Privacy and Randomness
Wallace He

Final Project Report

Assignment Handout
Dec 7, 2017

1 Introduction

Background

Deep learning based on artificial neural networks is a very popular approach to build model classifying and to recognize complex data input such as images, speech, and text. Training a deep learning model[1] would need to have many training data to generate the model for future prediction and self-update the model weight by back-propagation. Massive data collection required for deep learning training process presents some obvious privacy issues. After the training process, a model may inadvertently and implicitly store some of its training data; careful analysis of the model may, therefore, reveal sensitive information. From [FR15], individual face images with private information can be obtained from public facial recognition model through the Fredrikson Attack. It would cause both privacy and social issues since many deep learning models can be assessed easily from public API.

To protect the privacy of training data, [PA16] improves upon a specific, structured application of the techniques of knowledge aggregation. The mechanism is called Private Aggregation of TeacherEnsembles (PATE). First It created multiple machine learning models. The sensitive data-sets used to train the models can be seen as the training data-sets. Each model, considered as teacher model, is trained with a disjoint subset of training data-sets. Then by using auxiliary, public non-sensitive data, a student model is trained on the aggregate output of the ensemble of teachers, such that the students learn to accurately mimic the ensemble without access to the sensitive data. After finishing training, this student model would be our output model for predicting the data after a given unlabeled input.

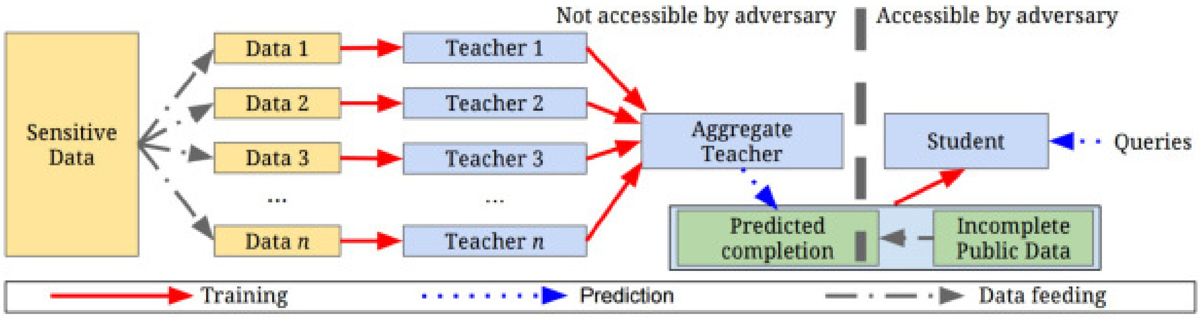


Figure 1: Overview of the approach: (1) an ensemble of teachers is trained on disjoint subsets of the sensitive data, (2) a student model is trained on public data labeled using the ensemble.

Analysis of privacy

From [PA16], suppose n would be number of the teachers and T is the number of public data that used for training the student model. At each sample of public data prediction, we use the aggregation mechanism with noise Lap(1/γ) which is (2γ, 0)-DP. Thus over T steps, we get (4Tγ^2 + 2γ $\sqrt{2Tlog(1/\delta)}$, δ)-DP by using the strong composition theorem. Therefore,the privacy guarantees can be enhanced by restricting student training to a limited number of teacher models' voting. Also, the apply of differential privacy can be seen as

[1] A ML model can be considered as large parameters of pattern based on the input training data-sets

adding noise of teacher models output for limiting the effect of any single sensitive data item on students learning. The student's privacy properties can be understood both intuitively (since no single teacher and thus no single data-set dictates the students training) and formally, regarding differential privacy.

Motivation

In last summer, I built a Convolutional Neural Network model with Dr.Barnett for images prediction. Convolutional Neural Network model is very similar to ordinary Neural Networks model: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single different score function: from the raw image pixels on one end to class scores at the other. They still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer [2]. In the high level training process, It can be simplified as the input of image and out put the prediction of the class(class scores). Based on the class score and correct label, the model will update the weight.

From reading and analysis the paper [PAE16], based on the given mechanism above, I implement the Private Aggregation of Teacher Ensembles (PATE) with my own data sets and own convolutional Neural Networks models to test whether the student model would be able to make good prediction through that training. Because the CNN code is written in Python and Keras back-end, this project would be written into a keras-backend. Also, the code of [PAE16] is posted on Github [3] by using a different deep learning Network model based on Tensorflow. In some parts of the final project, I took this code as a reference.

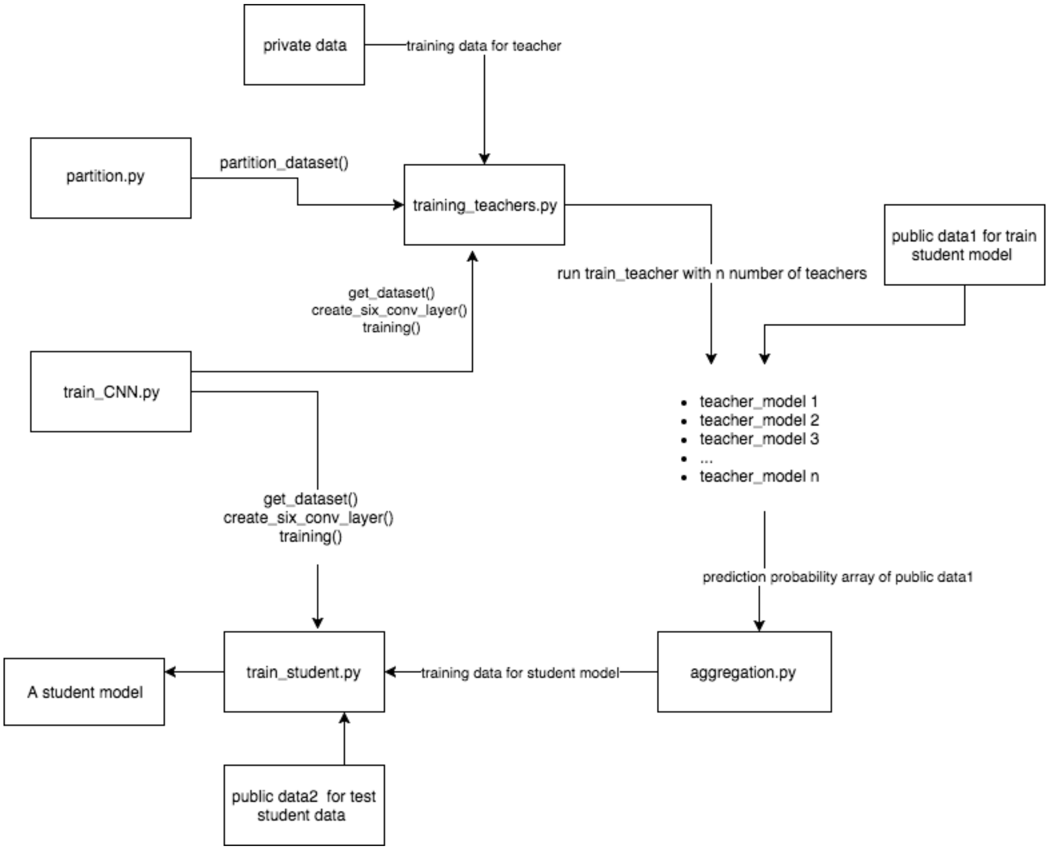
After implementing, the new student model was tested by the custom data sets (non-sensitive data) and comparing with the original training model based on mainly the accuracy loss since we already prove the privacy.

2 Detail of Implementation

The project has 6 files in total includes: trainTeachers.py, trainStudent.py, aggrgation.py, trainCNN.py, and partition.py. The the flowchart of whole program is listed below.

[2] CS231 course note from Standford

[3] https://github.com/tensorflow/models/tree/master/research/differential-privacy/multiple-teachers



Description of flowchart

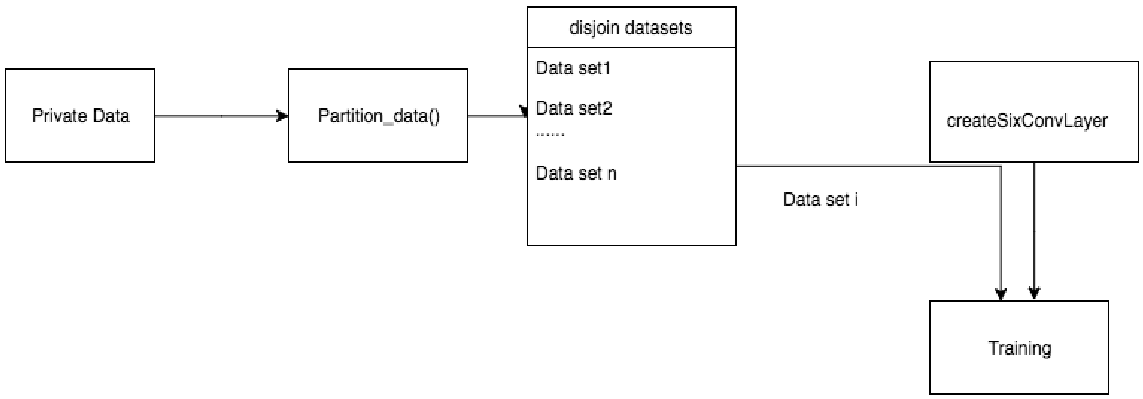
Input of whole program:
private dataset: P for training teacher model
public dataset 1: A for training student model
public dataset 2: B for testing student model
Number of teacher-models n and teacher-model's id i .

Step1: We will train a single teacher model based on the given private data-sets. the Inside of trainTeachers.py's **trainTeacher()** function, import **getDataset()** function from the trainCNN.py to get the private data by given directory. Then use **createSixConvLayer()** function from same file to create the a empty $model_i$. Then use the **partitionDataset()** function from the partition.py to partition the p into n part disjoint data-sets P_n with same length. Use P_i to train $model_i$. Then run trainTeacher() n times to create teacher models: $model_1 ... model_n$.

Step1: trainTeacher(n, i)

param: n: total number of teachers in the ensemble

param: i: id of the teacher being trained



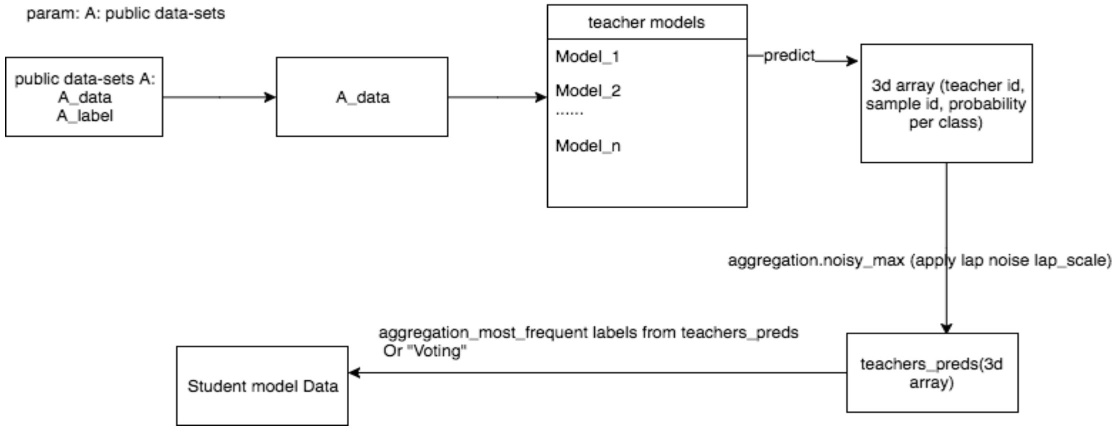
Step2: Then we will use each teacherModel $model_1 ... model_n$ and public data-sets A to prepare the training data-sets for the student-model. For public data-sets A, we only take the A's data part instead of label of A for creating the datasets. For each teacherModel, $model_1 ... model_n$ will make a prediction of each samples of A-data. This process will return a 3d array with teacher model's id, sample id, and probability per class. (**ensemblePreds()** fuction from trainStudent.py) Then we will apply the Laplace noise into probability of 3d array by the function noisymax in Aggregation.py. Then used the function **aggregation-most-frequent()** from Aggregation.py to get the label of A-data based on the 'most frequent Vote/highest probability of prediction' inside of 3d array.

Step2: PrepareStudent Data (n, lap_scale,A)

param: n: total number of teachers in the ensemble

param: lap_scale: id of the teacher being trained

param: A: public data-sets


public data-sets A:
A_data
A_label
A_data
teacher models
Model_1
Model_2
......
Model_n
predict
3d array (teacher id, sample id, probability per class)
aggregation.noisy_max (apply lap noise lap_scale)
teachers_preds(3d array)
aggregation_most_frequent labels from teachers_preds
Or "Voting"
Student model Data

Step3. Use the student model data as input to train the student model. Then test student model's accuracy by public datasets B. Return student model.

Implement changes from reference code and clarification

1.Most of the project is following the idea of reference code but most of the functions only used the name of reference code except for aggregation.py and partition.py .

2.Modified the partition-dataset() function inside of partition.py to fit the given data-sets and python3.

3.Modify the aggregation.py to fit in Keras version and python3.

3 Experiments

Methodology

Meth1. I used default data-sets of CNN for training . The default data-sets contains around 2500 of birds pictures with 11 different class. Those pictures can be considered as the private data-sets P. Then used the mechanism above, I created 10 teacher models, each teacher had around 250 pictures for training. Based on the 10 teacher models, I set up a extra 400 public picture data-sets as public data-set for training student-model1. Also, I used the same 400 pictures with 11 classes training a empty comparingModel1 as comparison. Extra 100 pictures for testing the accuracy of student-model1 and comparingModel1. We use Laplace scale of 20 to guarantee an individual query privacy bound of $\epsilon = 0.05$

Meth2. I modified the default data-sets for training. The new data-sets contains around 2500 of birds pictures with only 2 class for training: bird and nobird. Those pictures can be considered as the private data-sets P'. Then following the mechanism above, I created 10 teacher models, each teacher had around 250 pictures for training. Based on the 10 teacher models, I set up a extra 400 public pictures with labels as public data-set for training student-model2. In 400 pictures, 340 pictures set as training picture. All of 340 pictures got new label by teacher models' voting and trained for student-model2

Also, I used the same 340 pictures with 2 classes training a empty comparingModel2 as comparison. Extra 60 pictures used for testing the accuracy of student-model2 and comparingModel2. We use Laplace scale of 20 to guarantee an individual query privacy bound of $\epsilon = 0.05$

Meth3. Based on the same teacher models of 2, with holding same conditions, I changed up to 200 public pictures getting new labels for training student-model3 and rest 140 public pictures with labels also for training. We use Laplace scale of 20 to guarantee an individual query privacy bound of $\epsilon = 0.05$

Table 1: Experiment result

Experiments	ϵ	δ	Queries(# voting pics)	Student Accuracy	# of classes
Meth1	13.597	10ˆ5	340	66.34%	11
Meth2	12.248	10ˆ5	340	81.74%	2
Meth3	8.786	10ˆ5	200	84.51%	2

Improvement: After Talking with Thang, I realized that my previous implementation made some mistakes. My Meth1 and Meth2 data for training student-model actually all get the noisy labeled by teacher models. Based on the fact that the whole algorithm of PATE is (ϵ,δ)= (4Tγ^2 + 2γ $\sqrt{2Tlog(1/\delta)}$, δ)-DP and T as the number of labeling data for student-model by teachers' voting. If T is huge, the privacy loss will be huge. In Meth1 and 2 we have the $\epsilon = 12.24$. Therefore, I changed from Meth2 to Meth3, which would only take 200 public pictures for teacher model voting for labels and applying the Laplace noise. The rest of public pictures are also used for training student-model3.

Result and analysis: For Meth1, the accuracy of student-model1 is bad, around 0.6634 and the accuracy of comparingModel1 0.8615. The result indicates that applying PATE for protecting privacy on CNN model would in some point affect the accuracy of prediction. Also one of the reason caused this result is that We dont have a large training data-set for training teacher-models, the average accuracy of teacher-model is between 0.76-0.79. Since the accuracy of teacher-model is not high enough, the student-model's accuracy would not be too good.

For Meth2, the accuracy of student-model2 is around 0.8174. and the accuracy of comparingModel2 is over 90 percent. The result indicates that applying PATE for protecting privacy on CNN model would in some point affect the accuracy of prediction. Comparing with Meth1, We can see the average accuracy of teacher-model is increased to around 0.88-0.93, therefore the accuracy of student-model2 would increase this time. The gap between the accuracy of student-model2 and comparingModel2 can be considered as the loss caused by applying privacy.

For Meth3, the accuracy of student-model3 is around 0.8451. The result is surprise because the privacy bound is higher but the accuracy is higher than Meth2 too. One of the explanation is that there is not too many difference between public data (training for student-model) and private data (for teachers) since I basically created those two from the same data-set. Also, applying Laplace noise with more queries (public data needed to be labeled) would create more prediction accuracy loss. Therefore, if we train student model based on more public data with label and less public data with teacher models' voting, it would create higher accuracy and high privacy bound only under the circumstance that the private data and public data are highly homogeneous. If we are under the situation without labels of public data, we will have to label all the public data-set with teacher models' voting with adding Laplace noise. Under this situation, the privacy bound would be bad.

To solve this problem, the authors of [PA16] would choose subset of T to train a Generative Adversarial Network to generate student-model, which would be useful for getting smaller privacy loss. In future, I will think about implementing GAN to train student-model.

4 References

[PA16] Papernot, Nicolas, et al. Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data. ICLR 17 , arxiv.org/abs/1610.05755.

[FR15]Fredrikson, Matt, et al. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. ACM Digital Library, ACM, 16 Oct. 2015, doi.acm.org/10.1145/2810103.2813677.