mathematics

MDPI

Article

Adaptive Differential Privacy Mechanism Based on Entropy Theory for Preserving Deep Neural Networks

Xiangfei Zhang [1], Feng Yang [2], Yu Guo [2], Hang Yu [2], Zhengxia Wang [2] and Qingchen Zhang [2,*]

[1] School of Cyberspace Security, Hainan University, Haikou 570228, China
[2] School of Computer Science and Technology, Hainan University, Haikou 570228, China
* Correspondence: zhangqingchen@hainanu.edu.cn

Abstract: Recently, deep neural networks (DNNs) have achieved exciting things in many fields. However, the DNN models have been proven to divulge privacy, so it is imperative to protect the private information of the models. Differential privacy is a promising method to provide privacy protection for DNNs. However, existing DNN models based on differential privacy protection usually inject the same level of noise into parameters, which may lead to a balance between model performance and privacy protection. In this paper, we propose an adaptive differential privacy scheme based on entropy theory for training DNNs, with the aim of giving consideration to the model performance and protecting the private information in the training data. The proposed scheme perturbs the gradients according to the information gain of neurons during training, that is, in the process of back propagation, less noise is added to neurons with larger information gain, and vice-versa. Rigorous experiments conducted on two real datasets demonstrate that the proposed scheme is highly effective and outperforms existing solutions.

Keywords: deep neural networks; differential privacy; Laplace noise

MSC: 68P27; 68T01; 68T07

check for updates

Citation: Zhang, X.; Yang, F.; Guo, Y.; Yu, H.; Wang, Z.; Zhang, Q. Adaptive Differential Privacy Mechanism Based on Entropy Theory for Preserving Deep Neural Networks. *Mathematics* **2023**, *11*, 330. https://doi.org/10.3390/math11020330

Academic Editors: Liang Zou, Liang Zhao and Yonghui Xu

Received: 8 December 2022
Revised: 30 December 2022
Accepted: 6 January 2023
Published: 8 January 2023

1. Introduction

In recent years, deep learning has led to impressive successes in a wide range of applications, such as computer vision [1,2], natural language processing [3,4], Internet of Things [5,6], and healthcare [7–11]. However, since training deep neural networks usually requires the use of a large number of real data, despite the aforementioned progress, protecting model privacy has attracted public attention.

Deep neural networks have been proved to memorize training data [12,13] rather than learning the potential attributes of these data, so some attack methods against deep neural networks have been derived, such as membership-inference attack [14–17] and model-inversion attack [18–20]. For instance, Shokri et al. [14] proposed a membership-inference attack method against machine learning models. The authors used the adversarial learning strategy to train the inference model to identify the difference between the target model's prediction of its training input and that of its untrained input. Khosravy et al. [18] proposed a search strategy of the model-inversion attack, which uses the pre-trained depth-generation model to generate face images from random feature vectors and reduce the image search space to the feature vector space with low dimensions. In the semi-white box scenario, this method only obtains the model structure and parameters, without obtaining any user data, and verifies the severity of the model-inversion attack threat.

To defend deep learning-based systems against attacks, novel and existing privacy tools are constantly being explored by researchers to deal with these emerging issues. Differential privacy is a private information protection mechanism that prevents an adversary from inferencing any information about any particular record, even if the adversary holds all of the other records except the target one.

Since Dwork [21] proposed a mechanism to add noise to machine learning methods, differential privacy has gradually become a popular tool to protect machine learning models, including deep neural networks. Generally, noise can be added to the training neural network in four stages: input data, model parameters, loss function, and sample labels. For example, Wang et al. [22] proposed a differential-privacy-domain adaptation method, in which differential private noise is added to the gradient of the specific layer and training process to guarantee private training data. Their experimental results show that the proposed method achieves high accuracy with only a modest privacy loss in many adaptive tasks. Yu et al. [23] proposed an image privacy protection framework, which aims to identify the privacy sensitive content in an image using deep neural network technology and protect it with a generation with differential privacy against the synthetic content generated by the network. Their experimental results show that the suggested method can effectively protect user's privacy while maintaining the utility of IoMT images. In these methods, the magnitude of injected noise and the privacy budget are accumulated in proportion to the number of training epochs, so it may consume an unnecessarily large portion of the privacy budget.

Recently, scholars have focused on the research of adaptive differential privacy algorithm [24–27], whose main idea is to adaptively adjust the privacy budget during the training process according to the contribution of neurons or features to the model output results, adding less noise to more important neurons or features. For example, Phan et al. [25] proposed an adaptive Laplacian mechanism, which is completely independent of the training time consumed by the privacy budget, to preserve differential privacy for deep learning. Gong et al. [26] proposed a general differential privacy deep neural network learning framework that adaptively adds noise to the gradients according to the relevance between the neurons and the model output. These studies show that the adaptive differential privacy mechanism is superior to the traditional differential privacy mechanism.

In this paper, we propose an **En**tropy theory-based **A**daptive **D**ifferential **P**rivacy **P**reserving (EnADPP) framework for deep neural networks. The proposed EnADPP framework adds noise to the gradient adaptively according to the change of information gain of neurons during training. The main idea is to add less noise to neurons with larger information gain, and vice-versa. To achieve this, the training process of the neural network is regarded as a system, and the entropy theory is employed to calculate the information gain of neurons to estimate the amount of noise injected for the neuron gradients. Specifically, in each training iteration, a neural network interpretable algorithm (i.e., layer-wise relevance propagation, LRP [28]) is utilized to calculate the relevance between each neuron and the model outcome, and then the relevance value is used to calculate the information entropy of each layer and the information gain of each neuron. Noise is added to the gradient in light of the information gain. The experimental results show that, compared with the conventional differential privacy method, the proposed EnADPP framework provides strong privacy protection for the privacy information in the training data and improves the accuracy of the model.

2. Preliminaries

Representative data, which may contain sensitive information, are required for training deep neural networks. Therefore, the released model parameters may reveal private information on training data. As a promising privacy protection mechanism, differential privacy is widely used in deep neural networks to protect the privacy of trained models.

Definition 1 (ϵ-Differential Privacy [29])**.** *A random algorithm $\mathcal{M}$ takes $\mathcal{D}$ as the input set and $\mathcal{R}$ as the output set, formalized as $\mathcal{M} : \mathcal{D} \to \mathcal{R}$, which satisfies differential privacy if and only if the following inequality for all $\mathcal{O} \subseteq \mathcal{R}$ (i.e., $\mathcal{O}$ is the subset of $\mathcal{R}$) is established:*

$$\Pr[\mathcal{M}(D) = \mathcal{O}] \leqslant e^{\epsilon} \Pr\left[\mathcal{M}\left(D'\right) = \mathcal{O}\right], \tag{1}$$

where $D, D' \subseteq \mathcal{D}$ are two adjacent datasets that differ by at most one record, ϵ is the privacy budget that controls the strength of privacy protection, and the smaller privacy budget provides stronger privacy protection.

The Laplace mechanism is a general method for ϵ-differential privacy that employs global sensitivity (denoted as $GS_{\mathcal{F}}(D)$) of $\mathcal{F} : \mathcal{D} \to \mathcal{R}$ on two adjacent databases that differ at most by one tuple. The Laplace mechanism ensures that the algorithm $\mathcal{M} = \mathcal{F}(\mathcal{D}) + \eta$ satisfies ϵ-differential privacy by injecting noise η, where $\eta \sim Lap(GS_{\mathcal{F}}(D)/\epsilon)$.

3. Related Works

Deep neural networks are widely used in artificial intelligence fields, including Internet of Things, computer vision, and healthcare. As we all know, training neural networks requires representative data sets, which are usually collected from real samples. Therefore, the training data usually contain private information, such as passwords, clinical records, and face images. Moreover, a series of attack methods have been proposed against the neural network, such as gradient leak attack, model-inversion attack, and membership-inference attack. Therefore, it is imperative to protect the privacy of neural network models.

A promising approach to protect neural network patterns from attacks is to introduce differential privacy into neural network training. Ye et al. [30] proposed a differential privacy-defense method, which uses the differential privacy mechanism to modify and normalize the confidence score vector. This mechanism not only preserves privacy but also blurs membership and reconstructs data. By ensuring the order of fractions in the vector, the loss of classification accuracy is avoided. Xiao et al. [31] proposed a user-profile perturbation recommendation system scheme based on deep reinforcement learning. Deep reinforcement learning is employed in the scheme to select a differential privacy budget against inference attackers to protect users' privacy. Wei et al. [27] analyzed existing implementations of neural networks with differential privacy using fixed noise levels and proposed a deep neural network approach for differential privacy assurance by employing dynamic privacy levels. They suggest that the dynamic noise allocation strategy is superior to the traditional fixed noise variance strategy. Yu et al. [32] performed formal and refined privacy loss analysis for two different data-batch processing methods using centralized differential privacy and implemented a dynamic privacy budget allocator during training to improve the accuracy of the model. Xu et al. [33] reduced the privacy cost by using the adaptive learning rate to increase the convergence speed and mitigated the negative impact of differential privacy on model accuracy by introducing adaptive noise. In recent years, a growing number of works have proved that adaptive and dynamic differential privacy budget allocation strategies have obvious advantages [27,32–34].

4. Proposed Methods

In this section, we formally introduce the proposed EnADPP framework. We first introduce the calculation of system entropy and neuron information gain based on the LRP algorithm [28]. Then, the application of the proposed framework in SGD algorithm is introduced.

The LRP is a widely accepted algorithm, which is applied to compute the relevance of each input feature x_{ij} to the network outcome. For a neural network composed of l hidden layers, let $R^l_q(\mathbf{x}_i)$ denote the relevance of the neuron q at a certain layer l to the model outcome. For more details about LRP algorithm, please refer to [28].

4.1. Privacy Budget Ratio Based on Entropy Theory

Adding an equal amount of noise to the gradient may affect the utility of the model, resulting in an imbalance between the effectiveness of the model and the privacy protection of the training data. Therefore, we propose an adaptive differential privacy method to improve the model performance.

In the first place, we derive the information entropy of the certain layer, which can be calculated by

$$En(\mathbf{h}_l) = -\sum_{q\in\mathbf{h}_l} p\left(R_q^l(\mathbf{x}_i)\right)\log p\left(R_q^l(\mathbf{x}_i)\right), \tag{2}$$

where $\mathbf{h}_l$ represents the set of neurons in l-th layer and $p\left(R_q^l(\mathbf{x}_i)\right) = R_q^l(\mathbf{x}_i)/\sum_{q\in\mathbf{h}_l} R_q^l(\mathbf{x}_i)$. Then, we further estimate the information gain of each neuron in l-th layer, which is computed by

$$Ga(q) = \mathrm{En}(\mathbf{h}_l) - p\left(R_q^l(\mathbf{x}_i)\right)\log p\left(R_q^l(\mathbf{x}_i)\right). \tag{3}$$

To guarantee $Ga(q) \in [0,1]$, the $Ga(q)$ is normalized to

$$\frac{Ga(q) - \min_{q\in\mathbf{h}_l} Ga(q)}{\max_{q\in\mathbf{h}_l} Ga(q) - \min_{q\in\mathbf{h}_l} Ga(q)}. \tag{4}$$

Then, the privacy budget ratio α_q is introduced to adaptively add noise so that more noise is injected to gradients of neurons with smaller information gain value, while less noise is added to gradients of neurons, which is greater information gain at a certain layer. For the q-th neuron in a certain layer, the privacy budget ratio is

$$\alpha_q = Ga(q). \tag{5}$$

Furthermore, the adaptive privacy budget is expressed as $\epsilon_q = \alpha_q \times \epsilon$.

Finally, the Laplace noise with $Lap\left(\frac{\Delta\omega_t}{\epsilon_q}\right)$ is injected to gradients to preserve the differential privacy for neural networks. Algorithm 1 outlines the basic steps of the EnADPP framework.

Algorithm 1 the proposed EnADPP framework

```
Input: Training dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n\}$, the number of batches $T$, clipping bound $C$, batch size $|L|$, privacy budget $\epsilon$, and loss function $\mathcal{L}(\boldsymbol{\omega})$.
for $q \in \mathbf{h}_l$ do
    Calculate the relevance $R_q^l(\mathbf{x}_i)$ of the $q$-th neuron at $l$-th hidden layer.
    Calculate the entropy $En(\mathbf{h}_l)$ of the $l$-th hidden layer.
    Calculate the gain $Ga(q)$ of the $q$-th neuron at $l$-th hidden layer.
    The privacy budget ratio $\alpha_q = Ga(q)$.
    Compute the adaptive privacy budget $\epsilon_q = \alpha_q \times \epsilon$.
end for

for $t \in [T]$ do
    Take a random set of training samples $L$ with batch size $|L|$ on dataset $D$.
    Compute gradient: $g(\mathbf{x}_i) \leftarrow \bigtriangledown\mathcal{L}(\boldsymbol{\omega}_t, \mathbf{x}_i)$.
    if $\| g(\mathbf{x}_i) \|_1 > C$ then
        Gradient clipping: $\hat{g}(\mathbf{x}_i) \leftarrow \frac{C\cdot g(\mathbf{x}_i)}{\|g(\mathbf{x}_1)\|_1}$.
    end if
    Inject noise: $\tilde{g}(\mathbf{x}_i) \leftarrow \frac{1}{|L|}\left(\sum_{\mathbf{x}_i\in L_t} \hat{g}(\mathbf{x}_i) + Lap\left(\frac{\Delta\omega_t}{\epsilon_q}\right)\right)$.
end for
Output: The noisy gradient $\tilde{g}(\mathbf{x}_i)$.
```

4.2. Gradient Perturbation for SGD Optimization Algorithm

The proposed method is a general strategy, which can be applied to existing gradient descent methods. In this section, the proposed EnADPP framework is applied to the SGD optimization algorithm for minimizing loss functions.

Gradient perturbation for SGD. SGD is a useful optimization algorithm during the training of neural networks, which is used to optimize the loss function $\mathcal{L}(\boldsymbol{\omega})$ with respect to parameters $\boldsymbol{\omega}$ on T random training batches. At each training step, a random set of training samples L on dataset D is used. For mini-batch SGD, when we start at an initial point $\boldsymbol{\omega}_0$ and update the parameters $\boldsymbol{\omega}$ at t steps, we have

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma\left(\lambda\boldsymbol{\omega}_t + \frac{1}{|L|}\sum_{\mathbf{x}_i \in L_t} g(\mathbf{x}_i)\right), \tag{6}$$

where γ is the learning rate, λ is a regularization parameter, and $g(\mathbf{x}_i)$ is the gradient on the samples.

We apply the proposed framework to SGD optimization algorithm, as shown in Algorithm 2.

Algorithm 2 Adaptive differential privacy SGD optimization algorithm

```
Input: Training dataset $D = \{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\}$, learning rate $\gamma$, the number of batches $T$,
    clipping bound $C$, batch size $|L|$, privacy budget $\epsilon$, and loss function $\mathcal{L}(\boldsymbol{\omega})$.
Randomly initialize the model parameter $\boldsymbol{\omega}_0$.
Compute the noisy gradient: $\tilde{g}(x_i) = EnADPP(D, \epsilon, T, |L|, \mathcal{L}(\boldsymbol{\omega}), C)$.
for $t \in [T]$ do
    Gradient descent: $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma(\lambda\boldsymbol{\omega}_t + \tilde{g}(\mathbf{x}_i))$.
end for
Output: The noisy gradient $\tilde{g}(\mathbf{x}_i)$.
```

Proof. Suppose T training batches $L_1, \dots, L_t, \dots, L_T$ with same batch size $|L|$ are disjointed. Given two adjacent batches L_t and L'_t, let $\boldsymbol{\omega}_{t+1(L_t)}$ and $\boldsymbol{\omega}_{t+1(L'_t)}$ denote the parameters on batch L_t and L'_t, respectively. According to Equation (6), let $\Delta\boldsymbol{\omega}_t = \parallel \boldsymbol{\omega}_{t+1(L_t)} - \boldsymbol{\omega}_{t+1(L'_t)} \parallel_1$; we have the following inequality.

$$\begin{aligned}
&\Delta\boldsymbol{\omega}_t = \tfrac{\gamma_t}{|L|}\sum_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\left\|\sum_{\mathbf{x}_t\in L_t}\hat{g}(\mathbf{x}_i) - \sum_{\mathbf{x}'_i\in L'_t}\hat{g}(\mathbf{x}'_i)\right\|_1\\
&\leqslant \tfrac{\gamma_t}{|L|}\sum_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\left(\left\|\sum_{\mathbf{x}_t\in L_t}\hat{g}(\mathbf{x}_i)\right\|_1 + \left\|\sum_{\mathbf{x}'_i\in L'_t}\hat{g}(\mathbf{x}'_i)\right\|_1\right)\\
&\leqslant 2\tfrac{\gamma_t}{|L|}\max_{\mathbf{x}_i\in Lt}\sum_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\|\hat{g}(\mathbf{x}_i)\|_1\\
&\leqslant 2\tfrac{\gamma_t}{|L|}C.
\end{aligned} \tag{7}$$

where $\hat{g}(\mathbf{x}_i)$ is the gradient after gradient clipping and C is the pre-defined clipping bound in Algorithm 1 (lines 10–12).

The SGD algorithm with gradient perturbation can be expressed as

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma\left(\lambda\boldsymbol{\omega}_t + \frac{1}{|L|}\left(\sum_{\mathbf{x}_i\in L_t}\hat{g}(\mathbf{x}_i) + Lap\left(\frac{\Delta\boldsymbol{\omega}_t}{\epsilon_q}\right)\right)\right). \tag{8}$$

As in Algorithm 1 (line 13), the gradient perturbation is expressed as

$$\sum_{\mathbf{x}_i\in L_t}\hat{g}(\mathbf{x}_i) + Lap\left(\frac{\Delta\boldsymbol{\omega}_t}{\epsilon_q}\right). \tag{9}$$

Then, we have that

$$
\begin{aligned}
&\frac{\Pr\left\{\boldsymbol{\omega}_{t+1(L)}\right\}}{\Pr\left\{\boldsymbol{\omega}_{t+1\left(L'\right)}\right\}} \\
&= \frac{\prod\limits_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod\limits_{q\in\mathbf{h}_l}\exp\left(\frac{\epsilon q\,\frac{\gamma}{|L|}\left\|\sum\limits_{\mathbf{x}_i\in L_t}\hat{g}(\mathbf{x}_i)-\left(\sum\limits_{\mathbf{x}_i\in L_t}\hat{g}(\mathbf{x}_i)+Lap\left(\frac{\Delta\boldsymbol{\omega}_t}{\epsilon q}\right)\right)\right\|_1}{\Delta\boldsymbol{\omega}_t}\right)}{\prod\limits_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod\limits_{q\in\mathbf{h}_l}\exp\left(\frac{\epsilon q\,\frac{\gamma}{|L|}\left\|\sum\limits_{\mathbf{x}'_i\in L'_t}\hat{g}\left(\mathbf{x}'_i\right)-\left(\sum\limits_{\mathbf{x}'_i\in L'_t}\hat{g}\left(\mathbf{x}'_i\right)+Lap\left(\frac{\Delta\boldsymbol{\omega}_t}{\epsilon q}\right)\right)\right\|_1}{\Delta\boldsymbol{\omega}_t}\right)} \\
&\leqslant \prod_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod_{q\in\mathbf{h}_l}\exp\left(\frac{\epsilon_q\frac{\gamma}{|L|}}{\Delta\boldsymbol{\omega}_t}\left\|\sum_{\mathbf{x}_i\in L_t}\hat{g}(\mathbf{x}_i)-\sum_{\mathbf{x}'_i\in L'_t}\hat{g}\left(\mathbf{x}'_i\right)\right\|_1\right) \\
&\leqslant \prod_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod_{q\in\mathbf{h}_l}\exp\left(\frac{2\epsilon_q\frac{\gamma}{|L|}}{\Delta\boldsymbol{\omega}_t}\max_{\mathbf{x}_i\in L_t}\|\hat{g}(\mathbf{x}_i)\|_1\right) \\
&\leqslant \prod_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\prod_{q\in\mathbf{h}_l}\exp\left(\epsilon\frac{2\frac{\gamma}{|L|}}{\Delta\boldsymbol{\omega}_t}\max_{\mathbf{x}_i\in L_t}\|\hat{g}(\mathbf{x}_i)\|_1\right) \\
&\leqslant \exp\left(\epsilon\frac{2\frac{\gamma}{|L|}}{\Delta\boldsymbol{\omega}_t}\max_{\mathbf{x}_i\in L_t}\sum_{\boldsymbol{\omega}\in\boldsymbol{\omega}_t}\|\hat{g}(\mathbf{x}_i)\|_1\right) \\
&\leqslant \exp\left(\epsilon\frac{2\frac{\gamma}{|L|}C}{\Delta\boldsymbol{\omega}_t}\right) \\
&\leqslant \exp(\epsilon).
\end{aligned}
$$

□

5. Results

In this section, we carried out extensive experiments on the two well-known datasets (i.e., MNIST, ADNI) with several deep learning algorithms to verify the practicability of the model. First, the performances of the SGD optimization algorithm with the proposed framework on the two datasets are analyzed. Then, we investigate the impact of different privacy budgets (i.e., different noise levels) on model performances. In this work, we set the privacy budget $\epsilon \in [0.2, 0.4, 0.8, 1, 2, 4, 8]$. The accuracy is selected as the evaluation indicator, which is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

The proposed EnADPP framework is compared with three types of algorithms; both the proposed framework and the comparison methods are performed on the SGD optimization algorithm.

- The pSGD algorithm [35] is a classical differential privacy algorithm, which uses the same privacy level during the neural network training process.
- The APPDL algorithm [36] is an adaptive privacy preserving a deep learning algorithm, which injects noise with a specific decay rate based on the Gaussian mechanism into the gradient.
- The ADPPL framework [26] is an adaptive-noise-adding differential privacy algorithm, which dynamically adjusts the privacy budget according to the neuron's contribution to the model output during training.

5.1. The Accuracy on MNIST Dataset

The MNIST dataset [37] consists of 70,000 handwritten digits, including 60,000 training samples and 10,000 test samples, where each sample is a 28 × 28-size gray-level image. The architecture for MNIST uses two hidden layers with 512 and 256 neurons, respectively.

Table 1 shows the accuracy of the EnADPP framework and the comparison methods on the MNIST dataset. For the proposed framework, according to Table 1, we derive that the accuracy of MNIST dataset exceeds 93%. When $\epsilon = 0.2$, the accuracy is 93.03%, while when $\epsilon = 8$, the accuracy of the model reaches 95.89% and the difference is only 2.86%. When ϵ is between 0.2 and 8, the accuracy will be steadily improved. In addition, for comparison methods, when $\epsilon = 0.2$, the pSGD algorithm achieves 62.33% accuracy, which is the lowest accuracy among all methods. The APPDL algorithm achieves 91.52% accuracy, which is the highest accuracy among comparison methods. ADPPL achieves 84.10% accuracy. When $\epsilon = 8$, it is still the pSGD algorithm that obtains the lowest accuracy, and there is no significant difference between the accuracy obtained by APPDL and ADPPL.

It is easy to conclude from Table 1 that the accuracy of all methods increases with the increase in the privacy budget ϵ. This is because the smaller the privacy budget ϵ, the more noise is injected into the gradient, and the greater the protection of the model, which means the greater the impact on the model performance.

Table 1. The accuracy of EnADPP framework and comparison methods on MNIST dataset.

	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$
pSGD	62.33%	70.91%	73.51%	75.79%	83.68%	89.08%	91.66%
APPDL	91.52%	91.89%	92.06%	92.29%	92.77%	93.09%	93.12%
ADPPL	84.10%	86.14%	88.12%	90.11%	92.22%	93.06%	93.24%
EnADPP	93.03%	94.04%	94.21%	94.95%	95.16%	95.68%	95.89%

5.2. The Accuracy on ADNI Dataset

The ADNI (Alzheimer's Disease Neuroimaging Initiative) database collects data on Alzheimer's disease, including rs-fMRI and PET images, and genetics and cognitive tests. In this study, 319 rs-fMRI images (including 154 normal controls and 165 cases of early mild cognitive impairment) were selected for the experiment. The same data-processing flow as in [38] is adopted. The sliding window strategy is used to extract dynamic functional connectivity features from the data. The architecture for ADNI uses a Siamese network with two subnetworks sharing parameters, each of which is composed of an auto encoder.

Table 2 shows the classification results of 2way 1shot based on an EnADPP framework and comparison methods on the ANDI dataset. In Table 2, the accuracy of EnADPP framework exceeds 60%, when $\epsilon = 0.2$, the accuracy is 60.85%, and when $\epsilon = 8$, the accuracy is 66.50%. In the comparison methods, APPDL achieves the lowest accuracy when $\epsilon = 0.2$, and ADPPL achieves the highest accuracy when $\epsilon = 8$. From Table 2, we can drawn the same conclusion as with the experiments on the MNIST dataset: that the accuracy increases with the increase in the privacy budget. This is in line with the general expectations and similar conclusions that were reached in previous studies [26,27].

Table 2. The accuracy of EnADPP framework and comparison methods on ADNI dataset.

	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.8$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$
pSGD	58.11%	59.45%	59.91%	60.58%	61.51%	62.66%	63.86%
APPDL	58.10%	59.45%	60.81%	61.42%	62.16%	64.28%	65.15%
ADPPL	59.81%	60.79%	61.16%	62.50%	63.51%	64.86%	65.21%
EnADPP	60.85%	61.81%	62.16%	63.25%	64.86%	66.21%	66.50%

6. Discussion

From Tables 1 and 2, we can see that the adaptive-noise-adding methods achieve better accuracy than the fixed-amount noise-adding method. For the MNIST dataset, our approach has obvious advantages, especially when the privacy budget is small. When the privacy budget is 0.2, our method achieves an accuracy of about 7% higher than ADPPL and 30% higher than pSGD. With the increase in the privacy budget, the accuracy gap between the four methods becomes smaller. For the ADNI dataset, although the gap between accuracy has not changed much with the change of the privacy budget, our method is still more advanced.

The pSGD injects the same amount of Gaussian noise protection into the gradient during the training. As the number of training iterations increases, the noise superposition amount of each neuron gradient increases. In fact, for some neurons, it is unnecessary to add so much noise. Therefore, compared with the adaptive-noise-adding methods, the pSGD algorithm achieves lower accuracy. Although APPDL dynamically adjusts the noise added to the gradient during training through Gaussian noise with a specific decay rate, it adds the same amount of noise to all gradients in each training iteration. The variance of Gaussian noise decreases with the increase in training iterations, and the amount of noise added is not dynamically adjusted according to the relationship between the neurons and the model outcome. ADPPL uses the ratio of feature and outcome's relevance to the total relevance of the certain layer as the privacy budget ratio, while EnADPP uses the normalized value of the neurons' information gain as the privacy budget ratio. Figure 1 shows the privacy budget ratio of the two methods given the correlation between 10 features and the model output. It can be seen from Figure 1 that under the same relevance trend, compared with the ADPPL framework, the privacy budget ratio in the EnADPP method has a greater range of change. This means that when EnADPP adds noise adaptively, the amount of noise added by different neuron gradients varies greatly, which helps to fully consider the impact of different neurons on the model results when updating the model. Therefore, compared with the comparison methods, the proposed EnADPP method achieves better accuracy.

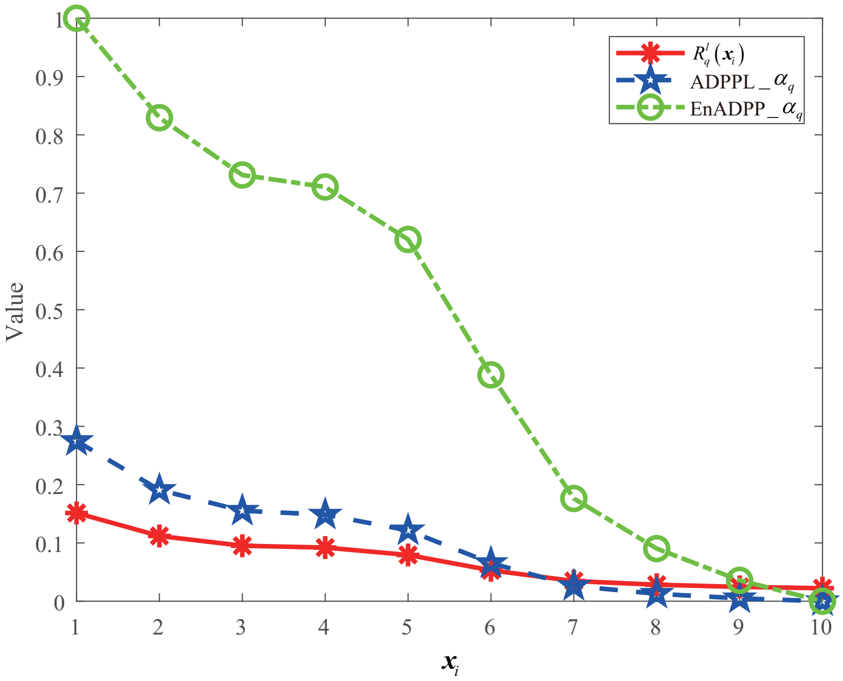

Figure 1. Comparison of the privacy budget ratio between ADPPL and EnADPP.

Moreover, Table 3 shows the accuracy without differential privacy on the two datasets, which is higher than that obtained by any differential privacy method. The accuracy of the proposed EnADPP algorithm is closer to this result.

Table 3. The accuracy on MNIST and ADNI dataset without differential privacy.

	MNIST	**ADNI**
Accuracy	97.60%	67.57%

7. Conclusions

A differential privacy mechanism with adaptive noise addition for deep neural networks is presented in this paper. This method adaptively adjusts the privacy budget according to the importance of the neural network to the training system in the process of training the neural network and allocates more of a privacy budget to the neurons that contribute more to the model output. Rigorous experiments on well-known MNIST and ADNI datasets using a fully connected neural network and Siamese network, respectively, verify the practicability of our method. The experimental results show that our method is superior to the comparison methods.In future work, we will explore more adaptive differential privacy algorithms to protect deep neural networks, such as an adaptive differential privacy policy based on dynamic gradient clipping.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z.; software, X.Z.; validation, H.Y.; formal analysis, X.Z.; investigation, X.Z.; resources, Q.Z.; data curation, Z.W.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z., F.Y. and Y.G.; visualization, H.Y.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, Z.W. and Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Major Science and Technology Plan of Hainan (ZDKJ2021016), the Key Research and Development Project of Hainan Province (Grants No. ZDYF2021GXJS017, No. ZDYF2021GXJS016), the National Natural Science Foundation of China (Grants No. 6216070295, No. 82160345), and the Key Science and Technology Plan Project of Haikou (Grants No. 2021-016).

Data Availability Statement: In this study, we used two public datasets, MNIST and ADNI. MNIST is available at http://yann.lecun.com/exdb/mnist/ accessed on 1 December 2022, ADNI is available at https://adni.loni.usc.edu/ accessed on 1 December 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jack, W.; Tatiana, K.; Chris, M. Vision Processing for Assistive Vision: A Deep Reinforcement Learning Approach. *IEEE Trans. Hum.-Mach. Syst.* **2022**, *52*, 123–133.
2. Ruotsalainen, L.; Morrison, A.; Mäkelä, M.; Rantanen, J.; Sokolova, N. Improving Computer Vision-Based Perception for Collaborative Indoor Navigation. *IEEE Sens. J.* **2022**, *22*, 4816–4826. [CrossRef]
3. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604–624. [CrossRef]
4. Yu, Y.; Chen, X.; Cao, S.; Zhang, X.; Chen, X. Exploration of Chinese Sign Language Recognition Using Wearable Sensors Based on Deep Belief Net. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 1310–1320. [CrossRef]
5. Yu, H.; Yang, L.T.; Zhang, Q.; Armstrong, D.; Deen, M.J. Convolutional Neural Networks for Medical Image Analysis: State-of-the-art, Comparisons, Improvement and Perspectives. *Neurocomputing* **2021**, *444*, 92–110. [CrossRef]
6. Zhou, X.; Liang, W.; Wang, K.I.K.; Wang, H.; Yang, L.T.; Jin, Q. Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things. *IEEE Internet Things J.* **2020**, *7*, 6429–6438. [CrossRef]
7. Yu, H.; Yang, L.T.; Fan, X.; Zhang, Q. A Deep Residual Computation Model for Heterogeneous Data Learning in Smart Internet of Things. *Appl. Soft Comput.* **2021**, *107*, 107361. [CrossRef]
8. Muhammad, K.; Khan, S.; Ser, J.D.; Albuquerque, V.H.C.d. Deep Learning for Multigrade Brain Tumor Classification in Smart Healthcare Systems: A Prospective Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 507–522. [CrossRef]
9. Hu, X.; Ding, X.; Bai, D.; Zhang, Q. A Compressed Model-Agnostic Meta-Learning Model Based on Pruning for Disease Diagnosis. *J. Circuits, Syst. Comput.* **2022**, *32*, 2350022. [CrossRef]
10. Zhang, X.; Shams, S.P.; Yu, H.; Wang, Z.; Zhang, Q. A pairwise functional connectivity similarity measure method based on few-shot learning for early MCI detection. *Front. Neurosci.* **2022**, *16*, 1081788. [CrossRef]
11. Wang, S.; Wang, S.; Liu, Z.; Zhang, Q. A role distinguishing Bert model for medical dialogue system in sustainable smart city. *Sustain. Energy Technol. Assess.* **2023**, *55*, 102896. [CrossRef]
12. Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. A Closer Look at Memorization in Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; Proceedings of Machine Learning Research; PMLR: Sydney, Australia, 2017; Volume 70, pp. 233–242.

13. Meehan, C.; Chaudhuri, K.; Dasgupta, S. A Non-parametric Test to Detect Data-copying in Generative models. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Palermo, Sicily, Italy, 26–28 August 2020.
14. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 3–18.
15. Shi, Y.; Sagduyu, Y. Membership Inference Attack and Defense for Wireless Signal Classifiers with Deep Learning. *IEEE Trans. Mob. Comput.* **2022**, 1. [CrossRef]
16. Salem, A.; Zhang, Y.; Humbert, M.; Fritz, M.; Backes, M. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In Proceedings of the Network and Distributed Systems Security Symposium 2019, Internet Society, San Diego, CA, USA, 24–27 February 2019.
17. Chen, H.; Li, H.; Dong, G.; Hao, M.; Xu, G.; Huang, X.; Liu, Z. Practical Membership Inference Attack Against Collaborative Inference in Industrial IoT. *IEEE Trans. Ind. Inform.* **2022**, *18*, 477–487. [CrossRef]
18. Khosravy, M.; Nakamura, K.; Hirose, Y.; Nitta, N.; Babaguchi, N. model-inversion attack by Integration of Deep Generative Models: Privacy-Sensitive Face Generation From a Face Recognition System. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 357–372. [CrossRef]
19. Alufaisan, Y.; Kantarcioglu, M.; Zhou, Y. Robust Transparency Against model-inversion attacks. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 2061–2073. [CrossRef]
20. Fredrikson, M.; Jha, S.; Ristenpart, T. model-inversion attacks That Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, Denver, CO, USA, 12–16 October 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1322–1333.
21. Dwork, C. Differential Privacy. In *Encyclopedia of Cryptography and Security*; van Tilborg, H.C.A., Jajodia, S., Eds.; Springer: Boston, MA, USA, 2011; pp. 338–340.
22. Wang, Q.; Li, Z.; Zou, Q.; Zhao, L.; Wang, S. Deep Domain Adaptation With Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3093–3106. [CrossRef]
23. Yu, J.; Xue, H.; Liu, B.; Wang, Y.; Zhu, S.; Ding, M. GAN-based Differential Private Image Privacy Protection Framework for the Internet of Multimedia Things. *Sensors* **2020**, *21*, 58. [CrossRef] [PubMed]
24. Phan, N.H.; Yue, W.; Wu, X.; Dou, D. Differential Privacy Preservation for Deep Auto-Encoders: An Application of Human Behavior Prediction (AAAI-16) [oral presentation]. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, AZ, USA, 12–17 February 2016.
25. Phan, N.; Wu, X.; Hu, H.; Dou, D. Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; pp. 385–394.
26. Gong, M.; Pan, K.; Xie, Y.; Qin, A.; Tang, Z. Preserving Differential Privacy in Deep Neural Networks with Relevance-based Adaptive Noise Imposition. *Neural Netw.* **2020**, *125*, 131–141. [CrossRef]
27. Wei, W.; Liu, L. Gradient Leakage Attack Resilient Deep Learning. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 303–316. [CrossRef]
28. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE* **2015**, *10*, e0130140. [CrossRef]
29. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Theory of Cryptography*; Halevi, S., Rabin, T., Eds.; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
30. Ye, D.; Shen, S.; Zhu, T.; Liu, B.; Zhou, W. One Parameter Defense—Defending Against Data Inference Attacks via Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1466–1480. [CrossRef]
31. Xiao, Y.; Xiao, L.; Lu, X.; Zhang, H.; Yu, S.; Poor, H.V. Deep-Reinforcement-Learning-Based User Profile Perturbation for Privacy-Aware Recommendation. *IEEE Internet Things J.* **2021**, *8*, 4560–4568. [CrossRef]
32. Yu, L.; Liu, L.; Pu, C.; Gursoy, M.E.; Truex, S. Differentially Private Model Publishing for Deep Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 332–349.
33. Xu, Z.; Shi, S.; Liu, A.X.; Zhao, J.; Chen, L. An Adaptive and Fast Convergent Approach to Differentially Private Deep Learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1867–1876.
34. Zhang, T.; Zhu, Q. Dynamic Differential Privacy for ADMM-Based Distributed Classification Learning. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 172–187. [CrossRef]
35. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
36. Zhang, X.; Ding, J.; Wu, M.; Wong, S.T.C.; Van Nguyen, H.; Pan, M. Adaptive Privacy Preserving Deep Learning Algorithms for Medical Data. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2021; pp. 1168–1177.

37. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
38. Kam, T.E.; Zhang, H.; Jiao, Z.; Shen, D. Deep Learning of Static and Dynamic Brain Functional Networks for Early MCI Detection. *IEEE Trans. Med. Imaging* **2020**, *39*, 478–487. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.