LETTER

Differentially Private Neural Networks with Bounded Activation Function*

Kijung JUNG†, **Hyukki LEE**†a), ***Nonmembers***, ***and*** **Yon Dohn CHUNG**†, ***Member***

SUMMARY Deep learning has shown outstanding performance in various fields, and it is increasingly deployed in privacy-critical domains. If sensitive data in the deep learning model are exposed, it can cause serious privacy threats. To protect individual privacy, we propose a novel activation function and stochastic gradient descent for applying differential privacy to deep learning. Through experiments, we show that the proposed method can effectively protect the privacy and the performance of proposed method is better than the previous approaches.

key words: *deep learning, activation function, differential privacy*

1. Introduction

Over the last decades, deep learning has gained enormous attention in various fields, such as voice recognition, image classification, and medical diagnosis [1]. Deep learning applications show considerable performance, however if an adversary can recover the original data used to train the deep learning model, a critical privacy issue may arise. For example, training data of a face recognition model could be reconstructed by the model inversion attack [2], and it causes a privacy breach. Moreover, the attacker could abuse the face images to break into other face recognition system. Therefore, protection of personal privacy should be considered during the deep learning process.

Differential privacy (DP) is the strongest privacy protection model for data processing [3], which provides a mathematically provable guarantee of protecting the privacy of individuals. The goal of differential privacy is that the output should not be considerably influenced irrespective of whether a single data point is added or removed. Noise addition is a typical method for satisfying the differential privacy.

Deep learning models can also preserve privacy by satisfying DP. Adding noise to output layer of the model could be an option to satisfy DP. Note that the intensity of noise is decided by the maximum influence of a single data point on result (i.e., sensitivity). Noisy outputs give more significant effects to all layer by back propagation than the forward process. This means that adding noise to the outputs seriously hinder the learning process.

There is another direction of research of adding noise to gradients in gradient descent step. Abadi [4] proposed a gradient clipping method that restricts the maximum gradient size by clipping the larger gradients than the given maximum threshold. However, it has a problem that the vector of a batch could change after gradient clipping. To solve this problem, the size of a gradient should be restricted in other way. Recently, there is research to bound sensitivity with bounded activation function such as sigmoid and tanh [5]. Our work proposes a novel bounded activation function that has the advantage of ReLU. In this paper, we present a novel method for differentially private neural networks with restricted gradient. A gradient is derived from (1) an input, (2) weight of a layer, (3) an activation function, and (4) the derivative of the activation function. To determine the intensity of the noise, four components of a gradient must be bounded. However, ReLU, the most popular activation function, is not a bounded function, and thus the maximum difference is theoretically infinite. We propose a new bounded activation function, called Bounded Exponential Linear Unit (BELU). By utilizing BELU as the activation function, the maximum size of gradients can be effectively controlled.

2. Preliminaries

2.1 Rényi Differential Privacy (RDP)

There are limitations to utilize the original DP because it does not allow any exceptions. Hence, relaxed DP is proposed to handle worst-case scenario in a statistical manner [6]. RDP defines a neighboring database via the notion of Rényi divergence. RDP is known to preserve the same level of DP as (ϵ, δ)-privacy [7]. It is known that RDP is a proper model for multiple compositions of DP mechanisms such as applying DP in deep learning in every gradient descent step [7].

Definition 1 (Rényi Divergence): For two probability distributions P and Q over R, the rényi divergence of order $\alpha > 1$ is

Manuscript received January 25, 2021.
Manuscript revised March 9, 2021.
Manuscript publicized March 18, 2021.

†The authors are with the Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea.

*This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00269, A research on safe and convenient big data processing methods). This research was also supported by National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (No. NRF-2019H1D8A2105513).

a) E-mail: hyukki@korea.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2021EDL8007

$$D_\alpha(P\|Q) = \frac{1}{1-\alpha}\log_{E_{x\sim Q}}\left(\frac{P(x)}{Q(x)}\right)^\alpha \tag{1}$$

Definition 2 (Rényi Differential Privacy): A randomized mechanism $f : D \longrightarrow R$ satisfies $(\alpha, \epsilon)-$ Rényi differential privacy, if for any adjacent $D, D' \in D$, it holds that

$$D_\alpha(f(D)\|f(D')) \leq \epsilon \tag{2}$$

3. The Proposed Method

In this chapter, we present the proposed activation function and differentially private stochastic gradient descent method with the proposed activation function.

3.1 Bounded Activation Function (BELU)

We propose a novel bounded activation function BELU. The key idea is to set the thresholds for ReLU, and restrict outputs if the inputs are out of thresholds. The lower threshold is 0, and the upper threshold is determined by a hyperparameter β. BELU returns an identical result as ReLU if a value is within the threshold. If an input value is out of threshold, BELU returns the result of an exponential function instead of a constant value. An exponential function has several benefits. First, the derivative of an exponential function is not zero. This makes training possible in out of the threshold area. Second, it is easy to calculate the derivative of exponential function. Hence, exponential boundaries give better performance. The definition of BELU is shown in Definition 2.

Definition 3 (Bounded Exponential Linear Unit (BELU)):

$$BELU(x) = \begin{cases} \alpha\left(e^x - 1\right), x < 0 \\ x, 0 \leq x \leq \beta \\ \alpha\left(-e^{-x+\beta} + 1\right) + \beta, x > \beta \end{cases} \tag{3}$$

The maximum bound of BELU is $\alpha + \beta$ and the minimum bound of BELU is $-\alpha$. The derivation of BELU is:

$$BELU'(x) = \begin{cases} \alpha e^x, x < 0 \\ 1, 0 \leq x \leq \beta \\ \alpha e^{-x+\beta}, x > \beta \end{cases} \tag{4}$$

Figure 1 is a concrete example of a BELU with $\alpha = 1$ and $\beta = 2$. The dotted lines represent the thresholds, 0 and 2, and the dashed lines describe the maximum bound and the minimum bound, -1 and 3, that are asymptotic line of the exponential functions.

We can bound the gradient size of all layers of a network by using BELU.

Lemma 1: The gradient size of the $i-th$ layer is bounded if the weight, loss function is bounded and the activation function is BELU.

Proof 1: The gradient of the $i-th$ layer is as follows:

$$\frac{\partial L}{\partial w_i} = y_i \cdot \gamma_{i+1} \tag{5}$$

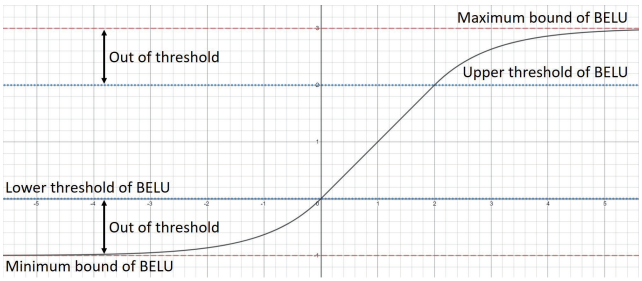

Fig. 1 An example of BELU ($\alpha = 1, \beta = 2$)

Algorithm 1: Differentially private SGD with BELU

Input : Training dataset $x_1, \ldots, x_N$, Loss function L, Learning rate η_t, Noise scale σ, Batch size b, Parameter of BELU α, β

1 Initialize θ_0 with He uniform initialization
2 B_i =*CalculateBound*$(L, \theta, \alpha, \beta)$
3 **for** $t = 0$ *to* $T - 1$ **do**
4 Take a random sample S_b
5 **foreach** $i \in S_b$ **do**
6 compute $g_t(x_I) \leftarrow \nabla_{(}\theta_t\ )\ L(\theta_t,\ x_i\ )$
7 $\widetilde{g}_t\left(x_i\right) \leftarrow g_t\left(x_i\right) + \mathcal{G}\left(0, \quad \sigma^2 B_i^2\right)$
8 **end**
9 $\theta_{t+1} \leftarrow \theta_t - \eta_t \widetilde{g}_t$
10 **end**

Output: θ_T and privacy cost (ϵ, δ) of the SGD Step

$$\gamma_i = \begin{cases} (y_i - \text{ Label }) \cdot BELU'\left(x_i\right), \text{ if i is the last} \\ \left(\sum \gamma_{i+1} \cdot w_i\right) \cdot BELU'\left(x_i\right), \text{ else} \end{cases} \tag{6}$$

w_i, x_i and y_i refers weights, an input and output of $i-th$ layer. In the equation (5), $(y_i - Label)$ is bounded, $BELU'(x) \leq 1$, and w can be bounded in the algorithm. Therefore, all the variables in the gradient are bounded. And thus, gradient size of $i-th$ layer is bounded. □

3.2 Differentially Private Stochastic Gradient Descent with BELU

Gradient clipping restricts the size of gradients by clipping gradients if the norm of a gradient is larger than the clipping size c. This method does not consider the ratio among gradients in the same batch. This means that the vector of the gradient can be changed after clipping. This could decrease the learning performance.

For the better performance, the proposed method restricts four components in gradients instead of gradient clipping. Differentially private SGD with BELU proceeds in the following order. It starts with the initialization of weights (line 1). He uniform initialization is used [1]. Bound is calculated with loss function, weights, and parameters of BELU to determine the intensity of noise (line 2). A batch is composed by sampling from the data set (line 4). After that, gradients are computed (line 6) and add Gaussian noise for each data from the batch (line 7). Subsequently, the descent step is proceeded with the noisy gradient of a batch (line 9). The entire algorithm is shown in Algorithm 1. Note that

BELU does not appear explicitly in the algorithm, nevertheless it is applied to compute bound and gradients.

4. Experiments

We experimentally evaluate the performance of the proposed method on classification using MNIST [9] that has 60,000 training data and 10,000 test data. The network has 32x32 size input, 32 size linear hidden layer, and 10 size output layer. We conduct experiments on Intel i9-9900x, 128GB RAM, GeForce RTX 2080 8GB. We use ReLU in the non-private version. Moreover, we evaluate two the previous method. First is an experiment using Tanh for activation function to bound the size of gradients. The second is gradient clipping, with ReLU varying clipping size. We measure the accuracy by averaging over 10 epochs. The parameters for the experiments are as follows. The noise multiplier σ is 1, δ is 10^{-5},and the batch size is 64. Calculating privacy losses by described in [8], the privacy loss of all private models (ϵ, δ) is $(1.10, 10^{-5})$.

As shown in Fig. 2, the result demonstrates that the proposed method gets higher accuracy than Tanh and gradient clipping. Figure 3 shows the effect of the threshold β of BELU. As the threshold of BELU increases, the gradient before noise addition is closer to the optimal. Therefore, the model converges faster. At the same time, the noise gets stronger as β increases. Consequently, the average accuracy continuously increases to a certain point (β=5). After the point, the accuracy decreases. The gap between the accuracy in the last epoch and average accuracy is closer as the threshold increases. This is because the model converges faster. At the same time, the accuracy decreases because of noise addition. Figure 4 represents the performance comparison of the proposed method and gradient clipping with ReLU. For each group, the clipping size is identical to the maximum bound of BELU. The proposed method outper-

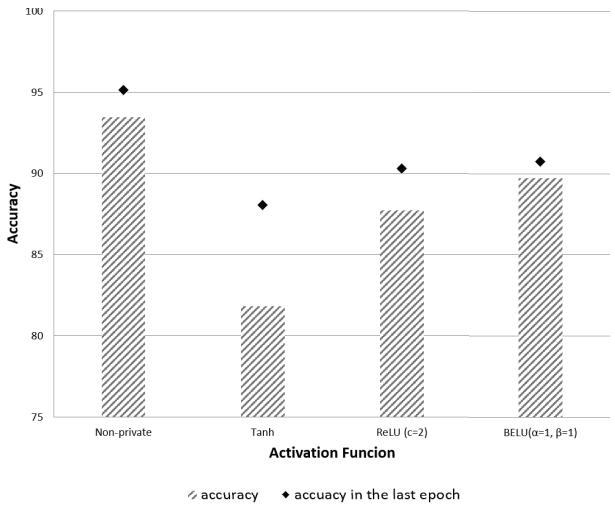

Fig. 2 Average accuracy of 10 epochs and accuracy in the last epoch varying activation function

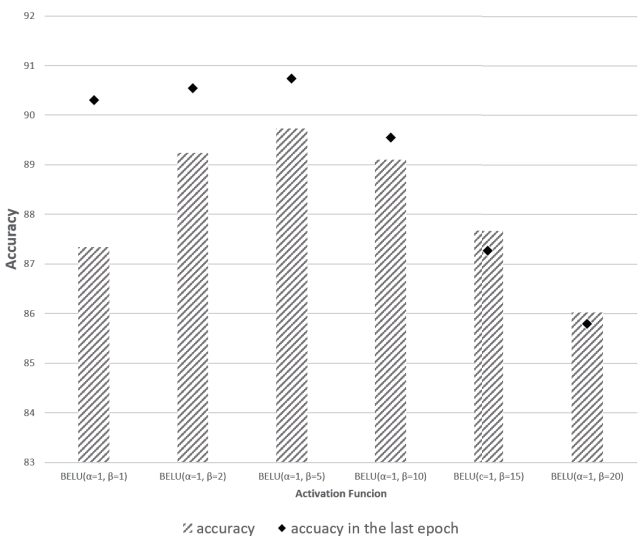

Fig. 3 Average accuracy of 10 epochs and accuracy in the last epoch varying β

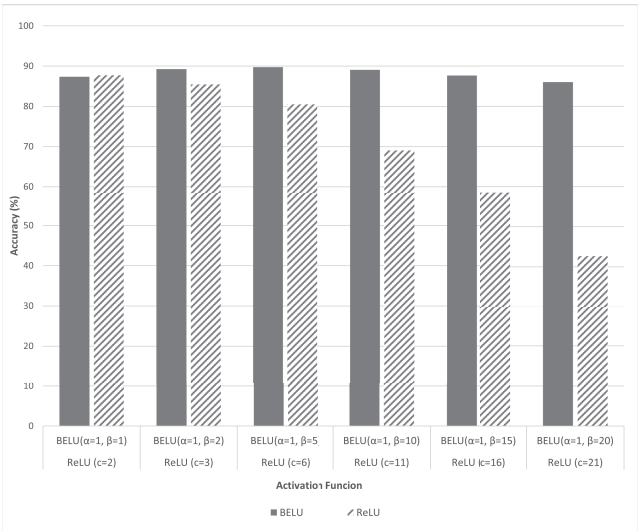

Fig. 4 Average accuracy of 10 epochs varying β and clipping size

Table 1 Average accuracy of 10 epochs and accuracy in the last epoch

Activation function	Average accuracy(%)	Accuracy in the last epoch(%)
Non-private	93.45	95.12
Tanh	81.81	88.07
ReLU(c = 2)	87.73	90.31
ReLU(c = 3)	85.46	85.68
ReLU(c = 6)	80.47	81.04
ReLU(c = 11)	69.05	66.23
ReLU(c = 16)	58.47	56.15
ReLU(c = 21)	42.67	33.89
BELU ($\alpha = 1, \beta = 1$)	87.34	90.3
BELU ($\alpha = 1, \beta = 2$)	89.24	90.54
BELU ($\alpha = 1, \beta = 5$)	89.73	90.74
BELU ($\alpha = 1, \beta = 10$)	89.1	89.55
BELU ($\alpha = 1, \beta = 15$)	87.67	87.27
BELU ($\alpha = 1, \beta = 20$)	86.03	85.8

forms gradient clipping with ReLU as the clipping size increases. This is because the noise gives significant effects to ReLU than BELU. To summarize, BELU converges faster with better accuracy than Tanh. Additionally, BELU shows better performance than ReLU if β is the appropriate value.

5. Conclusion

In this paper, we proposed a differentially private neural networks with gradient restriction. To restrict the size of gradients in a differentially private manner, we proposed a new bounded activation function BELU. Through experiments, we demonstrated that BELU effectively restricts the gradients and ensures high utility.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," Proceedings of the IEEE international conference on computer vision, pp.1026–1034, 2015.

[2] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.1322–1333, 2015.

[3] C. Dwork, "Differential privacy: A survey of results," International conference on theory and applications of models of computation. Springer, Berlin, Heidelberg, 2008.

[4] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp.308–318, 2016.

[5] N. Papernot, et al., "Tempered sigmoid activations for deep learning with differential privacy," arXiv preprint arXiv:2007.14191, 2020.

[6] Q. Geng and P. Viswanath, "The optimal mechanism in differential privacy," 2014 IEEE international symposium on information theory, IEEE, pp.2371–2375, 2014.

[7] I. Mironov, "Rényi differential privacy," 2017 IEEE 30th Computer Security Foundations Symposium, IEEE, pp.263–275, 2017.

[8] I. Mironov, et al., "Rényi differential privacy of the sampled Gaussian mechanism," arXiv preprint arXiv:1908.10530, 2019.

[9] http://yann.lecun.com/exdb/mnist/, accessed Jan. 2021.