DEEP LEARNING WITH DIFFERENTIAL PRIVACY

Sharique Shamim, 2018CS10388

Abstract

Area

Machine learning algorithms are widely used in the area of Big Data and its analysis. Neural Networks and Deep Learning have achieved exceptional results in various domains such as image classification, natural language processing, healthcare and many more. The advancement in these areas have been possible due to availbility of large and representative datasets which are used to train the neural network.

Problem

The datasets used to train the neural networks are often crowdsourced and may contain sensitive information. As a tremendous amount of data is freely available, it is possible for anyone to extract user data from the model weights by getting hold of any necessary information.

Model inversion attacks can exploit confidential information and can also lead to wrong predictions on the data. For instance, Friedrikson et al. used a model-inversion attack to extract images from a facial recognition model thus exploiting privacy.

Therefore it becomes very important to provide principled privacy guarantees while meeting the requirements and achieving the same results.

Solution

Addressing this problem, we will implement algorithms to train deep neural networks with non-convex objective to compute privacy costs within the framework of differential privacy under a modest privacy budget, feasible software and computational complexity, training efficiency while maintaining the quality and performance of the model.

Evaluation

We plan to use the deep learning framework Pytorch and Tensorflow for training the model and tensorflow_privacy and syft framework to ensure differential privacy. We will evaluate our algorithm on the MNIST Digit Dataset which is a standard image classification dataset and has been widely used in various machine learning tasks. We plan to do the simulation on Google Colab using GPU.

Tools Used

- Deep Learning Framework: torch, tensorflow
- Differential Privacy Framework: syft, tensorflow_privacy
- GPU: 12 GB NVIDIA Tesla K80
- Others: Google Colab, numpy, matplotlib

Takeaway

Differentially private algorithms are widely used in various applications. One of the biggest deployment of differential privacy is Google's RAPPOR Project which is used to report usage statistics for Google Chrome. They are also used by government agencies to publish information while ensuring confidentiality. These algorithms are used by various companies while collecting user data which is not visible to internal analysts.

Introduction

In this section we briefly overview basic principles of deep learning, definition of differential privacy and understand the differentially private SGD algorithm.

What is Differential Privacy ?

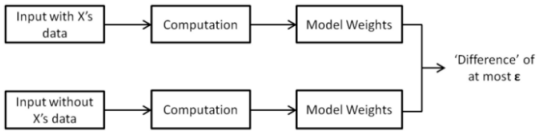
Differential privacy is a constraint on algorithms to limit the amount of the private information that is public. It mathematically guarantees us the privacy of user information. Differential privacy is a system by which a dataset can be publically shared by while describing patterns and trends in the dataset and preserving user related sensitive information.

Differential privacy is a mechanism which helps us to make the same predictions or inferences on a dataset irrespective of whether the individual's data was available during the analysis or not [1]. Differentially private algorithms helps to prevent exposure of user information and makes it immune to a wide range of privacy attacks.

Definition: (ϵ, δ) Differential privacy [2] states that the distribution of the output M(D) on the database D is nearly the same as M(D'):

$$\forall S \ \Pr[\mathrm{M(D)} \in \mathrm{S}] < e^{\epsilon} \cdot \Pr[\mathrm{M(D')} \in \mathrm{S}] + \delta$$

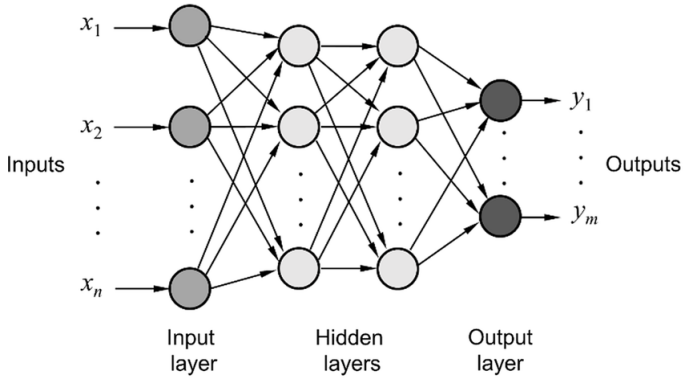
Here e^{ϵ} quantifies information leakage and δ denotes a small probability of failure.



Deep Learning

Deep Neural Networks are compositions of basic building blocks called units in a multi-layered structure. It makes use of parametrized functions which are linear or non-linear transformations to map a m-dimensional input space to some n-dimensional output space. Sigmoid and Rectified Linear Units (ReLUs) are common example of such transformations. The model is trained by varying the parameters of the units with the goal of fitting any finite set of input/output examples by making use of a backpropagation algorithm.

Deep Neural Networks are often trained by making use of Stochastic Gradient Descent which is used to minimize the average loss on the training data. The loss $L(\theta)$ on parameters θ is the average of the loss over the training examples $\{x_1, ..., x_N\}$, so $L(\theta) = 1/N \sum_i L(\theta, x_i)$. The model parameters are updated until the loss is acceptably small on the training data..



Differential Private SGD Algorithm

Input: Examples [1] $\{x_1, x_2, ..., x_N\}$, loss function $L(\theta) = \frac{1}{N}\sum L(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

`Initialize` θ_0 randomly

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

Take a random sample L_i with sampling probabiility L/N

`Compute Gradient`

for each $i \in L_t$ compute $g_t(x_i) \leftarrow \nabla_{\theta_t} L(\theta_t, x_i)$

`Clip Gradient`

$\tilde{g}_t(x_i) \leftarrow g_t(x_i)/max(1, \frac{||g_t(x_i)||_2}{C})$

`Add Noise`

$\tilde{g}_t \leftarrow \frac{1}{L}(\sum \tilde{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$

`Descent`

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}_t$

Output θ_T and compute the overall privacy cost (ϵ, δ) using a privacy accounting method.

Compute Gradient

In most neural networks the objective i.e loss function $L(\theta)$ is convex hence difficult to minimize. Mini-Batch Stochastic Gradient descent (SGD) is the most widely algorithm to minimize the loss function. In this algorithm at each step we randomly sample a batch of B examples and compute the gradient on this batch as $g_B = 1/|B| \sum_{x \in B} \nabla_\theta L(\theta, x))$.

Clip Gradient

The differential privacy of any algorithm is guranteed by providing a bound on the examples $\tilde{g}_t$. In this algorithm, there is no bound on the gradient computed by using the loss function. In order to provide the bounding influence we clip the gradients using the l_2 norm. By clipping the gradient we ensure that if $||g||_2 < C$, then g is preserved, whereas if $||g||_2 > C$, it gets scaled down to be of norm C where C is the clipping threshold.

Add Noise

By introducing distortion or randomness into the output of the neural network g_t we preserve the privacy as evident from the definition of differential privacy.

Descent

The parameters are updated until convergence as done in the gradient descent algorithm using a suitable learning rate.

Computing the privacy costs

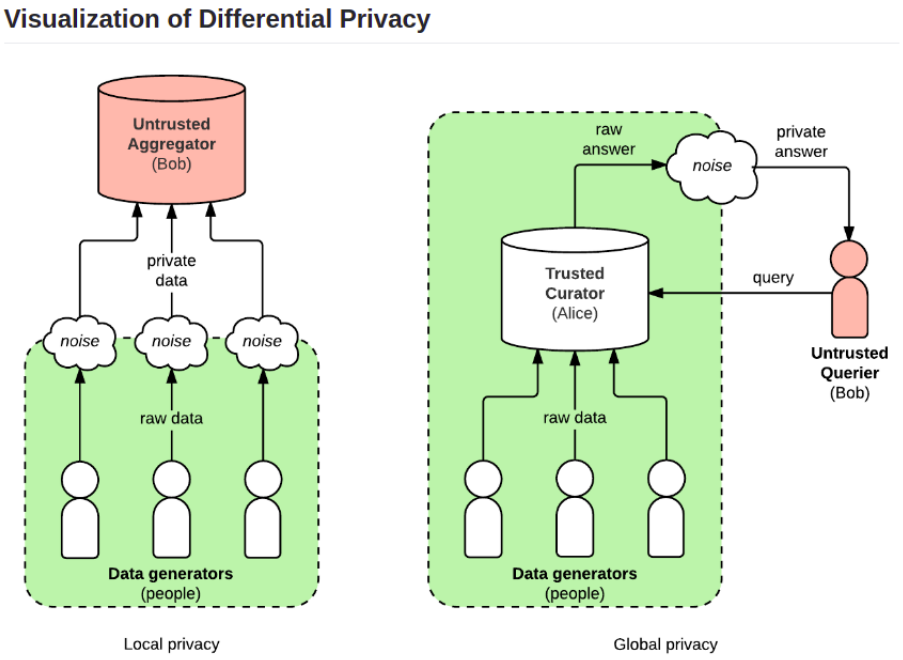
We use a privacy accounting method to compute the privacy cost of the differentially private model. During training the privacy cost is monitored by making used of a Privacy Accountant. For the gaussian noise, if we use $\sigma = \sqrt{2log\frac{1.25}{\delta}}/\epsilon$ then by standard definition [3], each step is differentially private. For neighboring databases d, d' $\in D^n$, a mechanism M, auxiliary input aux, and an outcome $o \in R$, define the privacy loss at o as:

$$\text{c(o; M, aux, d, d')} = \log \frac{Pr[M(aux, d) = o]}{Pr[M(aux, d') = o]}$$

We will use inbuilt methods to compute privacy costs and will not be implementing it.

Proposed Plan

Differential Privacy can be divided in two categories depending on where we introduce the distortion inorder to preserve the privacy. Local Differential Privacy [9] is when the distortion or randomness has been introduced in the data whereas Global Differential Privacy [4] is when the distortion or randomness has been introduced in the output on the examples obtained by the the parametrized functions. Noise is introduced in the data or the output to preserve the privacy while also preserving the general trends in the dataset for statistical analysis. Global Differential Privacy makes use of teacher models to find the labels on an unlabelled dataset.



- In this project we will use Global Differential Privacy which makes use of teacher models. We will use the teacher models that have been trained on unique datasets to evaluate the labels on an unlabelled test dataset which would be further used for training the differentially private model.
- We will make use of a mechanism called PATE analysis to evaluate the labels that have been generated on the unlabelled dataset by the teacher models. PATE analysis has been used to determine the trade-off between privacy and accuracy during training of the teacher models.
- We will randomize the labels generated by the teacher models on the unlabelled dataset by applying Laplacian noise. This has been done to purposely mislabel few examples to conserve privacy.
- After generating the labels, we will create and train a new model which will be differentially private.
- Hyperparameter tuning will be done to find the best values of ϵ and δ. Experiments will be done with η to find the best value.
- We will also vary the number of teacher models and epochs until a reasonable trade-off between privacy and accuracy is obtained.
- After this we plot different curves such as ϵ vs epochs, Accuracy vs epoch and Accuracy vs ϵ for varying values of δ to reproduce the results in the paper.

Related Work

The problem of privacy-preserving machine learning, deep learning or data-mining has been a focus of active work in several research communities since the late 90s. Some of the works that have been done this area are:

Privacy attacks that can be made on a deep learning model has been presented from three aspects: training data extraction, model extracting and membership inference in the paper [5]. The layers differential privacy mechanism has been deployed in this paper.

The main aspects of differential privacy such as interactive and non-interactive setttings, perturbation mechanism and its applications in recent research has been discussed in the paper [6].

The various threats and the defense mecanishm on deep learning privacy models has been presented in the paper [7]. It also discusses about the various aspects where random noise can be introduced into the data, gradient or the parametrized functions to protect the privacy model.

The paper [9] talks about local differential privacy, here the private data is regarded as a singletuple, so any different pairs of data are neighboring. Local differential privacy hides the complete data, not just each individual record, from the adversary.

The paper [10] presents views on the possibilties for improving performance with differential privacy techniques in the areas of machine learning, deep learning, multi-agent systems and artificial intelligence.

References

[1] Deep Learning with Differential Privacy. Martin Abadi, Andy Chu, Ian Goodfellow, Kunal Talwar, Li Zhang, Ilya Mironov, H. Brendan MnMahan
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45428.pdf

[2] A practical guide on how Differential Privacy is integrated into Deep Learning architectures for image classification
https://towardsdatascience.com/differential-privacy-in-deep-learning-cf9cc3591d28

[3] EE734 CMU - Deep Learning with Differential Privacy Lecture.
https://course.ece.cmu.edu/ ece734/fall2016/lectures/Deep_Learning_with_differential_privacy.pdf

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

[5] Deep Learning with Differential Privacy.
https://github.com/ateniolatobi/Differential-privacy-for-deeplearning-project

[6] Deep Learning with Differential Privacy.
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8683991

[7] Differential Privacy in Practice. Hiep Huu Nguyen, John Kim, Yoonho Kim
https://www.researchgate.net/publication/264029732_Differential_Privacy_in_Practice

[8] Differential Privacy in Deep Learning: An Overview. Trung Ha, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, Manh Tuan Nguyen.
https://ieeexplore.ieee.org/document/9044259

[9] Distributed Deep Learning under DifferentialPrivacy with the Teacher-Student Paradigm. Jun Zhao.
https://ieeexplore.ieee.org/document/9044259

[10] Local Differential Privacy. Kairouz, Oh, and Viswanath2014; Qin et al. 2016; Zhao and Zhang 2017; Kairouz, Oh,and Viswanath 2014
https://ieeexplore.ieee.org/document/9044259

[11] More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence. Tianqing Zhu, Dayong Ye, Wei Wang, Wanlei Zhou, Philip S Yu.
https://arxiv.org/abs/2008.01916