Differentially Private Collaborative Learning for the IoT Edge

Linshan Jiang
School of Computer Science and Engineering
Nanyang Technological University
linshan001@e.ntu.edu.sg

Xin Lou
Advanced Digital Sciences Center
Illinois at Singapore Pte Ltd
lou.xin@adsc-create.edu.sg

Rui Tan
School of Computer Science and Engineering
Nanyang Technological University
tanrui@ntu.edu.sg

Jun Zhao
School of Computer Science and Engineering
Nanyang Technological University
junzhao@ntu.edu.sg

Abstract

Collaborative learning based on training data contributed by many edge devices is a promising paradigm for implementing crowd intelligence. The collaboratively trained model generally provides superior classification performance due to the increased volume and expanded coverage of the training data. However, the data contribution may incur the concern of privacy breach. This paper presents the design of a privacy-preserving collaborative learning approach, in which the edge devices and the cloud train different stages of a deep neural network, and the data transmitted from an edge device to the honest-but-curious cloud is perturbed by Laplacian random noises to achieve ε-differential privacy. We apply the proposed approach to a case study of collaboratively training a convolutional neural network for handwritten digit recognition. The results show that our approach maintains 99% and 96% classification accuracy in implementing privacy loss levels of $\epsilon = 5$ and $\epsilon = 2$, respectively.

1 Introduction

Recent years have witnessed the performance breakthroughs of various pattern recognition tasks due to the research advances in machine learning. In the era of the Internet of Things (IoT), many edge devices distributed in urban areas will generate massive data, which can be used to further improve the performance of various machine learning systems. In particular, the *collaborative learning* that builds deep models based on the massive IoT data is envisioned as an important learning paradigm to implement crowd intelligence. In this paradigm, the increased volume and expanded coverage of the training data will significantly improve the quality of the learned model.

However, the training data contributed by the edge devices may contain privacy sensitive information. Data anonymoization can mitigate some concern about privacy breach; but it is inadequate, because cross correlations among different databases may be used to re-identify data [18]. Note that recent legislation (e.g., General Data Projection Regulation in European Union and Personal Data Protection Act in Singapore) imposes stricter requirements for privacy protection. To gain wide adoption in the era of IoT, the collaborative learning systems that rely heavily on the data contributed by the individual edge devices should be designed with proper privacy preservation mechanisms.

In this paper, we present the design of a collaborative learning approach that uses the computation capabilities of the edge devices and implements the differential privacy (DP) for the data transmitted to the cloud for building the machine learning model. Specifically, the edge devices collaboratively train a deep neural network, where the training of a number of front layers of the neural network is executed on each edge device and the training of the remaining layers is executed in the cloud. During the training phase, whenever any edge device contributes a training sample, it forward-propagates the training sample over the front layers and transmits the intermediate result data vector to the cloud. The cloud will further forward-propagates the remaining layers to compute the training loss. The training loss is finally fed back to all participating edge devices to update their front layers. Thus, the front layers at all the participating edge devices remain the same during the training phase. On the completion of the training, the layers maintained by the cloud can be disseminated to all the edge devices, such that the whole neural network can be executed locally on the edge devices.

In this paper, we consider a honest-but-curious cloud that aims to infer private information from the data uploaded by the edge devices during the training phase. We adopt the ε-DP [4] as our privacy definition, which gives quantifiable indistinguishability of different data vectors yielded by the edge devices against the honest-but-curious cloud. To implement ε-DP, a Laplacian random noise vector is added to

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

the data vector generated by the front layers before being transmitted to the cloud. In our design, we apply batch normalization to the data vector generated by the front layers at the edge device to attain an analytic upper bound of the normalized data. The bound is used as the global sensitivity in setting the Laplacian noise generator parameters to guarantee ε-DP.

We apply our proposed approach to a case study of collaboratively training a convolutional neural network (CNN) for image classification. We use MNIST [14], an image dataset of handwritten digits, to train the CNN. Two convolutional layers with max pooling are trained by the edge devices, while six dense layers are trained by the cloud. Results show that our approach maintains 99% and 96% classification accuracy in implementing privacy loss levels of $\varepsilon = 5$ and $\varepsilon = 2$, respectively. Note that, to provide good DP protection, the typical privacy loss level, i.e., ε, is often set to a value below 10. For example, in [9], to obtain the balance between system performance and data privacy, the ε is set to be 10. In [1], the ε is set to 0.5, 2, or 3. Thus, the case study based on MNIST shows that our approach can achieve good DP protection while maintaining satisfactory classification performance.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the design of the differentially private collaborative learning approach. Section 4 presents the performance evaluation results. Section 5 concludes the paper.

2 Related Work

Collaborative learning schemes have been proposed to protect the data owner's privacy. Existing approaches can be broadly classified into three categories of *distributed machine learning*, *data obfuscation/encryption*, and *partitioned model training*.

2.1 Distributed Machine Learning

In [9, 17, 21], the distributed collaborative training of a deep neural network is studied. In these studies, the gradients or the model parameters are exchanged among the collaborative learning participants. The exchange is orchestrated by a learning coordinator. The *federated learning* approach is proposed in [17], in which the training data is kept locally and each client trains a local deep model. The central server retrieves the local deep models from randomly chosen clients and returns the average deep model to them. This federated learning approach has been implemented in the Google's mobile App Gboard [7] that performs typing recommendation. In [21], each participant trains a local deep model using stochastic gradient descent (SGD) and uploads a selected portion of gradients to the coordinator for combining. Then, each participant downloads a selected portion of the global gradients to update its local deep model. In the Crowd-ML approach [9], a participant checks out the global classifier parameters from the coordinator and computes the gradients using its own training data. Then, the participants transmit the gradients to the coordinator to update the global gradients.

In the collaborative learning approaches, since the SGD or the parameters still contain the model information, to further protect the user privacy, in [9, 21], additive noises are added to the model parameters transmitted to the central server to achieve DP [4]. However, the DP protection is for each parameter. When the number of parameters in the model is large, the achieved privacy protection level is low [1]. In a recent work [10], the authors propose to use the generative adversarial networks that can generate the prototypical samples of the targeted training set to weaken the privacy protection achieved by [9, 21]. Moreover, in [2, 20], cryptographic primitives are integrated with the approaches in [17, 21] for better privacy protection. However, these cryptographic primitives, e.g., homomorphic encryption, often incur high computation complexity. Thus, these approaches are ill-suited for edge devices with constrained computing resources.

2.2 Data Obfuscation/Encryption

Data obfuscation/encryption approaches have also been proposed to protect data privacy. The approaches [8, 15] transmit the encrypted or obfuscated training data to the coordinator to build the machine learning models. The approach in [23] protects the privacy of both individual properties and group statistical properties by adding random noises to sensitive samples and augmenting the dataset with faked samples. This approach can counteract several privacy attacks without affecting the prediction accuracy of the trained model much.

2.3 Partitioned Model Training

In a recent work [16], instead of directly transmitting the noisy training data to the server, part of the CNN computation is performed at the edge nodes. The work shows that, if the edge nodes train the first layer of the CNN and add Gaussian noises to the output of the first layer, desirable learning performance can be maintained. However, the actual DP achieved at the server is much lower than that claimed by the edge nodes. In addition, the paper [16] formulates an optmization problem by considering privacy, resource cost and learning precision to find out the optimal partitioning of the CNN. However, the optimization model is based on an simplification assuming the same weights for all three factors.

In [19, 22], the transfer learning techniques are used for private inference across mobile devices and clouds. In this method, a number of front layers of the target neural network are deployed at the edge devices using a pre-trained neural network and the remaining layers of the target neural network in the cloud are trained using data from edge devices. To protect the privacy of these data, Siamese fine-tuning, dimensionality reduction, and noise addition mechanisms are used in [19]. In [22], the nullification technique is used to hide the sensitive data before adding the Laplacian noises to achieve DP. However, both approaches [19, 22] can only protect the data privacy at the fine-tuning stage but not the training stage. In addition, parts of the data have to be released for learning the pre-trained model in applications where no public data is available.

3 Approach Design

This section presents the design of the proposed differentially private collaborative learning approach. Our approach belongs to the partitioned model training category discussed in Section 2. We describe the system model and our approach in Section 3.1 and Section 3.2, respectively. Section 3.3 presents an analysis that guides the setting of the Laplacian noise generator to achieve ε-DP.

3.1 System Model

In this paper, we consider a collaborative learning system consisting of multiple *learning participants* and an honest-but-curious *learning coordinator* to realize a classification system. In practice, the learning coordinator and participants can be a cloud server and edge devices, respectively. In our model, we mainly consider the privacy contained in the original data due to potential leakage threat in which the data is used in unauthorized applications. For example, in an activity recognition system based on wearable devices, three-axis acceleration data can be used to infer human body activity. However, the acceleration data can be also exploited to infer the health status of the wearer. With such inferred health status, targeted advertisement can be performed. Thus, protecting the data privacy in a collaborative system is important. Privacy-preserving approach can prevent data abuse.

The coordinator in our model is honest but curious. Specifically, it honestly supports the collaborative learning process to compute correctly and send results truthfully. However, it is curious about the privacy contained in the data, since it may exploit the privacy for irrelevant applications. In this paper, we do not consider the privacy contained in the label of the contributed training data since we assume that the participant willingly contributes the labeled data to perform supervised learning and should have no expectation on the privacy contained in the labels. Our approach supports anonymization of the training features and labels. Specifically, the coordinator should not expect to know the participant's identity for the received training samples. Moreover, the coordinator cannot determine which two training samples come from the same participant. This can be achieved via an anonymous communication network [3] to transmit the training features and labels to the coordinator.

3.2 Approach Overview

In this paper, we propose an approach which can protect the privacy of the extracted features before being transmitted to the coordinator to preserve the privacy contained in the original data. Perturbing original data directly to protect privacy may lead to significant learning performance degradation which will be shown in Section 4. From our analysis in Section 3.3, we can perturb the results computed from the original data before being transmitted to the coordinator to protect the privacy contained in the original data.

To realize the advantages of collaborative learning, the classification computation during the learning phase is performed on the coordinator to make good use of various data from different participants. In this paper, we consider convolutional neural network (CNN) to design collaborative learning system, since CNN is an effective machine learning model. In CNN, convolutional layers fold data in several channels to extract features with specific pooling layers and activation layers. The dense layers (i.e., fully-connected layers) classify the extracted features to yield class labels. In our collaborative learning system, each participant runs convolutional layers to extract features that will be transmitted to the coordinator. The coordinator maintains the dense layers and forward-propagates them with the received features during the learning phase. Moreover, the participants will perturb the features before transmitting them to the coordinator.

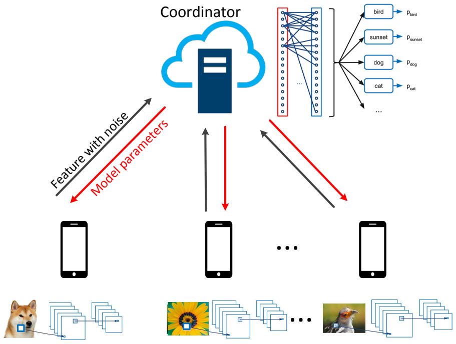

Figure 1. Overview of our proposed privacy-preserving collaborative learning approach.

Figure 1 illustrates the system architecture. Each participant collects data and extracts features locally. Under the privacy-preserving mechanism that will be presented in Section 3.3, participant sends privacy-preserving features and original labels to the coordinator such that the coordinator can train the fully-connected layers. The coordinator uses the backpropagation algorithm to update the fully-connected layer parameters and meanwhile sends back the propagated loss to the participants which will update convolutional layers accordingly. In the above process, the convolutional layers of all the participants are updated based on each contributed training data sample. Thus, the participants enjoy the advantages of collaborative learning, which help them better extract features.

We now discuss several design issues.

- During the classification phase after the completion of the collaborative learning, the participant can send testing data features to the cloud, which will then perform classification using the dense layers. Alternatively, on the completion of the collaborative learning, the coordinator can disseminate the dense layers to all participants. Then, each participant can run the full CNN to perform classification without transmitting testing data.
- In order to utilize the large volume of training data to improve the effectiveness of the convolutional layers, it is desirable to maintain the same convolution layers at all the participants. We adopt the following method to keep convolutional layer consistency among participants. After the coordinator updates dense layer parameters, it broadcasts propagated loss to all the participants. Thus, all the participants can update their own convolutional layers simultaneously. Since we can configure the same hyperparameters for the convolutional layers at all the participants, we can maintain the convolutional layers at all the participants consistent.

- The system will have significant overhead if each participant immediately sends new extracted features once it generates new data. To solve this issue, in our design, if the data exceeds a specified value, the participant starts to process data to extract feature and transmit it. This method matches well with our privacy-preserving approach which adopts batch normalization and Laplacian noisification, which will be presented in the next subsection.

3.3 Achieving ε-Differential Privacy

Differential privacy is an information-theoretic approach to protecting data privacy. It aims to confound the query results based on adjacent datasets. In our approach, we adopt ε-differential privacy [4] as our privacy definition. The ε-differential privacy (ε-DP) is formally defined as follows: *A randomized algorithm $\mathcal{A} : \mathbb{D} \to \mathbb{R}^t$ gives ε-DP if for all adjacent datasets $D_1 \in \mathbb{D}$ and $D_2 \in \mathbb{D}$ differing on at most one element, and all $S \subseteq Range(\mathcal{A})$*, $\Pr(\mathcal{A}(D_1) \in S) \le \exp(\varepsilon) \times \Pr(\mathcal{A}(D_2) \in S)$. Here, the differential privacy level ε, is a positive number which measures privacy loss. Smaller ε always means better protection: when ε is very small, $\Pr(\mathcal{A}(D_1) \in S) \approx \Pr(\mathcal{A}(D_2) \in S)$ for all $S \subseteq Range(\mathcal{A})$. Thus, the query results $\mathcal{A}(D_1)$ and $\mathcal{A}(D_2)$ are nearly indistinguishable, which prevents the attackers from recognizing the original dataset. We consider bounded differential privacy, where two neighboring databases have the same size, and have different records at only one of the positions. An approach to implementing ε-DP is to add Laplacian noise [6]. Concretely, for all function $\mathcal{F} : \mathcal{D} \to \mathbb{R}^t$, the randomized algorithm $\mathcal{A}(D) = \mathcal{F}(D) + [n_1, n_2, \dots, n_t]^\intercal$ gives ε-DP, where each n_i is drawn independently from a Laplace distribution $\mathrm{Lap}(S(\mathcal{F})/\varepsilon)$ and $S(\mathcal{F})$ denotes the global sensitivity of $\mathcal{F}$. Note that the global sensitivity $S(\mathcal{F})$ is $S(\mathcal{F}) = \max_{\text{neighboring databases } D, D' \in \mathbb{D}} ||\mathcal{F}(D) - \mathcal{F}(D')||_1$ while $\mathrm{Lap}(\lambda)$ is a zero-mean Laplace distribution with a probability density function of $f(x|\lambda) = \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}$.

A challenge in implementing ε-DP is the determination of the global sensitivity $S(\mathcal{F})$. It is hard to determine global sensitivity after convolutional layers. Theoretically, the output of convolutional layers can continuously increase or decrease during training epochs. However, too large or too small outputs of the convolutional layers may cause the *gradient exploding problem* or *gradient vanishing problem* [12]. Batch normalization (BN) [12] is developed to normalize the output of hidden layers to avoid the problems to support neural network training. Using each batch as a unit, BN normalizes the output of specific layers and then forwards it to the next layer. It limits the range of the output, enabling the determination of the global sensitivity $S(\mathcal{F})$. In our approach, we apply standard BN parameters: fixed variance 1 and fixed mean 0. In the following, we explain the method to compute the global sensitivity $S(\mathcal{F})$.

For simplicity, we assume there is only one channel in the CNN and the dimension of the output of the convolutional layers is $L \times W$. Denote the batch size in the convolutional neural network as N, the output of convolutional layers in a position $\langle i, j \rangle$ of element k in the batch as $X_{i,j,k}$. The difference between two adjacent datasets D and D' in our scenario is $X_{i,j,k}$ and $X'_{i,j,k}$, while the other elements are the same. The query request in our scenario is to read each element in the dataset because the coordinator can access all data sent from the participants. Thus, the global sensitivity $S(\mathcal{F})$ is equal to the maximum difference between $X_{i,j,k}$ and $X'_{i,j,k}$, $S(\mathcal{F}) = \max_{\langle i,j,k \rangle \in \langle L,W,N \rangle} \{X_{i,j,k} - X'_{i,j,k}\}$. Due to the constraint imposed by BN, we have $\sum_{k=1}^{N} X_{i,j,k} = 0$ and $\sum_{k=1}^{N} {X_{i,j,k}}^2 = N$.

To analyze $S(\mathcal{F})$, we now prove for any $\ell \in \{1, 2, \dots, N\}$ that $-\sqrt{N-1} \le X_{i,j,\ell} \le \sqrt{N-1}$ and both equal signs are applicable in special cases.

From the Cauchy–Schwarz inequality, we have

$$\left(\sum_{t \in \{1,2,\dots,N\} \setminus \{\ell\}} X_{i,j,t} \right)^2 \le (N-1) \sum_{t \in \{1,2,\dots,N\} \setminus \{\ell\}} {X_{i,j,t}}^2. \quad (1)$$

Applying $\sum_{t \in \{1,2,\dots,N\} \setminus \{\ell\}} X_{i,j,t} = -X_{i,j,\ell}$ and $\sum_{t \in \{1,2,\dots,N\} \setminus \{\ell\}} {X_{i,j,t}}^2 = N - {X_{i,j,\ell}}^2$ to Inequality (1), we obtain ${X_{i,j,\ell}}^2 \le (N-1) \cdot (N - {X_{i,j,\ell}}^2)$, which leads to

$$-\sqrt{N-1} \le X_{i,j,\ell} \le \sqrt{N-1}. \quad (2)$$

The equal sign in the first "≤" of Inequality (2) is applicable when $X_{i,j,\ell} = -\sqrt{N-1}$ and $X_{i,j,t} = 1/\sqrt{N-1}$ for $t \in \{1, 2, \dots, N\} \setminus \{\ell\}$. Similarly, the equal sign in the second "≤" of Inequality (2) is taken when $X_{i,j,\ell} = \sqrt{N-1}$ and $X_{i,j,\ell} = -1/\sqrt{N-1}$ for $t \in \{1, 2, \dots, N\} \setminus \{\ell\}$.

From the above analysis, $S(\mathcal{F})$ denoting $\max_{\langle i,j,k \rangle \in \langle L,W,N \rangle} \{X_{i,j,k} - X'_{i,j,k}\}$ is equal to $2\sqrt{N-1}$. By adding a random noise from $\mathrm{Lap}(S(\mathcal{F})/\varepsilon)$ [5], we can achieve ε-DP to protect original data privacy. It also succeeds when there are multiple channels in CNN. The detailed proof is omitted here due to space constraint.

4 Performance Evaluation

In this section, we evaluate our approach in an application of image-based handwritten digit recognition.

4.1 Evaluation Methodology and Settings

Our evaluation is based on a public dataset MNIST [14]. MNIST is a hand written dataset which consists of 60,000 training samples and 10,000 testing samples. Each sample is a 28×28 gray scale image showing a handwritten number within 0 to 9. It is widely used in machine learning literature as a basic benchmark dataset to eveluate the learning performance.

In our model, the CNN deployed at the participants has two convolutional layers with 30 and 80 channels, respectively. After each conventional layer, we apply max-pooling layers to reduce the size of the output. In neural networks, the max-pooling layer can accelerate learning with reduced parameter dimension while extracting features of subregion in a sample. In dense layers, the ReLU activation layer is used to increase the nonlinearity of the neural network. After the second conventional layer, we use a BN layer to accelerate the learning rate and prevent gradient vanishing problem and gradient exploding problem. Then, the DP technique is applied to perturb the output of the BN layer to preserve data privacy.

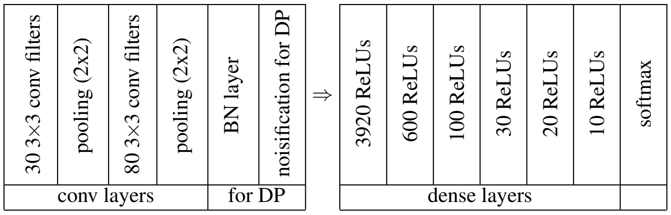

Figure 2. CNN structure.

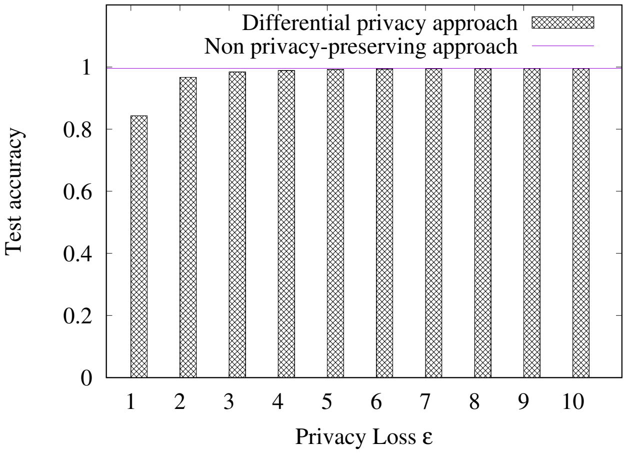

Figure 3. Impact of privacy loss level ε on the test accuracy of the collaboratively learned model with DP.

After adding the DP noise, the participants send perturbed features to the coordinator as the input for the dense layers. In our model, the dense layers contain four hidden layers for reducing the data dimension gradually and one output layer with a dimension of 10 which is the dimension of labels. Finally, we use softmax layer to predict label and compute loss. The structure of the CNN is shown in Figure 2.

In our experiments, we set the hyperparameters of CNN as follows: the learning rate is equal to 0.01 and the batch size is equal to 64. Thus, the global sensitivity $S(\mathcal{F})$ is equal to $\sqrt{63}$. Therefore, we apply various privacy loss levels ε to evaluate the performance of the differentially private collaborative learning based on MNIST.

4.2 Evaluation Results

For comparison, we use the centralized training approach without any privacy consideration as the baseline. The corresponding CNN excludes the noisification layer as shown in Figure 2. This centralized non-DP approach achieves 99.58% test accuracy. From Figure 3, with our differentially private collaborative learning approach, the test accuracy increases with the privacy loss level ε. Note that a large ε means less privacy protection. Thus, there exists a trade-off between the test accuracy and the degree of privacy protection. Generally, when the ε is chosen to be 5, which is often considered providing satisfactory privacy protection [1, 9], our system can still achieve 99.18% test accuracy. When ε is reduced to 1, the test accuracy decreases to 84.33%, because large DP noises start to undermine the performance of the classification system. However, when ε is around 2 to 5, the system shows good classification performance. Specifically, only 3% of accuracy reduction is observed when ε reduces from 5 to 2.

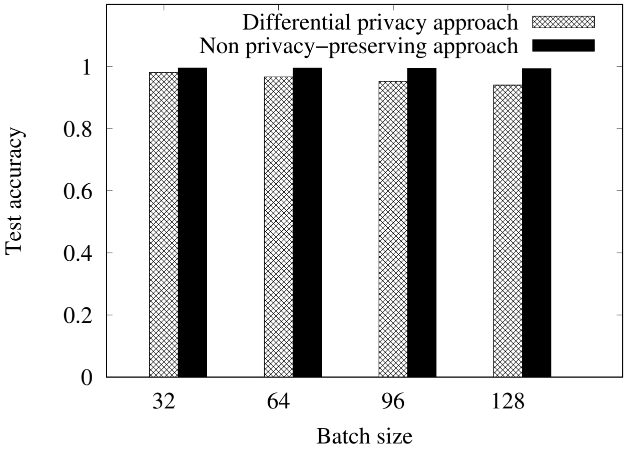

Figure 4. Impact of batch size on the test accuracy of the collaboratively learned model with DP (ε = 2).

In the second set of experiments, we investigate the impact of BN's batch size on the classification performance of the collaboratively learned model. For training CNN, a smaller batch size often results in more accurate estimation of the gradient descent, but longer convergence time of the training process. Moreover, in our approach, the batch size N determines the global sensitivity $S(\mathcal{F})$, i.e., $S(\mathcal{F}) = \sqrt{N-1}$. Thus, the smaller batch size also results in lower noise levels for the same ε setting. We set $\varepsilon = 2$. Figure 5 shows the test accuracy of the CNNs trained by our differentially private collaborative learning approach and the centralized learning approach without privacy preservation, under different batch size settings. When the batch size $N = 32$, the test accuracy is 99.5% and 98.1% for the centralized non-DP learning approach and our DP approach, respectively. When N increases to 128, the accuracy drops to 99.3% for the centralized non-DP approach and 94.0% accuracy for our approach. For our approach, with a larger batch size, both the global sensitivity and noise level become larger, leading to performance drop.

Adding Laplacian noises to the original data to achieve ε-DP is an alternative approach. In this section, we also investigate its effectiveness. Under this alternative approach, the global sensitivity $S(\mathcal{F})$ of the original data (i.e., the pixel values) is the maximum difference between any two pixels. Since the pixel value in MNIST is within the range of $(0, 255)$, the global sensitivity is a fixed value of 255. Figure 5 shows the test accuracy of the CNN trained by this alternative approach under various ε settings. We can see that, when $\varepsilon = 10$, the test accuracy is 11.35% only, which is close to the performance of random guessing (i.e., 10%). When $\varepsilon \geq 100$, although the approach can achieve good test accuracy, the privacy loss is too high to be meaningful in a collaborative learning system. Thus, the results show that adding Laplacian noises to the original data significantly degrades the learning performance. Moreover, by comparing

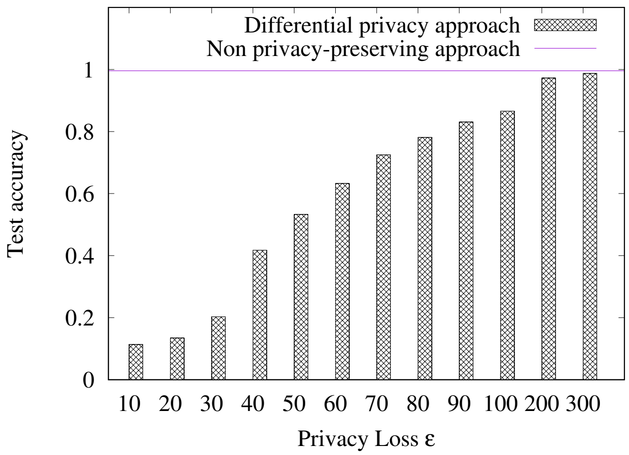

Figure 5. Impact of privacy loss level ε on the test accuracy of the collaborative learning approach that perturbs the original data for DP.

the results obtained with this alternative approach and our approach, we can see that the unsupervised feature learning performed by the convolutional layers is susceptible to the DP noises, whereas the classification boundary learning performed by the dense layers is more robust to the DP noises.

5 Conclusion and Future Work

This paper presents the design of a collaborative learning approach that trains different stages of a deep neural network at the edge devices and the cloud, respectively. The deep neural network model is constructed based on the training samples contributed by all the participating edge devices. To protect the privacy contained in the data communicated to the honest-but-curious cloud during the collaborative learning process, Laplacian random noises are added to the communicated data. We apply our approach to a case study of collaboratively learning a CNN for handwritten digit classification. Results show that collaboratively learned CNN with ε-DP has about 3% classification accuracy loss only, when the DP loss level ε is down to 2.

There are several potential future directions of this work. First, due to the page limit, we only evaluated the performance of our approach with one dataset. We will conduct more experiments using other benchmark datasets, e.g., CIFAR-10 [13] and LFW [11]. Second, we will implement the collaborative learning approach on edge devices to understand its performance and overhead in real IoT applications. Third, we will prove the privacy guarantee achieved by our approach and investigate the trade-off between privacy and learning precision under different settings. Finally, in our approach, the convolutional and dense layers are partitioned to the participants and cloud, respectively. In our future work, we will investigate other partition approaches and investigate the impact of different partitions on the trade-off between DP and learning performance.

6 Acknowledgments

This research is supported in part by two NTU Start-up Grants (M4081729.020 and M4082311.020). We acknowledge the support of NVIDIA Corporation with the donation of the Quadro P5000 GPU used in this research.

7 References

[1] M. Abadi, A. Chu, I. Goodfellow, H. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[2] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy preserving machine learning. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[3] G. Danezis and C. Diaz. A survey of anonymous communication channels. Technical report, Microsoft Research, 2008. MSR-TR-2008-35.

[4] C. Dwork. Differential privacy. *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2006.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Conference on Theory of Cryptography*, 2006.

[6] 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.

[7] Google AI. Federated learning: Collaborative machine learning without centralized training data, 2017. `https://bit.ly/2IHdmzw`.

[8] T. Graepel, K. Lauter, and M. Naehrig. Ml confidential: Machine learning on encrypted data. In *Intl. Conf. Inf. Security & Cryptology*, 2012.

[9] J. Hamm, A. Champion, G. Chen, M. Belkin, and D. Xuan. Crowd-ml: A privacy-preserving learning framework for a crowd of smart devices. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2015.

[10] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[11] G. B. Huang and E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Tech. Rep*, pages 14–003, 2014.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

[14] Y. LeCun, C. Corts, and C. J. Burges. The mnist database of handwritten digits, 2018. `http://yann.lecun.com/exdb/mnist/`.

[15] B. Liu, Y. Jiang, F. Sha, and R. Govindan. Cloud-enabled privacy-preserving collaborative learning for mobile sensing. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2012.

[16] Y. Mao, S. Yi, Q. Li, J. Feng, F. Xu, and S. Zhong. Learning from differentially private neural activations with edge computing. In *The Third ACM/IEEE Symposium on Edge Computing (SEC)*, 2018.

[17] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *The 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[18] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint arXiv:cs/0610105*, 2006.

[19] S. Osia, A. Shamsabadi, A. Taheri, H. Rabiee, N. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *arXiv preprint 1703.02952*, 2018.

[20] L. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Information Forensics and Security*, 13(5), 2018.

[21] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *ACM Conf. Computer and Communications Security (CCS)*, 2015.

[22] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. Yu. Not just privacy: Improving performance of private deep learning in mobile cloud. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[23] T. Zhang, Z. He, and R. Lee. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860*, 2018.