Gradient-Based Differential Privacy Optimizer for Deep Learning Model Using Collaborative Training Mode

Jing Xia
Wuhan Digital Engineering Institute
Wuhan, China
xiajing_csic@163.com

Weihua Huang
Wuhan Digital Engineering Institute
Wuhan, China
wh.h@foxmail.com

Zhong Ma
Wuhan Digital Engineering Institute
Wuhan, China
mazh@public.wh.hb.cn

Xinfa Dai
Wuhan Digital Engineering Institute
Wuhan, China
daixinfa@sina.com.cn

Li He
Wuhan Digital Engineering Institute
Wuhan, China
651305747@qq.com

***Abstract*—Deep learning model based on artificial neural network is one of the greatest pushers to realize intelligence of information system unprecedentedly. However, the risk of leaking user data privacy by attacking deep learning model exists in the training process, especially when multi-users concurrently utilize the service of cloud. Motivated by this observation, in order to protect the privacy of deep learning model, this paper proposes a gradient-based differential privacy optimizer using collaborative training mode based on CPU-GPUs hybrid system in cloud. In gradient-based differential privacy optimizer, random sampling, gradient tailoring, gradient-based random perturbation, and advanced privacy budget statistics together guarantee the usability and privacy of the model. Specifically, the effects of privacy budget, noise scale, and privacy deviation parameters and their combinations on accuracy of model are experimentally studied in this paper. To further improve the training efficiency of model, the CPU-GPUs hybrid system is explored as a collaborative training mode for gradient-based differential privacy optimizer. The experimental results indicate that using publicly available dataset MINIST and implemented in TensorFlow is proved to be feasible and efficient. Specifically, our implementation and experiments demonstrate that we can obtain approximately 96% of accuracy under a modest privacy budget. Furthermore, we can achieve up to 20%-30% speed-up ratio in the training process based on gradient-based differential privacy optimizer.**

Keywords—gradient, differential privacy, collaborative training, CPU-GPUs, deep learning model

I. INTRODUCTION

Along with the rise of deep learning concept and the corresponding training methodology first proposed by the Geoffery et al. [1], it is now customary to apply deep learning model to analyze complex information in various practical domains. Deep learning model builds deep neural network by simulating the way of human brain processing information. Deep learning model has made breakthrough development in applications of speech processing [2],[3], natural language processing [4],[5],[6], especially computer vision [7],[8] in recent years. However, many malicious users attack the deep learning model to obtain the privacy of training sample data. Therefore, how to protect the sensitive information of users in the process of training process becomes particularly important in cloud paradigm.

As the cloud provides Artificial Intelligence (AI) services for multi-users in a unified way, cloud customers are led to store, compute, and analyze their large-scale data on the third-party cloud providers generally. However, there are mainly three kinds of privacy leaks in this service mode, including privacy leak of user data, privacy leak of model itself (parameters, architecture) and privacy leak of training data. Existing solutions mainly include cryptography technology and differential privacy technology. Microsoft proposed CryptoNets [9], which rewrites the trained model through homomorphic encryption technology, so that the model can get ciphertext output based on ciphertext input prediction. The encryption model of cloud deployment guarantees the privacy of the model itself and the training data. At the same time, the user submits the encrypted data, and also guarantees the privacy of user data. However, the homomorphic encryption technology is faced with the common shortcomings of all encryption technologies. In special, the large amount of computation will lead to resource consumption and network communication overhead at the user end. Furthermore, the homomorphic encryption technology needs to approximate the non-linear activation function in the model, and faces the loss of accuracy. Therefore, this paper does not intend to adopt such a way of protecting data privacy using data encryption.

Differential privacy technology guarantees the privacy of training data by introducing randomness. Differential privacy-preservation further can not only protect the security of sensitive information in the training process, but also avoid the computational overhead caused by encryption schemes. Differential privacy-preservation prevents attackers from restoring the original data in the training data set by using generative countermeasure network, and consequently protects the sensitive information of training data set. In special, differential privacy-preserving method is to disturb data by adding noise to protect indistinguishability of individual samples in data sets. Its advantage is that it is easy to implement, which brings less extra computing and communication overhead, and provides privacy protection attributes with strict theoretical proof. Therefore, differential privacy-preservation can resist attacks against privacy data of model in cloud by preventing malicious users from obtaining model parameters and architecture in the application of deep learning model. Generally, there are two main ways to combine differential privacy with deep learning model. One is to treat deep learning as a black box and adds noise data to the final parameters of the trained deep learning model. The other is to treat the deep learning model as a white box and

add noise data in the parameter optimization stage of the model training process. In the former case, because feature parameters of some deep learning model depend on training data, adding too much noise to the parameters will destroy the availability of the model. On the contrary, when the noise data is added too little, it may not meet the need of privacy protection of the training data sets in cloud. Therefore, this paper proposes a gradient-based differential privacy optimizer in order to achieve a balance between privacy protection and data availability of deep learning model.

Nowadays, GPU-assisted accelerating is applicable and popular in many applications from the hardware perspective. Furthermore, with continuous architectural improvements, GPUs have evolved into a massively parallel, multithreaded, many-core processor system with tremendous computational power. Owing to introduction of the Compute Unified Device Architecture (CUDA) programming paradigm, a vast of computation problems outside of the graphics domain have benefited from the superior performance of GPUs. Maybe for this reason, many scholars utilize GPU to accelerate computational efficiency of various algorithms.

Differential privacy technology satisfies the parallel theory on data sets. That is, adding noise of the same privacy-preserving level on collaborative data sets separately will not lead to the accumulation of total privacy loss. Therefore, in order to make full use of heterogeneous computing resources, an efficient differential privacy optimizer is designed from the view of heterogeneous framework to improve computing efficiency.

Based on the above discussion and analysis, we combine the differential privacy-preserving technology for deep learning model and the parallel framework to create this paper's ideal. In this paper, we try to address one central question: How to design differential privacy-preserving for deep learning model using collaborative training scheme in cloud? That is, how to protect the privacy of deep learning model with high efficiency.

The purpose of this paper is to identify that the model training of privacy-preservation can be realized by applying the differential privacy optimizer based on CPU-GPUs hybrid system. We further do some experiments in order to quantitatively test the accuracy and the performance after adding the gradient-based differential privacy optimizer.

The main contributions of this paper are the following.(1) It is proved that a series of optimization methods for gradient-based differential privacy technology can achieve a balance between privacy protection and data availability, including random sampling, gradient tailoring, gradient-based random perturbation, and advanced privacy budget statistics technology. (2) Gradient-based differential privacy optimizer for deep learning model applied to publicly available dataset MINIST and implemented in TensorFlow are proved to be feasible and efficient. Furthermore, the CPU-GPUs hybrid system is explored as a collaborative training scheme for deep learning model using gradient-based differential privacy optimizer. (3) It is proved that the proposed method not only guarantee the accuracy and training efficiency of the deep learning model but also guarantee privacy of model by introducing the grandient-based differential privacy optimizer.

The rest of the paper is organized as follows. Section 2 reviews related work on differential privacy-preservation for deep learning model. The core of our paper, Section 3, a gradient-based differential privacy-preservation for deep learning model using collaborating training mode is proposed. Section 4 shows the experimental results, and we draw some conclusions and future works. Finally express our thanks in Section 5 and express our thanks in Acknowledgement.

II. Related Work

The deep learning model and corresponding training data are deployed to provide AI services for users in cloud, which may be crowdsourced and contain sensitive information. Because of the fact is that the privacy of sensitive personal information is an increasingly important topic as a result of the increased availability of cloud services, especially in the medical and financial field. Generally, many attackers utilize deep learning model to restore the original data by using generative countermeasure network. Consequently, the privacy of deep learning model remains major concerns in cloud computing paradigm. Recently, how to preserve the privacy of deep learning model is in a state of embryonic and positive development at home and abroad.

Dwork et al. [10][11] first put forward the formal definition of differential privacy concept, and relatively elaborate the basic principles and applications. Differential privacy has attracted much attention in machine learning and data mining since it was published in 2006. Abadi et al. [12] is the first time to link differential privacy technology with deep learning and a stochastic gradient descent optimization algorithm is designed to satisfy differential privacy. Specifically, they add appropriate Gaussian noise to the gradient update value of SGD algorithm to ensure that the final model parameters satisfy differential privacy attributes and protect the privacy of training data. However, the biggest problem of their method is that the differential privacy budget will be consumed with the number of iterations, so the privacy budget limits the number of iterations of the model, which will affect the accuracy of the deep learning model and consequently affect the normal use of the model. Furthermore, the increase of model parameters will lead to more consumption of privacy budget in each iteration process, so the privacy budget limits the size of the deep learning models, and limit the application of extreme-scale deep learning models subsequently. Based on the same idea, Xie et al. [13] introduced differential privacy into Generating Antagonistic Networks (GAN), which is unlike the original differential privacy with SGD algorithm, they only added noise to gradient updating in the process of discrimination, but they also had the problems of iteration number limitation and model capacity limitation. The differential privacy-preservation method for deep learning model based on deep convolutional generative adversarial networks (DCGAN) feedback proposed by Mao et al. [14] also achieves the purpose of privacy-preservation by adding noise to DCGAN training process.

Instead of the above methods of using differential privacy to interfere with the gradient value in the training process of deep learning models for privacy-preservation, some scholars have proposed a function mechanism based on differential privacy theory to interfere with the optimization objective function. This kind of differential privacy solution based on function mechanism avoids the problem of constant consumption of privacy budget caused by iteration process in deep learning model training. However, approximating a

non-linear objective function to a polynomial objective function will affect the accuracy of the objective function, which also affects the accuracy of the deep learning model. Papernot et al. [15] propose a novel Teacher-Student training paradigm, which divides data sets into disjoint data sets and uses a semi-supervised training method to train multiple Teacher models with open unlabeled data. The labels of open data are obtained by querying all Teachers models and aggregating differential privacy. Finally, only the Student model is released, which protects the privacy of the original sensitive data and the model.

Recently, because powerful computing capabilities enable deep learning model computation to be supported effectively, many scholars have proposed some solutions from the perspective of collaborative training. Shokri et al. [16] propose a deep learning method for privacy protection based on the idea of collaborative training through the coordination of the central server and participants trained the model on the local data set. In this collaborative training mode, how to receive the information returned by the participants and integrate it into the global model requires certain strategies. Therefore, McMahan et al. [17] propose the model averaging scheme in this scenario, and introduce the noise mechanism of differential privacy to ensure that the exposed gradient value does not reveal the privacy of training data, which strengthens the privacy protection of the training data.

Based on the investigation of the current situation at home and abroad, we can conclude that there are a few studies on privacy-preservation for deep learning model in general. Furthermore, we must notice that these existing works do not consider guaranteeing the accuracy of deep learning model after using the differential privacy technology. Moreover, to the best of our knowledge, there is no research trying to using CPU-GPUs hybrid system to design differential privacy for deep learning model based on collaborative training mode.

Based on the above research and analysis, we presents a gradient-based differential privacy optimizer using collaborative training mode based on CPU-GPUs hybrid system, for preserving the privacy of model and keeping the model available.

III. Proposed Method

In this section, we first describe the background knowledge of differential privacy. Then the detail of gradient-based differential privacy optimizer are illustrated. Based on the differential privacy optimizer, the collaborating training mode based on CPU-GPUs hybrid system is explained.

A. Background Knowledge of Differential Privacy

The main purpose of differential privacy is to limit the difference to a certain range between the query result of the original data set and that of the adjacent data set which differs by one record. Even if the attacker knows all the background knowledge of data set, he may not judge whether the target data exists in the data set by the result of the query of the data set, so the privacy of the data set is protected.

Definition One: Differential Privacy. If the result of random algorithm A with a value of $Range(A)$ on any data set D and $D^{'}$ with at most one record difference is O which satisfies Formula 1, then random algorithm A satisfies ε -differential privacy.

$$\Pr(A(D)=O) \le e^{\varepsilon} \times \Pr(A(D^{'})=O) \tag{1}$$

The privacy budget parameter determines the degree of preservation of differential privacy. The larger the ε is, the wider the constraints that can satisfy the Formula one and the lower the privacy consequently. Conversely, when the ε is smaller, the stronger the privacy is. In particular, when ε equals zero, the output of the algorithm is exactly the same for the two data sets with one record of any difference, and the privacy is the highest. However, in this case the output can't reflect any characteristics of the data set at all. The probability Pr( ) is controlled by the randomness of algorithm A, which indicates the risk of privacy disclosure.

Noise mechanism is the main technology to realize differential privacy. The amount of noise added is controlled by the privacy budget parameter. Too much noise will affect the availability of data, and too little noise will not provide enough privacy preservation. Moreover, sensitivity is a key parameter to determine the size of added noise, which refers to the maximum change in the result of deleting any record in the data set. Taking global sensitivity as an example, the role of sensitivity is illustrated as follows.

Definition Two: Global Sensitivity. For a query function f: $D \to R^d$, the output results on data set D and $D^{'}$ are $f(D)$ and $f(D^{'})$, the corresponding global sensitivity satisfies the Formula 2.

$$\Delta f = \max_{D,D^{'}} \left\| f(D) - f(D^{'}) \right\|_1 \tag{2}$$

The difference between D and $D^{'}$ is one record, D represents the query dimension of function f, R represents the real space mapped. The global sensitivity of a function is determined by the function itself. And different functions have different global sensitivity. For example, the counting function only needs a small amount of noise to cover up the impact of deleting records on the query results, which satisfies the differential privacy condition.

Laplace mechanism and exponential mechanism are the two most basic mechanisms for differential privacy-preservation. Laplace mechanism is suitable for numerical type. The exponential mechanism is applicable to non-numerical results. The noise required by the algorithm based on different noise mechanisms and satisfying differential privacy depends on the global sensitivity of the query function. Taking Laplace mechanism as an example, the role of it is illustrated as follows.

Definition Three: Laplace Mechanism. For any function f: $D \to R^d$, if the output of algorithm A satisfies Formula 3, then A satisfies the ε -differential privacy, where $lap_i(\Delta f/\varepsilon)(1 \le i \le d)$ is a mutually independent Laplace variable, and its corresponding probability density function is $P(x/b) = (1/2b)e^{-|x|/b}$. The noise is proportional to f and inversely proportional to E. That is, the greater the global sensitivity is, the greater the noise added.

$$A(D) = f(D) + < lap_1(\Delta f/\varepsilon), lap_2(\Delta f/\varepsilon), \ldots, lap_d(\Delta f/\varepsilon) > \tag{3}$$

Laplace mechanism mainly protects the privacy in data processing by selecting the data processing objects according to the proportion, in order to satisfy the need of differential privacy.

B. Gradient-Based Differential Privacy Optimizer

This paper intends to draw lessons from the differential privacy mechanism in classical privacy protection methods to enhance the security of the model. By adding random perturbations to the model parameters, in order to give uncertainty for the deep learning model. Differential privacy guarantees that an attacker with any strong background knowledge can not simply query the model and analyze the output to get information about a single sample.

However, differential privacy technology needs to conceal the real model parameters by introducing random noise, so as to shield the impact of a single training sample on the output of the model. Moreover, the scale of noise will have an important impact on the accuracy of the deep learning model. In order to solve the above problems, this paper proposes a series of optimization methods to reduce the accuracy loss caused by differential privacy itself, including random sampling, gradient tailoring based on pre-training threshold, gradient-based random perturbation, and advanced privacy budget statistics technology. According to the characteristics of the deep learning model and its actual training process, the differential privacy is adjusted appropriately to enhance the security of the model as far as possible without significant impact on the effect of the model in our research.

For the training process of neural network, there are two position values which directly determine the final model effect, that is, the error value calculated from loss function and the gradient update value. Therefore, the training of deep learning model based on differential privacy can consider adding noise from the two aspects, which is by adding noise to loss function or adding noise to gradient value in the process of back propagation.

In this paper, we add noise to the gradient value of the gradient back propagation process. Specifically, the normalized gradient is noised by using the Gaussian mechanism, and the gradient is updated after the disturbed gradient is obtained. Figure 1 illustrates the use of a Gaussian mechanism to add corresponding noise to each gradient calculated by back propagation.

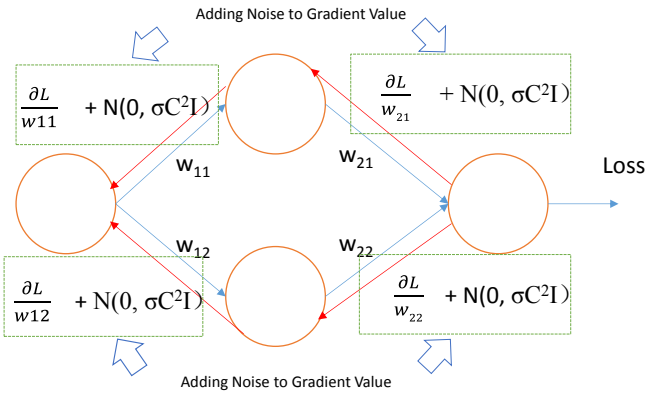


Fig. 1. Adding Noise to Gradient Value of Back Propagation Process

Based on the above discussion and analysis, the gradient-based differential privacy optimizer for deep learning model is elaborated by the following step.

Constructing the random sample. Firstly, each iteration process randomly sampled batch data B_t from the training set in a certain proportion q . Then the forward propagation process is executed normally. According to the output of the model and the classification label of the samples, as well as the task-related loss function, the loss value of the model L on the batch data B_t is calculated.

Gradient calculation. The gradient information of each sample x_i is separated from the total gradient information by using formula 4 ,when the gradient information is calculated according to the loss function.

$$g_t(x_i) = \nabla_{\theta_t} L(\theta_t, x_i) \tag{4}$$

Gradient tailoring. Because the gradient values of different parameters of loss function may vary greatly, the sensitivity value needed in differential privacy mechanism will be too large and noise will be introduced too large, so gradient information needs to be tailored. Through the pre-training model, the gradient information and the threshold parameters C can be statistically determined, and then the gradient can be tailored based on the gradient to ensure that the gradient values of all parameters are not more than C by using formula 5.

$$\overline{g}_i(x_t) = g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{C}) \tag{5}$$

Gradient interference. In order to avoid exposing information about a single training sample to model parameters, differential privacy algorithm based on Gauss mechanism is proposed to interfere with gradient information. According to the definition C of the previous step, the variance of the Gauss distribution $\partial = CI^2$ can be obtained. Therefore, the gradient value after interference is denoted by the formula 6, where I represents the unit matrix and $N(0, \sigma^2 C^2 I)$ represents a Gauss distribution with a mean value of 0 and a variance of $\sigma^2 C^2 I$. Note that in the previous step, the gradient tailoring is based on the gradient of the parameters of a single sample, where noise addition is added to the gradient of the parameters of a single sample. However, in order to ensure the stability of the training, the training algorithm uses SGD method, so the final gradient is the average value of the gradient of the disturbed samples in all batches.

$$\overline{g}_i = \frac{1}{|B_t|} \sum_i (\overline{g}_i(x_t) + N(0, \sigma^2 C^2 I)) \tag{6}$$

Parameter updating. Finally, the updated values of the parameters based on the gradient after interference are shown in formula 7, where η_t is the learning rate of this iteration.

$$\theta_{t+1} = \theta_t - \eta_t * \overline{g_i} \tag{7}$$

Privacy Budget Statistics. At the same time, we need to accumulate the privacy consumption caused by the iteration. By designing the gradient-based differential privacy optimizer in each iteration of deep learning training, the attacker can not get the information of any sample by collecting gradient-related information, which consequently protects the privacy of the model and training data. Specifically, the differential privacy mechanism adds noise δ to the gradient in the training process. According to the definition of differential privacy, it can be obtained as shown in formula 8.

$$\Pr[g(x) \in O] \leq \Pr[g(x \cup d) \in O] + \delta \tag{8}$$

It can be concluded that the smaller the privacy budget ε is, the more similar the gradient statistical behavior based on the training data set I and I', which makes it impossible for attackers to accurately judge whether the data set is I or I' according to the gradient information of the model. In addition, because data sets I and I' are adjacent data sets, attackers can not judge whether gradient information comes from adjacent data sets I or I', which means that the differential privacy mechanism guarantees that attackers can not determine every sample in training data sets I (or I'), which fundamentally guarantees the privacy of training data.

Based on the above analysis, no matter how strong the background knowledge of the attacker is, it can guarantee the indistinguishability of the individual in probability. Therefore, the above-mentioned method based on differential privacy can ensure that malicious attackers can not infer the sample information in the training set by querying the final model, thus ensuring the privacy of the model and training data, so as to enhance the security of the model.

C. Differential Privacy-Preservation Based on Collaborative Training Mode

In this subsection, we present gradient-based differential privacy optimizer for deep learning model from the aspects of physical and logic implementation.

Specifically, we design a hybrid CPU-GPUs system physically shown in the Figure 2, which utilize the parallelism scheme for gradient-based differential privacy optimizer. The gradient values trained on a single GPU can be distributed by adding noise using the gradient-based differential privacy algorithm to obtain the gradient values after adding perturbations. Finally, each GPU computing resource sends the calculated gradient value with noise back to the main computing unit for merging and completing an update process. Moreover, due to the support of differential privacy parallelism theory, the final model is guaranteed to satisfy differential privacy.

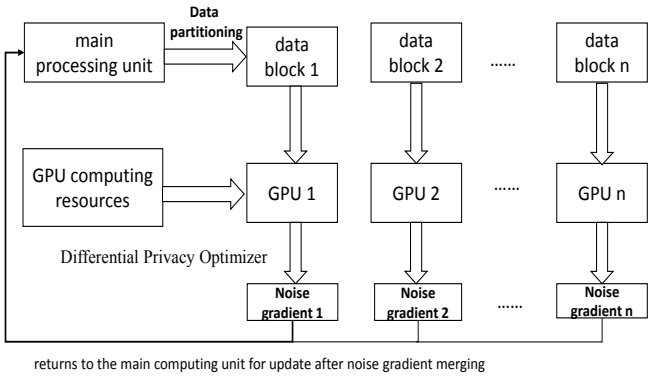


Fig. 2. Parallel Differential Privacy Training Process

Differential privacy technology satisfies the parallel theory on disjoint data sets, that is, adding noise of the same privacy level on disjoint data sets separately will not lead to the accumulation of total privacy loss. Therefore, a collaborative differential privacy training method logically is proposed logically, which is shown in Figure 3. By making full use of heterogeneous computing resources, the computing speed of the gradient-based differential privacy for model is accelerated. It is mainly divided into six processes as following.

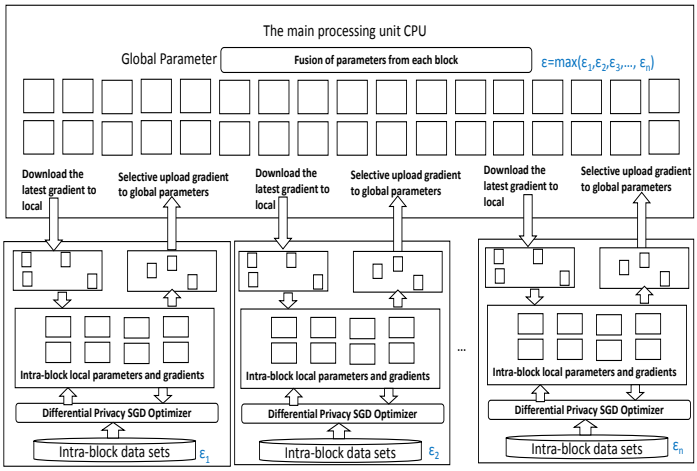


Fig. 3. Privacy Preservation for Deep Learning Model Based on Collaborative Training

Initialization. The main processing unit maintains a global model, but does not participate in the specific calculation. It downloads the global parameters for each computing resource, receives gradient updates from each computing resource, and completes the merging.

Data Partition. According to the actual situation of computing resources, the data are divided into appropriate sizes and distributed to the computing unit for calculation. In special, this step mainly assign data unit to the GPUs in order to perform computation of differential privacy-preservation in parallelism.

Downloading Global Gradient. Before each calculation, each computing unit downloads the latest global parameters from the main processing unit to local, replacing the local original parameters.

Updating Local Parallel. After receiving the data and the latest global parameters, each computing unit begins to

optimize the objective function by using the differential privacy optimizer in parallel. Then, the differential privacy optimizer is used to calculate gradients and ensure that each computing unit can satisfy its own differential privacy ε_i. Finally, the overall differential privacy budget ε of the deep learning model is satisfied as shown in Formula 9.

$$\varepsilon = \max_i \{\varepsilon_1, \varepsilon_2, ..., \varepsilon_n\} \quad (9)$$

Uploading Selective Gradient. After the calculation is completed, some or all gradient information is selected and uploaded to the main processing unit. In our study, we choose to upload the gradient values which vary greatly, so as to reflect the impact of data on the model.

Merging and Updating. The main processing unit receives the updates from all computing units in the current round, merges the gradients and updates them to the global parameters.

IV. Experimental Results and Analysis

A. Experimental Environment and Data Set

In this subsection, specific experiments are conducted on CPU-GPU hybrid systems. Analysis and verification of the gradient-based differential privacy optimizer for deep learning model are illustrated.

The configuration of hardware platform in the experiment is shown as Table 1. In special, NVCC is used in our experiment, NVCC is used as a compiler to compile GPU-related code, and GCC is used as a compiler to compile C++ code. TensorFlow is implemented in our experiment.

TABLE I. Experimental Environment

Item	*Specification*
CPUs	1*(Xeon E5)
System Memory	256 GB RAM
Operating System	Ubuntu14.04
GPUs	4*Tesla
#CUDA Cores	3584
GPU Core Frequency	1480MHz
GPU Memory	16GB
CUDA	7.5

The data set used in the experiment is MNIST (Handwritten Numbers). The content of MNIST data set is a handwritten number containing 10 categories, each corresponding to an Arabic number. The data set contains a total of 70,000 grayscale images (depth 1 and color image depth is 3), of which 60,000 training pictures, 10,000 test pictures. Each picture consists of 28*28 pixels, and each pixel is represented by a gray value.

The structure of the deep learning network used in the experiment is a feedforward neural network with depth of 3. The node of the hidden layer is 1000, the activation function is ReLU, and Softmax is a 10-class cross-entropy classifier(corresponding to 10 digits). The features of input layer are selected by principal component analysis (PCA), and the similarity threshold is less than 10%.

B. Experimental Analysis of the Proposed Method

In order to verify the effect of ε on accuracy, we set different parameters σ and ε in our experiment separately. In special, different combinations of σ, ε and δ are used to analyze the impact on accuracy.

Based on the above data sets, the gradient threshold C is set to 4 and the PCA is set to 60 dimensions. Based on the differential privacy theory, privacy budget reflects the degree of privacy protection. The smaller the value of privacy budget is, the higher the degree of privacy protection is, and the value range of privacy budget is 0-10 generally. The availability of the proposed method is measured by changing the privacy budget ε, privacy deviation δ, and noise adding scale σ.

The experiment was divided into three groups. The first group was fixed ε with 2, δ with 10^{-5}, and changed σ with value range of 1-10. The corresponding experimental results were shown in Figure 4. Specifically, Figure 4 shows that the accuracy increases firstly and then decreases with the increase of noise scale. That is, there is an inverted U-shaped relationship between them.

The second group was fixed σ with 4, ε with value range of 0.1-10, and δ with value range of 10^{-5}-10^{-1}. The corresponding experimental results were shown in Figure 5. Fig. 5 shows that there is a positive correlation between ε and δ, and the accuracy increases with the increase of ε and δ. Furthermore, the impact of ε on accuracy is greater than that of ε.

The third group was fixed δ is 10^{-5}, ε with 0.50, 2.00 and 8.00 respectively, σ with 8, 4 and 2 respectively. The experimental results of group 3 are shown in Fig. 6.

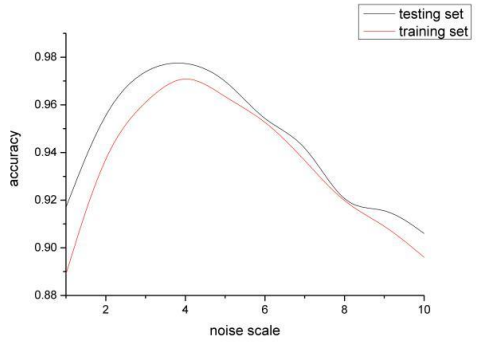


Fig.4 Effect of the noise scale change on accuracy

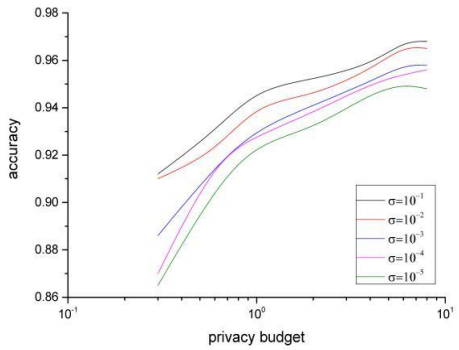
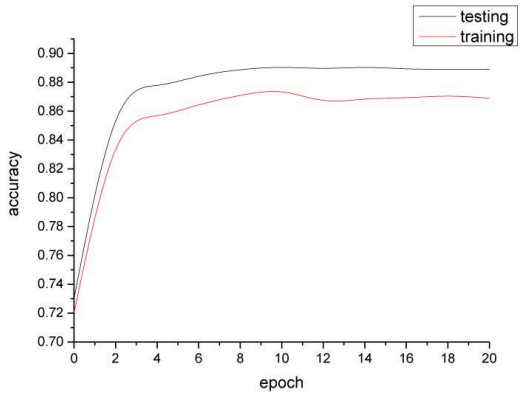
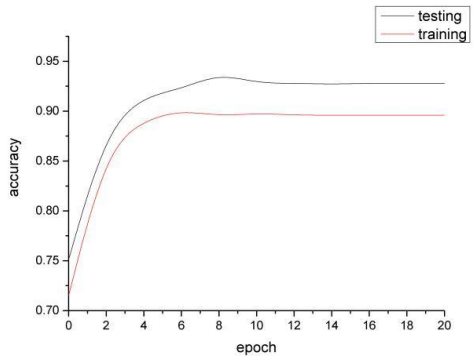


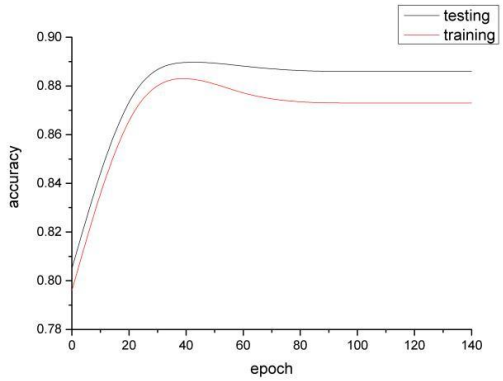
Fig.5 Effect of the noise scale change on accuracy



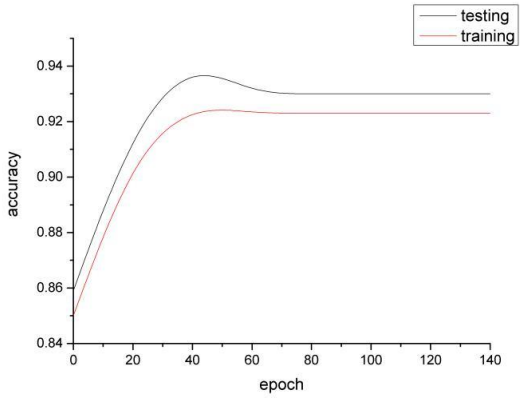
(1) $\sigma = 8, \varepsilon = 0.5$



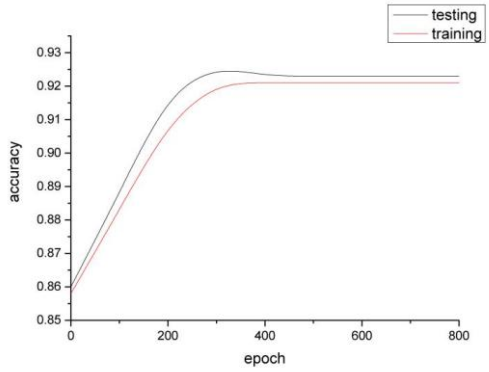
(2) $\sigma = 8, \varepsilon = 2.00$



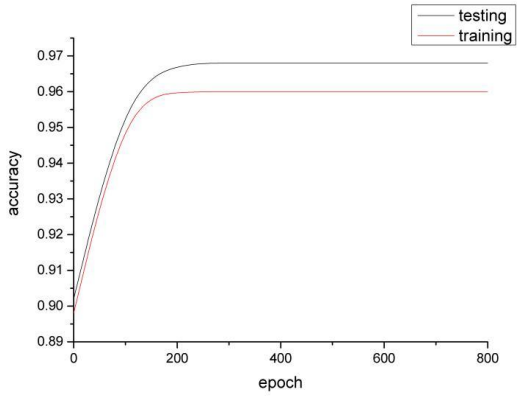
(3) $\sigma = 4, \varepsilon = 0.5$



(4) $\sigma = 4, \varepsilon = 2.00$



(5) $\sigma = 2, \varepsilon = 2.00$



(6) $\sigma = 2, \varepsilon = 8.00$

Fig.6 Effect of the noise scale and privacy budget change on accuracy

Figure 6 shows that the change of noise addition σ and privacy protection budget ε affect the accuracy of deep learning model. Under different scale of noise, when the parameters are $(8,2,10^{-5})$, $(4,2,10^{-5})$, and $(2,8,10^{-5})$, the accuracy reached 93.8%. 94.2% and 96.8% respectively. On the one hand, when σ is the same, ε is positively correlated with model accuracy.

According to the experimental results in Fig. 6, with the increase of differential privacy budget, the increment of noise is decreasing and the iteration period of training is increasing. Specifically, when the values of (ε , δ , σ) are (2, 10^{-5} , 8), the accuracy of model using gradient-based differential privacy reaches 96% compared with the accuracy of model without gradient-based differential privacy reaching 96.8%. The detailed experimental data show that the accuracy of model using gradient-based differential privacy is slightly decreased, but the availability of the model is still high. And the selected parameters can achieve a balance between privacy protection and data availability.

In our experiment, we utilize mini-batch style to train the deep learning model, for accelerating the convergence speed and the training efficiency. Figure 7 shows the speedup ratio of the MINIST data sets for collaborative differential privacy training on the CPU-GPUs hybrid system compared with the general CPU system. Overall, the data in Figure 5 show that the speedup ratio of the model has improved. Furthermore, we can conclude that the collaborative training method in this paper is efficient. Specifically, the speedup ratio can reach up to 23.5%. It also shows that when the batch value is 128, the speed-up ratio is significantly larger than other values.

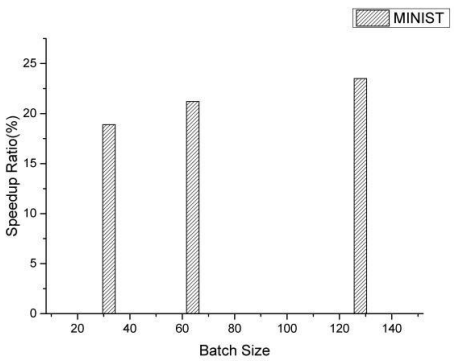


Fig. 7. Speedup ratio of the model

C. Discussions

The main problem of cloud security is the inefficiency of general data encryption scheme and the training efficiency due to the high complexity of the deep learning algorithm and the inefficient use of system architecture. In order to guarantee the security of deep learning model while improving the training efficiency of the system, this paper aims at utilizing the differential privacy-preserving for deep learning model. In special, we design a gradient-based differential privacy optimizer using collaborative training mode based on CPU-GPUs hybrid system in cloud. By utilizing the gradient-based differential privacy optimizer for deep learning model simultaneously guarantee the privacy of deep learning model and improve the training efficiency. Moreover, the experimental results show that the accuracy of the proposed method in our study is feasible and efficient. And the speedup ratio of 20% to 30% can be obtained. In the future, we may further research on how to adjust parameters to meet the needs of different users.

V. ACKNOWLEDGEMENT

We would like to thank Dr. Li at Huazhong University of Science and Technology for his suggestions on this paper. This paper is supported by National Key R&D Program No. 2018YFB1003605.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y.W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol.18, pp.1527-1554,2006.

[2] G. Hinton, L. Deng, and D. Yu, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol.29, pp.82-97,2012.

[3] W. Xiong, J. Droppo, and X. Huang, "The Microsoft 2016 conversational speech recognition system," in: Acoustics, Speech and Signal Processing (ICASSP), IEEE, Piscataway NJ ,2017, pp.5255-5259.

[4] R. Collobert, J. Weston, and L. Bottou, "Natural language processing (almost) from scratch," Journal of Machine Learning Research, pp.2493-2537,2011.

[5] W. Yih,, K. Toutanova, and J. Plat, "Learning discriminative projections for text similarity measures," in: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Stroudsburg PA, 2011, pp.247-256.

[6] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp.1-9, 2015.

[7] L. Yu, H. Chen, and Q. Dou, "Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks," IEEE Transactions on Medical Imaging, vol.36, pp.994-1004, 2017.

[8] Y. Lecu, L. Bottou, and Y. Bengio, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol.86, pp.2278-2324,1998.

[9] G. Bachrach, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," International Conference on Machine Learning, 2016.

[10] C. Dwork, "Differential privacy," in: Proceedings of the 33rd International Conference on Automata, Languages and Programming, pp.1-12, 2006.

[11] C. Dwork, "Differential privacy: A survey of results," in: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation, pp.1-19,2008.

[12] A. Martin, and C. Andy, "Deep learning with differential privacy," in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016.

[13] L. Xie, K. Lin, and S. Wang, "Differentially Private Generative Adversarial Network," arXiv preprint arXiv, 2018.

[14] M. Dianhui, L. Ziqin, and C. Qiang, "Deep Differential Privacy Protection Based on DCGAN Feedback," Journal of Beijing University of Technology, 2018.

[15] N. Papernot, M.Abadi, and U. Erlingsson, "Semi-supervised knowledge transfer for deep learning from private training data,"arXiv preprint arXiv, 2016.

[16] R. Shokri, and V. Shmatikov, "Privacy-preserving deep learning," in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, 2015.

[17] H. B. McMahan, E. Moore, and D. Ramage, "Federated learning of deep networks using model averaging," 2016.