How can differential privacy helps deep learning

SeungJae Lee
North London Collegiate School

Abstract

In today's scenario, deep learning has much application in daily living, such as health care, chatbots, entertainment, product recommendation, virtual sessions, etc. In the training phase, deep learning models train the datasets in which information privacy is stored locally through model parameters. However, some privacy concern issues still exist, so applying Differential privacy to the deep learning model is widely recognized for its traditional scenario in rigorous mathematical solutions. This paper revisits the Differential privacy stochastic gradient descent (SGD) method used to achieve good privacy protection. Then deploy the mechanism in the input, hidden, and output layer through pros and cons. Also, provide a broader outlook to this practice.

Keywords: Differential privacy, Deep learning, Risk, Model, Data privacy, DP-SGD.

1. Introduction

In this project, the state of the art that is machine learning methods are implemented with updated privacy-preserving policies and mechanisms with modest differential privacy. By demonstrating the track of detailed information of privacy loss, estimation on privacy loss empirically. Accuracy significantly improves when extensive data set trained from a data provider on a cloud server has significant and raised privacy concerns. However, preserving the privacy of data is a fundamental problem. Differential privacy has been a focal point in research and development in data privacy techniques. More recent DP applies to deep learning. A distortion method changes the existing raw data by adding statistical noise and data swapping while accessing processed data sets. The next session describes the fundamental terms on paper and the mechanisms used.

2. Overview

Differential privacy is a popular mechanism to train a machine learning model with bounded leakage about specific points in training data. Due to differential privacy, the model's accuracy was reduced. Here it demonstrates the data trained in neural networks through stochastic gradient descent (DP-SGD). Also, we overview the basic principles of deep learning.

a) Differential privacy:

Differential privacy is a theory that provides us with specific mathematical models with guarantees of user data privacy. It aims to reduce any impact of any personal data on the overall result; this means one can make the same interference though it is not in the input of analysis. The result of differential privacy computation is immune to broad privacy attacks. This is achieved by tuned noise during the calculation to make it difficult for a hacker to identify any user. This also leads to erosion of accuracy. Privacy is measured by epsilon and is inversely proportional to privacy protection. Achieving epsilon differential privacy is an ideal case and is difficult to achieve in reality, and hence, we used (ε, δ) differential privacy. By using this, the Algorithm is ε –differentially with probability (δ-1). Differential privacy has some properties like composability, group privacy, and robustness in an information system. Composability use for the modular design of the mechanism. Group privacy implies the degradation of privacy in the dataset. Robustness means privacy is not affected by side information. Hence δ is to 0, the better. Additive noise mechanism includes an approximation of functionality by bounded sensitivity function as above, choosing additive noise parameters, and performing privacy analysis of results.

b) Deep learning:

Deep learning is widely used in machine learning tasks that define parameterized functions from input to output with fundamental building blocks. Likewise, affine transformation and a more straightforward non-linear process. When individual data (e.g., User habits, clinical records, media) are used to train a deep learning model, some features are recognized. So for this profound learning, use specific traditional techniques. L2 (loss function) regularisation protects the privacy of training data to prevent over-fitting. To enhance the security of the deep learning model with the aid of a differential privacy model, how much to add and where it should locate the noise in the deep network is the primary thing that should be carefully considered. Because any minute change can differ the layer by layer abstraction, the more diverse the application strategy of the deep learning model, the more secured system will be.

3. Threats in deep learning:

Currently, the Deep learning system additionally faces security issues. There are various risks involved in training data of the DL model as data owned by multiple services while demonstrating with training. To attempt these issues by third party system, various attempts are managed to reduce threats by applying traditional privacy policies like Differential privacy or secure multiparty computation. To summarise the privacy issue and securities in deep learning execution, there are domains in which insecurities countered are **Attacks on the DL**

model: Two major types of attacks evasion and poison. Evasion attack related to inference phase where poison describes training phase.

Defense of DL model: Various defense techniques include two large groups, evasion, and poisoning, Eg. Gradient masking, robustness, detection.

Privacy attacks: These arise from service providers, information users.

Defense against privacy breach: That is homo-morphed encryption, Differential privacy.

Recently collaborative learning is employed in deep understanding, in which local and centralized users take action to train the data. They train the Generative Adversarial Networks(GAN) to avoid theft. It generates prototypical samples with the same distributed data. The training phase always has the active user. During the prediction phase, privacy attacks have been discussed. It has three aspects,

A. Membership interference attack:

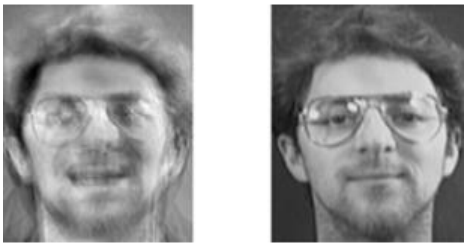
It is a black box threat. Membership interference has been studied in many different domains of study, from biomedical data to mobile services. It aims to detect the data generated is used for training or not. So this action can raise some privacy issues in the dataset as membership reveals the personal information. Training the attack model first forms the 'shadow model,' which is similar to the target model. Supervised training on shadow model and training by synthesized data with the same statistical feature as trained data. Furthermore, finally attack model was built. The principle of membership attack has a common aim as a differential privacy mechanism. Therefore most current privacy protection of differential privacy is used for membership attacks.

B. Training data extraction:

Privacy threat training data extraction in the white box in training. Several attacks in the Machine learning model based on network traffic classification by implementing support vector machine (SVM) and speech recognition software based on the Markov model. Extensive use of MLaaS in black box threats can become easily recognizable. In this setting, we should attempt to train our model for targeting the model's output into input data. We could recreate private data through this model inversion by intercepting output service data of the user's data and running it through the attacker. As we see, the left side is recreated data of the actual right side image.

C. Model Extraction:

This extraction model encounters the privacy issue when the model is trained on the user's privacy information. It extracts the parameter introduced on the model. The main intend of the attacker is to duplicate the model function, which prediction performance is the same as targeted, because of a close relation between model parameter and training data. Then



privacy is evaluated after a leak of parameters through a black-box model. A model f^ which is similar to target one f is built through a continuous sample to the black box and recording the prediction vector. And then according to pathfinding decision tree of original data obtained.

4. Preliminaries

A. Definition: A randomized mechanism M:D R where D is the domain, R is a range which satisfies (ε, δ)-differential privacy for any two adjacent inputs d, d' and subset output. The trade-off between any privacy leakage is controlled by privacy budget parameters. As smaller the privacy budget, the minute leakage and robust protection. The randomized mechanism gives ε differential protection, which is a stronger one.

B. Composition theorem: In which differential privacy has two privacy budgets that are sequential and parallel composition.

1) Sequential composition- In which random mechanism M sequentially performs dataset D, each tool provides ε-Differential privacy.
2) Parallel composition- In which random privacy mechanism M and data set D is divided into subparts (D1, D2, ...), the mechanism provides ε DP for every data set D and the final result will be of entire data set as max (ε1, ε2, ...) DP.

C. Sensitivity: Sensitivity defines query results on adjacent data sets, detecting output change due to a single sample in a hard case. Sensitivity f is based on query f and distributed data set.

D. Privacy loss: It causes due to addition of random uneasiness in the Algorithm. Privacy loss is calculated at each step of the Algorithm during execution, and it demonstrates the overall privacy loss at the end, which the privacy accountant controls.

E. Utility Measurement: It is measured by the amount of noise and errors in the set. The minute amount of noise also affects higher utility. Errors are tangled by the accuracy index phenomenon, which evaluates between private and non-private output.

5. Our approach

This is the central theory of the paper in which we focus on solution methods to privacy issues in deep learning. As differential privacy aims to ensure that regardless of the guarantee of personal user data, whether individual record included in the data or not, a query on result returns approximately the same data. Therefore, we need to know the maximum impact on personal data, which is possible by the Differential privacy algorithm.

Differential privacy and deep learning Mechanism: Differential privacy guarantees that the output of the deep learning model will not show statistical difference. In contrast, the model is being carried on adjacent datasets, which includes individual privacy. The mechanism's goal is training dataset privacy protection, Prevent information leakage in a black box and white box. Here is the approach of implementing differential privacy in deep learning.

Differential privacy with SGD algorithm: In this, we describe the more sophisticated approach used to control the influence of training data during training data, primarily in SGD method computation.

The Algorithm executes the primary methods to train the model with 0 parameters by minimizing empirical loss function L(0). At each stage SGD, computation gradient $\nabla\theta L(\theta, xi)$ for a random subset. Compute the L2 norm for each gradient, find the average addition of noise to attain privacy protection, and finally take a step opposite of average gradient noise computation. We can also find the privacy loss function, which is based on privacy accountants.

We normalize the running time of training data by expressing the number of epochs. Each epoch represents the expected bath number required to process the Algorithm; each epoch consists of N/L lots. Where N is dataset size and L is the lot size.

Privacy accounting: While considering DP-SGD, the issue is related to computing overall privacy cost to training datasets. The composability of DP enables to implement 'accountant' to compute cost and also accumulate the cost for future process.

The Moment Accountant Details: Moment accountant is the privacy spending by considering the privacy loss as any variable and using its moment generating functions to understand that variable distribution, hence known as moment accountant. There is a lot of research going to understand the privacy losses that account for a particular noise distribution and privacy losses. In consideration with Gaussian noise, if we used σ in Algorithm to be $\sqrt{}$ 2 logs 1.25 /δ / ε, we know standard argument hence each step becomes (ε, δ) differentially private concerning other. As the lot is a random sample pickup from the database, by applying the privacy amplification theorem, each step is $(O(q\varepsilon), q\delta)$, which is differential private concerning the whole database. Where $q = L/N$ is called as sampling ratio per lot and $\varepsilon \leq 1$. The strong composition theorem yields the best bond but does not take into account the particular noise distribution. A hence more robust accounting method is utilized, known as moments accountants.

Using this theorem, we can prove that is $(O(q\varepsilon\sqrt{T}), \delta)$ differentially private for the correctly chosen setting of noise scale and clipping threshold. Because of it, the bond is tighter in two ways it saves $\sqrt{\log(1/\delta)}$, a factor in the ε part and Tq factor in δ part. As we want δ to be small and $T \gg 1/q$, the saving at the end is significant. Through this combination theorem, by combining dataset libraries, we got the accuracy through epsilon DP-SGD privacy protection $\varepsilon \approx 0.94$.

Hyperparameter tuning: It is also known as hyperparameter optimization. It is a problem of choosing an optimal hyperparameter for algorithm learning purposes. It is a parameter whose value controls the learning process and tunes to balance accuracy, performance, and privacy. The optimization finds a hyperparameter that gives an optimal model to minimize predefined loss function on shared data and also return associated losses. If we try for some settings in hyperparameter, we can add privacy costs of all stages through the moment accountant. However, we are interested in account-model accuracy by modifying this setting; It will be better than previous ones.

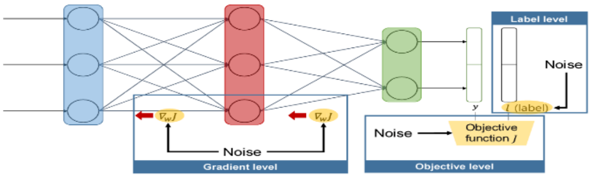
In our case, through model run with various datasets libraries by hyperparameter loop, the accuracy obtained as the trained model is ϵ value of 1.18.

6. Implementations:

We have implemented our approach that is DP-SGD, on a deep learning model by a Differential algorithm with TensorFlow. The source code is from Apache license version 2.0. Our implementation mainly consists of two methods: A sanitizer that pre-processes the gradient to attain privacy and a privacy accountant who keeps track of training data. The code contains a snippet of DPSGD_Optimizer that minimizes a loss function using a differentially private SGD and DP-Train dataset. Iteratively DPSGD optimizer works to bound the total privacy loss.

A model trained with DP-SGD provides provable differential privacy guarantees to input data. It having two vanilla SGD algorithm:

1. The sensitivity of each gradient is bounded by clipping the gradient for each training point. This limits how much each training point can impact model parameters.
2. Random noise is sampled and included to the clipped gradients to make it statistically impossible for each data point was added in the training dataset by comparing it with SGD when it operates with or without this specific data point training dataset.



As we use tf.Keras to train CNN to recognize the handwritten numbers with the DP-SGD model provided by TensorFlow. Furthermore, By setting the learning model with hyper-parameter which having three parameters,

a. l2_norm_clip (float) - To fix optimizer's sensitivity to personal training points.
b. noise_multiplier (float) - Noise is sampled and added to training data.
c. Micro batches (int) - Input data is split into micro-batches to improve utility. Input bath size should be a multiplier of the number of micro-batches.
d. learning_rate (float) - Higher learning rate, more the update matters. The learning rate is kept lower while training the data to coverage.

By using hyperparameter values, reasonable accuracy was obtained with the moment account. In comparison, we are building a learning model to a convolutional neural network with a sequential definition. After that, vector loss is obtained by defining the optimizer and loss function discussed in the learning model. Losses, for example, are computing rather than mean values to manipulate each training point.

This probability is sometimes called the privacy budget. A lower privacy budget ensures stronger privacy guarantees. If a single training point did not affect the outcome, then it is not memorized by Algorithm, and the individual's privacy is preserved.

Two metrics are used to express DP guarantee of Machine learning algorithm,

1. Delta (δ) - It bounds the probability of data. A rule of thumb is to set to be less than the inverse of the size of our training dataset. The model is developed to 10^-5 as the MNIST dataset has 60,000 training points.
2. Epsilon (ϵ) – Measures the robustness of data privacy. Provides the strength of privacy guarantee by including the probability of dataset through varying single training points. A smaller ϵ value implies better privacy protection.

Tensor flow provides the output values of training data through batch size, noise multiplier, epochs training, number of training points n.

Result:

DP- SGD outcome with sampling rate is 0.417% , and noise_multiplier is 1.3 iterated over 3600 steps, and then it satisfies the Differential privac y with epsilon is 0.942 and delta(δ) is 1e-05The optimal RDP order is 17.0. The tool reports tha t the hyperparameters which having the trained model have an ϵ value of 1.18.

7. Future scope in this work

Privacy guarantee: In today's scenario, the work is continuously developing in the privacy section. The frameworks are developing to secure functional evaluation and securing multiparty computation where input is splatted, and the focus is on avoid leakage of information. Another approach is k-anonymity which has theoretical solid and empirical limitations, but differential privacy provides an analytical framework to guess the datasets. So this can be applied to large machine learning tasks that differ from target models.

Model class: In recent work, the design and evaluation of a system for distributed neural network training. The sanitization relies on an additive noise mechanism, and as an additive mechanism based on sensitivity estimate, it can be improved to a strong sensitivity guarantee.

It can even be an attack model trying to specify user data from the model volume. The number of training data points in the model can identify appropriate and quantified as the risk to privacy. Once the reliability of the dataset is formalized, we can go towards differentially private optimizers to more complex and lengthy tasks in Computer network systems and Natural Language processing to next-level architecture.

8. Conclusion

The learning from this training model theory is seen in the tuning noise_multiplier mechanism, which attributed to a considerable depth with gradient manipulation is ineffective. Through the TensorFlow data library code, if we observe that the value of epsilon is independent of the training model and only depends on noise multiplier, batch size, epochs, and delta values we obtained from data. Epsilon has good strength to measure the amount of risk to privacy.

A new tool that may be of independent interest is the mechanism of privacy tracking, the moment accountant. It permits tight automated security to the complex composite mechanism that is beyond the composite theorem. There is no requirement metric for measuring the opaque model volume that can reveal small information about the user.

References:

1. Preventing Overfitting in Deep Learning Using Differential Privacy
   Khatri, Alizishaan Anwar Hussein.State University of New York at Buffalo. ProQuest Dissertations Publishing, 2017. 10622959.
2. Deep Learning Algorithms Design and Implementation Based on Differential Privacy Xuefeng Xu, Yanqing Yao, Lei Cheng. (November 11, 2020)
3. Differential Privacy Preservation in Deep Learning: Challenges, Opportunities, and Solutions JING WEN ZHAO1, YUN FANG CHEN1, AND WEI ZHANG 1,2, (Member, IEEE)
4. Deep Learning with Differential Privacy - Martín Abadi, Andy Chu, Ian Goodfellow† H. Brendan McMahan, Ilya Mironov, Kunal Talwar, Li Zhang October 25, 2016
5. PREVENTING OVERFITTING IN DEEP LEARNING USING DIFFERENTIAL PRIVACY by Alizishaan Anwar Hussein Khatri (SEPTEMBER 1 2017)
6. Diffprivlib: The IBM Differential Privacy Library A general-purpose, open-source Python library for differential privacy
   Naoise Holohan , Stefano Braghin
   Pol Mac Aonghusa ', Killian Levacher (IBM Research – Ireland)
7. Security and Privacy Issues in Deep Learning Ho Bae†, Jaehee Jang†, Dahlin Jung, Hyemi Jang, Heonseok Ha, Hyungyu Lee, and Sungroh Yoon* , Senior Member, IEEE
8. TensorFlow convolutional neural networks tutorial.
   www.tensorflow.org/tutorials/deep cnn
9. A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. Machine Learning, 94(3):401–437, 2014.
10. N. Phan, Y. Wang, X. Wu, and D. Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In AAAI, pages 1309–1316, 2016
11. C. Dwork and G. N. Rothblum. Concentrated differential privacy. CoRR, abs/1603.01887, 2016.

12. C. Dwork and J. Lei. Differential privacy and robust statistics. In STOC, pages 371–380. ACM, 2009.

13. K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. J. Machine Learning Research, 12:1069–1109, 2011.

14. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

15. CIFAR-10 and CIFAR-100 datasets. www.cs.toronto.edu/˜kriz/cifar.html.

16. C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In FOCS, pages 51–60. IEEE, 2010.