Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230330

Research on communication efficiency and privacy security of federated learning

Junhao Zhou

Liangjiang International College, Chongqing University of Technology, 400054, Chongqing, China

shenllyz@gmail.com

Abstract. In recent years, to surmount the challenges of data sensitivity and data silos, federated learning (FL), a collaborative decentralized privacy preservation machine learning framework was proposed and rapidly became a current research hotspot. In this paper, we present an overview of FL workflow, as well as the communication overhead and privacy leakage problems associated with FL. In order to alleviate these two pain points, several corresponding solutions are presented in this paper along with their advantages and disadvantages. To reduce the communication overhead, this paper suggests three reliable approaches by reviewing related research including increasing the amount of computing on the edge side, model compression, and improving parallelism. To address the privacy concerns, this paper introduces three prevailing privacy preservation techniques: differential privacy, homomorphic encryption, and secure multiparty computation (SMC), and shows how to efficiently incorporate these privacy-preserving techniques into a federated learning framework by summarizing relevant literature. On the whole, this paper contributes to explaining the overall workflow of a FL framework and helps advance the understanding of FL in improving communication efficiency and enhancing privacy security.

Keywords: federated learning, communication overhead, privacy preservation.

1. Introduction

An accurate machine learning model is based on bountiful high-quality data, however, most of the enterprises generally have insufficient data to train a reliable mode. Besides, under the traditional machine learning framework, the data center or cloud server can directly access the data, this method inevitably incurs uncontrollable data flow and privacy preservation problems of data that is sensitive. Therefore, data-sharing is challenging to be realized between different enterprises. Even within the same enterprise, limited by industry privacy and other issues, data may be opaque between different departments, making data exist in the form of isolated islands [1]. The centralized training data model also poses a great challenge to the computing and storage capacity of enterprise servers.

In order to overcome the constraints and challenges mentioned above under the traditional machine learning(ML) framework, Google first proposed and introduced the concept of FL in 2016[2][3]. In FL, virtual models are designed to solve the problem of collaboration without involving data exchange between different data holders. FL framework stores data locally, besides, the model training phase is also migrated to the local devices and the databases of each party still exist independently. FL allows

DOI: 10.54254/2755-2721/6/20230330

mobile devices to train models locally and only exchange model updates with the central server, and strict encryption algorithms are applied to communication among the parties. Therefore, FL ensures the security and privacy of information. In addition, the accuracy of the global model trained by FL is almost the same as that trained by traditional ML methods.

For the purpose of protecting user privacy and ensuring data security, FL solves the problem of siloed data, and enables all participants to effectively use their local model parameters to obtain a high-quality FL model. Nevertheless, there are still many challenges threats and to be solved in FL.

By analyzing and summarizing relevant literature, this paper stratifies the FL system and sorts out the current achievements of FL in the fields of communication efficiency and privacy security.

In this paper, section 2 specifies the basic workflow of the current federated learning framework. Section 3 presents the communication overhead problem in federated learning and proposes three effective solutions to reduce the cost. Section 4 mainly introduces the privacy security problem in federated learning, and analyzes three widely used privacy protection technologies.

2. Workflow of federated learning

Generally, one iteration of a FL system is comprised of four different steps:

(1) System initialization. The central server selects training tasks, target applications, and data requirements. After reaching an agreement with all parties, the central server specifies the global model and the hyperparameters of the training process, such as learning rate and batch size. Then the initial global model $w_G^{i=0}$ is issued by the central server to all parties.

(2) Local computation. Each participant trains the model locally and update local model parameters (gradients) w_j^i , where i denotes the number of iterations of the global model w_G^i,and j represents the serial number of the current participant. The goal of the jth participant is to minimize the loss function $L(w_j^i)$by applying appropriate algorithms and uploading the updated local model parameters $w_j^{i'}$ to the central server, where

$$w_j^{i'} = \arg\min_{w_j^i} L(w_j^i) \tag{1}$$

(3) Secure Aggregation. The central server aggregates model parameters after collecting the updated parameters from multiple data holders. In the process of aggregation, factors such as communication efficiency, systems heterogeneity, security and privacy need to be comprehensively considered. The server needs to perform secure aggregation without learning any information about the participants. Therefore, some encryption technologies such as Homomorphic Encryption [4] and SMC [5] are usually used to encrypt parameters to ensure that parameter aggregation can be carried out securely.

(4) Model updating. The center server updates the global model by minimizing the global loss function

$$L(w_G^i) = \frac{1}{N}\sum_{j=1}^{N} L(w_j^i) \tag{2}$$

After evaluating the performance of the updated model, the center server returns the updated model w_G^{i+1} to each party. Repeat the above steps until the global loss function converges or the performance of the global model meets the requirement. The well-trained global model will be retained in the central server for pragmatic applications including semantic location, autonomous vehicles, and so on.

DOI: 10.54254/2755-2721/6/20230330

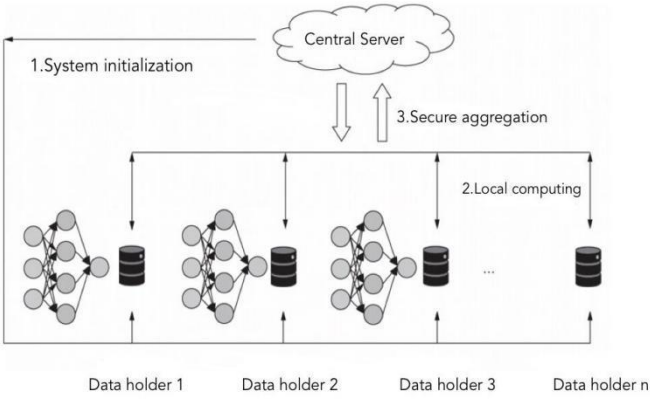

Figure 1. One iteration of federated learning.

3. Communication overhead in federated learning

In FL, the global model training time is generally composed of communication transmission time and data processing time. Recently, as computing power has increased, the time it takes to process data has greatly decreased, however, a FL network may consist of a large number of devices, thus the communication transmission time of federal learning has become a critical bottleneck limiting its training speed [6]. The central server needs to aggregate the model parameters from all parties in each round of training under the FL model, and then dispatch the updated global model parameters to each party. Therefore, communication overhead will become more and more expensive as the number of iterations and participants increase. With the continuous growth of the data set scale, the ML model also becomes more complex. For a convolutional neural network(CNN) model, each iteration may contain millions of parameters, and the continuous data transmission between clients and sever will greatly increase the communication cost and is considered an inefficient way from the perspective of resource utilization. Additionally, unstable network connection and inconsistent speed in the process of parameter upload and download may incur exorbitant model training costs as well.

Generally, there are several methods of reducing the communication overhead:

3.1. Increase the amount of computing on the edge side.

The edge side frequently exchanges model updates with the central server, which is the main reason for the high communication overhead. A feasible method to lower the communication overhead is to reduce the frequency of model uploading on the edge side and increase the amount of training on the local model. Liu et al. proposed a collaborative learning framework named Federated stochastic block coordinate descent (FedBCD), where parties perform more than one consecutive local gradient upgrade in parallel before communicating with the central server and only share a single value for each sample instead of model parameters and raw data [7]. The overall communication overhead can be greatly reduced by performing multiple local steps before aggregation. This method was finally proved to be more effective in lessening the number of communication rounds and the communication cost compared with (Federated stochastic gradient descent) FedSGD algorithm [8]. At present, the most commonly used aggregation algorithm is the FedAvg algorithm proposed by McMahan [9]. At each iteration of FedAvg,E epochs of SGD are performed on devices, where is a presented constant and denotes a fraction of the total devices involved. Compared with the FedSGD algorithm, on different CNN and recurrent neural network (RNN) architectures, the required communication rounds are reduced by 10-100 times without significantly influence on the convergence speed. However, the FedAvg algorithm also has several shortcomings. For example, the weight is allocated according to the amount of data on the client side, which makes the global model easily affected by clients with amounts of duplicate data. Additionally, FedAvg diverges empirically in settings where the data is non-identically distributed.

DOI: 10.54254/2755-2721/6/20230330

Since the FedAvg algorithm was proposed, a large number of follow-up studies have been carried out on this basis. SAHU A K et al. introduced a framework named FedProx, which can be regarded as a generalization and re-parametrization of FedAvg, to optimize systems and statistical heterogeneity in federated networks while maintaining the privacy and computational benefits originating from FedAvg. FedProx allows to dynamically update the number of local computations required by different clients in each round, which helps to improve the absolute testing accuracy by 22% on average in highly heterogeneous settings [10].

3.2. Model compression

Model compression is frequently applied to distributed ML to mitigate communication overhead. Unlike increasing the amounts of local computing, model-compression such as sparsification and quantization can significantly lower the size of messages communicated at each round [11]. KONEČNÝ J et al. proposed two methods for reducing the uplink communication costs which are structural updates and sketched updates. The structured update refers to uploading the model by defining the matrix structure (and random mask) of the uploaded model parameters ahead of time. Sketched updates learn the full model during local training and approximate or encode it in a lossy compressed form before sending it to the central server. Experiments on both convolutional and recurrent networks show that compared with FedAvg, the proposed methods can reduce the number of communication rounds by two orders of magnitude, however, its convergence speed decreases slightly. Model compression approach such as quantization is critical for reducing the communication overhead in the DNN model since it generally requires millions of parameters, nevertheless, it impairs the performance of the model. For example, gradient quantization quantizes the gradient into a low-precision value to reduce the communication bandwidth. Instead of directly uploading the raw gradient, it uploads the quantized gradient so as to reduce the number of communication bits per round. Unfortunately, the overall computing consumption may increase since the lack of accuracy. Shi et al. proposed a method known as Flexible Spar(flexible sparsification) for reducing the communication overhead by integrating their training algorithm with two state-of-the-art communication compression approaches: local computations and gradient sparsification [12]. Local computations are to perform more calculations on the edge side as we mentioned before, while gradient sparsification aims to upload a fraction of gradients with significant magnitudes. In addition, by accumulating the error that arises from only uploading sparse approximations of the gradient updates and then applying them to each party after model updates, the global convergence can be accelerated to a certain extent. According to Fig.2, the Flexible Spar method consumes less energy than other methods, especially to FedAvg, which indicates Flexible Spar achieved considerable progress in reducing communication overhead in the premise of reaching target accuracy. Besides, Fig.3 reveals that the convergence rate between Flexible Spar and Unified Spar are virtually identical, and also show similar performance. However, the performance of both methods is slightly lower than FedAvg. It reflects the disadvantage of model-compression approaches: inevitably impairs the final accuracy.

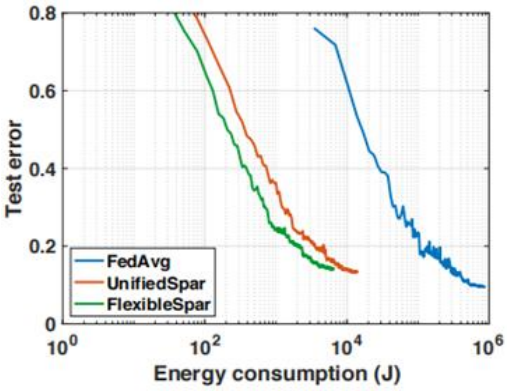

Figure 2. Energy consumption under the same target accuracy.

DOI: 10.54254/2755-2721/6/20230330

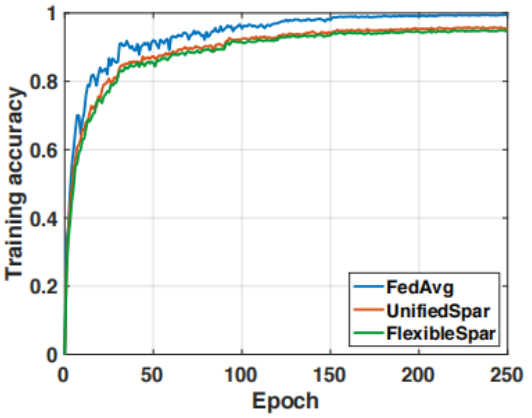

Figure 3. Communication rounds under the same target.

3.3. Improve parallelism

Parallel computing is an algorithm that which several processors can execute multiple instructions simultaneously and is considered an effective means for improving the computing efficiency and processing of complex problems. Traditionally, the process of global aggregation in FL is synchronous, namely, all devices are trained locally for a certain number of rounds in a fixed time interval, and then aggregated. However, parallel computing, especially synchronous parallel computing, is generally confronted with a significant short plate effect. When a participant makes an error and needs to recalculate, since the calculation time of this node generally exceeds that of other nodes, other nodes have to stall until the node completes the calculation, resulting in reducing the calculation efficiency. To lower the uploading latency arise from global model aggregation and the short plate effect caused by synchronization, Shi et al [13] presented a joint device scheduling and resource allocation policy. By means of an independently designed greedy algorithm, in which the devices with the shortest updating time are added to the scheduled device set successively until the convergence bound begins to increase, an ideal compromise is achieved between the per-epoch latency and the number of required epochs so as to minimize the global loss function for model parameters attained under a limited training time budget. Zhou et al [14] proposed an FL framework named Overlap-FedAvg, which allows continuous local model training without any blocking introduced by frequent communication. To improve the parallelism, a gradients compensation mechanism is applied to Overlap-FedAvg. This approach successfully alleviates the gradient staleness problem caused by parallelism. Extensive experiments show the parallelism of Overlap-FedAvg, which is capable to boost both the training phase and the communication phase on the premise of maintaining virtually identical accuracy with FedAvg. Besides, by applying a NVG algorithm [15], Overlap-FedAvg is suitable for scenarios with relatively large models and slow or unstable network connection of clients. On the whole, improving the parallelism in FL is an effective approach to reducing the communication overhead.

4. Security and privacy in federated learning

Even though neither raw data from users nor data sharing between users is required under the FL framework, which seemingly significantly enhance the privacy and security of data user, risks and threats still exist in FL. The privacy information leakage problem generally occurs in the aggregation stage and model updating stage. In the aggregation stage, a malicious central server can reconstruct the model according to the collected data characteristics of the participants through the parameters uploaded by the participants and then recover the original data of the target data holder [16], or initiate an attribute inference attack to infer whether the data of the target data holder contains some sensitive attributes. In the model updating stage, a malicious participant can initiate a membership inference attack [17] to infer whether a specific sample exists in the local database of a participant to obtain pertinent information

DOI: 10.54254/2755-2721/6/20230330

about the participant's original data. A lot of research has been conducted to enhance privacy security to resolve the issue of privacy leakage in FL. In general, there are three widely-used privacy preservation techniques in FL: *Differential Privacy* (DP), *Homomorphic Encryption*(HE), and *Secure multiparty computation* (SMC).

4.1. Differential privacy

DP was firstly proposed by Dwork C et al. [18] in 2006. It prevents information leakage by introducing a certain amount of noise into the original records, making an individual record lose its uniqueness, hidden in the datasets. In a nutshell, if the attacker has all the data records except the target data, differential privacy can still prevent the adversaries from obtaining the sensitive information of the target data.

Definition (Differential Privacy): let ε and δ be two positive real numbers and $\mathcal{A}$ be a randomized algorithm, S denotes any subset of the output of algorithm $\mathcal{A}$, given any adjacent datasets D and D' ,if $\mathcal{A}$ satisfies the following inequality:

$$\Pr[\mathcal{A}(D_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(D_2) \in S] + \delta \tag{3}$$

then $\mathcal{A}$ provides (ε, δ)-differential privacy. ε is a metric of privacy loss at a differentially change in data, it indicates the degree of privacy preservation. The smaller the value is, the better the privacy preservation. When ε approaches to 0, $\mathcal{A}$ converges in the probability distribution of outputting the same result on adjacent datasets D and D', which indicates $\mathcal{A}$ may leak less information. δ denotes the probability that the difference between the output results of $\mathcal{A}$ on adjacent datasets D and D'exceeds $\exp(\varepsilon)$. The ability of privacy preservation improves as δ decreases. If $\delta = 0$, we say that $\mathcal{A}$ is ε -differential privacy.

Typically the noise mechanisms of differential privacy algorithms are classified into exponential noise, Laplace noise, and Gaussian noise, where exponential noise is used for processing discrete datasets, while Laplace noise is used for processing continuous datasets.

In FL, privacy preservation methods implemented based on differential privacy techniques mainly include output perturbation [19], objective perturbation [20], and gradient perturbation [21]. As is shown in Fig.4, data holders will locally introduce a certain degree of noise to the model parameters, the central server then calculates the global model parameters on the perturbed parameters so that the attackers are prevented from obtaining the participant's local model parameters. Since differential privacy hides the individual record in the entire datasets by adding noise, making adjacent datasets difficult to distinguish, attackers may have difficulty in inferring whether a sample is a training sample. Therefore, differential privacy has a good defense effect against membership inference attacks.

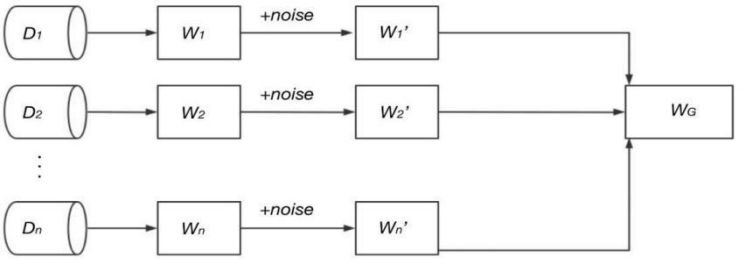

Figure 4. Secure aggregation based on differential privacy.

Although differential privacy has the characteristics of high computational efficiency and low communication overhead, introducing noise to the model parameters inevitably brings some accuracy loss to the global model. A hot research topic at the moment is how to lower the influence of noise on model training and improve model performance on the premise of preserving privacy information. Huang et al. [22] presented a DP-FL framework for unbalanced data between clients, which sets

DOI: 10.54254/2755-2721/6/20230330

different DP budgets ε based on the amount of data for each client. To improve performance, a novel DPAGD-CNN algorithm was developed to adaptively update training parameters for each user. Experimental results on several public datasets(e.g.MNIST) reveal that DP-FL has a better performance compared with the traditional FL framework under the unbalanced data scenario on the premise of preserving private information. Liu et al. [23] presented a sketch-based FL framework, which is promising in optimizing the trade-offs between performance and privacy. Sketch attains high accuracy with a concise data structure, it achieves ε-differential privacy while the amount of noise added to the sketch is significantly smaller than on the original parameters. Besides, as is proved by Nan Wu et al.[24], In FL, the difference between the fitness of the trained ML model using differential privacy gradient queries and the fitness of the model without any privacy concerns is inversely proportional to the square of the size of the training datasets and the square of privacy budgetε. Consequently, the usability of models that introduce differential privacy to process high-dimensional data needs to be further improved.

4.2. Homomorphic encryption

HE permits users to perform calculations on their encrypted data without having to decrypt it first. The plaintext obtained after HE is equivalent to the data obtained from the same calculation of the original plaintext data.

Definition (Homomorphic Encryption): Given plaintext data and,

If

$$\mathrm{Dec}\big(\mathrm{En}(\mathrm{m}_1) \odot \mathrm{En}(\mathrm{m}_2)\big) = \mathrm{Dec}\big(\mathrm{En}(\mathrm{m}_1 \oplus \mathrm{m}_2)\big) = \mathrm{m}_1 \oplus \mathrm{m}_2 \tag{4}$$

then we say the encryption function is homomorphic with respect to an operation $\oplus$ on the plaintext field, where Dec denotes the decryption operation, En denotes the encryption operation, and $\odot$ represents operation on the ciphertext fields.

In FL, HE is mainly used to preserve the gradients uploaded by data holders. By encrypting the local model parameters, the server can neither get the model parameters of the participants, nor the original data or the prediction results of the participants, thus protecting the data privacy in both the training stage and the prediction stage. Fig.5 depicts a typical FL framework that uses homomorphic encryption as the privacy preservation technique. Each data holder computes the local gradients and encrypts them with the public key. Common homomorphic encryption algorithms include Paillier encryption, RSA encryption, ElGamal encryption, etc. Then the encrypted local gradients are forwarded to the central server, where all the gradients are securely aggregated. Thereafter, the central server dispatches the aggregated gradients to all the data holders. Each data holder then decrypts the aggregated gradients with a private key and then updates the local model parameters.

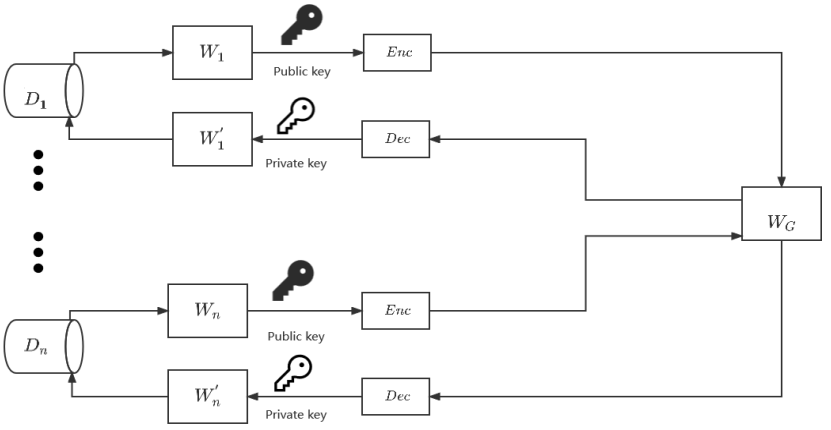

Figure 5. A FL framework based on HE.

DOI: 10.54254/2755-2721/6/20230330

Under the preservation of homomorphic encryption, all the computation in the aggregation process is performed on ciphertexts, which effectively averts the trained models being learned by external parties. Moreover, unlike differential privacy, homomorphic encryption hardly causes a loss of model accuracy since no noise is introduced to the uploaded gradients. Nevertheless, there is a significant computation overhead for FL systems that applies homomorphic encryption. It performs complex cryptographic operations such as exponentiation and modular multiplications on encrypted data to guarantee confidentiality, while they are extremely pricey to compute. To reduce the computation overhead caused by homomorphic encryption, Zhang et al. proposed a system for cross-silo FL named BatchCrypt [25], which is an efficient homomorphic encryption batching scheme. Each data holder in BatchCrypt quantizes the local gradients into low-bit integer representations and then encodes a batch of quantized values to a long integer, thus communication overhead is greatly reduced since the encryption is performed on batches. They also proposed a novel gradient clipping technique that allows them to choose optimal clipping thresholds with the minimum cumulative error. Experiments results on several widely used datasets (e.g.FMNIST, CIFAR10) show the overall network cost is reduced by an average of around 98% compared with stock FATE [26].

4.3. Secure multiparty computation

SMC enables multiple participants to collaboratively compute an objective function without a trusted third party, each party only obtains its own private calculation results, and cannot infer any additional information from any other participant through the calculation process. In FL, SMC can guarantee the privacy of all the participants in the process of exchanging gradients without compromising the accuracy of the model. At present, common SMC approaches include secret sharing [27], pairwise masking, and garbled circuit with oblivious transfer [28]. However, SMC relies on a large number of calculation on ciphertext and security proof, which seriously restricts the computing efficiency and burden the overall overhead. Besides, SMC-based privacy preservation approaches fails to preserve the privacy of the gradients in the model dispatching phase. In order to get rid of these weaknesses, Li et al. proposed a novel FL framework based on a chained SMC technique named Chain-PPFL [29]. To protect the gradients, each party adds a mask information to its gradients. The output of the current node is used by the descendent node as its mask information, which enables the gradients to be transferred between participants in a chained structure. Therefore, the malicious adversary cannot infer any information about the gradients from the participants, which implies Chain-PPFL is equivalent to DP with ε approaching zero. Experiments on datasets MINIST and CIFAR-100 indicate that the model accuracy is close to the FedAVG algorithm, while the computation and communication complexities of Chain-PPFL are much lower than the typical SMC based FL scheme. To reduce the communication overhead, XU et al. proposed an privacy preservation for FL named HybridAlpha [30], which defines a SMC protocol from a multi-input functional encryption scheme and applies a differential privacy mechanism to mitigate the risk of private information being inferred by an adversary. Experiments on the MINIST dataset show that HybridAlpha can lessen the training time by 68% and transfer volume by 92% against crypto-based SMC solutions while the data privacy is guaranteed. However, combing differential privacy with SMC will inevitably impair the model accuracy since noise are added to model parameters. KANAGAVELU proposed a Two-Phase MPC-enabled FL framework [31] which averts exchanging massive model parameters in the form of secret shares among all participants. The first phase uses peer-to-peer(P2P) MPC to elect a small array of participants as the model aggregation committee members. The second phase uses the MPC service provided by committee members to aggregate the local models in the form of secret shares of all the participants. Compared with the conventional P2P framework, Two-Phase MPC-enabled FL framework speeds up the execution time by 2 to 25 times both for common neural network models.

5. Conclusion

FL has made significant contributions to the resolution of data islands and data privacy issues in distributed machine learning. However, as the number of data holders involved in FL training tasks

DOI: 10.54254/2755-2721/6/20230330

grows, federal learning faces numerous challenges. This paper briefly introduces the workflow of a FL framework, and presents the communication overhead issues and privacy security problems faced by FL. Aiming at these two pain points, this paper puts forward several corresponding solutions and analyzes the advantages and disadvantages of each method.

Communication overhead occurs mostly as a result of the continuous growth in model parameters and instabilities in network transmissions. To lower the communication overhead, three effective approaches are proposed: increase the amount of computing on the edge side, model compression, and improve parallelism.

Increasing the amount of computing on the edge side can effectively reduce the communication overhead since more computations are performed locally, thus the overall iteration rounds are reduced. However, this approach achieves a poor optimization effect if the data is non-identically distributed.

Model compression mainly utilizes quantization and sparsification to lower the size of messages communicated at each round, therefore, the overall communication overhead is significantly reduced. Nevertheless, this approach will inevitably impair the final accuracy.

Improving parallelism can lower the communication overhead by reducing the uploading latency introduced by global model aggregation and the short plate effect caused by synchronization.

The privacy disclosure problem is mainly caused by the several attacks launched by malicious central servers and participants. This paper analyzes three effective privacy preservation approaches for the FL framework: DP, HE, SMC.

DP prevents information leakage by introducing a certain amount of noise to the raw data. This approach generally achieves privacy protection without burdening communication overhead. However, this approach will unavoidably bring some accuracy loss to the global model.

A homomorphic encryption-based FL framework can achieve high confidentiality on model parameters, making the central server and other data holders unable to infer any information about a specific participant. Although this approach does not affect the model accuracy, it burdens the computation overhead since complex cryptographic operations are required.

SMC-based FL framework can achieve the same effect as homomorphic encryption in terms of privacy preservation degree and complete the training tasks without a trusted third-party aggregation server, however, the costly communication overhead between gradients exchanging becomes the bottleneck of SMC-based solutions.

In addition to communication overhead and privacy issues, federated learning still faces many other challenges. Future work will focus on personalized FL, Asynchronous FL, Incentive mechanisms for FL, etc.

References

[1] Yang Qiang. AI and data privacy protection: the way to crack federal learning [J]. Information security research, 2019,5 (11): 961-965

[2] KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated learning: strategies for improving communication efficiency[J]. arXiv preprint, 2016, arXiv:1610.05492.

[3] J. Konecnˇ y, H. B. McMahan, D. Ramage, and P. Richt ` arik, “Federated optimization: Distributed machine learning for ´ on-device intelligence,” arXiv preprint arXiv:1610.02527, 2016.

[4] Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Foundations of Secure Computation, 1978, 4(11): 169−180.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS ’17). ACM, New York, NY, USA, 1175–1191.

[6] KONEČNÝ J, MCMAHAN H B, YU F X, et al. Federated learning: strategies for improving communication efficiency[J]. arXiv pre-print arXiv:1610.05492, 2016.

DOI: 10.54254/2755-2721/6/20230330

[7] Liu, Y. , Kang, Y. , X Zhang, Li, L. , Cheng, Y. , & Chen, T. , et al. (2019). A communication efficient vertical federated learning framework.
[8] CHEN J, PAN X, MONGA R, et al. Revisiting distributed synchronous SGD[J]. arXiv preprint arXiv:1604.00981, 2016.
[9] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.
[10] SAHU A K, LI T, SANJABI M, et al. Federated optimization for heterogeneous networks[J]. arXiv preprint, 2018, arXiv:1812.06127.
[11] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, “ATOMO: Communication-efficient learning via atomic sparsification,” in Proc. Advances in Neural Information Processing Systems, 2018, pp. 1–12.
[12] SHI D, LI L, CHEN R, et al. Towards Energy Efficient Federated Learning over 5G+ Mobile Devices [J]. IEEE Signal Processing Magazine, 2020, 37 (3), 50-60.
[13] SHI W, ZHOU S, NIU Z, et al. Joint device scheduling and resource allocation for latency constrained wireless federated learning[J]. IEEE Transactions on Wireless Communications, 2020, 20(1): 453-467.
[14] Zhou, Y., Ye, Q., & Lv, J..(2020). Communication-efficient federated learning with compensated overlap-fedavg.
[15] Z. Yang, W. Bao, D. Yuan, N. H. Tran, and A. Y. Zomaya, “Federated learning with nesterov accelerated gradient momentum method,” 2020.
[16] ZHU L，LIU Z，HAN S． Deep leakage from gradients [c]. International Conference on Advances in Neural Information Processing Systems． 2019: 14774 － 14784．
[17] SHOKRI R, STRONATI M, SONG C Z, et al. Membership inference attacks against machine learning models [C] //S&P 2017: 2017 IEEE
[18] Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Proc. of the Theory of Cryptography Conf. Berlin, Heidelberg: Springer, 2006. 265−284.
[19] Nissim K. Private data analysis via output perturbation[M]//Privacy-Preserving Data Mining. Springer, Boston, MA, 2008: 383-414.
[20] E. Nozari, P. Tallapragada and J. Cortés, "Differentially private distributed convex optimization via objective perturbation," 2016 American Control Conference (ACC), 2016, pp. 2061-2066, doi: 10.1109/ACC.2016.7525222.
[21] Yu D, Zhang H, Chen W, Liu TY, Yin J. Gradient perturbation is underrated for differentially private convex optimization. arXiv preprint arXiv:1911.11363, 2019.
[22] Huang, X., Ding, Y., Jiang, Z.L. et al. DP-FL: a novel differentially private federated learning framework for the unbalanced data. World Wide Web 23, 2529–2545 (2020). https://doi.org/10.1007/s11280-020-00780-4
[23] LIU Z X, LI T, SMITH V, et al. Enhancing the privacy of federated learning with sketching [EB/OL]. [2022-01-19].
[24] N. Wu, F. Farokhi, D. Smith and M. A. Kaafar, "The Value of Collaboration in Convex Machine Learning with Differential Privacy," 2020 IEEE Symposium on Security and Privacy (SP), 2020, pp. 304-317, doi: 10.1109/SP40000.2020.00025.
[25] Zhang C, Li S, Xia J, et al. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning[C]//2020 USENIX annual technical conference (USENIX ATC 20). 2020: 493-506.
[26] FATE (Federated AI Technology Enabler). https://github.com/FederatedAI/FATE, 2019.
[27] RIVEST R L, SHAMIR A, TAUMAN Y. How to share a secret[C]//Communications of the ACM. New York: ACM, 1979, 22(11): 612-613.
[28] YAO A C. Protocols for secure computations[C]//23rd annual symposium on foundations of computer science. Piscataway: IEEE, 1982: 160-164.
[29] Li Y, Zhou Y, Jolfaei A, et al. Privacy-preserving federated learning framework based on chained

DOI: 10.54254/2755-2721/6/20230330

secure multiparty computing[J]. IEEE Internet of Things Journal, 2020, 8(8): 6178-6186.
[30] XU R H, BARACALDO N, ZHOU Y, et al. Hybridalpha: an efficient approach for privacy-preserving federated learning[C]//Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. New York: ACM, 2019: 13-23.
[31] KANAGAVELU R, LI Z, SAMSUDIN J, et al. Two-phase multi-party computation enabled privacy-preserving federated learning[C]//2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing. Piscataway: IEEE, 2020: 410-419.