IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 February 2023, accepted 20 February 2023, date of publication 27 February 2023, date of current version 2 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3249575

RESEARCH ARTICLE

Image Classification Based on Layered Gradient Clipping Under Differential Privacy

CHUNMEI MA, XIANGSHAN KONG, AND BAOGUI HUANG
School of Computer Science, Qufu Normal University, Rizhao 276800, China

Corresponding author: Baogui Huang (hjbaogui@qfnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62272256, Grant 61771289, and Grant 61832012; in part by the Natural Science Foundation of Shandong Province under Grant ZR2022ZD03, Grant ZR2021QF050, and Grant ZR2021MF075; in part by the Shandong Natural Science Foundation Major Basic Research under Grant ZR2019ZD10; in part by the Shandong Key Research and Development Program under Grant 2019GGX1050; and in part by the Shandong Major Agricultural Application Technology Innovation Project SD2019NJ007.

ABSTRACT Convolutional neural networks (CNNs) are widely used in the field of image classification. At the same time, users face the risk of privacy leakage because adversaries can reverse private information from the training parameters of CNNs. Adding Gaussian noise to the training parameters is an effective means to prevent adversaries from stealing private, but this tends to reduce the utility of the models. Therefore, how to find a balance between privacy and utility has become a hot research topic. In this paper, to improve the image classification ability of CNN models under differential privacy protection, we propose an image classification algorithm based on layered gradient clipping under differential privacy, ICGC-DP for short. Firstly, the gradient tensor is layered according to the neural network model. Secondly, for each layered gradient tensor, the median of L_2 norms is used as the clipping threshold. Moreover, to prevent the sensitivity from converging to zero, we add a bound on the sensitivity to ensure that all gradients can be protected by differential privacy. To further improve the classification utility of ICGC-DP, we design an adaptive weighted fusion module for it. The module assigns weights to prediction tensors according to the variance between them. We conduct comprehensive experiments on the Mnist, FashionMnist and CIFAR10 datasets, respectively. The experimental results show that, when the privacy budget $\varepsilon = 2.0$, which indicates that the algorithm adds a large noise, ICGC-DP achieves 97.36%, 88.72% and 72.63% classification accuracy for the Minist, FasionMnist and CIFAR10 datasets, respectively; when the privacy budget $\varepsilon = 8.0$, which means the algorithm adds less noise, the classification accuracy of ICGC-DP for Minist, FasionMnist and CIFAR10 datasets reaches 97.81%, 89.49% and 74.41%, respectively.

INDEX TERMS Privacy preservation, deep learning, differential privacy, gradient clipping.

I. INTRODUCTION

In recent years, convolutional neural networks have been actively developed as fast and effective image classification methods and are widely used in image recognition-related industries, especially in the medical field, such as classification and analysis of patients with lung nodules [1], and optimization of network structure to improve the recognition accuracy of small-scale motion in medical motion images [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

Because adversaries can reverse training data from the training parameters, privacy protection has become an increasingly important topic that cannot be ignored [3]. When the privacy of CNN models is stolen, adversaries often obtain the overall parameters privacy by focusing on the different information of specific individuals. When stealing model privacy, adversaries not only attack the datasets for the model, but also use membership inference attacks or model inversion attacks [4]. Membership inference attacks [5] construct attack models by obtaining the training parameters of target models, and use the labels and prediction results of target models to

determine whether a sample belongs to the training dataset of the target model. Model inversion attacks [6] are usually used to attack shallow CNNs, where adversaries use the outputs and labels of target models to reconstruct samples of training datasets for the target models.

To address the above attacks, Dwork and Aaron [7] proposed differential privacy (DP) to protect data privacy by adding random noise to the data. Using the differential privacy protection method of deep learning, Abadi et al. [8] proposed the DP-SGD algorithm by adding Gaussian noise to the gradient to achieve differential privacy. Although differential privacy can effectively protect CNN models, the classification performance of the models decreases along with the perturbation of training parameters. Therefore, it is important to find a trade-off between the classification performance of models and privacy protection. In some previous works, the impact of noise on model performance was reduced by adaptively adding noise [9], [10], [11], [12], [13]. However, reducing the perturbation of training parameters with random noise by adjusting the sensitivity in stochastic gradient descent iterations has been a research direction in recent years.

Xu et al. [14] proposed an adaptive and fast convergent learning algorithm, which improves the convergence speed by adaptive learning rate, significantly reduces the privacy cost, and minimizes the negative impact of noise on model by introducing adaptive noise. Hu et al. [15] proposed a differential privacy deep learning model based on clustering technique, which obtains the L_2 norms of each gradient layer of the model, and uses the standard deviation function to quantify the L_2 norm set to obtain a tighter sensitivity, thus reducing the impact of noise on the classification performance of the model. There are also some methods that use changing sensitivity to reduce the impact of noise on model classification performance are proposed in algorithms [16], [17], [18].

However, there are still some issues to be solved in the above works [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. First, the clipping value C is determined manually. Therefore, it is difficult to obtain a more precise clipping value. Second, as the number of training steps increases, the gradient norm decreases gradually. Therefore, the clipping strategy with a fixed clipping value C leads to distortion of the gradient information, which eventually affects the classification performance of the model.

The main contributions of this paper can be summarized as follows.

1) First, we layer the gradient tensor of each sample according to the structure of the network model with differential privacy protection, and find the L_2 norm of each gradient tensor. Then, we obtain the median of L_2 norms to determine the clipping value C_M, so that a suitable C can be determined according to the change of gradient norms. The clipping value C_M is usually used as the sensitivity of Gaussian noise. To prevent C_M from being zero, we set a bound β for the sensitivity to ensure the privacy of all training parameters. Also, the influence of noise on parameters can be reduced due to the gradual decrease of C_M.
2) It is not feasible to improve the classification ability of the model under differential privacy through increasing model layers, because increasing the number of layers of the model also increases the number of training parameters and then the noise injected on the training parameters will also increase. Therefore, to further improve the classification performance of the model, an adaptive weighted fusion module (AWF) is designed. AWF consists of two identical CNNs. For each sample, AWF obtains two prediction tensors and then calculates the variance of the predicted probabilities of the two tensors. According to the variance, different weights are assigned to the two prediction tensors.
3) We have conducted comprehensive experiments and the results illustrate that ICGC-DP has superior classification performance on the Mnist, FashionMnist and CIFAR10 datasets.

The rest of this paper is organized as follows. In Section II, we present some backgrounds of knowledge and related works about differential privacy and deep learning. In Section III, we describe our proposed algorithm in detail. In Section IV, we experimentally analyze and demonstrate the better performance of our proposed algorithm over other algorithms mentioned in this paper. Finally, we conclude this paper in Section V.

II. BACKGROUND AND RELATED WORK

A. ATTACK METHODS FOR TRAINING PARAMETERS

In the previous contents of literature, indirect attacks to obtain model privacy by attacking the training parameters of models were mentioned. The main attacks are membership inference attack and model inversion attack.

Member inference attacks use training parameters to construct an attack model and determine whether a particular sample belongs to the dataset of the target model. Chen et al. [19] studied the fragility of machine learning to membership inference attacks and evaluated the effectiveness of using differential privacy as a defense mechanism. Shi and Yalin [20] developed an active defense against membership inference attacks. This defense can successfully reduce the accuracy of membership inference attacks and prevent information leakage from wireless signal classifiers.

The model inversion attacks reverse target samples by using labels and the outputs of the target models. Usynin et al. [21] proposed a new model inversion framework that builds on gradient-based model inversion attacks, allowing the adversary to obtain enhanced reconstructions while remaining stealth. Titcombe et al. [22] demonstrated that model inversion attacks can still be successful when the adversary has limited knowledge of the data distribution, and

proposed a simple additive noise method to defend against model inversion attacks.

B. DIFFERENTIAL PRIVACY

Differential privacy is often used to protect the privacy of important data. Dwork and Aaron [7] satisfies the privacy requirement by adding an appropriate size of noise to the query results to ensure that modifying an individual record in the dataset will not significantly impact on the query results.

Definition 1: (Sensitivity [23]). Given a query function $f : D \to R^d$, *where* D *is the given dataset and* R^d *is a* d *-dimensional real vector representing the query result of the query function* f *on dataset* D, *there exists a pair of adjacent datasets* D *and* D'. *The sensitivity is defined as* $\Delta f = \max ||f(D) - f(D')||_2$, *where* $||f(D) - f(D')||_2$ *is the Euclidean distance between* $f(D)$ *and* $f(D')$.

Definition 2: (Gaussian Mechanism). Add Gaussian noise to the query function $f : D \to R^d$ *to obtain a random algorithm A. A is defined as* $A(D) \triangleq f(D) + N(0, \Delta f^2 \cdot \sigma^2)$ *where* $N(0, \Delta f^2 \cdot \sigma^2)$ *is the normal (Gaussian) distribution,* 0 *is the mean and* $\Delta f \cdot \sigma$ *is the standard deviation. The randomized algorithm A satisfies* (ε, δ) *- differential privacy, if* $\Pr[A(D) \in O] \leq e^{\varepsilon} \Pr[A(D') \in O] + \delta$ *holds, where* ε *is the privacy budget,* O *is the output set,* δ *is an additive term, and* δ *is preferably smaller than* $1/|D|$, $|D|$ *is the number of samples.*

C. DEEP LEARNING UNDER DIFFERENTIAL PRIVACY PROTECTION

Combining differential privacy with deep learning is a popular means of protecting deep learning models and their training parameters.

Abadi et al. [8] proposed the DP-SGD algorithm in order to prevent the privacy of a model from being leaked. In the training process of CNN, to prevent individual gradient from leaking privacy to the overall training parameters, a gradient threshold C is set, and for each sample of a batch, the model calculates the L_2 norm $\|g\|_2$ of its gradient g. If $\|g\|_2$ is greater than C, the gradient vector g is replaced by $g \Big/ \max\left(1, \frac{\|g\|_2}{C}\right)$. Then, all gradients of samples in the batch are aggregated and then Gaussian noise is added.

Yu et al. [16] proposed the Improved-DP-SGD algorithm, using noisy gradient as momentum during the training process to facilitate training. The gradient magnitude converges to zero when the algorithm converges. Based on this fact, the noise injected on the gradient is reduced by reducing the gradient norm during training.

Liu et al. [17] proposed the DPL-GGC algorithm, a strategy for clipping the gradient tensor of each sample by dividing it into a given group, introducing a smoothing sensitive mechanism with differential privacy protection. In this way, it imposes a limit on the joined Gaussian noise.

The above proposed methods are some related algorithms that combine differential privacy with deep learning.

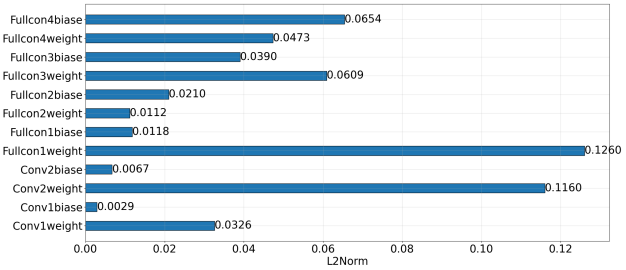


FIGURE 1. L_2 **norms of the gradient tensor for each layer.**

III. OUR ALGORITHM

In this section, the details of our algorithm will be explained. We choose the Gaussian noise for differential privacy protection of CNN.

A. LAYERED GRADIENT CLIPPING

Some previous algorithms that protect model privacy by adding Gaussian noise to the gradient usually set a fixed C to clip the gradient, the clipping bound strategy is set in lemma 1.

Lemma 1 (Clipping Bound [8]): To satisfy differential privacy, we set a clipping threshold C, *constraining the effect of each* $g(x_i)$ *on the whole. We clip the gradient of each sample with* L_2 *norm. If* $\|g(x_i)\|_2 \leq C$, $g(x_i)$ *is retained, otherwise, the gradient vector* $g(x_i)$ *is replaced by* $g(x_i) / \max\left(1, \frac{\|g(x_i)\|_2}{C}\right)$. *The clipping process can be denoted by the following formula.*

$$\bar{g}_t(x_i) = g_t(x_i) / \max\left(1, \frac{\|g_t(x_i)\|_2}{C}\right). \tag{1}$$

However, [16] shows the norms of gradients tend to decrease as the number of training steps t increases, and as shown in Figure 1, the L_2 norm of each gradient layer is different.

Therefore, it is not reasonable to choose a fixed C value for global gradient clipping. Because, first, the fixed C value cannot adaptively clip gradually decreasing gradients. Second, the fixed C value does not provide a suitable clipping according to the different gradient norm of each layer.

As shown by the above reasons, global gradient clipping with fixed clipping value leads to distortion of gradient information. Moreover, a model with differential privacy protection accumulate loss in the backpropagation process, leading to a decrease in the classification performance of the model.

So, it is important to set a suitable clipping value to perform layered clipping of gradients in the process of training step accumulation.

To solve the above problem, we propose a layered gradient clipping method. As shown in Figure 2, we divide the global gradient tensor g_i of sample x_i into a total of k layers according to the model structure and denote them as $\{g_i^1, \cdots, g_i^k\}$.

Algorithm 1 Get C_M (Get the Clipping threshold)

Input: The gradient tensor $g_t(x_i)$ of sample x_i;
Output: C_{Mi};
Layer gradient tensor;
1: Divide $g_t(x_i)$ into k layers;
$g_t(x_i) = \left[g_t^1(x_i), \cdots, g_t^k(x_i)\right]$;
2: For each $g_t^j(x_i) \in g_t(x_i)$, obtain the L_2 norm $\left\|g_t^j(x_i)\right\|_2$;
3: Sort $g_t^j(x_i)$ in ascending order of $\left\|g_t^j(x_i)\right\|_2$, $j \in \{1, \cdots, k\}$, obtain the set S of L_2 norms;
$S = \{\left\|g_i^1\right\|_2, \cdots, \left\|g_i^k\right\|_2\}$;
Calculate clipping threshold;
4: $C_{Mi} = \frac{1}{2}\left(\left\|g_i^{\frac{k}{2}}\right\|_2 + \left\|g_i^{\frac{k}{2}+1}\right\|_2\right)$;
5: **return** C_{Mi}.

After that, L_2 norm is obtained for each of these gradient tensors. Then, sorting these gradients according to their L_2 norm to obtain the L_2 norm set $S = \{\left\|g_i^1\right\|_2, \cdots, \left\|g_i^k\right\|_2\}$, where $\left\|g_i^k\right\|_2 \geq \left\|g_i^{k-1}\right\|_2 \geq \cdots \geq \left\|g_i^1\right\|_2$.

To prevent excessive gradients in each layer from leaking parameter privacy, the median of the L_2 norm set of gradients in each layer is set as the clipping threshold C_{Mi}.

$$C_{Mi} = \frac{1}{2}\left(\left\|g_i^{\frac{k}{2}}\right\|_2 + \left\|g_i^{\frac{k}{2}+1}\right\|_2\right). \tag{2}$$

Since each layer of the network model corresponds to a weight gradient and a bias gradient, k is even. Next, each layer of the gradient tensor g_i^k will be clipped by C_{Mi}:

$$\bar{g}_t^k(x_i) = g_t^k(x_i) / \max\left(1, \frac{\left\|g_t^k(x_i)\right\|_2}{C_{Mi}}\right). \tag{3}$$

The above process of getting the clipping threshold C_{Mi} is shown in Algorithm 1.

B. ADDING NOISE

In order to prevent leakage of model training parameters, Gaussian noise is added to the parameters. DP-SGD [8] is a classical deep learning noise addition algorithm. It uses a clipping threshold C as fixed sensitivity to add noise on the aggregated gradient tensor.

$$\bar{g}_{t\_noise} = \frac{1}{L}\sum_i (\bar{g}_t(x_i) + N(0, \sigma^2 C^2 I)). \tag{4}$$

where L is the size of the sample batch, σ is the noise multiplier, I is the unit matrix and N is the average noise of the injected aggregation gradient.

However, the consumption of privacy budget ε increases as the training step t increases, resulting in more and more noises injected into the training parameters. This eventually leads to a decrease in the classification ability of the model.

As the number of training steps increases, the gradient norm shows a decreasing trend, and the C_{Mi} will gradually decrease. So, in order to reduce the impact of noise on the classification performance of the model, we can use this property of C_{Mi} to replace C as the sensitivity of the noise.

Also, to prevent sensitivity from converging to zero, resulting in no noise added on the gradient parameter, we set a tiny hyperparameter β and let.

$$\overline{C_i} = C_{Mi} + \beta. \tag{5}$$

The noise addition of $\overline{C_i}$ is shown in Equation (6), where $\overline{N}$ is the average noise injected on the aggregation gradient after changing the sensitivity.

$$\bar{g}_{t\_noise} = \frac{1}{L}\sum_i (\bar{g}_t(x_i) + \bar{N}(0, \sigma^2 \overline{C}_i^2 I)). \tag{6}$$

C. ADAPTIVE WEIGHTED FUSION MODULE

Since adding more network layers to the model adds more gradient parameters, resulting in more noise injected into the model. Thus, it is not an effective approach to improve the classification performance of the model under differential privacy by increasing the number of model layers.

To further improve the classification performance of the model with differential privacy protection, we design an adaptive weighted fusion module, shortly named AWF, the overall flow of the AWF module is shown in Figure 3.

The AWF module performs an adaptive weighted fusion of the prediction tensor y_i obtained form two identical models with differential privacy protection.

It assigns weights w to the prediction tensors according to the magnitude of their variance V, Equations (7) and (8) represent the calculation of the predicted probabilities variance in the prediction tensor.

$$\overline{p} = \frac{1}{n}\sum_{b=1}^{n} p_b. \tag{7}$$

$$V = \frac{1}{n}\sum_{b=1}^{n} (p_b - \overline{p})^2. \tag{8}$$

where n is the number of prediction categories. Because the prediction tensor with more accurate prediction results has more dispersion of the prediction probabilities p_b in its matrix structure. And the prediction tensor with poorer predictions has a less discrete prediction probabilities p_b in the matrix structure. Due to the magnitude of the variance is proportional to the dispersion, the variance of the prediction tensor can reflect the accuracy of its prediction results.

Next, the prediction tensor is assigned a weight w, which is calculated according to the variance ratio of the prediction tensor, see Equation (9). Where, V_1 and V_2 represent the variances of the prediction tensors y_i^1 and y_i^2, respectively. Then, the weighted prediction tensors $w_1 y_i^1$ and $w_2 y_i^2$ are fused to obtain the fusion tensor Y, see Equation (10).

$$w_1 = \frac{V_1}{V_1 + V_2}, w_2 = \frac{V_2}{V_1 + V_2}. \tag{9}$$

$$Y = w_1 y_i^1 + w_2 y_i^2 \tag{10}$$

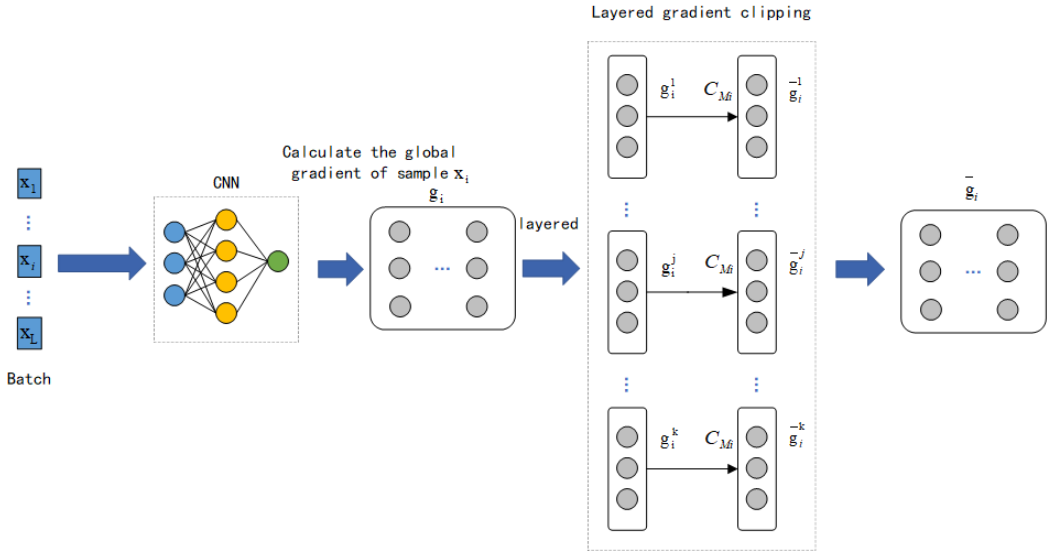


FIGURE 2. Gradient tensor layered clipping flow diagram.

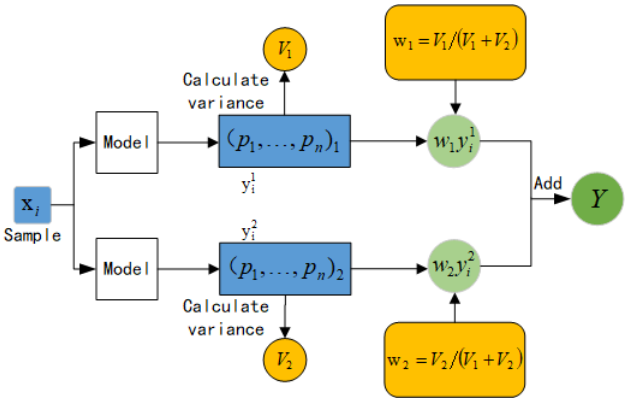


FIGURE 3. AWF module flow diagram.

At this point, a more accurate prediction result can be obtained for the model with differential privacy protection.

The overall flow of the above is shown in Algorithm 2.

IV. EXPERIMENT

In this section, we conducted several experiments on different datasets as well as privacy budgets to evaluate the performance and privacy of ICGC-DP. We compare DP-SGD [8], DPL-GGC [17], Improved-DP-SGD [16] and ICGC-DP, respectively, under the same number of training steps. In the following, we first describe the experimental settings, including the experimental datasets, models, and implementation details. Then, we compare the classification performance of ICGC-DP with the other algorithms mentioned in this paper.

A. EXPERIMENTAL SETTINGS

We perform experimental evaluation on three public datasets respectively.

a) *Datasets*. The Mnist dataset [24] contains 60,000 training samples and 10,000 test samples with a total of 10 categories. The FashionMnist dataset [25] also contains 60,000 training samples and 10,000 test samples, with 10 categories. The CIFAR10 dataset [26] consists of 32 × 32 colour images with 10 categories. It contains 50,000 training samples and 10,000 test samples, respectively.

b) *Models*. In our experiments, we set up two models, Model1 and Model2, respectively.
Model1 is used to train and test the Mnist and FashionMnist datasets, and it has 2 convolutional layers with 10 and 20 neurons, each with a dimension of 5×5, and 4 fully connected layers, fllowing by 1 softmax layer.
Model2 is used to train and test the CIFAR10 dataset, and its model structure has 2 convolutional layers with 64 and 64 neurons, each with a dimension of 5 × 5, 2 fully connected layers and 1 softmax layer.

c) *Parameter settings*. In the next experiments, we set the parameters for Model1, Model2 and the privacy environment, respectively. For Model1, we set the sampling rate q to 0.01, the learning rate α to 0.005, δ to 10^{-5}, and β to $1/T^2$. For Model2, we set the sampling rate q to 0.01, the learning rate α to 0.01, δ to 10^{-5}, and β to $1/T^2$.

All our experiments are run in the pytorch environment.

B. PRIVACY COST

To evaluate the trade-off between privacy cost and accuracy of DP-SGD, DPL-GGC, Improved-DP-SGD and ICGC-DP algorithms, we compare the accuracy of the algorithms by consuming a certain privacy budget ε. To ensure the fairness of the experiments, the model structures of different algorithms are the identical on the same dataset.

Algorithm 2 ICGC-DP Algorithm

Input: Training dataset $X = \{x_1, x_2, \ldots x_U\}$, where U represents the number of training samples, loss function $f(\theta, x_i)$, Parameters: Noise multiplier σ, batch size L, learning rate α, hyperparameter β, sampling rate q, privacy parameters:(ε, δ);

Output: training parameters with differential privacy protection θ_t;

for $t \in [T]$ **do**

Randomly select L samples with sampling probability L/U, all samples constitute the sub-dataset L_t;

Adaptive weighted fusion Module;

For each $x_i \in L_t$, obtain the prediction tensor y_i^1 and y_i^2;

$\overline{p} = \frac{1}{n}\sum_{b=1}^{n} p_b$;

$V = \frac{1}{n}\sum_{b=1}^{n} (p_b - \overline{p})^2$;

$w_1 = \frac{V_1}{V_1+V_2}, w_2 = \frac{V_2}{V_1+V_2}$;

$Y = w_1 y_i^1 + w_2 y_i^2$;

Compute gradient;

For each $x_i \in L_t$, $g_t(x_i) = \nabla_{\theta_t} f(\theta_t, x_i)$;

Clip gradient;

$C_{Mi} = \text{Get } C_M\ (g_t(x_i))$;

$\bar{g}_t^k(x_i) = g_t^k(x_i) / \max\left(1, \frac{\|g_t^k(x_i)\|_2}{C_{Mi}}\right)$;

Add noise;

$\overline{C_i} = C_{Mi} + \beta$;

$\bar{g}_{t\_noise} = \frac{1}{L}\sum_i (\bar{g}_t(x_i) + \bar{N}(0, \sigma^2 \bar{C}_i^2 I))$;

Gradient descent;

$\bar{\theta}_{t+1} = \bar{\theta}_t - \alpha \bar{g}_{t\_noise}$;

end for

return θ_t.

TABLE 1. Privacy cost comparisons on Mnist, $\delta = 10^{-5}$.

Algorithm	ε	Accuracy(%)
DP-SGD	1.92	93.46
DPL-GGC	1.72	94.67
Improved-DP-SGD	0.51	96.71
ICGC-DP	**0.51**	**97.05**

As shown in Table 1, on the Mnist dataset, ICGC-DP and Improved-DP-SGD only consume a privacy budget of 0.51, and their accuracy reaches 97.05% and 96.71%, respectively. DPL-GGC consumes a privacy budget of 1.72, and the accuracy of DPL-GGC reaches 94.67%. DP-SGD consumes a privacy budget of 1.92, and the accuracy of DP-SGD reaches 93.46%.

As shown in Table 2 and Table 3, on the CIFAR10 and FashionMnist datasets, ICGC-DP consumes privacy budgets of 3.29 and 1.40, its accuracy reaches 73.13% and 89.11%, respectively. DP-SGD consumes privacy budgets

TABLE 2. Privacy cost comparisons on CIFAR10, $\delta = 10^{-5}$.

Algorithm	ε	Accuracy(%)
DP-SGD	6.85	60.97
DPL-GGC	4.68	68.46
Improved-DP-SGD	3.38	70.85
ICGC-DP	**3.29**	**73.13**

TABLE 3. Privacy cost comparisons on FashionMnist, $\delta = 10^{-5}$.

Algorithm	ε	Accuracy(%)
DP-SGD	3.97	82.46
DPL-GGC	2.57	85.65
Improved-DP-SGD	1.79	88.09
ICGC-DP	**1.40**	**89.11**

of 6.85 and 3.97, its accuracy reaches 60.97% and 82.46%, respectively. DPL-GGC consumes privacy budgets of 4.68 and 2.57, its accuracy reaches 68.46% and 85.65%, respectively. Improved-DP-SGD accuracy reaches 70.85% and 88.09%, and it consumes privacy budgets of 3.38, 1.79 respectively.

It can be seen from Tables 1, 2 and 3 that ICGC-DP improves the accuracy of the model and reduces the overall privacy consumption. This is because our method adds less noise in the same steps.

C. ACCURACY COMPARISON

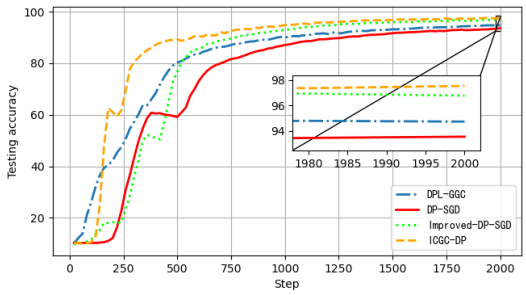
In this subsection, we compare the accuracy of different algorithms on the Mnist, CIFAR10 and FashionMnist datasets under different privacy budgets ε.

Figure 4 shows the comparison results of DP-SGD, DPL-GGC, Improved-DP-SGD and ICGC-DP on the Mnist dataset.

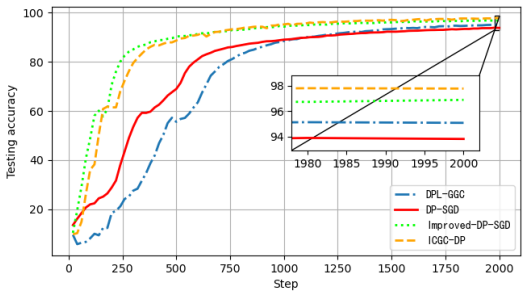
Figure 4(a) shows that ICGC-DP has better classification performance than DP-SGD, DPL-GGC, and Improved-DP-SGD for the same model. When the privacy budget $\varepsilon = 2.0$, which indicates that a large Gaussian noise is injected on the training parameters, ICGC-DP achieves test accuracy of 97.36%, while DP-SGD, DPL-GGC and Improved-DP-SGD obtain 93.53%, 94.72% and 96.77% accuracy, respectively.

When the privacy budget $\varepsilon = 8.0$, as shown in Figure 4(b), ICGC-DP exceeds the test accuracy by 3.92%, 2.68%, and 0.90% on average compared to DP-SGD, DPL-GGC, and Improved-DP-SGD. When the privacy budget is set large, which is equal to injecting less noise on the training parameters, the test accuracy gaps among the algorithms decreases.

Figure 5 shows the test accuracy of each algorithm on the CIFAR10 dataset. Figure 5(a) shows that ICGC-DP outperforms DP-SGD, DPL-GGC and Improved-DP-SGD. When the privacy budget $\varepsilon = 2.0$, the test accuracy of ICGC-DP is 72.63%, which exceeds the test accuracy by 15.50%,

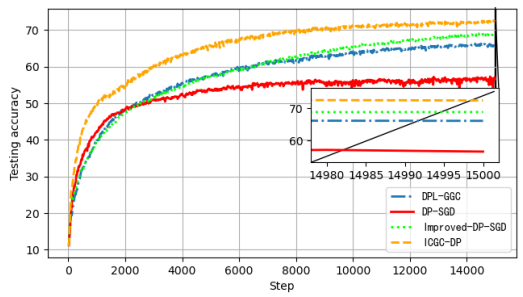


(a) $\varepsilon = 2.0$

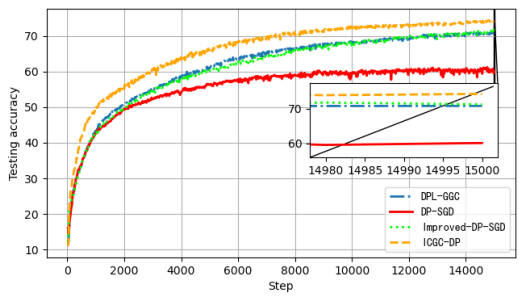


(b) $\varepsilon = 8.0$

FIGURE 4. Accuracy comparisons on Mnist.

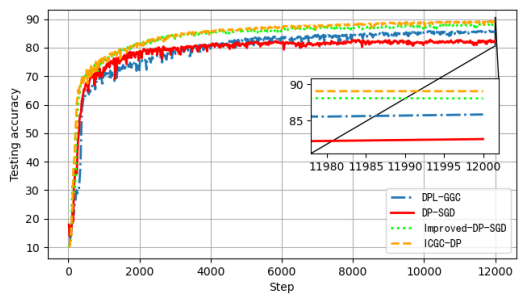


(a) $\varepsilon = 2.0$

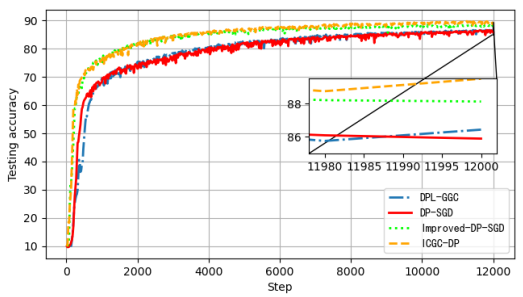


(b) $\varepsilon = 8.0$

FIGURE 5. Accuracy comparisons on CIFAR10.



(a) $\varepsilon = 2.0$



(b) $\varepsilon = 8.0$

FIGURE 6. Accuracy comparisons on FashionMnist.

6.51%, and 3.67% compared to the DP-SGD, DPL-GGC and Improved-DP-SGD.

In Figure 5(b), when the privacy budget $\varepsilon = 8.0$, the test accuracy of ICGC-DP, DP-SGD, DPL-GGC and Improved-DP-SGD is 74.41%, 61.15%, 70.82% and 71.27%, respectively.

Figure 6(a) shows the comparison of the test accuracy among the algorithms for privacy budget $\varepsilon = 2.0$ on the FashionMnist dataset. We conclude that ICGC-DP outperforms the other algorithms, compared to DP-SGD, DPL-GGC and Improved-DP-SGD, the final test accuracy of ICGC-DP increases by 6.60%, 3.16% and 0.98%, respectively.

Moreover, as shown in Figure 6(b), it is clear that ICGC-DP achieves 89.49% test accuracy on the FashionMnist dataset if privacy budget $\varepsilon = 8.0$, while the accuracy of DP-SGD, DPL-GGC and Improved-DP-SGD algorithm is 85.89%, 86.43% and 88.12%, respectively.

Figures 4, 5 and 6 show that the larger the allocated privacy budget, the less noise affects the model, and also demonstrate that ICGC-DP outperforms other algorithms mentioned in this paper on different datasets.

D. COMPUTATIONAL EFFICIENCY COMPARISON

Furthermore, we evaluate the average time consumption of DP-SGD, DPL-GGC, Improved-DP-SGD and ICGC-DP on the Mnist, FashionMnist and CIFAR10 datasets, respectively.

DP-SGP is a classical algorithm, and the accuracy of the other algorithms mentioned in the paper is still growing when it converges at the same number of steps. Therefore, we take the accuracy of DP-SGD when it reaches convergence as

TABLE 4. Comparison of the average time consumption (minutes).

Dataset	DPL-GGC	DP-SGD	Improved-DP-SGD	ICGC-DP
Mnist	9.73	9.93	7.06	**8.47**
FashionMnist	18.86	19.17	12.74	**33.60**
CIFAR10	17.43	34.09	20.74	**30.70**

a reference object and use the time required for other algorithms to reach the similar accuracy as it as a measure of computational efficiency.

For example, On the Mnist dataset, DP-SGD converges with accuracy of 93.43% and its average time consumption is 9.93 minutes. DPL-GGC achieves accuracy of 93.49% and its average time consumption is 9.73 minutes. Improved-DP-SGD achieves accuracy of 93.53% and its average time consumption is 7.06 minutes. ICGC-DP achieves accuracy of 93.66% and its average time consumption is 8.47 minutes.

The comparison of the time consumption on the different datasets is shown in Table 4. ICGC-DP has a longer time consumption than other algorithms on FashionMnist and CIFAR10 datasets. This is because the AWF module in the ICGC-DP needs to calculate the number of parameters of both models, which brings a large amount of computation for each iteration of the ICGC-DP. In fact, the AWF module can run in parallel. This way, the time consumption of ICGC-DP can be theoretically reduced by about half compared to that on Table 4. As shown in Figure 4, on the Mnist dataset, ICGC-DP requires fewer steps to achieve similar accuracy as when DP-SGD converges, thus its time consumption is less.

V. CONCLUSION

In this paper, our proposed algorithm improves the problems that previous works failed to address. First, we layer the gradient tensor and obtain L_2 norm for the gradients to determine the clipping value C_M. In addition, in order to prevent the gradient norm from gradually converging to zero as the number of training steps increases, resulting in zero sensitivity, we set a sensitivity bound β. Then, we design an adaptive weighted fusion module, which is based on the variance of predicted probabilities to assign the weights of fusion. With this module, we further improve the classification ability of the model. Finally, we experimentally demonstrate that our proposed algorithm have better classification performance than previous state-of-the-art related algorithms mentioned in this paper. However, our algorithm has a much higher time cost due to the need to compute more training parameters, which is a shortcoming that needs to be addressed in our future work.

REFERENCES

[1] P. Monkam, S. Qi, H. Ma, W. Gao, Y. Yao, and W. Qian, "Detection and classification of pulmonary nodules using convolutional neural networks: A survey," *IEEE Access*, vol. 7, pp. 78075–78091, 2019.

[2] Y. Zhou and Z. Gao, "Intelligent recognition of medical motion image combining convolutional neural network with Internet of Things," *IEEE Access*, vol. 7, pp. 145462–145476, 2019.

[3] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48901–48911, 2019.

[4] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," 2020, *arXiv:2005.03915*.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[6] Z. Zhang, Q. Liu, Z. Huang, H. Wang, C. K. Lee, and E. Chen, "Model inversion attacks against graph neural networks," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 19, 2022, doi: 10.1109/TKDE.2022.3207915.

[7] C. Dwork and R. Aaron, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[8] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.

[9] Y. Wang, M. Gu, J. Ma, and Q. Jin, "DNN-DP: Differential privacy enabled deep neural network learning framework for sensitive crowdsourcing data," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 1, pp. 215–224, Feb. 2020.

[10] J. Yang, J. Wu, and X. Wang, "Retracted: Convolutional neural network based on differential privacy in exponential attenuation mode for image classification," *IET Image Process.*, vol. 14, no. 15, pp. 3676–3681, Dec. 2020.

[11] Y. Xie, P. Li, C. Wu, and Q. Wu, "Differential privacy stochastic gradient descent with adaptive privacy budget allocation," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 227–231.

[12] D. Li, J. Wang, Z. Tan, X. Li, and Y. Hu, "Differential privacy preservation in interpretable feedforward-designed convolutional neural networks," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 631–638.

[13] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive Laplace mechanism: Differential privacy preservation in deep learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 385–394.

[14] Z. Xu, S. Shi, A. X. Liu, J. Zhao, and L. Chen, "An adaptive and fast convergent approach to differentially private deep learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 1867–1876.

[15] Y. Hu, D. Li, Z. Tan, X. Li, and J. Wang, "Adaptive clipping bound of deep learning with differential privacy," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 428–435.

[16] D. Yu, H. Zhang, and W. Chen, "Improve the gradient perturbation approach for differentially private optimization," in *Proc. NIPS*. 2018, pp. 1–7.

[17] H. Liu, C. Li, B. Liu, P. Wang, S. Ge, and W. Wang, "Differentially private learning with grouped gradient clipping," in *Proc. ACM Multimedia Asia*, 2021, pp. 1–7.

[18] D. Xu, W. Du, and X. Wu, "Removing disparate impact on model accuracy in differentially private stochastic gradient descent," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1924–1932.

[19] J. Chen, W. H. Wang, and X. Shi, "Differential privacy protection against membership inference attack on machine learning for genomic data," in *Proc. Pacific Symp.*, Nov. 2020, pp. 26–37.

[20] Y. Shi and Y. Sagduyu, "Membership inference attack and defense for wireless signal classifiers with deep learning," *IEEE Trans. Mobile Comput.*, early access, Feb. 7, 2022, doi: 10.1109/TMC.2022.3148690.

[21] D. Usynin, D. Rueckert, and G. Kaissis, "Beyond gradients: Exploiting adversarial priors in model inversion attacks," 2022, *arXiv:2203.00481*.
[22] T. Titcombe, A. J. Hall, P. Papadopoulos, and D. Romanini, "Practical defences against model inversion attacks for split neural networks," 2021, *arXiv:2104.05743*.
[23] C. Dwork, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2006, pp. 265–284.
[24] Y. LeCun, C. Cortes, and C. Burges. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[25] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[26] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

CHUNMEI MA received the M.S. degree from the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2007. She is currently an Associate Professor with the School of Computer Science, Qufu Normal University, China. Her current research interests include big data, the Internet of Things, and machine learning.

XIANGSHAN KONG is currently pursuing the master's degree with the School of Computer Science, Qufu Normal University, China. His research interests include machine learning and differential privacy.

BAOGUI HUANG received the M.S. and Ph.D. degrees from Qufu Normal University, Rizhao, China, in 2008 and 2020, respectively. He is currently an Associate Professor with the School of Computer Science, Qufu Normal University. His current research interests include wireless networks, distributed computing, and machine learning.