YOU ARE NOT ALLOWED TO SHARE THE CONTENT WITH OTHERS OR DISSEMINATE THE CONTENT

NUS CS-CS5562: Trustworthy Machine Learning	October 13, 2023
Assignment 4	
Lecturer: Reza Shokri	*Student:*

The objective of this assignment is to let you understand and implement privacy-preserving machine learning methods such as differentially private stochastic gradient descent (DP-SGD). The assignment contains the following parts:

1. **Warm up: Non-private Logistic Regression:** Implement the non-private SGD algorithm to get familiar with the procedure for training logistic regression models.

2. **Private Training via DP-SGD:** Modify the non-private SGD algorithm to implement a private SGD algorithm. Reason about the effect of algorithm design (noise, step-size, batch-size, number of iterations) on test accuracy.

3. **Membership inference on DP models:** Observe how differential privacy bounds limit the success of membership inference attacks.

You will need to implement the code in `assignment_4.ipynb` and write a report about the tasks. Details about the exact items to be reported are given in the corresponding task descriptions in this document.

The code in `assignment_4.ipynb` was tested using `tensorflow=2.4`, please use the same version to avoid any errors.

- You need to use the LaTeX template we provided in the `report/report.tex`.

- **The report should ONLY contain FOUR pages. Anything that exceeds four pages will be ignored.**

- You are required to submit the completed `assignment_4.ipynb`, your LaTeX file(s) and the compiled PDF file for the report (name it `report.pdf` and zip it with the notebook for submission).

1 Warm up: Non-private Logistic Regression

A training algorithm for logistic regression takes a dataset $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, and returns a logistic regression classifier for approximately (or exactly) solving the following optimization problem:

$$\arg\min_{\theta} L(\theta; D) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(\frac{exp(\langle\theta, \bar{x}_i\rangle)}{1 + exp(\langle\theta, \bar{x}_i\rangle)}\right) + (1 - y_i)\log\left(\frac{1}{1 + exp(\langle\theta, \bar{x}_i\rangle)}\right)\right), \tag{1}$$

where $\theta \in \mathbb{R}^{d+1}$ is the model parameter to be estimated and $\bar{x}_i^T = (1, x_i^T)$.

You are required to train a non-private logistic regression model on a binary classification dataset e.g. Adult, Arcene. The code for loading both datasets has been provided. Here is the workflow:

1. Complete the method `sgd` in the notebook that uses SGD to solve the logistic regression problem.
2. The method `custom_logistic_obj_and_grad` computes the loss value and gradient of the logistic regression problem.
3. The class `CustomSGDLogisticRegression` trains your logistic regression model.
4. Load the Adult (or Arcene) dataset.
5. Use the SGD algorithm ($B = 1, \gamma = 0.25, E = 20, \theta_0 = 0$) to train a logistic regression model on the chosen training set in the notebook.
6. Plot the train and test accuracies of the model under a variety of training set size = 5000, 10000, 20000, 30000 (numbers given for Adult dataset) in the notebook.

This question will help you set up the code for the next part of the assignment.

2 Private Training via DP-SGD

2.1 DP-SGD

You are required to modify the private SGD function to include gradient clipping and additive noise. The pseudocode for DP-SGD with noise and clipping is given below in Algorithm 1.

1. Complete the method `dp_sgd` in the notebook that performs differentially private stochastic gradient descent, by clipping the mini-batch gradient and adding noise to it.

2. Class `CustomPrivateSGDLogisticRegression` trains your private logistic regression model.

3. Load the Adult (or Arcene) dataset.

4. In the notebook, use the DP-SGD algorithm ($\sigma = 0.05, B = 1, \gamma = 0.25, E = 20, \theta_0 = 0, C = 1$) to train private logistic regression models on the chosen training sets with size 10000 and compute the train and test accuracies of this model.

5. In the **report**, write down how do they compare to the train and test accuracies of the non-private model on dataset with size 10000?

2.2 Computing Privacy Parameters of DP-SGD using Moments Accountant

An important issue for DP-SGD is computing the overall privacy cost of training a model. Abadi et al. [2016] give a method to compute the privacy cost by utilizing the

Algorithm 1: Pseudocode for the DP-SGD algorithm with noise and clipping.

```
Data: Dataset (X, y), loss function L, initial model θ_0, number of epochs E,
      mini-batch size B, noise multiplier i.e. scale σ, learning rate γ,
      clipping norm C.
Result: θ_priv
Set θ_e ⟵ θ_0;
Set (X_B, y_B) ⟵ randomly batched (X, y) according to mini-batch size B;
for epoch e from 0 to E do
    for batch (X_b, y_b) in (X_B, y_B) do
        /* Compute mini-batch gradient                                  */
        Compute per-example gradient g_b(X_{b,i}) ⟵ ∇_{θ_e} L(θ_e, X_{b,i}, y_{b,i});
        /* Clip mini-batch gradient                                     */
        ĝ_b(X_{b,i}) ⟵ g_b(X_{b,i}) / max(1, ||g_b(X_{b,i})||_2 / C);
        /* Add noise to mini-batch gradient                             */
        ĝ_b ⟵ (1/B) Σ_i (g_b(X_{b,i}) + N(0, σ^2 C^2 I));
        /* Update theta with noisy gradient                             */
        θ_e ⟵ θ_e − γ ĝ_b
    end
end
Set θ_priv ⟵ θ_e;
return θ_priv
```

composability property of differential privacy. They propose a "moments accountant" procedure that computes the privacy cost at each access to the training data, and accumulates this cost as the training progresses. We reproduce Theorem 1 from the paper here:

Theorem 1. There exist constants c_1 and c_2 so that, given the sampling probability $q = \frac{B}{N}$ (where B = mini-batch size, N = total number of data points) and the number of steps E, for any $\epsilon < c_1 q^2 E$, Algorithm 1 is (ϵ, δ)-differentially private for any $\delta > 0$ if we choose the noise scale to be:

$$\sigma \geq c_2 \frac{q\sqrt{E \log \frac{1}{\delta}}}{\epsilon} \tag{2}$$

In `report.pdf`, using the mini-batch size $B = 1$, dataset size 10000, noise scale $\sigma = 0.05$, and number of epochs $E = 20$ specified for training the private model using Algorithm 1, compute the resulting ϵ for $\delta = 10^{-5}$.

Hints

- Following the proof of Theorem 1 given by Abadi et al. [2016], compute the moments accountant $\alpha(\lambda)$ of the DP-SGD algorithm with $\sigma = 0.05, B = 1, \gamma = 0.25, E = 20, \theta_0 = 0, C = 1$ and datasets size 10000. (You can ignore the $O(q^3\lambda^3/\sigma^3)$ term in [Abadi et al., 2016, Lemma 3].)

- Putting the $\alpha(\lambda)$ value computed into Theorem 2.2 given by Abadi et al. [2016], you can solve the optimization problem to compute δ for different values of ϵ.

- Using the previous result (e.g. by using binary search), you can compute the ϵ for your algorithm for $\delta = 10^{-5}$.

2.3 Effect of Clipping Norm on Accuracy

In this question you will observe how the true gradient g and privatized gradient $\hat{g}$ diverge for your private model trained using Algorithm 1, and the effect of the clipping norm on the accuracy of the model.

1. Choose a clipping norm e.g. $C = 1.0$. Select a batch of points for which you will store the gradient before and after clipping and adding noise, i.e., store the

true gradient g and the privatized gradient $\hat{g}$. You will need to compute and store these values for multiple epochs at epoch 1, 2, ..., 20. Observe how the trajectories of g and $\hat{g}$ differ. In `report.pdf`, write down your observations. You can include plots to show your findings.

2. Repeat the experiment above for multiple clipping norms, e.g., $C = [0.1, 0.5, 1.0, 2.0]$. Observe how the clipping norm affects the difference between g and $\hat{g}$. How does the clipping norm affect the accuracy of the final model? Write down your observations in `report.pdf`. You can include plots to support your observations.

3 Membership inference on DP models

In this question, you will conduct membership inference attack on private models trained with the DP-SGD algorithm. You will observe how the differential privacy bound limits the success of membership inference.

- Randomly sample a subset D_{tr} of size 10000 from the Adult dataset, and randomly select one record z from the remaining Adult dataset (excluding D_{tr}).
- Train 10 private model on D_{tr} and $D_{tr} \setminus \{(x, y)\}$ each (in total 20 models) using Algorithm 1 with $\sigma = 0.05, B = 1, \gamma = 0.25, E = 20, \theta_0 = 0, C = 1$.
- Complete the `compute_in_out_loss_for_target` function in the notebook to compute (non-member) loss value for model trained on D_{tr} on the record z; as well as (member) loss value of model trained on $D_{tr} \cup \{z\}$ on the record z.
- Compute a histogram of 10 member loss values and 10 non-member loss values, and store the membership value in the notebook.
- Use the `roc_curve` function in the notebook to compute the TPR and FPR scatter plot for membership inference in the notebook.

- In the lecture, we talked about the promise of differential privacy in bounding the membership leakage. We repeat the theorem below.

  Theorem 1 *Let $D, D \cup z$ be an arbitrary pair of neighboring datasets. If the algorithm $\mathcal{T}$ is (ε, δ)-differentially private, then the TPR and FPR of any attack algorithm $\mathcal{A}$, over random trials of the membership inference game, satisfy the following equation.*

  $$FPR + e^{\varepsilon} \cdot (1 - TPR) \geq 1 - \delta \tag{3}$$

  $$e^{\varepsilon} \cdot FPR + (1 - TPR) \geq 1 - \delta \tag{4}$$

  Prove Theorem 1 in the report.

- Plug the ϵ value you computed in Task 2.2.1 under $\delta = 10^{-5}$ into the above theorem, and plot the upper bound for TPR given arbitrary FPR$\in [0, 1]$ in the notebook.

- Answer this question in the **report**. Do the TPR FPR values for distinguishing two histograms satisfy the inequalities in Theorem 1? If not, give a possible explanation of why.

  (Hint: the TPR and FPR in Theorem 1 refer to their average over a lot of trials.)

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.