ISBN: 978-981-18-1791-5
2021 the 11th International Workshop on Computer Science and Engineering (WCSE 2021)
doi: 10.18178/wcse.2021.06.006

Adaptive Privacy-preserving and Shuffling Aggregation in Federated-Learning

He Huixian [+], Cao Zhenfu

East China Normal University, China

Abstract. Deep learning models are usually trained on data sets containing sensitive information, such as personal shopping transactions, personal contacts, and medical records. Therefore, more and more important work attempts to train neural networks subject to privacy constraints, which are specified by differential privacy or divergence-based relaxation. However, these privacy definitions have weaknesses in handling certain important primitives (synthesis and sub-sampling), which makes the privacy analysis of training neural networks loose or complex. Federated learning is a popular privacy protection method, which collects local gradient information instead of real data. One way to achieve strict privacy guarantee is to apply differential privacy to federated learning. However, previous work did not give a practical solution. This paper proposes a new type of adaptive privacy-preserving and shuffling aggregation in federated-learning mechanism design. It can make the data more different from its original value and introduce lower variance. In addition, the proposed mechanism is updated through the split and shuffle model, avoiding the curse of dimensionality. A series of empirical evaluations conducted on the three commonly used data sets of MNIST, Fashi-MNIST and CIFAR-10 show that our solution can not only achieve excellent deep learning performance, but also provide strong privacy protection.

Keywords: federated learning, privacy preserving

1. Introduction

With the brilliant success of AlphaGo, big data-driven artificial intelligence (AI) is expected to soon be applied to all aspects of our daily lives [1-3].When integrating artificial intelligence into various IoT applications, distributed machine learning (ML) is very effective for many data processing tasks by defining a parameterised function from input to output as a composition of the basic building blocks. The latest advancement in distributed ML is presented in the form of Federated Learning (FL). In this approach, data is retrieved and processed locally on the client side, and updated ML parameters are sent to a central server for aggregation. Federated learning is a distributed machine learning approach that trains a lossless learning model based on local training and parameter passing by the participants without direct access to the data source.

A prominent advantage of FL is that it enables local training without any exchange of personal data between the server and the client, thus preventing the client's data from being eavesdropped on by hidden adversaries. Despite the obvious advantages of federation learning and its development in line with the times, its security should be tested before it is put into practice. In recent years, numerous research results have shown that there are still security issues in the federated learning mechanism, where an attacker can still compromise some personal data by analysing the differences between the relevant parameters trained and uploaded by the client, such as the weights of the neural network training. There are also poisoning attacks, counter-sample attacks, etc.Some existing approaches to prevent information leakage on federation learning include the addition of artificial noise, differential privacy techniques. Existing works based on differential privacy algorithms include local differential privacy [4], distributed SGD algorithms based on differential privacy, and DP meta-learning.

There are several obvious problems and challenges in the previous approach. First, noisy data approaches its original value with high probability and does not substantially reduce the risk of information exposure [5]. Secondly, the estimated mean introduces a large variance, leading to poor accuracy. Therefore,

[+] Corresponding author. Tel.: + 18157970561;
E-mail address: 271401697@qq.com.

more communication between the cloud and the client is required to achieve convergence of the algorithm. Thirdly, the privacy budget explodes due to the high dimensionality of the weights in the deep learning model. Finally, to the best of our knowledge, existing works have not demonstrated excellent deep learning performance on popular datasets. In this paper, we propose a new neural network perturbation mechanism to address the above issues. Predictive probability- based data perturbation on the neural network in the model and local model weight splitting and shuffling are applied to a typical federated learning system. Our main contributions are manifold. First, we propose a new perturbation mechanism for neural networks and show how it can be applied to federated learning. By making the perturbed data clearer than the original values, we demonstrate that the ROIT mechanism is more secure than existing mechanisms and significantly reduces the risk of information leakage. Second, we apply splitting and shuffling to each client's gradient to mitigate privacy leaks caused by high-dimensional data and multiple query iterations. Third, we demonstrate that our solution is able to present better model training performance with less variance over a mountain of average computation. Finally, we evaluate ROIT-FL on three datasets commonly used in our previous work, namely MNIST [6], Fashion-MNIST and CIFAR-10, respectively. The proposed mechanism achieves a privacy budget of = 1 with an accuracy loss of 0.97% on MNIST, = 4 with an accuracy loss of 1.32% on Fashion-MNIST, and = 10 with an accuracy loss of 1.09% on CIFAR-10.

2. Preliminary

In this section, we review the concept of federated learning, differential privacy and layer-wise relevance propagation algorithm, which serve as the underlying structure of our ROIT.

2.1. Federated Learning

Suppose there are n participants and client Ci has local dataset Di, i belongs to 1, 2...n. Now it is necessary to train the model M_{global} in the total dataset D1 D2 ... Dn. Federated learning refers to a distributed learning approach, i.e., it does not directly integrate all the data together training to get the model M_{sum}, but instead each participant trains the local data to get new parameters based on the initial parameters passed from the server. On the server side, its goal is to train a global model on N aggregated datasets. An active client, training the model on the local dataset minimizes a particular loss function and gets the corresponding weights. The server then collects the weights from the N clients and aggregates them:

$$w = \sum_{i=1}^{N} p_i w_i \tag{1}$$

where wi is the training vector of the i-th client and w is the vector of parameters after client aggregation. Such an optimization problem can be formulated as

$$w^* = \arg\min_{w} \sum_{i=1}^{N} p_i F_i(w) \tag{2}$$

3. Our Approach

3.1. Overview

In this section, we introduce the federated learning approach with ROIT consists of two steps, as shown in Algorithm 1.

Cloud Update: First, the cloud initializes the weights randomly at the beginning. Let n be the total number of local clients. Then, in the r-th communication round, the cloud will randomly select $n_r \leqslant n$ clients to update their weights for local-side optimization. Unlike the previous works where they assume that the aggregator already knows the identities (e.g., IP addresses) of the users but not their private data, our approach assumes the client remains anonymous to the cloud. For example, the client can leverage a changing IP address or the same IP address for all clients to send the local weights back to the cloud. This approach can provide us more robust privacy bound and practical solution and more details in section 3.2.

Local Update: For each client, it contains its own private dataset. In each communication, the selected local clients will update their local model by the weight from the cloud. Next, the local model uses

Stochastic Gradient Descent (SGD) to optimize the distinct local models' weights in parallel. In order to provide a practical privacy protection approach, we split and shuffle each local model's weights and send each weight through an anonymous mechanism to the cloud. In this case, we can provide more reliable and give a practical solution with available results in the final.

3.2. Randomized Privacy-preserving Adjustment Technology

The following diagram shows the transformation process for each hidden neuron in the training model:

$$y = a(x * \omega + b) \tag{3}$$

Here, x is the input vector, y is the output, b and () is the activation function used to combine linear and non-linear transformations. $z(\omega) = x * \omega + b$ is the linear transformation part.

$\mathcal{M}$	A randomized mechanism for DP
x, x'	Adjacent databases
ϵ, δ	The parameters related to DP
$\mathcal{C}_i$	The i -th client
$\mathcal{D}_i$	The database held by the owner $\mathcal{C}_i$
$\mathcal{D}$	The database held by all the clients
$\lvert \cdot \rvert$	The cardinality of a set
N	Total number of all clients
K	The number of chosen clients $(1 < K < N)$
t	The index of the t -th aggregation
T	The number of aggregation times
$\mathbf{w}$	The vector of model parameters
$F(\mathbf{w})$	Global loss function
$F_i(\mathbf{w})$	Local loss function from the i -th client
μ	A presetting constant of the proximal term
$\mathbf{w}_i^{(t)}$	Local uploading parameters of the i -th client
$\mathbf{w}^{(0)}$	Initial parameters of the global model
$\mathbf{w}^{(t)}$	Global parameters generated from all local parameters at the t -th aggregation
$\mathbf{w}^*$	True optimal model parameters that minimize $F(\mathbf{w})$
$\widetilde{\mathbf{W}}$	The set of all local parameters with pertubation

Algorithm 1 ROIT

Data: $T, \mathbf{w}^{(0)}, \mu, \epsilon$ and δ
Initialization: $t = 1$ and $\mathbf{w}_i^{(0)} = \mathbf{w}^{(0)}, \forall i$
while $t \leq T$ **do**
  Local training process:
  while $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ **do**
    Update the local parameters $\mathbf{w}_i^{(t)}$ as

$$\mathbf{w}_i^{(t)} = \arg\min_{\mathbf{w}_i} \left(F_i(\mathbf{w}_i) + \frac{\mu}{2} \left\| \mathbf{w}_i - \mathbf{w}^{(t-1)} \right\|^2 \right)$$

    Add noise and upload parameters

$$\tilde{\mathbf{w}}_i^{(t)} = \mathbf{w}_i^{(t)} + \frac{1}{|D_i^t|} \mathrm{Lap}\left(\frac{GS_l}{\epsilon_j} \right)$$

  end while
  SplitShuffle: $\mathbf{W} \leftarrow \mathbf{W} \cup$ Split Shuffle $\left(\tilde{\mathbf{w}}_i^{(t)} \right)$
  Model aggregating
end while

Since in the neural network structure, the output of the previous layer is the input of the next layer, we can obtain that the original data are only used by the linear transformation of the first hidden layer. Intuitively speaking, in order to obtain a learning model with privacy protection, we can inject noise into the data of the first hidden layer [7]. A traditional linear transformation method consists of injecting noise with the same confidentiality budget into the original data, and the improved version consists of injecting noise with different confidentiality budgets. But our work is more competitive.

We creatively propose a Randomized Noise-Injection Technology (ROIT), which can improve the accuracy and availability of the system. In particular, we introduce two adjustment factors: f and p (f [0, 1], p [0, 1]) [1] where f represents a threshold to decide whether the contribution of the attribute to the output is high or low, whose value is defined by users. i.e., the attribute classes, whose contributions which exceeds threshold f , have a greater contribution to output. Then, we inject adaptive Laplace noise to all these attributes. While the contribution is lower than threshold f , a probability selection is made for such attributes. i.e., we choose the original data with probability 1 p, and to inject adaptive Laplace noise to some attributes

$$\tilde{x}_{i,j} = \begin{cases} \ddot{x}_{i,j} \beta \geq f \\ \bar{x}_{i,j} \beta < f \end{cases} \tag{4}$$

with probability p. The formula is as follows:

where represents the ratio of contribution: $\beta = \frac{|\ddot{c}_j|}{\sum_{j=1}^{u} |\ddot{c}_j|}$.

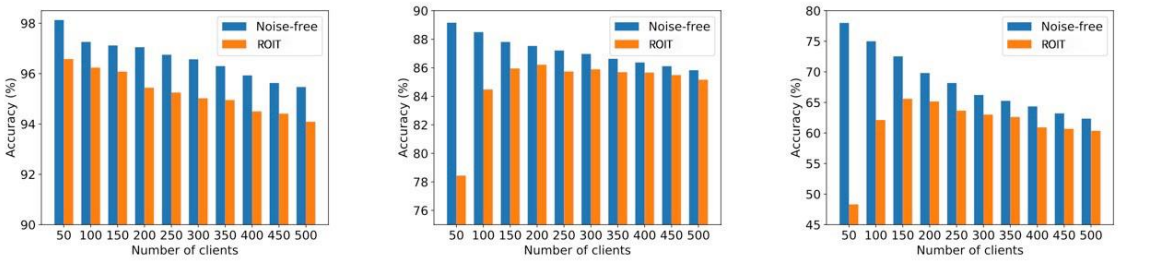
Algorithm 2 Data Perturbation	Algorithm 3 Split&Shuttle
Input: Original w	Input: Perturbed w after Algorithm1
Output: w^* after data perturbation	Output: w^* after split and shuffle
for each weight $p \in w$ do	for each weight $p \in w$ do and its id $(p, id) \in w$
if $\beta \geq f$ then	$t \leftarrow$ Random Sample a Time;
$x'_{i,j} = x_{i,j} + \frac{1}{\lvert D_i^l \rvert} \mathrm{Lap}\left(\frac{GS_l}{\epsilon_j}\right)$	$w^* \leftarrow w^* \cup (p, id, t)$
else	end for
$x'_{i,j} = \ddot{x}_{i,j} \beta \geq f$ replace p as $p*$	return w^*
end if	
end for	
After perturb all $p \in w$, then we have $w \rightarrow w^*$	
return w^*	

3.3. Split & Shuffle

There are two parts to the shuffle mechanism: split and shuffle. The main goal is to enhance privacy protection while using data perturbation in coalition learning. Split breaks the private connection weight of each local client model, while shuffle breaks the privacy of communication between the local client and the cloud. In order to better shuffle the weights, in Algorithm 3, each client will also randomly sample and send each weight to the cloud. Our method is to first divide the weight of each local model by each model. Then, each weight is shuffled through the client anonymity mechanism, and the id of each weight is sent to the cloud [8]. Finally, split and shuffle allows the cloud to ensure that the correct weight values are collected to update the central model without having to know the relationship between each weight and the local client.

We use split and shuffle to bypass the curse of dimensionality. Since the weights are split and uploaded anonymously, the cloud cannot associate different weight values of the same client, and therefore cannot infer more information about a certain client. Therefore, ROIT is sufficient for each weight. Also, due to anonymity, the cloud cannot connect weights from the same client in different iterations. Without splitting and shuffling, the privacy budget will grow to T d, where T is the number of iterations and d is the number of weights in the model.

4. Experiments



(a) Effect of n on MNIST ($\epsilon = 1$) (b) Effect of n on Fashion-MNIST ($\epsilon = 10$) (c) Effect of n on CIFAR ($\epsilon = 10$)

Fig. 1: Comparison of accuracy on MINIST and CIFAR.

In this section, the effectiveness of ROIT is evaluated using an image classification task as experimental examples. First, the effect of different weights is verified using MNIST and Fashion-MNIST image reference data, followed by performance validation using CIFAR-10 and three previous datasets. Here, for MNIST [9] and Fashion-MNIST, (c, r) = (0, 0.075) and r = 0.015.For CIFAR-10, due to the complexity of the network, c = 0 and r is set within the range of weights for each layer. the learning rate is 0.03 for MNIST and 0.015 for CIFAR-10. taking into account the randomness when perturbing, the test experiment were run 10 times independently to obtain the average value. In order to explore the impact of the specific value of the privacy level on the quality of the images, we have conducted several experiments on datasets. In these experiments, we trained by setting different privacy parameters and got several models of privacy protection levels. The generated images are shown in Fig. 1, Fig. 2 and Fig. 3, corresponding to different levels of privacy parameters. It can be seen that we can generate clear images when the privacy level is high. And large

privacy parameters correspond to high-quality images, which indicates the distortion of the image is caused by noise rather than a poor quality training set. According to [10], large privacy parameter means great risk of privacy breaches, but it also means better generated data. This is a trade-off between privacy and performance.

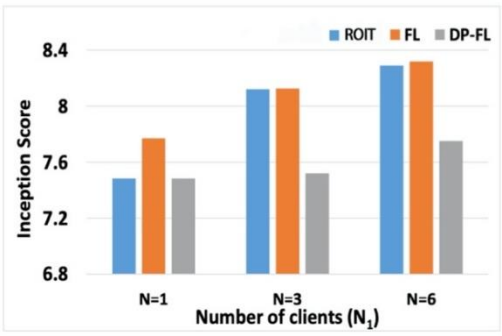


Fig. 2: Comparison of IS on MNIST dataset

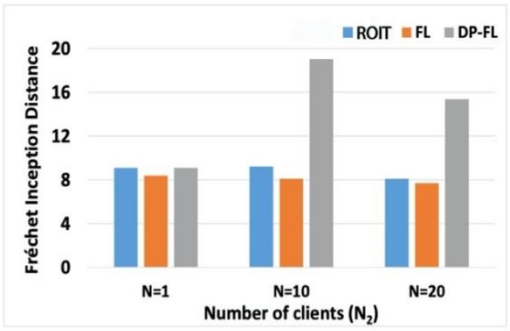


Fig. 3: Comparison of FID on CelebA dataset

5. Conclusion and Future Plan

In this paper we propose a new mechanism for ROIT and show how it can be applied to protect coalition learning, applying partitioning and shuffling to each client's gradient to mitigate the privacy degradation caused by large data sizes and many query iterations. Empirical studies show that our system performs better than previous related work on the same image classification task. This is expected to greatly accelerate the practical application of NOIT in collaborative learning. In the future, several research questions can be explored, such as preventing client anonymisation from side-channel attacks, improving data perturbation mechanisms, and applications to natural language processing, speech recognition, and graph learning. Furthermore, it is very important to generalise the proposed privacy protection techniques to other scenarios.

6. References

[1] H. Lee, S. H. Lee, and T. Q. S. Quek. Deep learning for distributed optimization: Applications to wireless resource management. IEEE Journal on Selected Areas in Communications, 37(10):2251-2266, 2019.

[2] J. Li, S. Chu, F. Shu, J. Wu, and D. N. K. Jayakody. Contract-based small-cell caching for data disseminations in ultra-dense cellular networks. IEEE Transactions on Mobile Computing, 18(5):1042-1053, 2019.

[3] Z. Ma, M. Xiao, Y. Xiao, Z. Pang, H. V. Poor, and B. Vucetic. High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies. IEEE Internet of Things Journal, 6(5):7946-7970, 2019.

[4] U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. page 1054-1067, 2014.

[5] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. Journal of Machine Learning Research, 12(29):1069-1109, 2011.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, pages 265-284. Springer, 2006.

[7] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4):211-407, 2014.

[8] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963, 2017.

[9] Y.Mizukami,K.Tadamura,J.Warrell,P.Li,andS.Prince.Cudaimplementationofdeformablepatternrecognitionanditsapplication to mnist handwritten digit database. In 2010 20th International Conference on Pattern Recognition, pages 2001-2004. IEEE, 2010.

[10] M. A. Pathak, S. Rane, and B. Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In NIPS, pages 1876-1884. Citeseer, 2010.