

The Improvement of Federated Learning in communication efficiency

Yimeng Yang *

College of computer and information science, college of software, Southwest University, 2
Tiansheng Road, Beibei District, Chongqing China

* Corresponding Author Email: 201900110064@mail.sdu.edu.cn

Abstract. In recent years, the need to protect user privacy and data security as well as multi-party data cooperation has become increasingly strong. Privacy computing is considered a fundamental solution that is growing rapidly and gaining unprecedented popularity. Currently, federated learning is the most popular product in privacy computing. With its lightweight technology route and deployment, it has become a mainstream solution and product of choice for many privacy computing applications, such as providing advanced medical image collection segmentation technology in the medical field to protect patient privacy, where it can play a unique role. But Federated learning faces a need to improve communication efficiency. This paper introduces FedAvg, the most basic algorithm for federated learning, including its basic process, advantages and disadvantages. We analyze algorithms that optimize federated learning by adding local computation, Communication Compression (including model Compression, Periodic Averaging and Quantization Compression, and gradient Compression), computation and communication overlap, and compare the strengths and weaknesses of representative algorithms for each approach. At the end of this paper, the summary and prospect are made.

Keywords: Federated learning; communication efficiency; local computation; communication compression

1. Introduction

As the public and policymakers increasingly aware of the importance of protecting privacy, demanding of data in practice to protect the privacy of machine learning is rising. Thus, the federated learning is more and more active and becoming research hot. Federated learning can protect the privacy of people and corporation, by using the distributed machine learning. Federated learning is a core technology of privacy computing and is regarded as a typical distributed machine learning technology. The main idea of this approach is to conduct distributed model training among multiple data sources with local data. On the premise of not exchanging local individual or sample data, Federated Learning only exchanges model parameters or intermediate results. A global model based on virtual fusion data is constructed to realize data privacy protection.

Federated learning is still a work in progress, with problems and challenges waiting to be addressed by scholars [1-2]. The typical major problems and challenges of federated learning focus on the following three areas [3]: The communication of federated learning is slow to a certain extent, and there are also unstable situations [4]; Federated learning participants heterogeneous devices, different devices have different computing power. For example, there is a big computing power gap between mobile phones and computers [5]. Because federated learning is a private computing technology, it is more concerned with data security and privacy [6]. Most current research assumes that clients and servers are trusted, which may not be true in real life, which means federated learning has a privacy threat problem.

At the algorithmic level, the accelerated optimization of federated learning mainly includes methods such as reducing the number of communication rounds between the client and server, model compression, computation and communication overlap. The original algorithm is FedAvg, however, FedAvg has two drawbacks: device heterogeneity (different devices have different communication and computing power) and data heterogeneity (data from different source may be non-independent and identically distributed). At present, many researches have been proposed to solve these problems.

In order for the problem of Non-IID and communication efficiency to be solved effectively, scholars in related fields have proposed various solutions, but the review papers on the analysis of improving the communication efficiency of federated learning are not comprehensive enough.

This paper is mainly to analyze and compare the federated learning algorithms which can accelerate the convergence speed and improve communication efficiency. These algorithms reduce the communication overhead from different angles. However, some algorithms, such as FedProx and SCAFFOLD, reduce the training accuracy while increasing the communication efficiency. Other algorithms also bring some costs. For specific algorithm comparison analysis, see Table The Algorithm Comparison of Federated Learning involved in this paper, at the end of this paper. First introduced the most basic federated learning algorithm, FedAvg, explained its algorithm framework and process, as well as its advantages and disadvantages, and then the improved algorithms are introduced respectively from the perspectives of increasing local calculation, communication compression, calculation and communication overlap, and the representative algorithm of each method is introduced, including the process of each algorithm and how to improve the communication efficiency. The advantages and disadvantages of the representative algorithms of these methods are analyzed and compared. (The structure of the paper is illustrated in figure 1 The framework and typical algorithms mentioned in this paper)

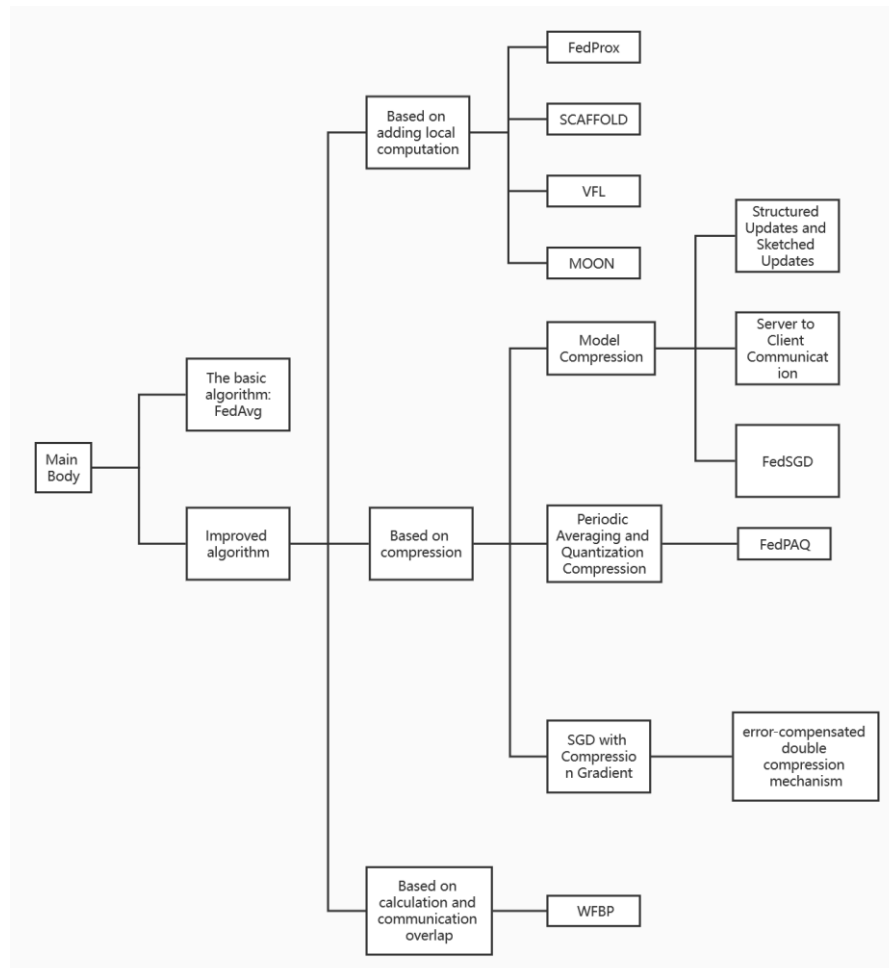


Figure 1. The framework and typical algorithms mentioned in this paper

2. The basic algorithm: FedAvg

FedAvg [7] is the most basic and classical federated learning algorithm. FedAvg is a distributed machine learning framework. This algorithm can realize multiple clients to train machine learning models at the same time. No private data is uploaded to the server during FedAvg training. The server

aggregates the local model weighted by the local client to produce the global model. The local client is responsible for training the local data to produce the local model. A model that resembles the outcomes of centralized machine learning is eventually created after numerous cycles of repeated iterations, which significantly lowers the number of privacy issues brought on by the combination of standard machine learning source data.

(1) First, the client downloads the model parameters from the server, and the client updates the local model parameters locally. This process is the local machine learning training.

(2) Secondly, the accuracy of the model is constantly updated by local stochastic gradient descent among users. When the predetermined local training times are reached, the model parameters that locally trained by the local clients are uploaded to the server.

(3) The server randomly selects user devices and receives model parameter gradients uploaded by local users for aggregation. Represents the gradient aggregation of server model. The server will average the sampled user model parameters gradients and add them to the model parameters after the previous round of aggregation to update the global model parameters. Finally, the aggregated model parameters are passed back to the sampling user device, and then the operation of Step 1 is continued. Repeat the preceding steps until the communication times reach t .

Under the framework of typical FedAvg: The server gives the clients taking part in the training access to the global parameters during each cycle of communication between the client and the server. Then each client participating in the training uses local data to train for the same epoch, and the client uploads the gradient obtained from the local training to the server for aggregation to form the updated parameters.

3. Based on adding local computation

FedProx algorithm [8] optimized the federated learning algorithm from the perspective of adding local computation, fine-adjusts FedAvg, and adds a correction term to the loss of the clients. By adding the correction term of proximal term, the stability of the overall convergence is improved, and the model converges better and faster, increasing the computation cost. Non-IID data can be optimized, but at the cost of reduced accuracy.

The idea of SCAFFOLD algorithm [9] is similar to FedProx. It also modifies gradient and adds a correction term, so that each local update is pulled back to the ideal update path, and the convergence speed is fast. SCAFFOLD improves the local update strategy and uses control variates to overcome client drift. It has a fast convergence rate and is not affected by the distribution of client data. It can effectively reduce the communication frequency, which can improve communication efficiency.

Different from the previous two algorithms, MOON algorithm [10] is designed to optimize the training effect on the image dataset. By combining the idea of the contrast learning framework SimCLR [11], MOON algorithm obtains inspiration from NT-Xent Loss and proposes Model-Contrastive Loss, which makes the effect of solving data heterogeneity in image datasets better. Traditional comparative learning is at the data level. FedAvg's local model training stage is improved by MOON; throughout the training phase, the client model can learn a representation that is more similar to the global model and learn a better representation than the prior iteration of the client local model. The global model's representation is better than the local model's, according to the authors of the work on the MOON method, so training should take place far from the local model and near the global model. Finally, the author of this paper confirmed this idea through experiments. In conclusion, the federated learning model trained on the Non-IID image dataset performed well because the author added a Model-contrastive loss when training the local model.

In addition, vertical federated learning [12] is optimized from the perspective of sample features and can also improve the privacy of federated learning. VFL is a verifiable privacy preserving federated learning for efficient and secure industrial intelligence model training. Specifically, VFL uses Lagrangian interpolation and blinding techniques to achieve safe gradient aggregation. At the same time, an effective verification mechanism is designed, in which participants can detect falsified

results with great probability. Based on the characteristics of Lagrangian interpolation, VFL proposes a verification mechanism to verify the correctness of aggregation results. In this scheme, each participant can detect falsified results independently and effectively.

Compared with existing schemes, the number of participants has no effect on the computational complexity of this verification technique. This verification mechanism also requires to reduce the overhead for each participant, so this algorithm can not only guarantee the gradient will not be leaked, but also increase the communication efficiency and accelerate the convergence speed.

4. Based on Model Communication Compression

4.1. Model Compression

Kjonečny J et al. proposed a structured model update method to update the server parameters. The size of the model update parameters that each participant submitted to the server during a parameter communication round was decreased in order to decrease communication. Kjonečny J et al. studied two methods, these are Structured Updates and Sketched Updates. Both methods are efficient at cutting down on upload overhead.

Structured Updates: The space of variables that control updates is inherently small. Direct learning to update results from a reduced parameter using fewer variables such as low-rank and random masks. **Sketched Updates:** Prior to submitting the update to the server, the update must first be compressed. The complete model is learned, updated and compressed using quantization, random rotation down sampling before sending to the server.

Also optimized from the perspective of model compression, but using a different approach is that, Caldas et al. proposed a method that can effectively reduce the cost of downlink Communication (Server to Client Communication) and seamlessly integrate with existing methods to reduce the cost of uplink Communication [13], which specifically includes the use of lossy compression on the global model of Server to Client. And Federated Dropout, which lowers both client-to-server communication costs and local computation costs by enabling users to effectively complete local training on a smaller portion of the overall model. Reducing the cost of communication comes at the expense of some degree of model accuracy.

Similar to the ideas of the above two algorithms, Rothchild El al. Proposed a method FedSGD that uses Count Sketch to compress the update of the client model [14]. Since Count Sketch is linear, momentum and error accumulation can be calculated through Sketch, so the calculation task of momentum and error accumulation can be transferred from the client to the central server, which overcomes the problem of sparse clients participating in updating, while maintaining high compression rate and good convergence. Compared with FedAvg, this method reduces the traffic by increasing the local computation of nodes.

4.2. Periodic Averaging and Quantization Compression

Reisizadeh et al., proposed FedPAQ [15], a processing method of periodic averaging and quantization, and quantization itself is also a way of compression. FedPAQ allows clients in the network to perform local training before synchronizing with a central server, sending only updates from active clients back to the central server, and sending back only a quantized version of the local information. In generally, FedPAQ uses three modules to lessen the communication load: cycle averaging, partial client participation, and quantization processing. More training iterations are needed since the problem of decreasing convergence accuracy is brought on by this communication loss.

4.3. SGD with Compression Gradient

In addition to the most frequently used model training algorithm "local random gradient" in federated learning, Hongchang Gao et al. proposed a new distributed random gradient algorithm for efficient communication [16], which can compress the gradient exchanged between workers and

servers through the error compensation dual compression mechanism. It solves the problem of large communication overhead in the effective training of federated learning large-scale models, and significantly reduces the communication cost under the condition of ensuring convergence performance and model accuracy. Compressing gradients sent from the node to the central server and from the central server to the node using the α -compression operator (such as top-K scarification operator) is the fundamental concept behind local SGD with compression gradients. Compression of gradients can also improve the privacy of federated learning.

5. Based on calculation and communication overlap

For hierarchical and stacked structure of neural network, typical work is like wait-free Backward Propagation [17], which finds communication process and computation process without dependence and implements them in parallel through Scheduling. As shown in the figure 2 The process of WFBP, the blue block of Scheduling represents the gradient computation process. The orange block represents the communication process, where the dark part is the communication start time, and the light part is the data transmission time. Assuming that the gradient calculation of the current layer (L) has been completed at this time, the gradient communication process of the current layer (L) can be executed in parallel with the gradient calculation process of the next layer (L-1).

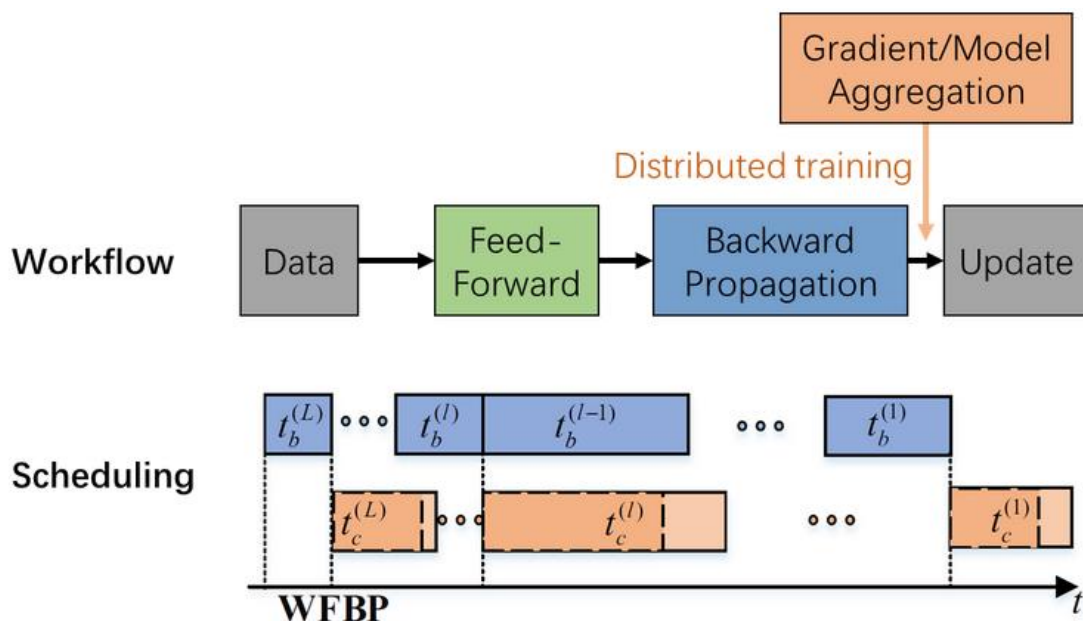


Figure 2. The process of WFBP

There are usually three cases when WFBP is used, as shown in figure 3. Three cases of WFBP. The ideal Case is Case 1. The communication time is shorter than the gradient calculation time, and the communication process can be completely parallel with the gradient calculation process. In the Case of Case 2, although some communication processes are time-consuming and cannot be well covered in a single scheduling, no additional communication overhead is generated on the whole. In the Case of Case 3, there is an overall additional communication overhead due to more time-consuming communication processes. Considering that every communication process has startup time, the situation of Case 3 is relatively common, so there is MG-WFBP(merged-Gradient WFBP) [18], that is, multiple communication processes are Merged into one, thereby reducing the proportion of startup time and thereby reducing the overall communication time, as shown in figure 4. The case of MG-WFBP.

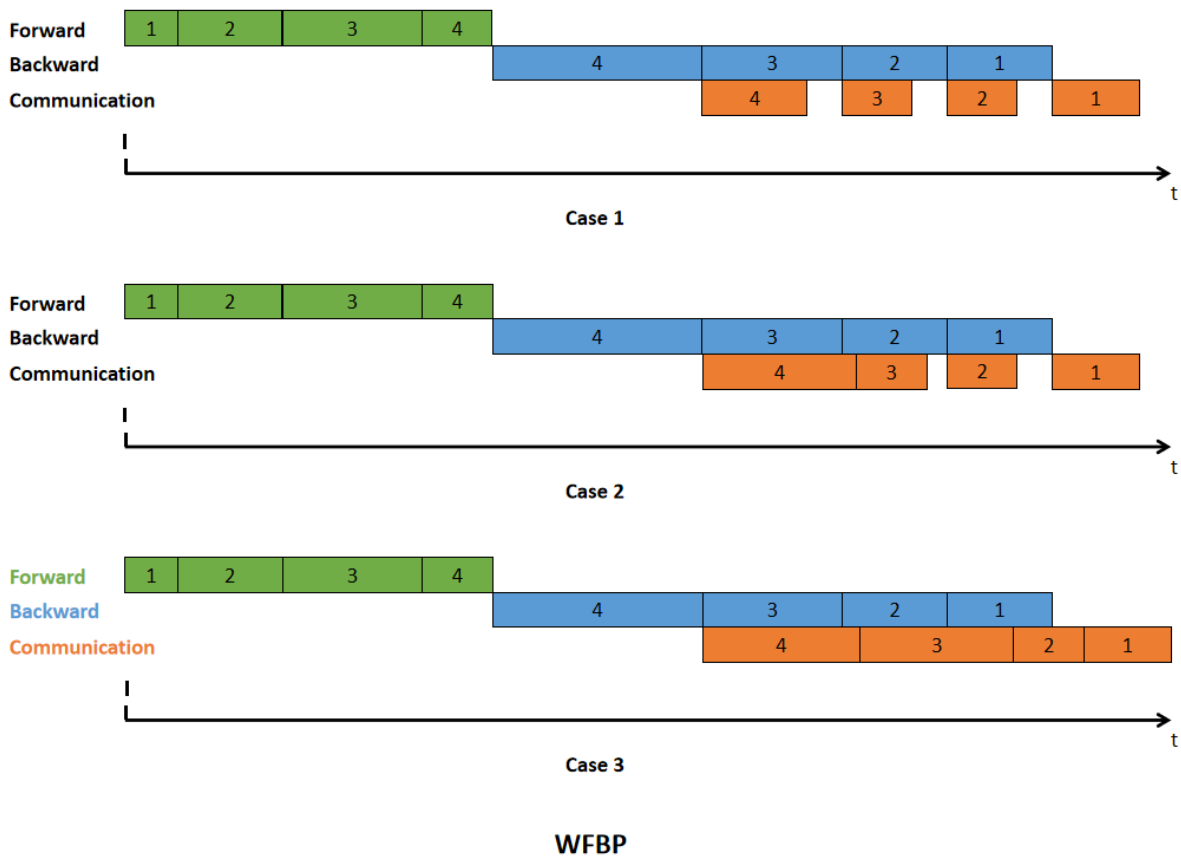


Figure 3. Three cases of WFBP

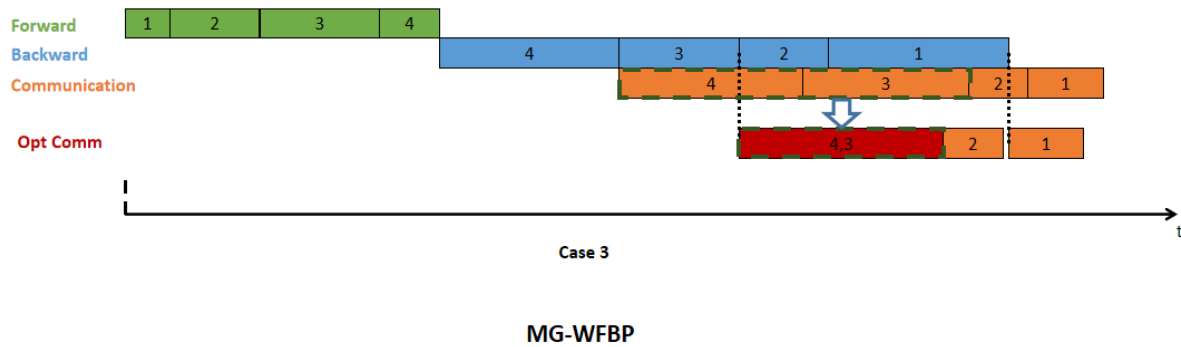


Figure 4. The case of MG-WFBP

Table 1. The algorithm comparison of federated learning involved in this paper

Method of optimization	Algorithm	advantages and disadvantages
Adding Local Computation	FedProx	It increases the computational cost and can optimize Non-IID data at the cost of reduced accuracy.
	SCAFFOLD	Improve the convergence speed, not affected by the client data distribution, can effectively reduce the communication frequency, the disadvantage is that it will increase the amount of calculation.
	MOON	The federated learning model trained on Non-IID image datasets has a good effect of improving the convergence rate, but the disadvantage is that the training accuracy is not good enough.
	VFL	Increasing computational cost comes at the cost of decreasing accuracy

	Structured Sketched Updates	To increase the effectiveness of communication between the participants and the server, the transmission model is compressed, but at the expense of convergence issues in the intricate model structure.
Model Compression	Server - Client updates	At the expense of decreased accuracy and potential convergence issues, the compressed transmission model increases the communication efficiency between the server and the participant.
	FedSGD	The problem of sparse clients participating in the update is overcome, while maintaining high compression rate and good convergence, at the cost of reduced accuracy and possible convergence problems.
Periodic Averaging and Quantization Compression	FedPAQ	Under the condition of ensuring convergence performance and model accuracy, the communication overhead is significantly reduced, but the accuracy is reduced.
SGD with Compression Gradient	error-compensated double compression mechanism	The communication overhead is significantly reduced while the convergence performance and model accuracy are guaranteed. The cost is that there may be convergence problems.
Calculation and Communication Overlap	Wait-Free Backward Propagation	Combining multiple communication processes can further reduce the communication time, which will bring information loss during data transmission.

6. Conclusion

One of the major challenges of federated learning is communication efficiency. This review paper analyzes, compares and summarizes the optimization algorithms of federated learning. At the algorithm level, the accelerated optimization of federated learning mainly includes methods such as lowering the number of rounds of communication, model compression, computation and communication overlap. These methods, like FedProx, VFL, and other algorithms, can improve the communication efficiency of federated learning, accelerating training and convergence speed at the expense of accuracy. How to balance communication efficiency and accuracy at the same time is also a challenge of federated learning. It is expected that more scholars with challenging spirit will solve this problem in the future, and make federated learning play better. It is anticipated that federated learning will soon be able to offer more applications with security services that can share and preserve privacy, as well as more diverse application situations, thanks to the ongoing study and efforts of academics.

References

- [1] Liu Zaiyi, Shi Zhenwei, Liang Changhong. Application of federated learning technology in medical image artificial intelligence. 2021, *Chin J Med*,102(05):318-320.
- [2] Zhou Chuanxin, Sun Yi, Wang Degang, Ge Huanhui. 2021 *J. net. Inf. Sec.*,7(05):77-92.
- [3] Liang Tiankai, Zeng Bi, Chen Guang. Federal study reviews: concepts, technology, application and challenges. 2022, *Compute. Appl.*, 1-13.
- [4] Qiu Xinyuan, Ye Zecong, Cui Tei long, Gao Zhiqiang. A survey on the Communication Cost of Federated Learning. 2021, *J. Compute. Appl.*, 222,42(02):333-342.
- [5] Wang Zhuangzhuang, Chen Hongsong, Yang Limin, Chen Lifang. Federated learning and data security. 2021, *Intel. Compute. Appl.*,11(01):126-129+133.
- [6] Wang Kunqing, Liu Jing, Li Chen, Zhao Yuhang, Lu Haoran, Li Peng, Liu Bingying. A review of Federated Learning security threats.2020, *Inf. Sec. Res.*, 222,8(03):223-234.

- [7] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas: Communication-Efficient Learning of Deep Networks from Decentralized Data. 2017, *AISTATS*: 1273-1282
- [8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, Virginia Smith: Federated Optimization in Heterogeneous Networks. 2020 *ML Sys* 71 36
- [9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, Ananda Theertha Suresh: SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. 2020, *ICML*: 5132-5143.
- [10] Qinbin Li, Bingsheng He, Dawn Song: Model-Contrastive Federated Learning. Proceedings of the 2021, *Conf. Compute. Vis. Pat. Rec.* 10713-10722
- [11] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey E. Hinton: A Simple Framework for Contrastive Learning of Visual Representations. 2020 *ICML*: 1597-1607
- [12] Yanci Zhang, Han Yu: Towards Verifiable Federated Learning. 2022, *IJCAI*: 5686-5693
- [13] Sebastian Caldas, Jakub Konečný, H. Brendan McMahan, Ameet Talwalkar. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. 2018, *CoRR* abs/1812.07210
- [14] Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., et al. FetchSGD: communication-efficient federated learning with sketching, 2020, *ICML* 891
- [15] Wang Jian-zong, Kong Ling-wei, Huang Zhang-cheng. A survey of federated learning algorithms. 2020 *Big data*, 6(06):64-82.
- [16] Hongchang Gao, An Xu, Heng Huang: On the Convergence of Communication-Efficient Local SGD for Federated Learning. 2021 *AAAI*: 7510-751
- [17] Hao Zhang, Zeyu Zheng, Shizhen Xu, Wei Dai, Poseidon: An Efficient Communication Architecture for Distributed Deep Learning on GPU Clusters. 2017 *CoRR* abs/1706.03292
- [18] Shaohuai Shi, Xiaowen Chu, MG-WFBP: Efficient Data Communication for Distributed Synchronous SGD Algorithms. 2018 *CoRR* abs/1811.11141