Differentially Private Stochstic Gradient Descent with Curricula

Haozhe An, Kaiyan Shi
{haozhe, kshi12}@umd.edu
Fall 2021

ABSTRACT

The increasingly popular stochastic gradient descent with differential privacy tends to protect data privacy at the expense of model accuracy. In this project, we investigate the possibility of incorporating curriculum learning into stochastic gradient descent with differential privacy, so as to strike a better balance between data privacy and model performance. We experiment different schedules for noise injection, which we call noise curriculum; we also carry out experiments that train a deep learning model with re-arranged sample ordering, which is sample curriculum. Our current results indicate the great potential of using noise curriculum for improvements in accuracy but give little evidence that model performance could benefit from sample curriculum.

KEYWORDS

Differential privacy; Gradient Descent; Curriculum learning

1 INTRODUCTION

In recent years, we have witnessed the rapid development of machine learning and along with that, people increasingly focus on data privacy due to widespread applications of machine learning models. More work now aim at designing machine learning models under certain privacy guarantees. The mainstream optimization technique in machine learning models is stochastic gradient descent (SGD) and its variants[5, 24, 42]. SGD allows fast, stable, and almost globally optimal convergence in many machine learning models. Despite its effectiveness, SGD is prone to data leakage as the gradient calculation can reveal a considerable amount of knowledge of the training dataset, putting data privacy at risk. For example, previous research demonstrates the possibility of recovering facial images through a model-inversion attack [13].

Data privacy could be better protected with the introduction of differential privacy (DP) [1, 10, 11]. By injecting noises to training sample data or to the gradients computed in SGD, the training algorithm could achieve ϵ-differentially private guarantees, where ϵ is known as the privacy budget. When two neighboring datasets that are only different by a single data point are used for training machine learning models, an ϵ-differentially private algorithm guarantees that the log-likelihood ratio of the model outputs are at most ϵ. Intuitively, an adversary cannot make meaningful inferences about if an individual data point takes part in the training or not when ϵ is small. DP has demonstrated its effectiveness through a broad range of deployments.

However, one common limitation of DP is that the preservation of data privacy is achieved at the expense of deteriorated model performance as a result of noise injection. To make up for the accuracy loss, we propose to leverage curriculum learning (CL) [4, 12] in order to mitigate the adverse effects brought by the noise. CL results in training strategies that organize training samples in a meaningful order for enhanced model performance. We hypothesize that training with curricula would better preserve accuracy given a fixed privacy budget.

In this work, we propose a general differentially private SGD training scheme which utilizes curriculum learning. We explore two types of curricula, namely *noise curriculum* and *sample curriculum*, which will be followed during training. We evaluate the effectiveness of using these two curricula in differentially private SGD trainings.

Noise curriculum. We conduct experiments if varying the noise schedule in every epoch will affect model performance. The noise schedule adjusts the magnitude of noise multiplier so that the noise added to training samples follow a pre-defined mathematical function. We make attempts to use a wide range of functions to define noise schedules, including but not limited to constant schedules, linearly decaying or increasing schedules, quadratically decaying or increasing schedules and so on. When carrying out these experiments, we keep the default sample ordering in the original implementation of SGD training.

Sample curriculum. We design a strategy that learns sample "difficulty levels" using a logistic regression model. We re-arrange the sample ordering by the ascending samples' losses produced by the trained logistic regression model. We hypothesize that this "easy-to-hard" arrangement of sample ordering would preserve differential privacy while causing smaller accuracy loss in comparison to a model where no privacy is guaranteed.

We have obtained the following major observations so far.

- The noise curriculum gives rise to marginally better model performance on the validation set given a fixed amount of privacy budget.
- Experiments of noise curriculum reveal a potential logarithmic correlation between the privacy budget and the model performance (in terms of validation accuracy).
- Current experimental results do not show any evidence that sample curriculum is effective in improving model prediction accuracy at a fixed privacy budget. We offer a few possible reasons that explain this phenomenon.

2 RELATED WORK

2.1 Stochastic Gradient Descent with Differential Privacy

There are many works on differentially private machine learning. Two main methods would be output perturbation [17, 19, 39] to add noise to the output and target perturbation [18] to add noise to the objective function. This area gains increasing popularity in the past decade, beginning from differentially private stochastic gradient descent(DP-SGD) [1, 3, 28] to privatize SGD. Then, many gradient-based models have been studied under differential privacy, to name a few, [2, 31, 33, 34, 38, 40] and even Renyi differential privacy [7]. Most of the research in this area only focuses on constant noise addition, and we discuss such work with adaptive noise addition in Sec. 2.3.

2.2 Curriculum Learning

Humans tend to learn from simple examples first and then gradually deepen their understandings to adapt to more complex examples. Curriculum learning (CL) [4, 12] in neural network's training is inspired by this phenomenon, where training samples are arranged in ascending orders by their difficulty levels and then fed into the model. CL has gained increasing attention in machine learning and computer vision. It is widely used in solving various real-world problems [6, 14–16, 22, 30, 37]. A shared challenge in CL is how to quantify the level of difficulty for each sample. In early attempts [4, 29], the curriculum is handcrafted and fixed during training. Self-paced Learning framework [20] is proposed later to optimize the curriculum jointly with the model parameters.

However, CL has not been widely used to improve the performance of SGD with DP. In our project, we plan to use CL to mitigate the effects of privacy-preserving noises that cause accuracy drops, while still maintaining differential privacy.

2.3 Curriculum Learning in Differentially Private Stochastic Gradient Descent

Increasing attention in recent years is paid to developing differentially private SGD with adaptive noise addition [8, 9, 27, 36]. All these work consider adding smaller noise during the training procedure, but according to different noise schedules. Their noise schedules either comes from naive arithmetic sequence [36], or through sampling techniques [9] or from theoretical analysis on the optimal schedule [8, 27].

Compared to those previous work, we propose several different noise schedules for injection, covering basic function types. Also, all these work measure their differential privacy budget according to the composition theorem of standard differential privacy, which means that they just linearly add privacy budgets spent in each iteration. However, none of them analyze the privacy budget through Renyi differential privacy, which allows a tighter analysis of composite theorem [25]. We actually utilizes this new notion and then transforms them into the standard differential privacy.

As described above, there is also no prior work on combining differentially private SGD with only sample curriculum learning, i.e. ordering samples according to their difficulty levels. Therefore, Alg. 2 is the first differentially private SGD combining the notion of both adaptive noise injection and normal curriculum learning (sample curriculum).

3 PRELIMINARIES

3.1 Notations

Let $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ be a training set containing n labelled samples and C classes. Each sample is denoted by (x_i, y_i) where $x_i \in \mathbb{R}^d$ is the features with dimensionality d, and $y_i \in \{1, 2, \dots, C\}$ is the label.

A machine learning model is a function $\mathbf{f}(x;\theta) : \mathcal{X} \mapsto \mathcal{Y}$ parameterized by θ. A loss function between the model prediction and true labels is used for backpropagation that gives the gradient for optimization direction. The parameters are typically updated iteratively by gradient descent. Usually a model will go through the dataset for several passes for gradient updates. We refer each pass as one epoch. The total number of epochs to train a model T is a hyper-parameter. The step size in each gradient update is known as the learning rate. We denote the learning rate as η. Since the size of a dataset could be very large, and it is more feasible to break down the whole dataset into smaller batches and update the gradients batch by batch. We use $B \subset \mathcal{D}$ to denote one arbitrary batch.

3.2 Privacy Accounting

For differentially private SGD, one important step is to compute the privacy budget (ϵ, δ) of the whole training procedure. The privacy accounting method can be summarized in the following three steps.

Step 1. Calculate Renyi divergence for each epoch in the training procedure based [25, 26].

Step 2. Add all those divergence accumulated from each epoch and calculate the corresponding overall Renyi privacy budget (α, ϵ').

Step 3. Transform the (α, ϵ') Renyi differential privacy budget to (ϵ, δ) differential privacy budget.

Detailed explanations for each step are described below. We write MNIST dataset as S and its neighbouring dataset as S'.

Renyi divergence for each epoch. Consider the output distribution of the i^{th} epoch on S and S', denoted as $\mathcal{M}_i(S)$ and $\mathcal{M}_i(S')$. The Renyi divergence of order $\alpha > 1$ is calculated by definition as

$$D_\alpha(\mathcal{M}_i(S)\|\mathcal{M}_i(S')) = \frac{1}{\alpha - 1} \log \mathrm{E}_{x\sim\mathcal{M}_i(S')} \left(\frac{\mathcal{M}_i(S)(x)}{\mathcal{M}_i(S')(x)}\right)^\alpha .$$

Note that the noise addition step in Alg. 1 is actually a sampled Gaussian mechanism, we then can compute the divergence for each epoch based on [26]. It is proved in Sec. 3.1 of [26] that

$$\mathrm{E}_{x\sim\mathcal{M}_i(S')} \left(\frac{\mathcal{M}_i(S)(x)}{\mathcal{M}_i(S')(x)}\right)^\alpha \leq \mathrm{E}_{x\sim\mathcal{M}_i(S)} \left(\frac{\mathcal{M}_i(S)(x)}{\mathcal{M}_i(S')(x)}\right)^\alpha \quad \text{for all } \alpha \geq 1.$$

As in Step 3., we need to calculate the Renyi privacy budget, which is defined as the upper bound of the divergence, then the actual divergence to calculate would be

$$D_\alpha(\mathcal{M}_i(S)\|\mathcal{M}_i(S')) = \frac{1}{\alpha - 1} \log \mathrm{E}_{x\sim\mathcal{M}_i(S)} \left(\frac{\mathcal{M}_i(S)(x)}{\mathcal{M}_i(S')(x)}\right)^\alpha .$$

Such expectation value can be numerically stably calculated based on the formula provided in Sec. 3.3 in [26], which would then give the divergence of each epoch.

Linear accumulation of Renyi divergence and (α, ϵ') Renyi differential privacy. With output distributions in epoch $\mathcal{M}_1(S), \dots, \mathcal{M}_1(S)$, and $\mathcal{M}_1(S'), \dots, \mathcal{M}_1(S')$, we would have the output distributions after T epochs as $\mathcal{M}(S)^T = \mathcal{M}_1(S) \times \dots \times \mathcal{M}_T(S)$ and $\mathcal{M}(S')^T = \mathcal{M}_1(S') \times \dots \times \mathcal{M}_T(S')$, through composition of probability distributions.

Then by the additivity of Renyi divergence [32], we would have

$$D_\alpha\left(\mathcal{M}(S)^T \| \mathcal{M}(S')^T\right) = \sum_{i=1}^{T} D_\alpha\left(\mathcal{M}_i(S)\|\mathcal{M}_i(S')\right),$$

for $\alpha \in [0, \infty]$ and $T < \infty$, which demonstrates the linear accumulation of Renyi divergence.

Then through the definition of Renyi differential privacy

$$\mathrm{D}_\alpha \left(\mathcal{M}_i(S) \| \mathcal{M}_i\left(S'\right)\right) \leq \epsilon_i',$$

we compute the overall Renyi differential privacy ϵ:

$$\begin{aligned}\epsilon' &:= \left\lceil D_\alpha \left(\mathcal{M}(S)^T \| \mathcal{M}(S')^T\right)\right\rceil \\ &\leq \sum_{i=1}^{T} \left\lceil D_\alpha \left(\mathcal{M}_i(S) \| \mathcal{M}_i(S')\right)\right\rceil \\ &= \sum_{i=1}^{i=T} \epsilon_i'\end{aligned}$$

(ϵ, δ) differential privacy. With fixed δ, the privacy budget in differential privacy can be directly computed from ϵ' obtained in previous step, according to Thm. 3.1.

THEOREM 3.1. *(From RDP to (ϵ, δ)-DP [25]) If f is an (α, ϵ')-RDP mechanism, it also satisfies $\left(\epsilon' + \frac{\log 1/\delta}{\alpha-1}, \delta\right)$-differential privacy for any $0 < \delta < 1$.*

3.3 Experimental Setup

MNIST dataset. MNIST [21] dataset contains handwritten digits from 0 to 9. It consists of a training set of 60,000 examples, and a validation set of 10,000 examples. Each image is a grey-level image whose size is 28×28. MNIST has been a popular benchmark dataset in research about differential privacy.

Model architecture. We train a simple convolutional neural network (CNN) model on MNIST. The model composes two convolution layers, with max pooling after each convolutional layer and relu activation function. A fully connected layer is used as the last layer to produce logits for predictions.

Hyperparameters. We train the CNN model for 20 epochs for quick evaluation, although further training could bring marginal improvements in both training and validation accuracy. The batch size is 600. We use constant learning rate $\eta = 0.15$. We implement the training procedure using Keras.

4 NOISE CURRICULUM

4.1 Algorithm

Under the noise curriculum, we investigate how the schedule of noise addition affects the performance of differentially private SGD. Instead of using a constant noise multiplier to scale the standard deviation of the noise addition in [1], we use different noise multiplier in each epoch determined by the noise schedule function $F_{noise}(t)$. The corresponding algorithm is formally presented in Alg. 1.

Note that in this design, sample ordering for each epoch is random according to default SGD setting. Also for differentially private SGD alone, we can just set $F_{noise}(t) =$ constant to reproduce DP-SGD in [1].

For more comprehensive investigation, we consider both decreasing and increasing trends. For both trends, four functions with typical shapes are considered: piecewise, linear and exponential (logarithmic) and quadratic. The eight noise schedules are show in Figure 1 and Figure 2. The corresponding formula are given in Table 1 .

Algorithm 1 Training with noise curriculum

```
Input: Training data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ in default sample ordering, number of epochs $T$, noise schedule function $F_{noise}(t)$ given epoch $t$, learning rate $\eta$, deep learning model loss function $\mathcal{L}(\theta, B) = \frac{1}{|B|} \sum_{x_i \in B} \mathcal{L}(\theta, x_i)$, gradient norm bound $C$
Output: Trained model $\theta_T$, the overall privacy cost $(\epsilon, \delta)$ using a privacy accounting method
  $\theta_0 \leftarrow$ random initialization                          ▷ Beginning of step 2
  for $t \in [1, T]$ do
    for $B_i \subset \mathcal{D}'$ do          ▷ $B_i$ is one batch of samples, $\cup_i B_i = \mathcal{D}'$
      $\bar{\mathrm{g}}_t(B_i) \leftarrow \frac{\mathrm{g}_t(B_i)}{\max(1, \|\mathrm{g}_t(B_i)\|_2/C)}$          ▷ Gradient clip
      $\bar{\mathrm{g}}_t(B_i) \leftarrow \bar{\mathrm{g}}_t(B_i) + \mathcal{N}(0, F_{noise}(t)^2 C^2 \mathbf{I})$          ▷ Add noise
      $\theta_{t+1} \leftarrow \theta_t - \eta \bar{\mathrm{g}}_t(B_i)$          ▷ Gradient descent
    end for
  end for
```

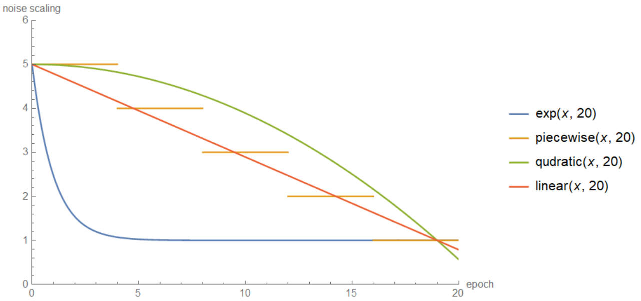

Figure 1: Decreasing noise schedules, with $T = 20$, .

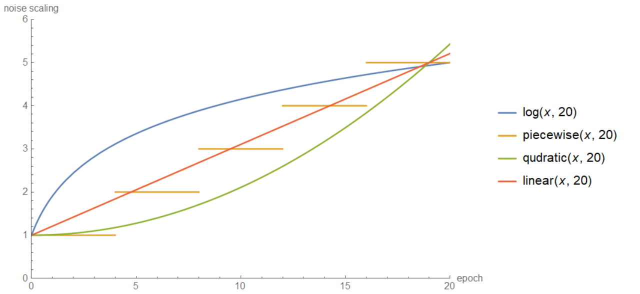

Figure 2: Increasing noise schedules, with $T = 20$.

4.2 Experimental Results

Experiments under the proposed noise curriculum is conducted with 20 epochs, i.e. $T = 20$, and noise schedule presented in Fig. 1 and 2. Comparison between increasing and decreasing trends of noise schedule is shown in Tab. 2. Note that in order for better and clearer comparison, we add one more noise schedules, which is "increasing exponential" with $\left(1 - \frac{e^{T-1}-5}{e^{T-1}-1}\right) e^t + \frac{e^{T-1}-5}{e^{T-1}-1}$. Although its

Trend	Noise schedule	$F_{noise}(t)$
Decreasing	Exponential	$\left(5 - \frac{1-5e^{-(T-1)}}{1-e^{-(T-1)}}\right) e^{-t} + \frac{1-5e^{-(T-1)}}{1-e^{-(T-1)}}$
	Piecewise	$-\left\lfloor \frac{t}{T/5} \right\rfloor + 5$
	Linear	$-\frac{4}{T-1}t + 5$
	Quadratic	$-\frac{4}{(T-1)^2}t^2 + 5$
Increasing	Logarithmic	$\frac{4}{\log(T+1)}\log(t+1) + 1$
	Piecewise	$\left\lfloor \frac{t}{T/5} \right\rfloor + 1$
	Linear	$\frac{4}{T-1}t + 1$
	Quadratic	$\frac{4}{(T-1)^2}t^2 + 1$

Table 1: Example noise schedules for noise multiplier between 1 and 5. "Decreasing" refers to a noise schedule whose noise multiplier begins with 5 and gradually decreases to 1, whereas "increasing" refers to a schedule increasing from 1 to 5.

Noise schedule	Trend	ϵ	Validation accuracy
Exponential	decreasing	2.61	**0.9572**
	increasing	2.61	0.9442
Piecewise	decreasing	1.64	**0.9517**
	increasing	1.64	0.9252
Linear	decreasing	1.40	**0.9463**
	increasing	1.40	0.9236
Quadratic	decreasing	1.32	**0.9425**
	increasing	1.69	0.9313

Table 2: MNIST validation set accuracy comparison between decreasingly and increasingly scheduled noise multipliers under similar set of noise multipliers. These results show a preliminary conclusion that decreasing noise curriculum performs better than increasing noise curriculum.

shape is similar to "increasing quadratic", its privacy budget is the same as "decreasing exponential" and therefore is worth reporting. The resulting validation accuracy indicates that under similar set of noise multipliers and similar privacy budget, it would be better to add more noise first and gradually decrease. Possible reasons are provided in Sec. 6.1.

Since integer constant noise multipliers $1 - 5$ lead to different overall privacy budget ($\epsilon, \delta = 10^{-5}$) from decreasingly scheduled noise multipliers according to the privacy accounting method described in Sec. 3.2. We adjust the constant noise multipliers to make ϵ between theirs and scheduled's equal up to four decimal places. Results under nearly equal privacy budgets' cases are reported in Fig. 3. We can see that validation accuracy under noise curriculum outperforms that under constant noise (multipliers). Detailed analysis will be presented in Sec 6.1.

ϵ	Noise type	Validation accuracy
2.61	decreasing exponential	**0.9572**
	constant 1.05	0.9529
1.64	decreasing piecewise	0.9517
	constant 1.38	**0.9565**
1.40	decreasing linear	**0.9463**
	constant 1.54	0.9434
1.32	decreasing quadratic	**0.9425**
	constant 1.32	0.9419

Table 3: MNIST validation set accuracy comparison between constant noise multipliers and decreasingly scheduled noise multipliers under equal privacy budgets. These results show a preliminary conclusion that noise curriculum usually leads to accuracy gains.

5 SAMPLE CURRICULUM

Besides the noise curriculum, we also propose to investigate the effectiveness of sample curriculum in differentially private SGD training. In sample curriculum, we re-arrange training sample ordering so that the model is fed with easier data near the beginning epoch and receives increasingly difficult samples afterwards. We hypothesize this easy-to-hard curriculum could help the model better adapt to noisy training samples.

5.1 Algorithm

We illustrate the idea of sample curriculum in Fig. 3. The algorithm composes two two steps.

Step 1 **Obtain sample ordering:** We first use the training data to train a multi-class logistic regression model L. The logistic regression model is a less powerful model compared to the CNN model we will use in the next stage and therefore, it is a potentially good candidate to assess the difficulty levels of each training sample. This step is illustrated in Fig. 3 step 1.

Step 2 **Train CNN with the new sample ordering:** Shown in Fig. 3 step 2, the same training data is then passed to the trained logistic regression model to obtain the individual losses for each training sample. We sort the samples by their log loss in the logistic regression model. The binary log loss is defined as

$$Cost(L(x), y) = \begin{cases} -\log(L(x)) & \text{if } y = 1 \\ -\log(1 - L(x)) & \text{if } y = 0 \end{cases}$$

and for a multi-class logistic regression model, this loss is generalized to

$$Cost(L(x), y) = -\sum_{i=1}^{C} t_i \log(p_i) \tag{1}$$

where $C = 10$ is the number of classes in MNIST (digits 0 to 9), $t_i \in \{0, 1\}$ is the ground truth label indicating the

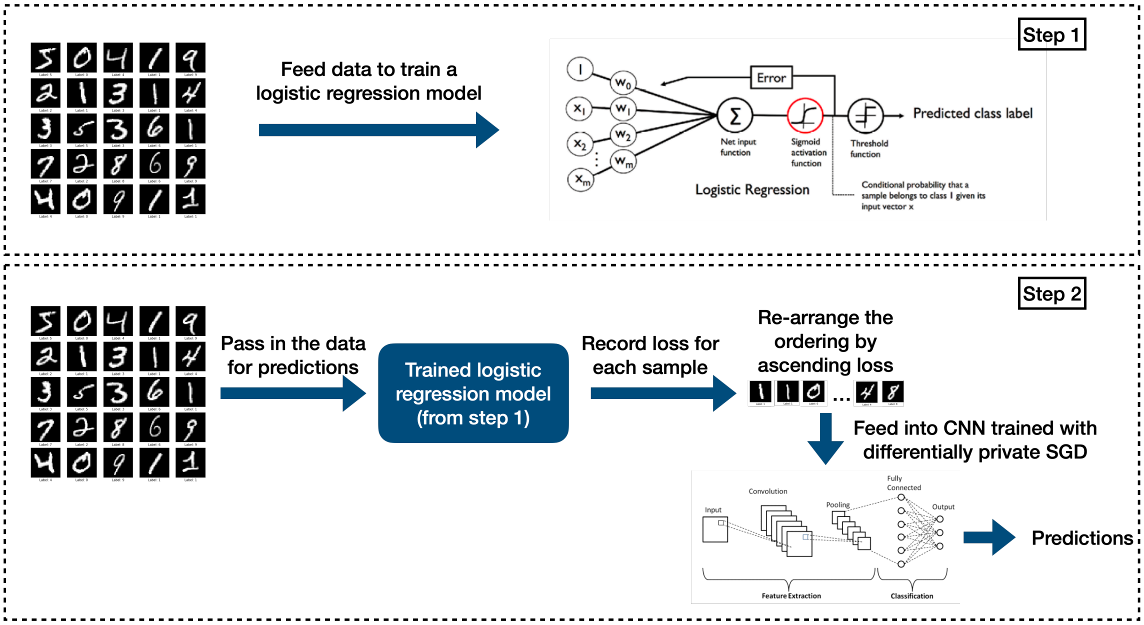
Step 1
Feed data to train a logistic regression model
1
x1
x2
xm
w0
w1
w2
wm
Error
Net input function
Sigmoid activation function
Threshold function
Predicted class label
Logistic Regression
Conditional probability that a sample belongs to class 1 given its input vector x
Step 2
Pass in the data for predictions
Trained logistic regression model (from step 1)
Record loss for each sample
Re-arrange the ordering by ascending loss
Feed into CNN trained with differentially private SGD
Input
Convolution
Pooling
Fully Connected
Output
Feature Extraction
Classification
Predictions

Figure 3: An illustration of the algorithm for sample curriculum in differentially private SGD training.

class of a sample ($t_i = 1$ if $y = i$ otherwise $t_i = 0$), and p_i is the softmax probability for class i.

The intuition behind this is that a sample with a greater magnitude of loss indicates it is closer to the decision boundary and thus harder for the model to make a prediction; in contrast, a sample with a smaller loss indicates the model could easily fit the data point. We then re-arrange the ordering of training samples by ascending loss values. We feed this new sequence of training data into the CNN model for training. Finally, we test the model performance on the validation set when the training completes.

We formally present the algorithm in Alg. 2. Note that in this design, the sample ordering is fixed after we sort the training samples by their log loss produced by the logistic regression model. We train the CNN model with the same sample ordering in each epoch. In addition, the proposed sample curriculum is orthogonal to the noise curriculum. The two curricula could possibly complement each other and bring enhanced model performance when used simultaneously.

5.2 Experimental Results

We evaluate the effectiveness of the sample schedule by conducting an experiment using the proposed sample curriculum only and compare it with the default training with differential privacy. We also run experiments using the sample curriculum combined with a variety of noise schedules. These settings will indicate if combining the two curricula is helpful in improving model validation accuracy.

We report our current experimental results in Table. 4. We see that the proposed sample curriculum does not bring any improvements since the setting without sample curriculum consistently outperforms the model trained with sample curriculum in all experiments. We offer a few explanations that may have caused these results in Sec. 6.3.

Algorithm 2 Training with sample curriculum

```
Input: Training data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ in default sample ordering, a trained logistic regression model $L$, number of epochs $T$, noise schedule function $F_{noise}(t)$ given epoch $t$, learning rate $\eta$, deep learning model loss function $\mathcal{L}(\theta, B) = \frac{1}{|B|} \sum_{x_i \in B} \mathcal{L}(\theta, x_i)$, gradient norm bound $C$
Output: Trained model $\theta_T$, the overall privacy cost $(\epsilon, \delta)$ using a privacy accounting method
  $C \leftarrow \emptyset$                                  ▹ Beginning of step 1
  for $(x_i, y_i) \in \mathcal{D}$ do
      $C \leftarrow C \cup \{((x_i, y_i), Cost\,(L(x_{a_i}, y_i)))\}$
  end for
  Sort $C$ by $Cost\,(L(x_{a_i}, y_i)))$ in ascending order
  $\mathcal{D}' = \{(x_{a_1}, y_{a_1}), (x_{a_2}, y_{a_2}), \dots (x_{a_n}, y_{a_n})\}$
  where $\forall i < j, Cost(L(x_{a_i}), y_i) < Cost(L(x_{a_j}), y_j)$
  $\theta_0 \leftarrow$ random initialization                ▹ Beginning of step 2
  for $t \in [1, T]$ do
      for $B_i \subset \mathcal{D}'$ do        ▹ $B_i$ is one batch of samples, $\cup_i B_i = \mathcal{D}'$
          $\mathbf{g_t}(B_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, B_i)$        ▹ Compute gradient
          $\bar{\mathbf{g_t}}(B_i) \leftarrow \frac{\mathbf{g_t}(B_i)}{\max(1, \|\mathbf{g_t}(B_i)\|_2 / C)}$        ▹ Gradient clip
          $\bar{\mathbf{g_t}}(B_i) \leftarrow \bar{\mathbf{g_t}}(B_i) + \mathcal{N}(0, F_{noise}(t)^2 C^2 \mathbf{I})$        ▹ Add noise
          $\theta_{t+1} \leftarrow \theta_t - \eta \bar{\mathbf{g_t}}(B_i)$        ▹ Gradient descent
      end for
  end for
```

Noise schedule	Noise multiplier	W/O SC_{lr}	SC_{lr}
Constant	0	0.9577	0.9470
	1	0.9557	0.9469
	2	0.9514	0.9332
	3	0.9389	0.9260
	4	0.9158	0.9075
	5	0.9091	0.9003
Linear	5-1	0.9463	0.9437
Piecewise	5-1	0.9441	0.9356
Quadratic	5-1	0.9423	0.9382

Table 4: MNIST validation set accuracy when combining various noise schedules with the proposed sample curriculum using logistic regression as the difficulty level metric (SC_{lr}). These results do not show that the tested sample curriculum is helpful in improving validation accuracy given the same privacy budget.

6 DISCUSSION

6.1 Effectiveness of Noise Curriculum

First it is indicated in Tab. 2 that validation accuracy is higher under decreasing noise curriculum with the same privacy budget. We could then conclude that decreasing noise curriculum outperforms increasing noise curriculum. Possible reason behind such phenomenon could be that noise disturbance is more likely to affect the performance in the end than at the early stage. With increasing epochs, the trained model is closer to the optimal solution. So the trained model would be more "delicate" in the end, and small noise addition would not disturb it as much as large noise would, leading to better performance.

Then through the comparison in Tab. 3 on constant noise and (decreasing) noise curriculum, we observe that under fixed privacy budget, adding gradually decreased noise would lead to better validation accuracy for "linear", "quadratic" and "exponential". However, we did not repeat our experiments for multiple times and the number of epochs T used are not enough to get convergence for validation accuracy $> 98\%$. So we can only have a preliminary conclusion that the noise curriculum can improve the model performance, under fixed privacy budget.

6.2 The Potential Correlation between ϵ and Validation Accuracy

Throughout experiment, we also observe a logarithmic relationship between privacy budget ϵ and validation accuracy, at least for relatively large noise (noise multiplier is at least 1). The correlation has been shown in Fig. 4. Note that the correlation is drawn in log plot for better demonstration. It is shown that there exists a relatively linear relationship between validation accuracy and $\log \epsilon$, especially for large noise addition, i.e. noise multiplier > 1.0. To formally assess the linear relationship, we use pearson correlation coefficient and its value is 0.848 with p-value 0.016. As the p-value is lower than the conventional 5%, the coefficient can be considered as statistically significant. Therefore we can conclude that the observed logarithmic relationship is valid.

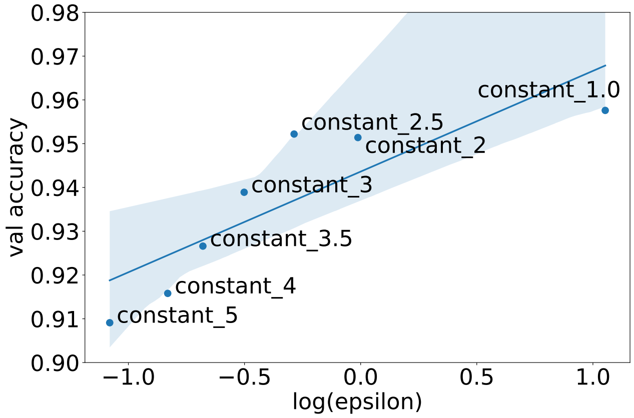

Figure 4: A log plot of MNIST validation set accuracy versus privacy budget ϵ obtained through the privacy accounting method described in Sec. 3.2. These results can lead to a rough result that for relatively large noise multiplier, i.e. > 1.0, the validation accuracy is proportional to $\log \epsilon$. The plot indicates the best fitting line with confidence interval 0.95.

We ascribe such relationship to the stability of the optimal solution. With relatively smaller noise, the effects noise can bring are diminishing. This means that the model is more likely to reach the optimum early and then become stable. For example, for constant noise multipliers 2.5 and 3.0, they might both reach some critical accuracy level early. Then because it is normally slower to achieve better accuracy after that level, the accuracy differences between those two cases would not be as large as cases with larger noise. Therefore the accuracy increases more slowly with smaller noise added, which would give a logarithmic relationship.

However, we could also see that with smaller noise multipliers, i.e. noise multipliers approaching 1.0, the linearity between accuracy and $\log \epsilon$ is disappearing. We think this is also caused by the the stability of the optimal solution. With absolutely small noise, the found solution at $T = 20$ might be very close to the optimal one, and then it's even harder for the loss function to decrease more. So their accuracy is expected to be very likely to be very close to each other and difference might be more influenced by the randomness, instead of by real performance differences.

6.3 (In)effectiveness of Sample Curriculum

Our current experimental results do not show any evidence that sample curriculum could improve model performance on the validation set given a fixed privacy budget ϵ. We argue that several possible factors could contribute to this phenomenon. More experiments should be carried out using additional datasets and more dynamic sample curriculum to reach a conclusion.

Reason 1: The proposed sample curriculum using a logistic regression model is not optimal. Our current sample curriculum relies on the log loss from the logistic regression model. Since CNN models extract features from data samples differently from how a logistic regression model works, the log loss might not be able to reflect the accurate "difficulty level" of a sample when making predictions using a CNN model. A potentially better alternative would be arrange the samples in a dynamic order using the CNN model's cross entropy loss before each epoch starts. The cross entropy loss is defined as

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} t_i \log(p_i) \tag{2}$$

where $C = 10$ is the number of classes in MNIST (digits 0 to 9), $t_i \in \{0, 1\}$ is the ground truth label indicating the class of a sample, and p_i is the softmax probability for class i. Using the CNN model's loss as a metric to evaluate the samples' difficulty level would allow us to have a more dynamic sample ordering that vary in each epoch, since the model gets updated throughout training. The dynamic schedule could be more advantageous because the sample ordering is more adaptive to the CNN model parameters.

Reason 2: The fixed ordering in each epoch adversely affects the performance of SGD.. Previous research shows that the noise produced by stochastic sampling in SGD contributes largely to its effectiveness [23, 35, 41]. However, our current design of sample curriculum uses the same sample ordering after step 1. The repeated, fixed sample ordering for every epoch reduces the diversity of stochastic noise in comparison to the original default training procedure. This may have caused the deteriorated performance of the CNN model.

Reason 3: MNIST might be a too easy dataset to benefit from sample curriculum. It is worth noting that the logistic regression model could already perform quite well on the MNIST dataset, with training accuracy 0.9346 and validation accuracy 0.9256. The high accuracy imply that most samples would be classified as "easy" samples by our design of sample curriculum. Thus, the resultant ordering would only affect a small subset of the samples, which is not an impactful change overall.

The above three reasons may have contributed to our negative finding of the sample curriculum. Given that we only had the opportunity and resource to test one sample curriculum on one dataset, it remains inconclusive as to whether sample curriculum could bring improvements in model performance with a fixed privacy budget.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose to use curriculum learning to improve model performance on the validation set while preserving the same level of privacy guarantee. We consider two orthogonal curricula, namely noise curriculum and sample curriculum. The noise curriculum dynamically change the magnitude of noise multiplier in every epoch, resulting in higher validation accuracy than using a constant noise schedule in most cases. The sample curriculum aims to re-arrange sample orders in the training set based on their level of difficulty for a model to fit, so that the model gets to learn easier samples before they encounter the harder ones. We hypothesize that these two curricula could each have their own schedule during training and complement each other to bring improvements in validation accuracy.

We conduct experiments for various noise schedules and one sample curriculum depending on the output of a logistic regression model for sample ordering. On one hand, we find that the noise curriculum is beneficial as using it gives rise to marginally higher validation accuracy. However, due to resource constraints, we need to carry out additional runs of experiments to make sure these results are statistically significant. On the other hand, we do not have any conclusion about the effectiveness of sample curriculum yet. We suspect that there exists better curriculum design than using a logistic regression model as an assessment of sample difficulty level. Additionally, we find that it is necessary to perform experiments using more complex and larger datasets than MNIST because the difficulty levels of sample in MNIST are not evenly distributed.

Besides the observations on model performance, we also see that there might be a logarithmic correlation between the privacy budget ϵ and the validation accuracy. Our observation suggests that there might be theoretical upper bound for validation accuracy given the amount of noise we add in differentially private SGD training. This bound would allow people to approximate a model's performance given the magnitude of noise multiplier without even training and testing the model.

We thus propose the following for future research directions.

- Repeat the experiments on noise curriculum to make sure that the gain in validation accuracy is consistent throughout different settings. Ideally, there should be about 3 to 5 runs of each setting. Average values and standard deviations should be reported for statistical significance.
- Design more dynamic and adaptive sample curriculum based on the CNN model loss instead of the logistic regression model loss.
- Investigate the theoretical bound between the noise multiplier magnitude and the validation accuracy.

REFERENCES

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

[2] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Thakurta. 2019. Private stochastic convex optimization with optimal rates. *arXiv preprint arXiv:1908.09970* (2019).

[3] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 464–473.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[5] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[6] Qi Cai, Yingwei Pan, Yu Wang, Jingen Liu, Ting Yao, and Tao Mei. 2020. Learning a Unified Sample Weighting Network for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.

[7] Chen Chen, Jaewoo Lee, and Dan Kifer. 2019. Renyi differentially private erm for smooth objectives. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2037–2046.

[8] Junhong Cheng, Wenyan Liu, Xiaoling Wang, and Xingjian Lu. 2020. Adaptive Distributed Differential Privacy with SGD. (2020).

[9] Jian Du, Song Li, Moran Feng, and Siheng Chen. 2021. Dynamic Differential-Privacy Preserving SGD. *arXiv preprint arXiv:2111.00173* (2021).

[10] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[12] Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition* 48, 1 (1993), 71–99.

[13] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[14] Alex Graves, Marc G Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated Curriculum Learning for Neural Networks.

In *Proceedings of the International Conference on Machine Learning(ICML)*.

[15] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. 2020. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.

[17] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. 2012. Differentially private online learning. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 24–1.

[18] Hua Jingyu, Xia Chang, and Zhong Sheng. 2015. Differentially Private Matrix Factorization. In *Proc of the 24th Int Conf on Artificial Intelligence. Palo Alto: AAAI Press*. 1763–1770.

[19] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. 2012. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 25–1.

[20] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems(NeurIPS)*.

[21] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* 2 (2010), 18.

[22] Siyang Li, Xiangxin Zhu, Qin Huang, Hao Xu, and C-C Jay Kuo. 2017. Multiple instance curriculum learning for weakly supervised object detection. *arXiv preprint arXiv:1711.09191* (2017).

[23] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. 2018. Don't use large mini-batches, use local SGD. *arXiv preprint arXiv:1808.07217* (2018).

[24] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[25] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 263–275.

[26] Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. R\'enyi Differential Privacy of the Sampled Gaussian Mechanism. *arXiv preprint arXiv:1908.10530* (2019).

[27] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. 2019. AdaCliP: Adaptive clipping for private SGD. *arXiv preprint arXiv:1908.07643* (2019).

[28] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 245–248.

[29] Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2009. Baby Steps: How "Less is More" in unsupervised dependency parsing. In *Advances in Neural Information Processing Systems(NeurIPS)*.

[30] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. 2012. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems(NeurIPS)*.

[31] Florian Tramèr and Dan Boneh. 2020. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660* (2020).

[32] Tim Van Erven and Peter Harremos. 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.

[33] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. 2020. On differentially private stochastic convex optimization with heavy-Tailed data. In *International Conference on Machine Learning*. PMLR, 10081–10091.

[34] Di Wang, Minwei Ye, and Jinhui Xu. 2018. Differentially private empirical risk minimization revisited: Faster and more general. *arXiv preprint arXiv:1802.05251* (2018).

[35] Colin Wei, Sham Kakade, and Tengyu Ma. 2020. The Implicit and Explicit Regularization Effects of Dropout. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. PMLR, 10181–10192. https://proceedings.mlr.press/v119/wei20d.html

[36] Yun Xie, Peng Li, Chao Wu, and Qiuling Wu. 2021. Differential Privacy Stochastic Gradient Descent with Adaptive Privacy Budget Allocation. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. IEEE, 227–231.

[37] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. 2020. Curriculum Manager for Source Selection in Multi-Source Domain Adaptation. In *Proceedings of the European Conference on Computer Vision(ECCV)*.

[38] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. 2021. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *arXiv preprint arXiv:2102.12677* (2021).

[39] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. 2017. Efficient private ERM for smooth objectives. *arXiv preprint arXiv:1703.09947* (2017).

[40] Yingxue Zhou, Zhiwei Steven Wu, and Arindam Banerjee. 2020. Bypassing the ambient dimension: Private sgd with gradient subspace identification. *arXiv preprint arXiv:2007.03813* (2020).

[41] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. 2018. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195* (2018).

[42] Martin Zinkevich, Markus Weimer, Alexander J Smola, and Lihong Li. 2010. Parallelized stochastic gradient descent.. In *NIPS*, Vol. 4. Citeseer, 4.