Comparing Differentially Private Gradient Descent Algorithms

Joshua Childs
jhchilds@uvm.edu
University of Vermont

Samuel Clark
University of Vermont
samuel.clark@uvm.edu

Abstract

In recent years, large machine learning models have become ubiquitous in the tech stacks of many large companies. These models take in users' data in order predict certain outcomes for individual users as well as groups of users. With this data usage comes the concern of privacy. Differential privacy specifically has become a quickly growing area of research that has the potential to address privacy concerns of machine learning models. This paper builds three custom differentially private gradient descent optimizers and compares the effect they each have on the accuracy of the training of original model. The analysis shows significant differences in the effects

Keywords: differential privacy, neural networks, Renyi, Zero Concentration, epsilon, delta, accuracy, epochs, machine learning

1 Introduction

This paper describes a systematic comparison between three differentially private optimizers using unique strategies. The strategies include pure Epsilon differential privacy, Epsilon-Delta differential privacy, Renyi differential privacy, and Zero Concentrated differential privacy. All of these strategies were implemented as optimizers in the Tensorflow library. In order to show the effects of the differential privacy noise on the accuracy of a machine learning model, we utilized the well known MNIST dataset.

2 Methods

The first step in creating an environment suitable to carry out these tests was to download the MNIST dataset. After downloading this dataset, a non differentially private convolutional neural network was built. This model was meant to be a baseline to confirm and compare the effects of the varying degrees of noise added to the dataset. The next step was to create custom differentially private optimizers using the python API for Tensorflow.

2.1 Custom Differentially Private Optimizers

In order to train the convolutional neural networks with differential privacy, we chose to implement custom optimizers that add noise with the respective differentially private mechanisms during every epoch. The optimizers inherit Tensorflow's Optimizer class, but extend them to include the necessary parameters in each unique optimizer.

In order to abstract the implementation, we created the parent class DPOptimizer that included the necessary attributes for use with the Tensorflow library. We also added useful parameters like epochs and sensitivity that was used by all of the differentially private optimizers.

The epochs parameter was utilized in order to take advantage of sequential composition in our optimizers. This was achieved by dividing the incoming privacy arguments (which differ depending on the strategy) and using the divided version to add noise during any one epoch. Therefore, when these operations were aggregated, the final privacy cost was that of the parameter handed in to the optimizer upon fitting the model.

2.2 Accuracy Experiments

In order to compare the effects of the unique differentially private training strategies on the accuracy of the model, we implemented a training loop for each of the differentially private optimizers. Each optimizer was trained and tested on 100 varying values of their privacy parameters. 100 values of Rho were sampled from 10^{-7} to 10^{-4}, which were used in the Zero Concentrated differentially private optimizer. These values were then converted to the equivalent privacy parameters for the Epsilon, Epsilon-Delta, and Renyi differentially private optimizers. The privacy parameters and testing accuracies were both stored to compare to the original, non differentially private model. The non differentially private model was also trained with a custom optimizer, similar to the differentially private optimizers, to ensure the optimizers were fairly compared. Each optimizer was trained with 10 epochs for each of the 100 privacy parameters tested.

3 Results

The results from our experiment are as follows, and are demonstrated in Figure 1.

Of all the optimizers, the Renyi differntially private optimizer performed the best for a given privacy cost. A low enough privacy cost was not used to determine when the noise introduced by the Renyi algorithm affected the accuracy. At all costs tested, the Renyi optimizer performed just as well as the non differentially private algorithm.

Figure 1. Noise effects on Accuracy

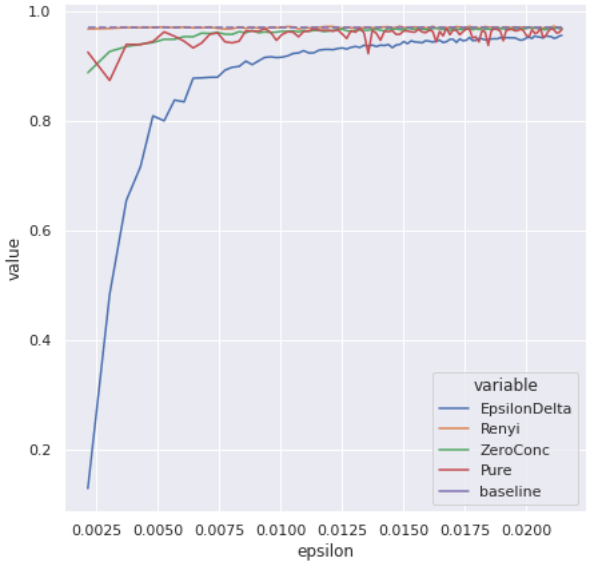

Table 1. Privacy Parameters

Parameters	Differential Privacy Strategy
ϵ	Pure
(ϵ, δ)	Epsilon Delta
$(\alpha, \bar{\epsilon})$	Rényi
ρ	Zero Concentrated

The pure and Zero Concentrated differentially private optimizers performed similarly, and reached an accuracy very close to the baseline at around $\epsilon = 0.01$. The Epsilon-Delta optimizer performed the worst, getting close but not reaching the baseline at around $\epsilon = 0.02$. The Epsilon-Delta optimizer also took longer than the other optimizers to reach a practical accuracy.

Although the optimizers use the variant specific parameters as shown in Table 1, these values were all converted in a privacy budget of ϵ for the purposes of comparison and visualization.

4 Conclusion

This project attempted to highlight the differences that varying differential privacy strategies have on the the training of a convolutional neural network. Using our custom optimizers, we showed how modifying privacy parameters can have a huge effect on the accuracy of the model.