer with the longest training time to complete the training, which also leads to the problem of slower convergence and very long training time. How to get through training without losing efficiency with a tight training schedule is a difficult problem to solve.

From the above analysis, we see the necessity to use the quality of the data to determine the aggregation weights and the limited training time of the device, which motivates us to propose a new method to optimize the fairness of the aggregation performance and the training efficiency by using data quality assessment. In this paper, we propose a new aggregation method that can improve the data quality problem when aggregating and design a client-side data filtering method to prioritize the training of high-quality data to improve the training efficiency; our main contributions are as follows:

We improve the federated learning framework by more accurately measuring the weight of user aggregation. We add both data quantity and data quality into the weights of aggregation, replacing the original method of aggregation by data volume to make aggregation more accurate. However, a data complexity criterion for aggregation, applying to the CG scores, ensures that the users with high data quality have a greater aggregation weight and thus improve the performance of the global model to a specific extent performance.

We design a client-side data filtering method, which applies the data evaluation method to rank the data points according to their complexity before training and prioritizes the data with higher quality in case of tight training time, which can save time to a certain extent and achieve faster aggregation results.

We experiment our method on a real dataset and a synthetic dataset and compare it with some fairness-optimized traditional methods. The experimental results show that our method does not lose to existing methods in model accuracy and guarantees some performance fairness. It also improves in training efficiency.

The rest of the paper is organized as follows. Section II reviews previous work on data quality and aggregation, and fairness. In Section III, we introduce the problems in federated learning, the system model and the algorithmic ideas we propose. In Section IV, experiments will be conducted to demonstrate the effectiveness of our approach. Section V is a summary and overview of the whole work.

2. Related Works

2.1. Data Quality and Aggregation

In federated learning, the first algorithm that appeared is FedAvg, proposed in [1]. The gradient updated by the selected clients is taken as the mean value, and this mean value becomes the model parameter updated by the server; such a way of operation is very efficient and straightforward in general scenarios, and the global model can reach an optimal value, but for individual users, due to the part of the client loss function is large, the highest training accuracy cannot be achieved, and thus fairness cannot be guaranteed. Li et al. [6] proposed a new modeling framework in which the model of a single client can achieve optimal accuracy, that is, the variance of the user's local model accuracy deviating from the global accuracy is minimized, which can be considered fair. Huang et al. [7] proposed a new modeling framework based on training accuracy and participation frequency. Gao et al. [8] proposed a fair incentive mechanism based on blockchain, which evaluates the revenue based on the customer's contribution to the overall model; with the addition of blockchain, it has a more robust reliability. These approaches, while doing quite well in terms of their respective fairness, neglect the issue of data quality at the time of aggregation, which is taken into account in our work, and aggregation can be guaranteed to be done by data quality and quantity at the time of aggregation.

2.2. Performance Fairnesss

The performance fairness of federated learning models was initially proposed by Li et al. [6], which is measured by the model performance variance among clients, and the smaller performance variance means the fairer, later there are also many related works mentioning this point and improving it, such as [4][7][9], these works have a big improvement in the improvement of performance fairness, but they don't take into account the effect of data quality. Whereas, the effect of data quality is considered in our approach, which also indirectly improves the performance fairness to a certain extent.

2.3. Optimization of Training Efficiency

In federal learning, there are two channels to improve efficiency. One is by reducing the consumption of communication, and the other is by improving the speed of model training, but need to lose as little as possible the accuracy of the model; such a scheme will need to rely on several fast-converging algorithms. Gradient optimization algorithms like SGD have been improved in different ways such as [8,10]. These optimization methods can be applied to federated learning to improve efficiency and save communication resources. Liu et al. [11] proposed the method of momentum gradient optimization to shorten the training time to improve efficiency. [12] proposed an adaptive federated learning framework. Karimireddy et al. [13] proposed a gradient compression method with error feedback to further improve communication efficiency. Zhou et al. [14] proposed to improve the model efficiency through the staleness of the gradient, Yang et al. [15] proposed an efficient adaptive compression algorithm. Liu et al. [16] proposed an efficient method with active sampling capabilities. There are also methods that improve both robustness and efficiency, such as [17-19]. However, all of the above methods have improved in efficiency, but have some limitations: the convergence is slightly worse in some exceptional cases, and the gradient is sometimes unstable, especially the gradient with stochasticity. If the training time is very tight, the training accuracy obtained may not reach the ideal effect. Our approach differs from the above in that it utilizes data quality assessment to predict high-quality data in advance, which is effective and stable in improving local training efficiency even when training time is very limited.

3. System Model and Method

In traditional Federated learning, the amount of data is the determinant of the aggregation weights, and model aggregation takes place server-side. Ideally, this would produce excellent training results. However, the reality is not so ideal; in general, the quality of data can vary greatly from client to client. Undifferentiated aggregation by data volume may lead to considerable deviations from the global model, while at the same time the local training time should be limited and cannot be kept in a waiting state. Therefore, not only the data quality issue but also the local training time needs to be taken into account when aggregating models.

The flowchart of federal learning with the addition of data evaluation methods can be shown in Figure 1

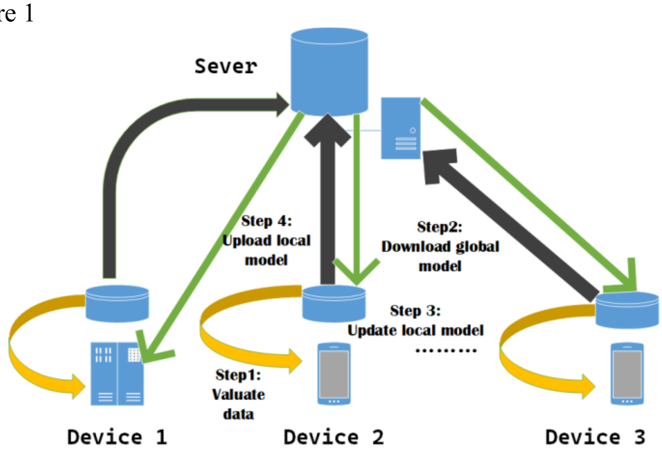


Figure 1. Federated learning framework based on data assessment

3.1. System Model

In the classical federated learning setting, there are K clients, where each client k has a dataset D_k of size n_k and after the client's current round of training is completed, the client sends the model parameters to the server, which causes the server to iterate on the model update, such that the loss of the objective is minimized [1]:

$$Minimize\text{: } F(x) = \sum_{k=1}^{K} p_k f_k(x) = \sum_{k=1}^{K} \frac{n_k}{\sum_{i=1}^{K} n_i} f_k(x), \tag{1}$$

The standard algorithm for optimizing the above objective is FedAvg, where p_k denotes the probability of a client being selected, and the server takes m client devices for training in each round, updating the global model using a weighted average of the numbers. This algorithm achieves fast and stable convergence in most ideal cases. However, since only the quantity of data is taken into account in the aggregation and not the quality of data, some users with less quantity of data but higher quality of data are not given a fair weight in the aggregation, which makes there is a great deficiency in the performance fairness. Moreover, in terms of local computational efficiency, if a large number of redundant data are trained, a lot of time and energy are wasted. Therefore, to improve the performance fairness and local training efficiency of the model, the optimization objective becomes:

$$Minimize\text{: F(x)} = \sum_{k=1}^{K} p_k f_k\,(x,t),$$
$$Subject\ to\text{: t} \leq T \tag{2}$$

Here T is the local time threshold of the user's device, in the limited time, to train quickly, so need to prioritize the training of high-quality data.

Under this condition, minimizing the loss translates into maximizing the data quality of the selected data, as demonstrated below:

Proof 1. Here to minimize the global loss function, it is necessary to minimize each local loss function, i.e., minimize all the f_k(x,t), Since time is finite, under the same time constraints, the magnitude of the decline of f_k(x,t) in each round is mainly affected by the quality of its data. So the problem can be transformed into maximizing the selected high quality user data, which can be expressed in the following form:

$$Maximize\text{: Val}(S_t) = \sum_{i=1}^{n_k} Val(p_k^i D_k^i), \tag{3}$$

Where S_t denotes the batch of data randomly sampled by the client for each round of training, and p_k^i is the probability that each client's data point is selected.

While improving the local efficiency, changing the aggregation weights so that both data quality and quantity are used as weights to improve the performance fairness, the total score of the user data can be expressed as the product of the quality of the user's data and the data, which can be expressed by the mathematical expression as:

$$Val_k = \sum_{i=1}^{n_k} Val^i, \tag{4}$$

Where, Val_kdenotes the total data score of the user k, while n_k denotes the amount of data for that user's k, and Val^i denotes the data score of the user's i-th data point.Thus, $p_k$in Eq. 2 can be rewritten as:

$$\text{F(x)} = \frac{Val_k}{\sum_{k=1}^{K} Val_k} f_k(x,t), \tag{5}$$

Definition 1. (Training Time Threshold). Due to the different performance of each user's device, the size of the data volume is also different, resulting in the training time having an enormous difference. Therefore, when the situation is more urgent, according to the actual situation, you can set a threshold for the training time to reach the time to upload the parameters to participate in the model aggregation.

Definition 2. (User Local Data Sampling Probability). The user data sampling probability is determined by the ratings of the data points. The higher the rating, the higher the probability of winning, which can be expressed mathematically in the following form:

$$p_k^i = \frac{Val_k^i}{\sum_{i=1}^{n_k} Val_k^i}, \tag{6}$$

Where p_k^i denotes the probability that the ith data point in client k's dataset will be extracted, and Val_k^i denotes the ith data point score for client k.

Definition 3. (Model Performance Fairness). For two global models w_1 and w_2, w_1 is considered to be fairer than w_2, if the accuracy distribution of w_2, is more homogeneous than that of w_1 on m devices.

3.2. Aggregation Method

In our method, the evaluation method used is the CG data evaluation method proposed by Nohyun et al.[20], which can be performed before the model is trained. The mathematical expression for the calculation of CG evaluation score is shown below:

$$CG^i = y^T (G^\infty)^{-1} y - y_{-i}^T (G_\infty^{-i}) y_{-i}, \tag{7}$$

The meaning of this mathematical expression is the complexity gap value obtained by removing the data point (x_i,y_i) in a definite data set, where y_{-i} denotes the labeling vector obtained by removing the ith data sample in the dataset, and G_∞^{-i}, similarly, denotes the (n - 1) × (n - 1) matrix obtained by removing the i-th row and the i-th column elements in G_∞.

Assessment method 1 (data quality assessment based on data quality score). Here, the data assessment score is equal to the sum of the scores of the user data, and the mathematical expression is shown below:

$$Val_k = CG_k = \sum_{i=1}^{n_k} CG^i, \tag{8}$$

The expression is the sum of all the data scores of customer k. Here, not only the quality of the data is considered, but also the quantity of the data, which has a high degree of fairness.

Evaluation Method 2 (User Data Evaluation Based on Data Score and Frequency of Participation)CGF The evaluation score for this method, is not only related to the data quality score but also adds a certain weight to the frequency of participation. When filtering clients, it is based on the amount of data. In other words, the greater the frequency of user participation, which means that it contains more data, and there is a reason to have a higher score.

First, after the client passes the frequency of participation to the server, the server performs a normalization calculation:

$$f_k = \frac{f_k}{\sum_{k=1}^{m} f_k}, \tag{9}$$

Where f_k denotes the participation frequency of user k, and m denotes the m clients selected in each round

Since a higher participation frequency indicates a larger amount of information carried by the user, f_k is processed as follows:

$$f_k^{val} = \begin{cases} -log_2(1-f_k), & 1-f_k \neq 0 \\ -log_2(1-f_k+e), & 1-f_k = 0 \end{cases} \tag{10}$$

To prevent computational anomalies, e is added, where e is a very small number, infinitely close to 0.The final server-side calculation of the resulting frequency of participation of user k is.

$$f_k^{val} = \frac{f_k^{val}}{\sum_{k=1}^{m} f_k^{val}}, \tag{11}$$

Finally the evaluation score calculation is performed.

$$Val_k = \alpha \frac{CG_k}{\sum_{k=1}^{m} CG_k} + \beta f_k^{val} \tag{12}$$

Where α+β = 1, different values can be set in different cases.

The following new aggregation algorithm is proposed by us considering the data quality

Algorithm 1 Federated learning data valuation aggregation (FedDva).

Input parameters: K, S_t, l, E, η, T, k

Out parameters: well-trained w

Pre-operation: each device calculate the Val

while: t=0,1,.......,T-1

Server selects a subset S_t from K client with probability p_k.

Server sends w_t to all chosen client

Each chosen client do : $w_{t+1}^k = w_t^k - \eta\, \Delta\omega$ to Sever

Sever :

update the weights as: $weight_k = \frac{Val_k}{\sum_{k \in S_t} Val_k}$

Aggregate the w : $w_{t+1} = \sum_{k \in S_t} weight_k * w_{t+1}^k$

end while

When conducting synchronized federated learning, due to the extensive data of some customers or the performance of the equipment, the training time may be longer, in which case a longer waiting time is required, thus reducing the efficiency of the overall learning.

This module evaluates the data quality score of each user's dataset before training. When a customer is selected, the data with a high data quality score from the customer can be directly selected for training, which can save time and improve specific efficiency.

First, the data evaluation is performed locally, which can be affected by this evaluation module since the CG scores can assess the complexity of the data and can be done before performing model training. These can be done before training.

Then, their local data quality scores are ranked, and data points with high scores are prioritized for training.

After the data screening by the local evaluation module, in the particular case of tight training time, the high-quality data is prioritized and trained, and the training can be ended early and uploaded to the server for aggregation. The client-side data filtering algorithm is shown in Algorithm 2.

Algorithm 2 Client-side training time threshold algorithm

Input parameters: training time threshold T

Out parameters: w_{t+1}^{k}

Pre-operation: each device calculate Val, calculate p_k^i by Equation 6

while time < T do:

Client selects some data S_t from their dataset with probability p_k^i

Each chosen client updates w_{t+1}^{k}

If local training has been completed:

break

end while

send the w_{t+1}^{k} to sever

4. Experiment

In this section, we present the results of the experiments with FedDva algorithm. In the first sub-section, we describe the parameter settings and the dataset used for the experiments; in the second section, we show the results of this algorithm compared to other baselines and finally discuss the results.

4.1. Experimental Setup

Datasets. In our experiments, three main datasets are used: the MNIST dataset, the FMNIST dataset and the synthetic dataset, where the MNIST dataset [21] is an image classification of handwritten digits 0-9, a 784 dimensional 28*28 image with output labeling results between 0 and 9, a total of 60,000 data. The other is a synthetic dataset, where the degree of independent or non-independent homography can be set to achieve different experimental results. These two datasets are from previous work on federated learning [22-24].

Training model setup. CNN model, the learning rate is set to 0.01, the number of users is generally defaulted to 100, the number of customers extracted in each round is 10, the batch size is defaulted to 10, the two parameters α and β in the above text are defaulted to 0.9 and 0.1, and that very small parameter e in Eq. 10 is defaulted to 0.0001.

Baseline. We mainly use the classical baseline FedAvg, which is almost optimal in terms of accuracy and efficiency in most of the ideal cases, as well as several baselines with fair performance such as FedFa, q-Fl. and Fedprox, which takes system heterogeneity into consideration for reweighting. The learning rate is set to 0.01, the parameter q in q-FL is set to 1, and the two parameters in FedFa are set to (0.5,0.5).

Metrics. The main measures are the test accuracy and training loss of different aggregation methods on the same dataset, the performance fairness in Definition 3, and the convergence speed of the same methods under time constraints.

4.2. Accuracy and Loss Test Results

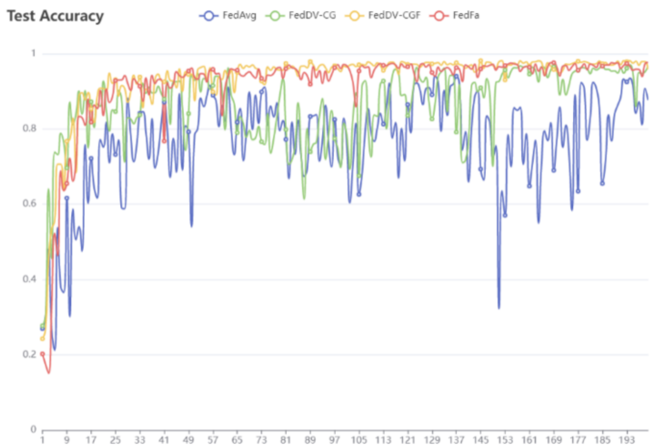


Figure 2. Test Accuracy with Equal Distribution of the non-iid MNIST Dataset

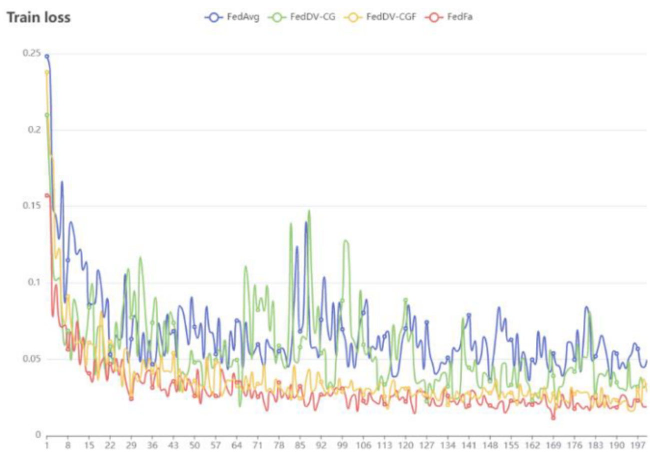
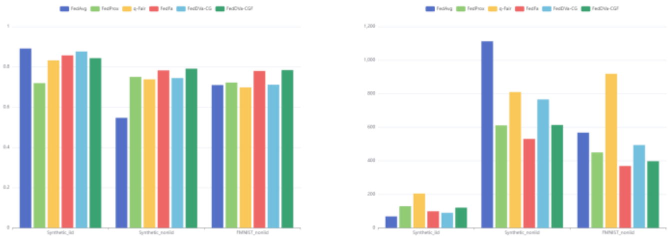


Figure 3. Train loss with Equal Distribution of the non-iid MNIST Dataset

The horizontal axis of the above figure indicates the number of running rounds in which the participation frequency and accuracy weight parameter of FedFa is set to 0.5, and it can be seen that after many rounds of training, in terms of testing accuracy, FedDV-CG, and FedDV-CGF are both slightly better than FedAvg, not much different from FedFa, but in terms of the convergence speed of training loss is not as good as that of FedFa, we analyze This is because our gradient optimization method uses SGD, while FedFa uses a scalar gradient descent method, so the convergence speed is faster and more stable.

4.3. Performance Difference Test Results



(a)Average accuracy (b)performance variance

Figure 4. Model Performance

From Figure 4, it can be noticed that our method has good average accuracy on both datasets, very close to the highest performing federal average in the independent identically distributed case, and the average test accuracy measured by our method is not weaker than that of several other schemes in the non-independently identically distributed case. It is found that according to Figure 5, compared with the other methods, our method guarantees a certain degree of performance fairness at the same time. Although our method is not optimal in terms of both fairness and performance, it achieves good results under both criteria and achieves a certain balance.

4.4. Results of the Time Thresholding Algorithm.

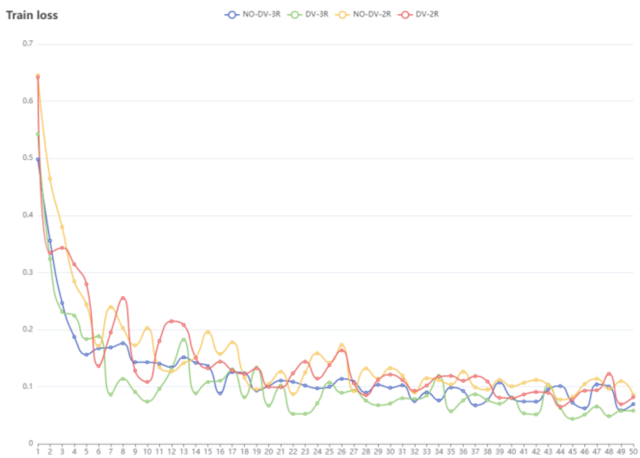


Figure 5. Training loss at time threshold

The horizontal axis in Figure 5 represents the global number of running rounds tested on the MNIST-no-iid dataset, where the basic algorithm uses FedAvg, where NO-DV-3R represents the result of the training loss obtained without data evaluation data filtering and with the client-qualified number of rounds of 3. In contrast, the DV-2R similarly represents the result of the run obtained after data evaluation data filtering and the running results obtained in the case where the number of client-qualified rounds is 2. It is evident that in the time-thresholded data screening algorithm, in the case of the same number of rounds (time), the training loss decreases significantly faster and more efficiently, proving that our method is effective.

5. Conclusion

In this paper, we have proposed new aggregation methods, FedDV-CG and FedDV-CGF, and have used data evaluation methods to have the effect of accelerating local efficiency. Not only the effect of data quality is considered in the aggregation, but also the effect of data quality, so that the algorithms can guarantee a certain degree of performance fairness. Experiments show that it also has a performance that is extremely close to FedAvg, achieving a certain trade-off between fairness and performance. However, there is still much room for improvement in our work, e.g., the evaluation of customer data before model aggregation can be changed to a randomized evaluation method, which can save more effort and time. In our future work, we can apply the idea of data evaluation to customer selection, and select high-quality customers through data quality and equipment performance to achieve better training results.

Acknowledgment

This paper is indebted to Mr. Jianfeng Lu for his guidance, to the school for providing the environment and resources for this work, and to his classmates for their enthusiastic help.

This research was funded by the National Natural Science Foundation of China (No.62372343, 62072411), the Zhejiang Provincial Natural Science Foundation of China (No. LR21F020001) and the Key Research and Development Program of Hubei Province (No. 2023BEB024).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Agueray Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics. PMLR, 2017, 1273-1282.

[2] S. -A. Naas and S. Sigg. Fast converging Federated Learning with Non-IID Data," 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 2023, 1-6. doi: 10.1109/VTC2023-Spring57618.2023.10200108.

[3] H. Wang, Z. Kaplan, D. Niu, & B. Li. Optimizing federated learning on non-iid data with reinforcement learning. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications. 2020, 1698-1707.

[4] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, & V. Smith. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2, 429-450.

[5] Z. Zhou, Y. Li, X. Ren and S. Yang. Towards Efficient and Stable K-Asynchronous Federated Learning With Unbounded Stale Gradients on Non-IID Data. In IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12). doi: 10.1109/TPDS.2022.3150579.

[6] T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. arXiv:1905.10497, 2019.

[7] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, & T. Huang. Fairness and accuracy in horizontal federated learning. Information Sciences, 2022, 589, 170-185.

[8] L. Gao, L. Li, Y. Chen, C. Xu, & M. Xu. FGFL: A blockchain-based fair incentive governor for Federated Learning. Journal of Parallel and Distributed Computing, 2022, 163, 283-299.

[9] Z. Zhao, & G. Joshi. A dynamic reweighting strategy for fair federated learning. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing. 2022, 8772-8776). IEEE.
[10] Z. Xie, X. Wang, H. Zhang, I. Sato, & M. Sugiyama. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In International Conference on Machine Learning 2022, 24430-24459. PMLR.
[11] W. Liu, L. Chen, Y. Chen, Y., & W. Zhang. Accelerating federated learning via momentum gradient descent. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(8), 1754-1766.
[12] Y. Gong, Y. Li and N. M. Freris. FedADMM: A Robust Federated Deep Learning Framework with Adaptivity to System Heterogeneity. 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022, 2575-2587. doi: 10.1109/ICDE53745.2022.00238.
[13] S. P. Karimireddy, Q. Rebjock, S. Stich, M. Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In International Conference on Machine Learning. 2019, 3252-3261. PMLR.
[14] Z. Zhou, Y. Li, X. Ren, & S. Yang. Towards Efficient and Stable K-Asynchronous Federated Learning with Unbounded Stale Gradients on Non-IID Data. IEEE Transactions on Parallel and Distributed Systems. 2022.
[15] W. Yang, Y. Yang, X. Dang, H. Jiang, Y. Zhang and W. Xiang. A Novel Adaptive Gradient Compression Approach for Communication-Efficient Federated Learning. 2021 China Automation Congress (CAC), Beijing, China, 2021, 674-678. doi: 10.1109/CAC53003.2021.9728013.
[16] L. Liu, J. Zhang, S. H. Song and K. B. Letaief. Communication-Efficient Federated Distillation with Active Data Sampling. ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, 201-206. doi: 10.1109/ICC45855.2022.9839214.
[17] X. He, J. Zhang and Q. Ling. Byzantine-Robust and Communication-Efficient Personalized Federated Learning. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, 1-5. doi: 10.1109/ICASSP49357.2023.10095468.
[18] F. Sattler, S. Wiedemann, K. -R. Müller and W. Samek. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. in IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(9), 3400-3413. doi: 10.1109/TNNLS.2019.2944481.
[19] Z. Zhang, L. WL, D. He, J. Li, S. Cao and X. Wu. Communication-Efficient and Byzantine-Robust Federated Learning for Mobile Edge Computing Networks. in IEEE Network, 2023, 37(4), 112-119. doi: 10.1109/MNET.006.2200651.
[20] K. Nohyun, K., Choi, H., & Chung, H. W. Data valuation without training of a model. In The Eleventh International Conference on Learning Representations. 2022
[21] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 1998, 86 (11) 2278–2324.
[22] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks.Proceedings of Machine Learning and Systems, 2020, 2:429–450.
[23] T. Li, S.Y. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, 6357–6368. PMLR.
[24] Z. Fan, H. Fang, Z. Zhou, J. Pei, M.P. Friedlander, C. Liu, & Y. Zhang. Improving fairness for data valuation in horizontal federated learning. In 2022 IEEE 38th International Conference on Data Engineering (ICDE). 2022, 2440-2453. IEEE.