ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205

GPriS: A Differentially Private Deep Learning Method with Gradient Sparsification

Huiqi Zhang

Guangdong Provincial Key Laboratory of IRADS, Beijing Normal-Hong Kong Baptist University, Zhuhai 519087, China

t330201606@mail.uic.edu.cn

Abstract. Privacy protection has become a core issue in machine learning, with differential privacy widely regarded as an effective method for preserving privacy in stochastic gradient descent. We propose the GPriS (Gradient Privacy Sparse) method, a novel method that enhances differential privacy in deep learning by incorporating sparsity-based pruning. By leveraging the Lottery Hypothesis, GPriS identifies and preserves a lucky "sub-network" of critical parameters, reducing the number of parameters and improving both efficiency and privacy. Experiments on the MNIST dataset demonstrate that GPriS achieves 98.83% test accuracy with only 60% of gradient updates under a privacy budget of $\varepsilon = 4$, a performance level comparable to non-privacy settings. Compared to traditional DP-SGD, GPriS offers a better trade-off between privacy, model accuracy, and training efficiency. Our results show that GPriS effectively allocates privacy budgets, ensuring optimal performance even under tight privacy constraints.

Keywords: Differential Privacy; Deep Neural Networks; Gradient Sparsification; Importance-based Updates.

1. Introduction

In recent years, deep learning has made significant progress in tasks like image recognition [1], natural language processing [2], and recommendation systems [3]. In federated learning [4,5] for user data on mobile devices, model gradients are shared or accessible, making privacy protection crucial. The deep learning process typically consists of two stages (Fig. 1): training the model with data to achieve good generalization, followed by user access via APIs or local deployment. Model interaction can be classified as black-box or white-box access. In black-box access, attackers can only observe the input-output relationship, while in white-box access, attackers can also access the model's structure, parameters, and gradients, which may contain sensitive information. Existing studies have shown that attackers can reconstruct training samples from model gradients using techniques like gradient inversion [6] or gradient matching [7,8].

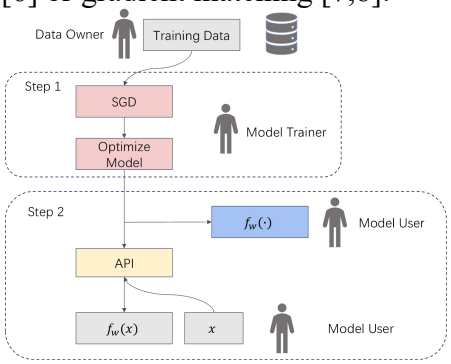

Fig. 1 Deep Learning Pipeline

To address privacy leakage, differentially private stochastic gradient descent (DP-SGD) [9] is widely used. It ensures privacy by clipping gradients and adding Gaussian noise, satisfying the (ε, δ) -differential privacy $\left((\varepsilon, \delta) - DP\right)$ constraint. However, DP-SGD struggles with high-dimensional models. For modern models with millions of parameters, it often results in significant accuracy loss at practical privacy levels [10]. Additionally, the accumulated privacy budget makes optimization more challenging. Current DP-SGD algorithms add noise uniformly

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205
across all parameters, causing unnecessary perturbations in irrelevant directions, which weakens the model's ability to capture important patterns.

To understand how to achieve these goals, we first analyze the gradient distribution in a typical training scenario using the MNIST dataset. We found that the gradient distribution is sparse (Fig. 2a), with over 99% of values in the [-0.01, 0.01] range, and only a few gradients showing significant updates. Since most gradients are near zero, adding uniform noise is inefficient. Focusing noise injection on the important gradients, and treating near-zero gradients differently, allows us to maintain privacy with less impact on accuracy. The gradient histogram also approximates a Gaussian distribution (Fig. 2b), supporting the use of Gaussian noise in DP mechanisms. Based on these findings, we propose the GPriS method, which reduces noise perturbation and improves model performance while maintaining differential privacy by dynamically selecting and protecting key gradient dimensions.

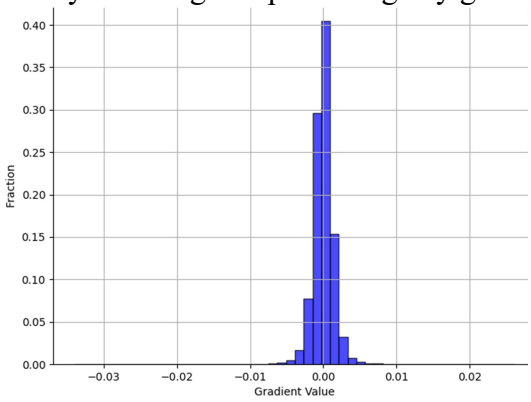

Fig. 2a Gradient sparsity and distribution characteristics under DP-SGD

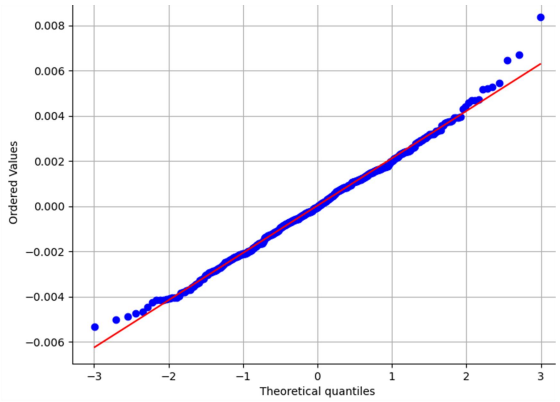

Fig. 2b Quantile-Quantile plot of preselected MNIST gradient vector

Fig. 2 Gradient sparsity and distribution characteristics under DP-SGD

Based on these findings, we propose GPriS, a method that reduces noise perturbation and improves model performance while ensuring differential privacy by dynamically selecting and protecting key gradient dimensions. In summary, our contributions are: (1) GPriS, a new improvement to DP-SGD that dynamically selects important gradient dimensions for more efficient noise injection; (2) empirical results demonstrating improved accuracy under strict privacy budgets.

2. Our Method

2.1 Differential Privacy in Deep Learning

The core idea of differential privacy is that even if an attacker can access the models output or gradients, they cannot determine whether a particular sample belongs to the training set. By injecting noise into the gradient updates, differential privacy ensures that the model's sensitivity to individual training samples is not leaked, thus protecting user data privacy:

Def 1: $(\varepsilon, \delta) - DP$[11] Let $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ be a randomized algorithm, with parameters $\varepsilon > 0$ and $\delta \in [0,1]$, If for any pair of neighboring datasets $D, D^{'} \in \mathcal{D}$ (which differ by at most one element), and for any measurable subset $\mathcal{S} \in \mathcal{R}$, the following holds:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr\left[M\left(D^{'}\right) \in S\right] + \delta$$

then $\mathcal{M}$ is said to satisfy $(\varepsilon, \delta) - DP$. In simpler terms, ε quantifies how much the model's output can change when a single training example is added or removed, and δ controls the probability that the privacy guarantee may not hold.

Lemma 1: Post-Processing Mechanism[12] Let $M: \mathcal{D} \to \mathcal{R}$ be an algorithm satisfying $(\varepsilon, \delta) - DP$, and let $f: \mathcal{R} \to \mathcal{R}^{'}$ be any function. Then, the composite function $f \circ M: \mathcal{D} \to \mathcal{R}'$ is also $(\varepsilon, \delta) - DP$.

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205

This lemma shows that post-processing operations do not break the differential privacy guarantee, allowing us to compose privacy budgets over multiple training cycles.

2.2 Lottery Hypothesis

The Lottery Hypothesis suggests that a sparse, trainable sub-network, known as the "lucky sub-network" [12], exists in a randomly initialized neural network. By preserving key connections early in training, convergence similar to the full network can be achieved without using all parameters. This implies that not all parameters are essential for learning. GPriS leverages this idea by pruning less important parameters and focusing on the "lucky sub-network" that contains the most relevant information for accurate learning under differential privacy.

2.3 GPriS Algorithm

We first present an overview of the GPriS algorithm. Trained under differential privacy constraints, the algorithm evaluates the importance of pretraining parameters by calculating their absolute mean values to select the "winning tickets". These selected tickets continue training to meet the differential privacy requirements. Compared to traditional DP-SGD, GPriS trains only a "sub-network", significantly reducing the number of parameters and enhancing training efficiency. The overall process is shown below.

Algorithm1: GPriS	
Input	Training data $D = \{x_1, ..., x_N\}$; loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i l(\theta, x_i)$; model parameters $\theta \in \mathbb{R}^d$; learning rate η; noise multiplier σ; batchsize B; clipping bound, pruning ratio r;
1	Randomly initialize model $\theta_0 \sim \mathcal{N}(0, I_d)$; $\lvert\theta_0\rvert = d$
2	Initialize selection mask $m \leftarrow 1^d$; initial mask all-ones
3	*Pretraining - Importance Parameter Selection (Lottery-Based)*
4	Run standard DP-SGD for T_{pre} steps
5	Compute importance score for each parameter: $s_j = \frac{1}{T_{pre}}\sum_{t=1}^{T_{pre}} \lvert\theta_{t,j}\rvert, \forall j \in \{1, ..., d\}$
6	Sort s_j in descending order, get indices $I_{as} = [j_1, j_2, ..., j_d]$
7	Define the set of randomly selected unimportant parameters: $\lvert I_{ns}\rvert = \lfloor(1-r)\cdot d\rfloor$, $\lvert I_{ns}\rvert \leftarrow \{j_{d-\lvert I_{ns}\rvert 1}, ..., j_d\}$
8	Final selected mask index set: $I_s \leftarrow I_{as} \setminus I_{ns}$
9	Define binary mask: $m[j] = \begin{cases} 1, & if\ j \in I_{ns} \\ 0 & otherwise \end{cases}$
10	Parameter Reset (Weight Rewinding): $\theta_p \leftarrow \theta_0 \odot m (\odot denotes\ element - wise\ multiplication)$
11	*Main Training Loop*
12	For $t \in \{1,2, ..., T\}$ do
13	Sample minibatch $B_t \subseteq D$ with probability $q = \frac{B}{N}$
14	Compute per-sample gradients (only on selected parameters): $g_t \leftarrow \nabla_{\theta_t}\mathcal{L}(m \odot \theta_t, x_i)$
15	Clip gradients for each $i \in L_t$, $\overline{g}_t \leftarrow \frac{g_t}{\max\left(1, \frac{\lVert g_t(I_s, x_i)\rVert_2}{C}\right)}$
16	Add Gaussian noise and aggregate: $\widetilde{g}_t \leftarrow \frac{1}{N}(\sum_i \overline{g}_t + \mathcal{N}(0, \sigma^2 C^2 \mathrm{I}))$
17	Gradient descent update: $\theta_{t+1} \leftarrow \theta_t - \eta\widetilde{g}_t$
Output	θ_T, use a privacy accountant to compute cumulative (ε, δ)

In the differential privacy pretraining stage, the model initializes its parameters through the differential privacy mechanism and learns the distribution of high-weight connections. This stage

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205

uses $(\varepsilon_{pre}, \delta)$-DP to ensure privacy, preventing gradient updates from leaking sensitive data. The selection mask m is initially set to 1 for all parameters, indicating that all parameters are involved in training. DP-SGD is used for pretraining for T_{pre} iterations, during which the importance score of each parameter is computed. For each parameter $\theta_{t,j}$, the importance score s_j is calculated as the absolute mean of its updates over all training iterations.

Based on the scores s_j, the parameters are sorted by importance, and the least important parameters are pruned. A binary mask $m[j]$ is then set so that only the most important parameters are retained, and others are excluded from further updates in the subsequent training.

During each training iteration, the algorithm follows these steps to update the model parameters: *Gradient Computation:* A minibatch B_t is sampled from the dataset, and gradients are computed only for the selected parameters. *Gradient Clipping:* To protect privacy, gradients are clipped to prevent large values from causing privacy leakage. *Noise Addition:* Gaussian noise is added to meet differential privacy requirements. The noise standard deviation is determined by the privacy budget ε and other training parameters. *Gradient Update:* The model parameters are updated using gradient descent with the noise-adjusted gradients, and the cumulative privacy budget (ε, δ) is tracked using a privacy accounting method (e.g., Moment Accountant). The clipping bound C limits the impact of outlier gradients, and the noise multiplier σ controls the amount of noise added to ensure privacy. Both are typically set based on the desired privacy budget ε and the sensitivity of the model.

GPriS draws on the Lottery Hypothesis by identifying key sub-networks of parameters that are critical for achieving high performance. This allows us to prune less important parameters, making the model more efficient while still adhering to privacy constraints.

3. Experiments

3.1 Experimental Setup

We conducted experiments on the MNIST dataset to validate the significant advantages of the proposed GPriS method over DP-SGD. The MNIST dataset is widely used in deep learning research, consisting of 60,000 training images and 10,000 test images. All experiments were performed on an NVIDIA GeForce RTX 4080 GPU, using the PyTorch framework.

3.2 Impact of Pruning Rate on Accuracy and Efficiency

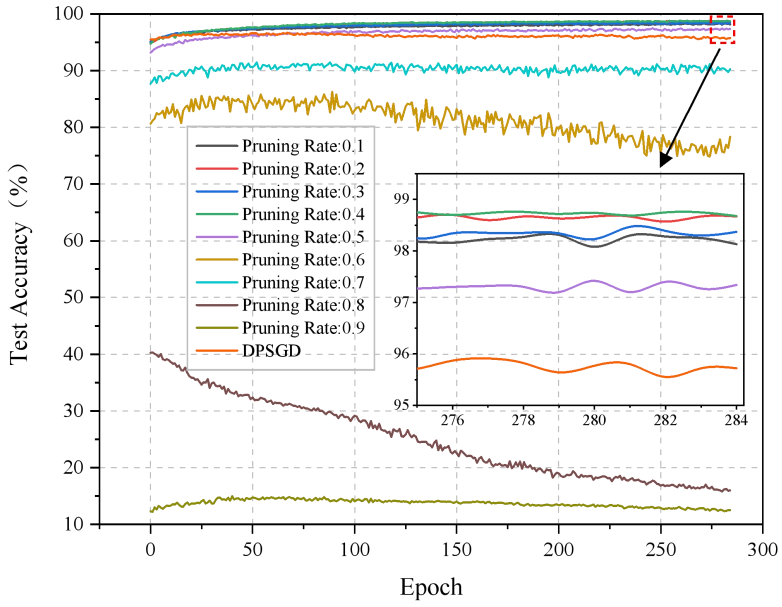

Fig. 3 Test Accuracy vs Epochs for Different Pruning Rates

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205

Fig. 3 illustrates the trend of model accuracy on the test set as the number of training epochs varies under different pruning rates. Overall, pruning rates in the range of 0.1–0.4 had minimal impact on model performance and led to significant improvements in test accuracy compared to standard DP-SGD. Notably, a pruning rate of 0.4 (retaining 60% of the parameters) consistently outperformed other rates across multiple experiments, yielding a stable performance ranking: 0.4 > 0.2 > 0.3 > 0.1. These results suggest that moderate pruning not only helps mitigate overfitting and improves the model's generalization ability, but also demonstrates that our method effectively preserves the learning capacity of critical parameters while maintaining a sparse network structure. Therefore, we recommend a pruning rate of 40% for optimal performance under differential privacy constraints.

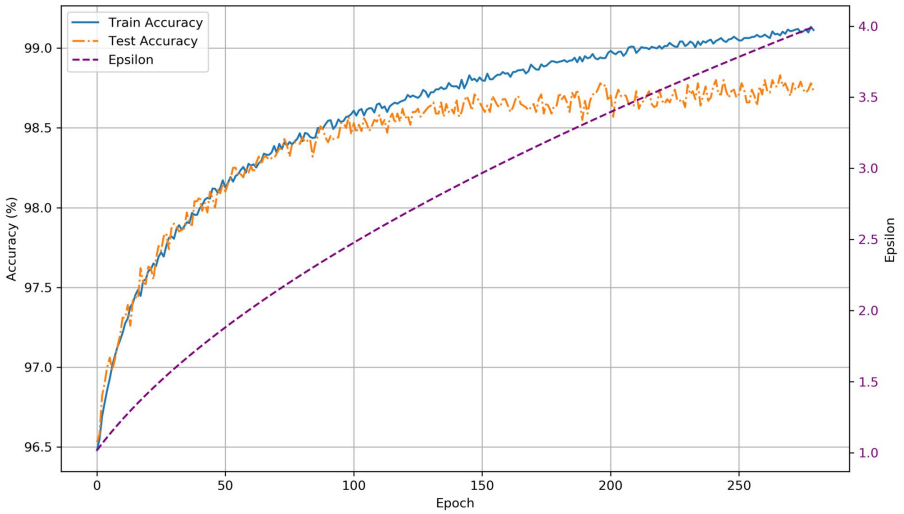

Fig. 4 Test Accuracy Evolution for Different Pruning Rates

Fig. 4 illustrates the evolution of training accuracy, test accuracy, and privacy budget (ε) during training. As training progresses, both accuracies rise, and the privacy budget accumulates, relaxing privacy protection constraints. With $\varepsilon = 4$, the model achieved 98.83% test accuracy on the MNIST test set, demonstrating effectiveness under a reasonable privacy budget.

In non-private settings, traditional models with handcrafted features reach 99.11% accuracy [13]. Our method achieves 98.83% accuracy on MNIST using only 60% of gradient updates, closely matching the performance of non-private models.

To improve privacy budget efficiency, we allocate a larger portion to training the "winning ticket" (the critical sub-network) and a smaller budget for its identification. Our results show that this mechanism effectively selects high-performance sub-networks even with low privacy budgets, while ensuring no sensitive information is exposed during the selection phase.

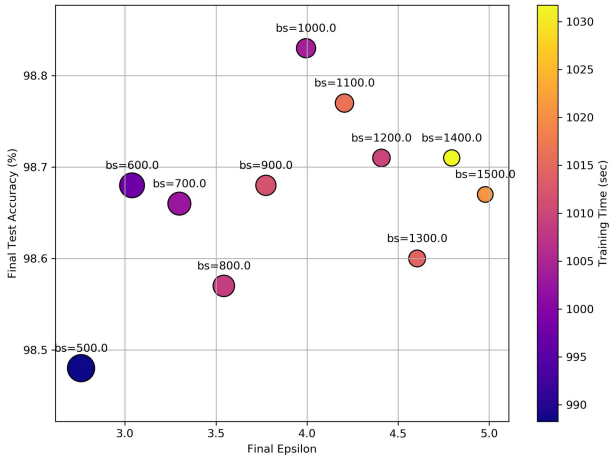

Fig. 5 Visualization of the privacy–utility–cost trade-off under DP-SGD

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205

We tested batch sizes ranging from 500 to 1500 and learning rates from 0.01 to 0.2, tuning the configurations for optimal performance on the MNIST dataset. Fig. 5 shows the relationship between final privacy budget ε, test accuracy, and training cost under different batch size and learning rate configurations. Each point represents a specific hyperparameter set, with color indicating training time (darker colors for longer times) and point size reflecting the number of training steps (larger points correspond to smaller batch sizes).

The results show that smaller batch sizes improve privacy protection by reducing privacy budget consumption, but they also increase training time and cost. In contrast, medium batch sizes (around 900–1200) strike a good balance between training efficiency, privacy budget consumption, and model accuracy, with the final privacy budget ranging from $\varepsilon \approx 3.7\ to\ 4.4$, achieving high accuracy while maintaining reasonable training time and computational cost.

These results indicate that adjusting batch size and learning rate configurations can effectively improve performance, balancing privacy protection with training resource consumption.

4. Summary

This paper proposes GPriS, a method that combines differential privacy and sparsity-based pruning to improve deep learning performance while preserving privacy. Leveraging the Lottery Hypothesis, GPriS trains a sparse “lucky sub-network”, reducing the number of parameters and enhancing efficiency and privacy. Experiments show that GPriS achieves similar accuracy to non-private models while meeting strict privacy constraints, outperforming DP-SGD in both accuracy and efficiency.

Acknowledgements

H.Q. Zhang was supported in part by the Guangdong Basic and Applied Basic Research Foundation (grant number 2023A1515110469), in part by the Guangdong Provincial Key Laboratory IRADS (grant number 2022B1212010006), and in part by the grant of Higher Education Enhancement Plan (2021-2025) of “Rushing to the Top, Making Up Shortcomings and Strengthening Special Features” (grant number 2022KQNCX100).

References

[1] Lundervold A S, Lundervold A. An overview of deep learning in medical imaging focusing on MRI[J]. Zeitschrift fuer medizinische Physik, 2019, 29(2): 102-127.

[2] Chen M X, Lee B N, Bansal G, et al. Gmail smart compose: Real-time assisted writing[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2287-2295.

[3] Bhowmick A, Duchi J, Freudiger J, et al. Protection against reconstruction and its applications in private federated learning[J]. arXiv preprint arXiv:1812.00984, 2018.

[4] Geiping J, Bauermeister H, Dröge H, et al. Inverting gradients-how easy is it to break privacy in federated learning?[J]. Advances in neural information processing systems, 2020, 33: 16937-16947.

[5] Lü L, Medo M, Yeung C H, et al. Recommender systems[J]. Physics reports, 2012, 519(1): 1-49.

[6] Zhu L, Liu Z, Han S. Deep leakage from gradients[J]. Advances in neural information processing systems, 2019, 32.

[7] Geiping J, Bauermeister H, Dröge H, et al. Inverting gradients-how easy is it to break privacy in federated learning?[J]. Advances in neural information processing systems, 2020, 33: 16937-16947.

[8] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures[C]//Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015: 1322-1333.

ISSN:3007-9640 DOI: https://doi.org/10.70114/acmsr.2025.3.1.P205
[9] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016: 308-318.

[10] Zhou Y, Wu Z S, Banerjee A. Bypassing the ambient dimension: Private sgd with gradient subspace identification[J]. arXiv preprint arXiv:2007.03813, 2020.

[11] Dwork C. Differential privacy[C]//International colloquium on automata, languages, and programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 1-12.

[12] Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4): 211-407.

[13] Frankle J, Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks[J]. arXiv preprint arXiv:1803.03635, 2018.

[14] Tramer F, Boneh D. Differentially private learning needs better features (or much more data)[J]. arXiv preprint arXiv:2011.11660, 2020.