Local Differential Privacy Preservation in Split Learning: Random Response or Laplace Mechanism?

Anonymous Author(s)

ABSTRACT

Deep learning has gained wide popularity in recent years, and it shows great performance in many tasks. Traditionally, to achieve great performance in deep learning tasks, we need to gather decentralized data in a central site to train a model with massive amount of data. However, to gather data from decentralized data owners may raise serious data privacy issues. Split learning, as a distributed collaborative learning approach was proposed to address these issues while local differential privacy mechanisms were widely adopted to provide theoretical guarantee for preserving data privacy. In this paper, we provide a systematic study of the use of local differential privacy mechanisms in split learning. We specifically focus on two local differential privacy mechanisms, laplace mechanism[1] and random response[13], which are both widely used local differential privacy mechanisms. By conducting extensive experiments, we show that random response mechanism can provide better utilities under the same privacy guarantee. Meanwhile, random response shows less computation time and lower communication complexity when used in split learning. Finally, we use two common attacks against deep learning models to show that random response can better preserve the privacy of the data owner in face of possible attacks while maintaining high utilities in comparison with laplace mechanism.

CCS CONCEPTS

• **Security and privacy** → **Data anonymization and sanitization**; **Privacy protections**;

KEYWORDS

local differential privacy, laplace mechanism, random response, split learning, federated learning

1 INTRODUCTION

Deep Neural Networks (DNN) have been successfully applied to a wide range of areas including computer vision, speech recognition, and natural language processing [3][7][11]. The superior prediction ability of DNN models relies on large amounts of data. Meanwhile, with the increase of mobile and IoT technology, tremendous valuable data are generated by edge devices but are isolated from each other. There is an increasing demand to learn from the dispersed data, so as to better support deep learning (DL) tasks at the edge. However, in the most conventional training paradigm, models are placed at a central site. It requires collecting training data from users, which raises concerns about data privacy and violates data protection laws. As a result, privacy issues turn to be a barrier for empowering edge intelligence, and collaborative training without sharing the input training data is highly desired. Recently, split learning[12], as a new learning paradigm, has emerged to address this issue. Split learning trains the first few layers of the neural network at the edge device and transmit the intermediate features to cloud servers with abundant computational resources to facilitate the rest of the training. In this paper, we adopt a modified version of split learning. We use a technique commonly used in transfer learning[14], that is, we deploy the first few layers of a pretrained model trained on public datasets from a similar domain on edge devices and freeze the parameters. Then the edge devices transmit the intermediate features to cloud servers and the cloud server finishes the rest of the training. Although exposing the intermediate features instead of the training data is assumed to be safer, but recent study[8] proposed an inversion attack to recover the inputs from the intermediate features without the need to know the parameters or structure of the model at the edge. Meanwhile, exposing the intermediate features fail to provide quantitative measure of privacy leakage. In this paper, we adopt local differential privacy[2] mechanism in split learning, which provides better privacy guarantee to the data owners in theory and practice. By applying local differential privacy mechanism on the intermediate output, we can reduce the privacy leakage with theoretical privacy guarantee and make the inversion attack[8] invalid without sacrificing the performance of our model. We compare two most commonly used local differential private mechanisms, laplace mechanism[1] and random response[13]. We show that in the split learning setting, random response can better preserve the privacy of data owners while losing less utility. Moreover, random response mechanism will greatly save communication cost and computation time. We summarize our contributions as follows.

- Adopt the idea from transfer learning, and propose a modified version of split learning.
- Apply local differential privacy mechanism in split learning, and preserve the privacy of data owners in theory and practice.
- Compare the performance of two local differential privacy mechanism in multiple ways, and show that random response outperforms laplace mechanism in many aspects.

2 BACKGROUND

2.1 Split Learning

Split learning[12] was proposed as a collaborative training approach. The basic idea is to partition a neural network between the cloud and the edge. A couple of the shadow layers are loaded on the edge devices and the deep layers which have heavy parameters are deployed on the cloud server with supreme computing power. During the forward process, The edge device feeds the training data into the network upto the partition layer and sends the intermediate feature map of the partition layer to the cloud server. The cloud server gets the feature map and use the feature map as input to train the remaining layers. In the backward process, the cloud server passes the gradients of the intermediate layer to the edge device and

the edge device uses the gradients to update local model parameters. Split learning can preserve the privacy of data owners (usually the edge devices) to some extent because only the intermediate feature map of the partition layer is sent to the cloud server instead of the raw data which may contain sensitive information related to the data owner. However, split learning can not provide any theoretical guarantee of privacy preserve, and the possible evil cloud server or the evil third party who can get the intermediate feature map can still get information about the raw data with the method used in [8]

2.2 Local Differential Privacy

Local differential privacy is a statistical definition of privacy with quantitative measure of privacy[2]. It is used to collect data from decentralized data owners while preserving the privacy of data owners. It assumes that the data collector can be curious about the sensitive information of data owners, so the data owners add noise to their own data and send the noisy data to the data collector to reduce the privacy leakage of their data. Mathematically, the local differential privacy mechanism is defined as follows.

Definition 2.1 (ϵ-LDP). A random mechanism $\pi : \mathcal{D} \rightarrow \mathcal{Y}$ satisfies ϵ-local differential privacy, where $\epsilon \geq 0$, if and only if for any inputs $d, d' \in \mathcal{D}$ and $y \in \mathcal{Y}$, we have $Pr[\pi(d) = y] \leq e^{\epsilon} Pr[\pi(d') = y]$.

The philosophy behind local differential privacy can be illustrated as follows: The curious data collector is more unlikely to distinguish two different data samples from different data owners with theoretical guarantee. We then introduce two local differential private mechanisms. The Laplace Mechanism[1] is defined as $\mathcal{L}_f = \pi(d) + Lap(\frac{\Delta\pi}{\epsilon})$, where $Lap(\frac{\Delta\pi}{\epsilon})$ is a random variable sampled from Laplace distribution with scale $\frac{\Delta\pi}{\epsilon}$. $\Delta\pi$ is called global sensitivity, which is the maximal value of $\|\pi(d) - \pi(d')\|$ among any pair of d and d'. Random response[13] is the another widely known local differential privacy mechanism. It originates from a survey technique when the survey questions may ask for sensitive or private information related to the people surveyed. The people surveyed choose to fill in the correct answer with probability p and choose the opposite answer with probability $1 - p$. The mechanism satisfies ϵ-LDP when $p = \frac{e^{\epsilon}}{1+\epsilon}$.

2.3 Attacks against ML Models

We focus on two common attacks against DL models which are closely related to split learning and may make use of privacy leakage to steal the private information of data owners. Feature inversion attack[8] is recently devised for the edge-cloud collaborative learning system. The adversary aims to recover the input raw data instances from the intermediate output of the partition layer on the edge. The adversary is assumed to know the structure and parameters of the model on the edge, and performs gradient descent to approximate the input raw data. Membership inference attack[4] tries to infer whether a given sample is used to train a model or not. Basically, the attacker first trains multiple "shadow" models to imitate the behaviours of the target model and then trains a binary attack model with the labeled inputs and outputs of the shadow model. Given the prediction output query from the target model as input, the binary attack model infers whether the data queried is a member of the target model's training dataset or not.

3 MODEL

3.1 Workflow

In this section, we will detail the workflow of the use of local differential privacy mechanisms in split learning model. To facilitate the training process and improve the performance, we adopt the commonly used method in transfer learning. We load the parameters of model pretrained on large public datasets on the edge device and freeze the parameters of the model on the edge device during the training process. The insights behind the method are that features from the shadow layers are more general than the deeper layers. So it is more flexible to adapt to a wider range of related datasets and tasks. The shadow layers of the model pretrained on a similar application domain can also perform well. With the fixed model parameters on the edge devices, edge devices transform the input training data into feature representations in parallel. The edge devices do not need to repeatedly send the feature representations of the same data samples, nor receive backward gradients to update the model. The edge devices then apply the local differential privacy mechanisms (laplace mechanism or random response) on the intermediate feature representations. With the noisy intermediate output sent by the edge devices, the cloud server label the intermediate ouput as input and the ground truth target as output to train a classifier. The cloud server doesn't send the backward gradients to the device, and only updates the model on the cloud.

3.2 Laplace Mechanism Representations

In this section, we will show how to apply laplace mechanism to protect sensitive information of the feature representations from edge devices. We denote the intermediate feature map as a vector with r elements for simplicity, though the intermediate fearure map may be a tensor in practice. We denote the feature map as $A = (A_1, A_2, ..., A_r)$. To control the global sensitivity $\Delta = \max_{i,j} \|A_i - A_j\|$, we need to trunctae the feature map to the range $(-\frac{\Delta}{2}, \frac{\Delta}{2})$. We define $B = (B_1, B_2, ..., B_r)$ as follows.

$$B_i = \begin{cases} A_i & \text{if } \frac{\Delta}{2} > A_i > -\frac{\Delta}{2}; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We then apply laplace mechanism to B. The local differentially private representations is denoted as $\tilde{A}$, which can be defined as $\tilde{A}_i = B_i + Lap(\frac{\Delta}{\epsilon})$, and $Lap(\frac{\Delta}{\epsilon})$ is laplacian noise sampled from laplace distribution with scale $\frac{\Delta}{\epsilon}$. We have that each $\tilde{A}_i$ satisfies ϵ-LDP, and according to the composition theorem of LDP, the entire vector $\tilde{A}$ satisfies $r\epsilon$-LDP.

3.3 Random Response Representations

In this section, we will detail how to use random response to protect sensitive information of the feature representations from edge devices. We denote the feature map as $A = (A_1, A_2, ..., A_r)$. And we then binarize the feature map and get output $B = (B_1, B_2, ...B_r)$. The binarize process is shown as follows.

$$B_i = \begin{cases} 1 & \text{if } A_i > 0; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We then apply random response on B to hide sensitive information. The local differentially private representation is denoted as $\tilde{A}$. And it can be computed as follows.

$$\tilde{A}_i = \begin{cases} B_i & \text{with probability } p; \\ B_i \oplus 1 & \text{otherwise.} \end{cases} \quad (3)$$

The edge devices then send the locally differential private representations $\tilde{A}$ to the cloud server. We set $p = \frac{e^\epsilon}{e^\epsilon + 1}$. From the definition of local differential privacy we can have that $\tilde{A}_i$ satisfies ϵ-LDP. From the composition theorem of LDP, the vector $\tilde{A}$ satisfies $r\epsilon$-LDP.

4 EVALUATION

4.1 Experiment Settings

We evaluate the local differential privacy mechanisms in split learning on image classification tasks. We conduct our experiment on two datasets, CIFAR10[9] and SVHN[10]. CIFAR-10 has 10 classes and contains 60,000 32x32 color pixel images with 3 RGB channels (50,000 training images and 10,000 testing images). SVHN is also composed of 32x32 images, including 73,257 training samples and 26,032 testing samples. We use ImageNet32x32 images extracted from CINIC dataset[5] to pretrain the model and load the first few layers onto the split learning model on edge devices. CINIC dataset are part of the original ImageNet[6] images under-sampled from 224x224 to 32x32 resolution with the Box algorithm from the Pillow Python library2. These Imagenet32x32 images have the same 10 classes as CIFAR-10 (the number of images for train/validation/test is 70,000/70,000/70,000 respectively) but do not include any image in neither CIFAR-10 dataset nor SVHN dataset. We choose VGG-16[11] and ResNet-18[7], which are typical deep learning models, to evaluate the local differential privacy mechanisms.

4.2 Utilities Under The Same ϵ

In this section, we detail the utilities of laplace mechanism and random response in deep split learning under the same privacy budget ϵ. We choose the basic block (including 2 convolution layers) as the unit to partition ResNet-18 and partition VGG-16 after a specific layer where the feature map size changes. In practice we may partition at different layers according to the capacity of edge devices, So we partition at different layers from shadow to deep to systematically compare the utilities of laplace mechanism and random response. We measure the utilities of local differential privacy mechanism based on the test accuracy of our split learning model after 400 epochs of training. As shown in Figure 1-4, on both datasets and both models, random response outperforms laplace mechanism, which means that random response will better preserve the utilities of split learning model. We also find that in Table 1 by cutting at an early layer and setting $\epsilon = 2$, the split learning model with random response mechanism can achieve almost as high accuracy as the model without adding noises, which means that our model with random response can be very useful in practice.

4.3 Computation Time Analysis

In this section, we will show the computation time comparison between laplace mechanism and random response. The computation time of laplace mechanism can be split into three steps: truncate

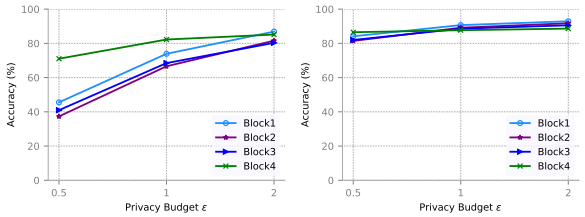

Figure 1: Comparison of laplace mechanism (left) and random response (right) in ResNet-18, CIFAR10

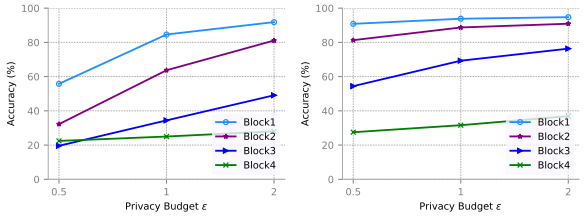

Figure 2: Comparison of laplace mechanism (left) and random response (right) in ResNet-18, SVHN

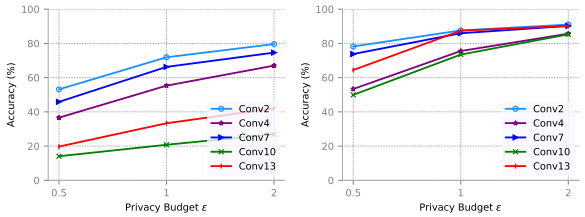

Figure 3: Comparison of laplace mechanism (left) and random response (right) in VGG16, CIFAR10

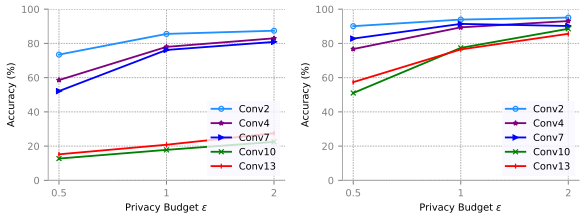

Figure 4: Comparison of laplace mechanism (left) and random response (right) in VGG16, SVHN

Table 1: Random response utility (measure by test accuracy) in split learning when $\epsilon = 2$ and cut at an ealy layer

Model,Dataset	$\epsilon = 2$,cut early	Without LDP
ResNet-18,CIFAR10	92.91%	94.86%
ResNet-18.SVHN	94.73%	95.57%
VGG-16,CIFAR10	90.96%	92.12%
VGG-16,SVHN	94.97%	95.42%

Table 2: Comparison of the computation time of laplace mechanism and random response.

Method	Random Response	Laplace Mechanism
Step1(s)	0.2241(57%)	0.1170(17%)
Step2(s)	0.1367(35%)	0.5660(82%)
Step3(s)	0.0351(8%)	0.0080(1%)
Total(s)	0.3958	0.6911

the feature map, generate the laplacian noise and adding the noise to the feature map. Similarly, the computation time of random response can be split into three steps: binarize the feature map, generate noise with bernoulli distribution with parameter p, add noise to the feature map via the $\oplus$ operation. We evaluate the computation time by applying local differential privacy mechanism on 1000 random vectors (similar to the total batches) with 10000 elements (similar to the size of feature map per batch) to compare the total time. The results are shown in Table2. The reasons behind the long computation time of generating the laplacian noise can be described as follows. The basic techinique for generating random numbers satisfying a given distribution $F(x)$ is to first generate random numbers u satisfying uniform distribution U. Then we can have that $X = F^{-1}(U)$ satisfies the given distribution $F(x)$. As the inverse function of Laplace distribution function is logarithmic, it is more computationally expensive than generating bernoulli distribution. Moreover, the third step of random response can be implemented in the hardware level efficiently, which may make the computation faster.

4.4 Communication Comlexity Analysis

In this section, we will evaluate the communication complexity of using local differential privacy mechanism in split learning. In our approach, the intermediate feature map only needs to be transferred once. Assuming there are n data samples while the intermediate feature map has r elements. In the case of laplace mechanism, all elements are float point numbers. In computer system one float point number is usually represented by 32bits. So the total bits to be transmitted is $32nr$ bits. In the case of random response, all the elements are 0 or 1, which can be represented by 1bit. So the edge devices just need to transmit nr bits, which will greatly save the communication cost by 32 times.

4.5 Privacy Leakage Against Attacks

We use two kinds of commonly used attacks and compare the performance of two LDP methods in face of possible attacks. We use model inversion attack under the white box setting, since white box attacks are harder to defend than the black box setting. The metrics we use are SSIM (structural similarity index measure) and PSNR (peak signal to noise ratio), which measure the reconstruction quality of images. Usually SSIM less than 0.3 means the image is unrecognizable. Our experiment shows that without LDP mechanisms, the first 3 layers intermediate output of ResNet-18 can be inverse to the original image with high quality. With laplace mechanism and random response, the performance of the inversion algorithm is greatly reduced. We observe that partitioning ResNet-18 with ϵ = 0.5 (random response) at layer 2 achieves a good trade-off between accuracy and privacy. Moreover, we perform membership inference attacks on the trained model under different settings. The results are shown in Table4. As membership inference attack is a binary classification task, we use F1-score to measure the performance of our membership inference attack method. In the setting without using LDP mechanisms, the F1-score of attack is 0.6657, which means that it is very likely for our attack to find whether a given data sample is in the training set. After apply LDP mechanisms in split learning, the F1-score of attacks decrease significantly in both random response and laplace mechanism setting, which means that both methods can well defend membership inference attack.

Table 3: Comparison of Laplace Mechanism (LM) and Random Response (RR) in face of model inversion attack

Metric	**SSIM**					
Privacy Budget	$\epsilon = 0.5$		$\epsilon = 1$		$\epsilon = 2$	
Method	RR	LM	RR	LM	RR	LM
Layer1	0.354	0.120	0.576	0.260	0.728	0.517
Layer2	0.211	0.084	0.306	0.086	0.453	0.205
Layer3	0.165	0.062	0.170	0.039	0.205	0.081
Layer4	0.155	0.068	0.154	0.054	0.169	0.066
Metric	**PSNR**					
Privacy Budget	$\epsilon = 0.5$		$\epsilon = 1$		$\epsilon = 2$	
Method	RR	LM	RR	LM	RR	LM
Layer1	12.941	12.202	14.216	12.854	15.531	14,528
Layer2	12.466	11.996	12.918	11.959	13.621	12.399
Layer3	12.311	11.958	12.422	11.894	12.538	12.019
Layer4	12.274	12.058	12.296	11.968	12.299	12.020

Table 4: Comparison of Laplace Mechanism (LM) and Random Response (RR) in face of membership inference attack

Metric	**F1-score**					
Privacy Budget	$\epsilon = 0.5$		$\epsilon = 1.0$		$\epsilon = 2.0$	
Method	**RR**	**LM**	**RR**	**LM**	**RR**	**LM**
Layer1	0.5230	0.5012	0.5742	0.5415	0.6315	0.5933
Layer2	0.4923	0.4907	0.5403	0.5257	0.6099	0.5625
Layer3	0.4978	0.4985	0.5261	0.5134	0.5681	0.5307
Layer4	0.4986	0.5013	0.5029	0.5017	0.5010	0.4989

5 CONCLUSIONS

In this paper, we explore local differential privacy preservation in split learning. We systematically study the performance of two local differential privacy mechanism in the modified split learning model. We show that using random response for local differential privacy will achieve better utility than laplace mechanism while preserving the sensitive information. Future works may further show that random response, as a local differential privacy mechanism, can have more applications in federated/collaborative learning.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications*

Security (CCS '16). Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Björn Bebensee. 2019. Local Differential Privacy: a tutorial. *arXiv preprint arXiv:1907.11908* (2019).

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[4] Daniel Bernau, Philip-William Grassal, Jonas Robl, and Florian Kerschbaum. 2019. Assessing differentially private deep learning with Membership Inference. *arXiv preprint arXiv:1912.11328* (2019).

[5] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. 2018. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505* (2018).

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model Inversion Attacks against Collaborative Inference. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19)*. Association for Computing Machinery, New York, NY, USA, 148–162. https://doi.org/10.1145/3359789.3359824

[9] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[10] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[11] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[12] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. 2018. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564* (2018).

[13] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.