Machine Learning with Laplacian Smoothing Gradient Descent

Bao Wang [1]
Department of Mathematics, UCLA

[1] Joint work with Lisa Kreusser and professor Stan Osher's Group.

Disadvantages of SGD

- Weaker theoretical guarantee mainly comes from the variance of SGD.
- Cannot avoid saddle points.
- Cannot protect the privacy of the training data, at least not a good trade-off between privacy and utility.

Laplacian Smoothing Gradient Descent

For any differentiable function $\mathcal{L}(\mathbf{w})$, consider

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \gamma(I - \sigma L)^{-1}\nabla\mathcal{L}(\mathbf{w}^k).$$

Laplacian Smoothing Gradient Descent (LS-GD)

High order schemes

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \gamma(I + (-1)^n\sigma L^n)^{-1}\nabla\mathcal{L}(\mathbf{w}^k).$$

Osher, Wang, Yin, Luo, Pham, and Lin, arXiv:1806.06317, 2018

Variance Reduction

Softmax Regression

For a given instance $\mathbf{x}$, the probability belonging to k-th class is modeled by

$$P(y = k|\mathbf{x}, \mathbf{w}) = \frac{\exp\left(\mathbf{w}_k^T \cdot \mathbf{x}\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^T \cdot \mathbf{x}\right)},$$

where $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K)$.

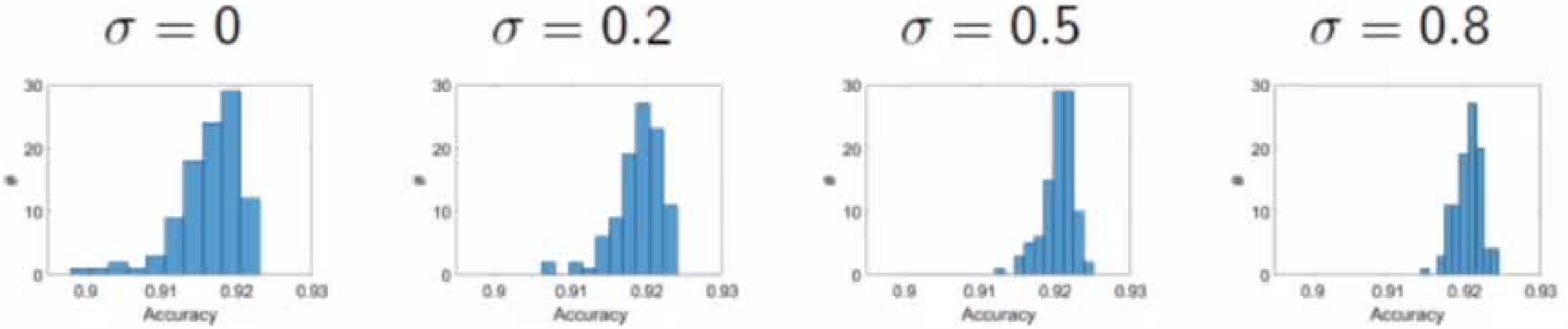
To learn the weights $\mathbf{w}$, we consider the cross-entropy loss function (maybe with regularization terms)

$$\mathcal{L}(\mathbf{w}) = \sum_i - \log(P(y = y_i|\mathbf{x}_i, \mathbf{w})),$$

where the sum is taken over the training data $\{(\mathbf{x}_i, y_i)\}$.

SGD v.s. LS-SGD - Softmax Regression

Consider the MNIST hand written digits recognition by using the Softmax regression. The models are trained by running 100 epochs of SGD and LS-SGD respectively on the 60000 training instances with batch size 100 and learning rate 0.05.



$\sigma = 0$ $\sigma = 0.2$ $\sigma = 0.5$ $\sigma = 0.8$

Figure: The histogram of generalization accuracies of the softmax regression model trained with LS-SGD over 100 independent experiments by using different σ.

Training LeNet with Small Batch Size

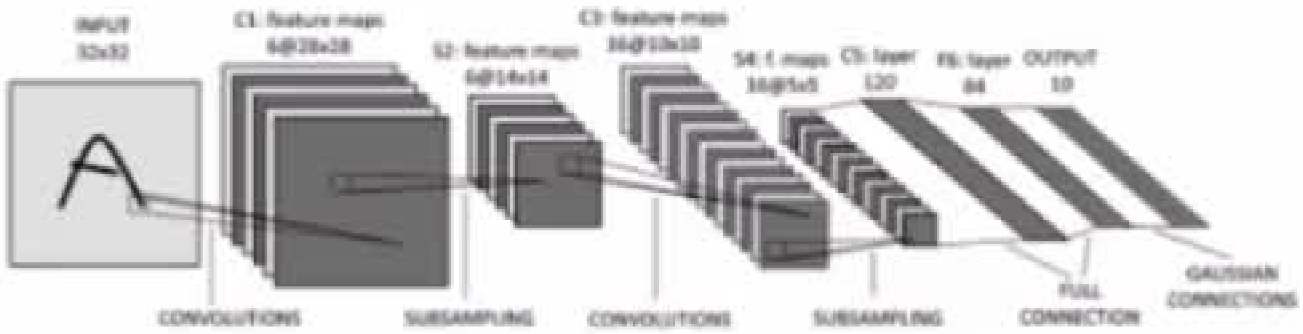


Figure: Architecture of LeNet5.

LeCun et al., Proc. IEEE, 1998

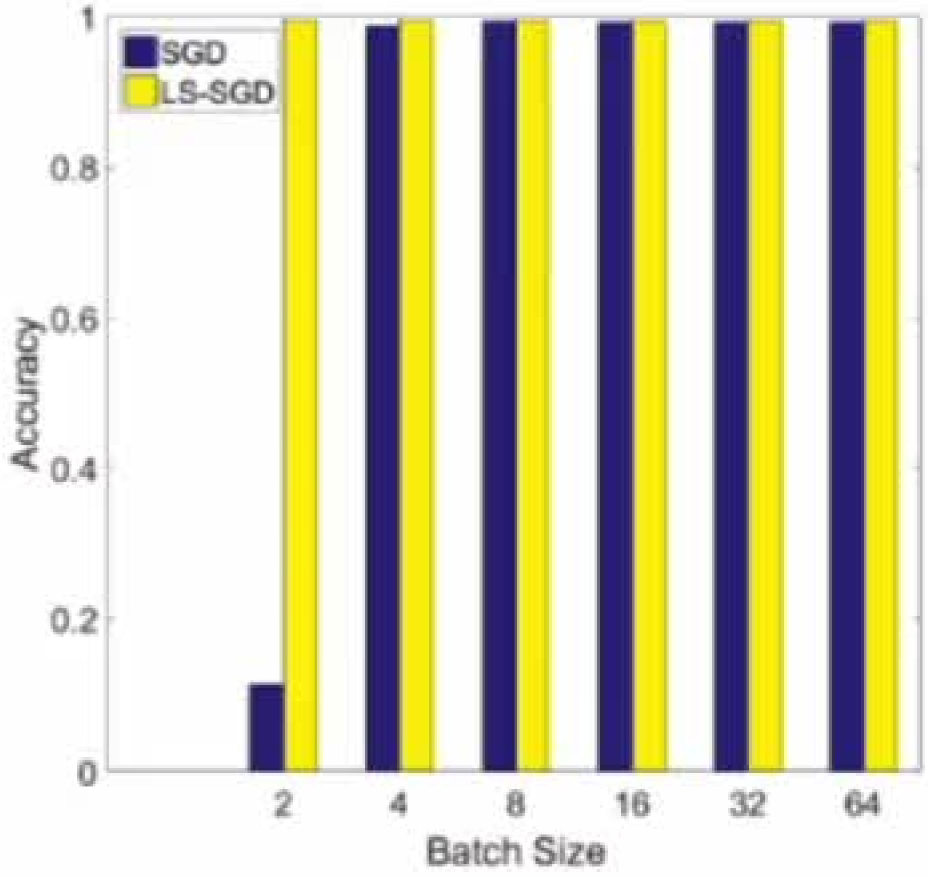


Figure: Generalization accuracy of LeNet5 trained with different batch sizes by SGD and LS-SGD.

Avoid Saddle Point

Avoid Saddle Point

Consider the quadratic function

$$f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x},$$

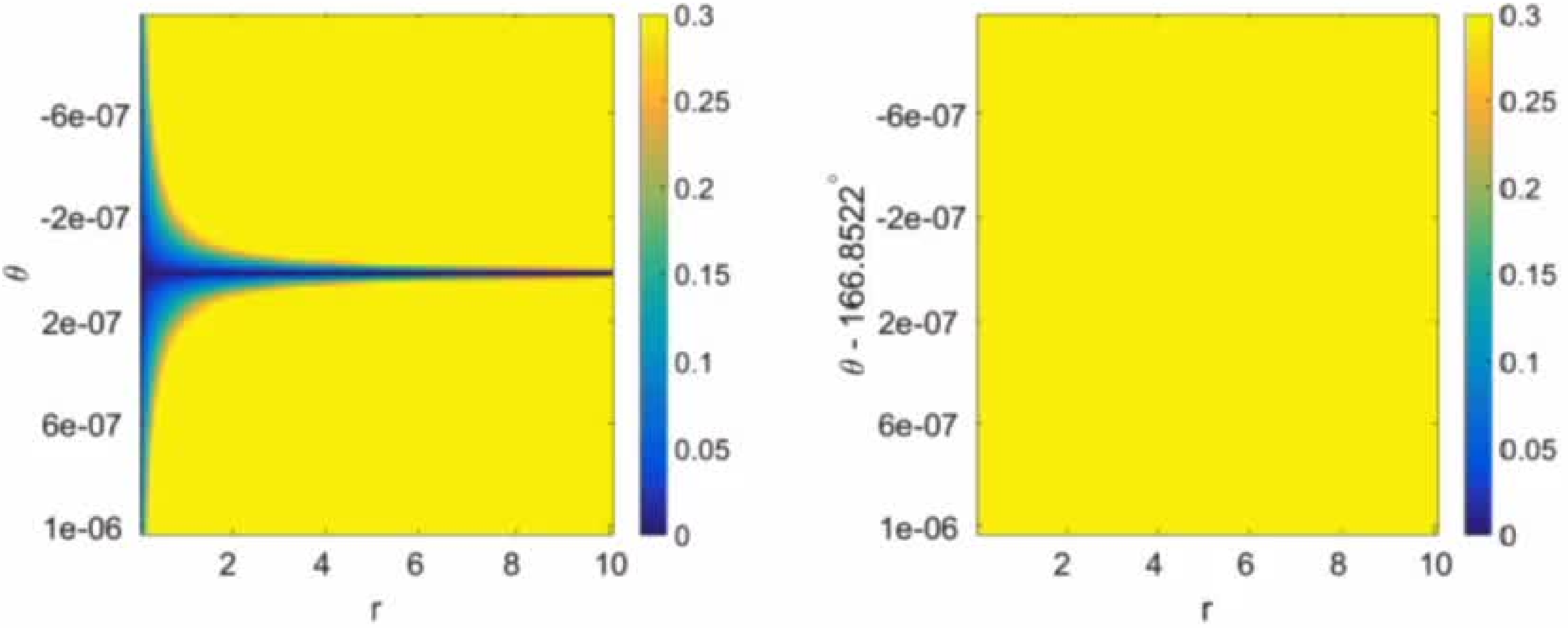
where the matrix $A = diag\{k, k, \cdots, k, -k\}_{n\times n}$ with $k > 0$, f has a saddle point at $\mathbf{0}$, then we have

$$\dim\{\mathbf{x}_0 | \mathbf{x}_{k+1} = \mathbf{x}_k - \eta\nabla f(\mathbf{x}_k), \ \ \mathbf{x}_\infty = \mathbf{0}\} = n - 1$$

$$\dim\{\mathbf{x}_0 | \mathbf{x}_{k+1} = \mathbf{x}_k - \eta\left(I - \frac{k+1}{k+2}L\right)^{-1}\nabla f(\mathbf{x}_k), \ \mathbf{x}_\infty = \mathbf{0}\} = \left\lfloor \frac{n-1}{2} \right\rfloor$$

Kreusser, Osher, and Wang, 2018

Avoid Saddle Point



(a)

(b)

Figure: Plot of distant field to the saddle point $(0, 0)$ after running 100 iterations of GD (a) and LSGD (b) with learning rate 0.1 on the function $f(x, y) = x^2 - y^2$. The coordinate of every pixel denotes the starting point. For both (a) and (b) we show the same sized region starting from where GD and LSGD will be closest the the saddle point.

LSGD never converge to saddle point.

LSGD escape the "stuck region" faster.

Differential Privacy

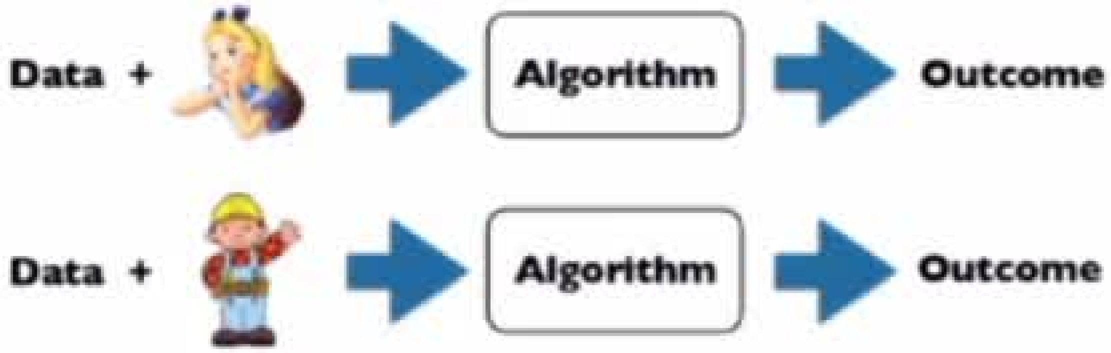
Differential Privacy

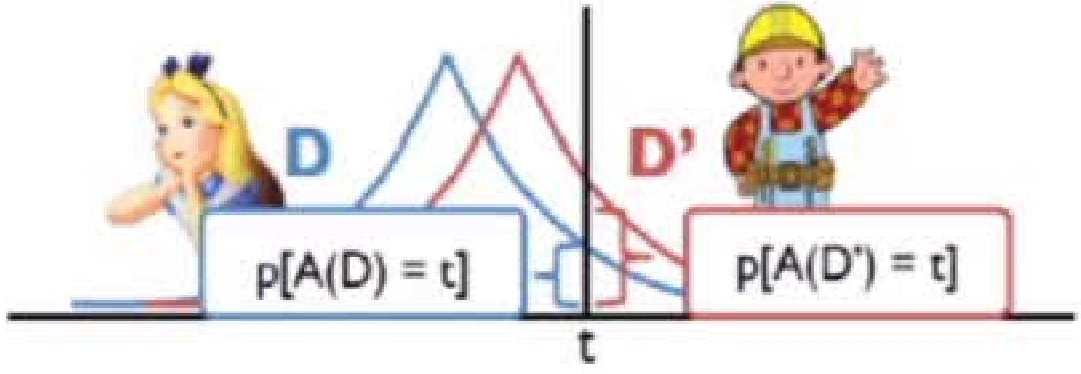
Differential Privacy:
A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all neighboring datasets D, D' and for all events S in the output space of $\mathcal{A}$, we have

$$Pr(\mathcal{A}(D) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(D') \in S) + \delta,$$

when $\delta = 0$ and $\mathcal{A}$ is ϵ-differentially private [3].

What DP Guarantees?





Participation of a person does not change the outcome much! We cannot infer whether the boy or the girl is participated in the dataset or not based on the outcome. Therefore the privacy of the participant is guaranteed.

For all D, D' that differ in one person's value, if $\mathcal{A}$ is (ϵ, δ)-DP, then:

$$\max_{S, Pr(\mathcal{A}(D) \in S) > \delta} \left\{ \log \frac{Pr(\mathcal{A}(D) \in S) - \delta}{Pr(\mathcal{A}(D')) \in S} \right\} \leq \epsilon$$

[3] C. Dwork and A. Roth, The Algorithmic Foundation of Differential Privacy, 2014.

Differential Privacy Mechanism

To obtain the DP solution of the empirical risk, $\mathcal{L}(\mathbf{w}, \{\mathbf{x}_i, y_i\}_{i=1}^n)$, we can do:

- **Output Perturbation:**

$$\mathbf{w}^{\mathrm{priv}} = \mathrm{argmin}\mathcal{L}(\mathbf{w}, \{\mathbf{x}_i, y_i\}_{i=1}^n) + \mathbf{b}_{\mathrm{priv}}^{\alpha}.$$

- **Objective Perturbation:**

$$\mathcal{L}_{\mathrm{priv}}(\mathbf{w}, \{\mathbf{x}_i, y_i\}_{i=1}^n) = \mathcal{L}(\mathbf{w}, \{\mathbf{x}_i, y_i\}_{i=1}^n) + \frac{1}{n} < \mathbf{b}_{\mathrm{priv}}^{\beta}, \mathbf{w} > .$$

- **Gradient Perturbation:**

$$\nabla\mathcal{L}_{\mathrm{priv}}(\mathbf{w}^t, \{\mathbf{x}_i, y_i\}_{i=1}^n) = \nabla\mathcal{L}(\mathbf{w}^t, \{\mathbf{x}_i, y_i\}_{i=1}^n) + \mathbf{b}_{\mathrm{priv}}^{\gamma}.$$

Laplace Mechanism: the noise is sampled from Laplace distribution, which typically guarantees ϵ-DP.
Gaussian Mechanism: the noise is sampled from the Gaussian distribution, which typically guarantees (ϵ, δ)-DP.

Our Goal: develop algorithms to perform empirical risk minimization, which can improve the utility and privacy guarantee.

Definition

L_1/L_2-**Sensitivity.** For any given loss function $\mathcal{L}(\mathbf{w}, \{\mathbf{x}, \mathbf{y}\})$, the L_1-sensitivity of $\nabla\mathcal{L}$ is:

$$\Delta_1(\nabla\mathcal{L}) = \max_{\|\mathbf{x}-\mathbf{x}'\|_1=1} \|\nabla\mathcal{L}(\mathbf{w}, \{\mathbf{x}, \mathbf{y}\}) - \nabla\mathcal{L}(\mathbf{w}, \{\mathbf{x}', \mathbf{y}'\})\|_1,$$

and the L_2-sensitivity of $\nabla\mathcal{L}$ is:

$$\Delta_2(\nabla\mathcal{L}) = \max_{\|\mathbf{x}-\mathbf{x}'\|_1=1} \|\nabla\mathcal{L}(\mathbf{w}, \{\mathbf{x}, \mathbf{y}\}) - \nabla\mathcal{L}(\mathbf{w}, \{\mathbf{x}', \mathbf{y}'\})\|_2,$$

where $\|\mathbf{x} - \mathbf{x}'\|_1$ means the data sets $\{\mathbf{x}\}$ and $\{\mathbf{x}'\}$ differ in only one entry.

Dwork et al, 2004

Laplace and Gaussian Mechanism

Theorem

Laplace Mechanism *Given any function* $f : \mathbf{x} \rightarrow \mathbf{y}$*, the following Laplace mechanism guarantees* $(\epsilon, 0)$*-differential privacy.*

$$\mathcal{M}_L(\mathbf{x}, f(\mathbf{x}), \epsilon) = f(\mathbf{x}) + \mathbf{Y},$$

where $\mathbf{Y}$ *is the Laplace noise with the same dimension as* $\mathbf{y}$ *and each coordinate is sampled i.i.d. from* $\mathrm{Lap}(\frac{\Delta_1(f)}{\epsilon})$.

Theorem

Gaussian Mechanism *Given any function* $f : \mathbf{x} \rightarrow \mathbf{y}$*, the following Gaussian mechanism guarantees* (ϵ, δ)*-differential privacy* $(0 < \epsilon < 1)$*.*

$$\mathcal{M}_L(\mathbf{x}, f(\mathbf{x}), \epsilon) = f(\mathbf{x}) + \mathbf{Y},$$

where $\mathbf{Y}$ *is the Gaussian noise with the same dimension as* $\mathbf{y}$ *and each coordinate is sampled i.i.d. from* $\mathcal{N}(0, \sigma^2)$ *with* $\sigma \geq \frac{c\Delta_2(f)}{\epsilon}$ *for* $\forall\ c^2 > 2\ln\left(\frac{1.25}{\delta}\right)$.

Two DP Strategies

- **Strategy I**

$$\mathbf{w}^{k+1} = \mathbf{w}^{k} - \eta \mathbf{A}_{\sigma}^{-1} \left(\nabla \mathcal{L}(\mathbf{w}, \{\mathbf{x}, \mathbf{y}\}) + \mathbf{n} \right). \tag{3}$$

- **Strategy II**

$$\mathbf{w}^{k+1} = \mathbf{w}^{k} - \eta \left(\mathbf{A}_{\sigma}^{-1} \nabla \mathcal{L}(\mathbf{w}, \{\mathbf{x}, \mathbf{y}\}) + \mathbf{n} \right). \tag{4}$$

Privacy Guarantee

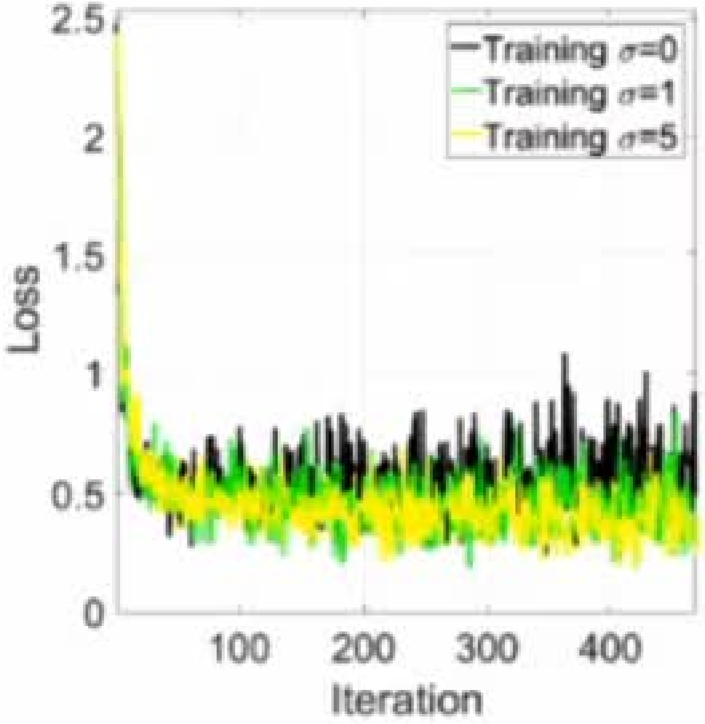
Theorem

For a given noise $\mathbf{n}$*, if it guarantees* (ϵ, δ)*-differential privacy for the (S)GD. Then with the same noise* $\mathbf{n}$*, the scheme in the* **Strategy I** *is also guaranteed to be* (ϵ, δ)*-differentially private.*

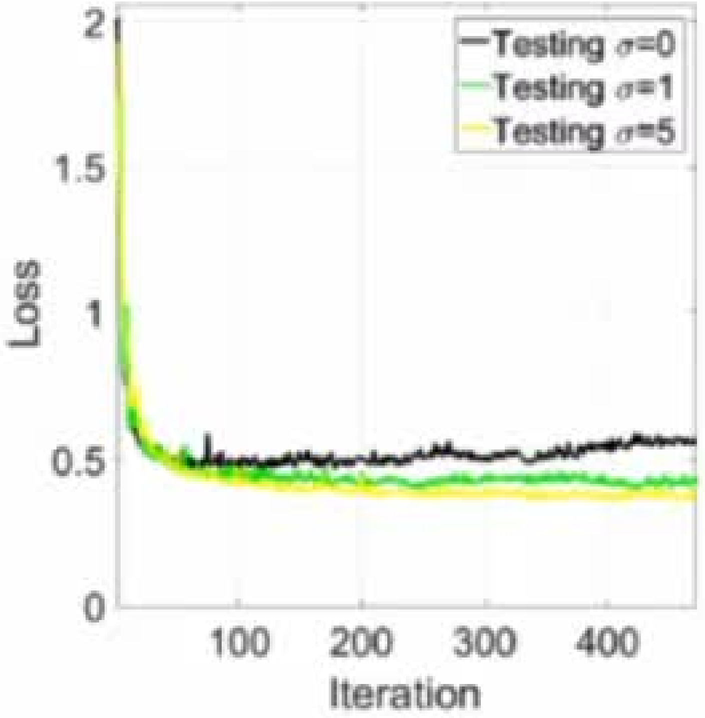
Remark

For **Strategy II***, we can use a fixed noise to guarantee different level of differential privacy with high probability. This leads to multi-level differential privacy guarantee.*

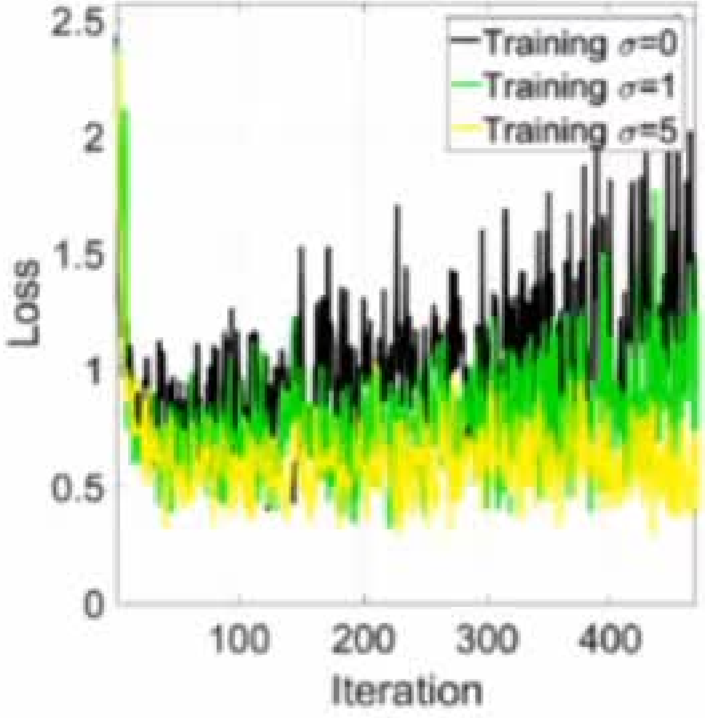
Training/Testing Loss of the Softmax Regression Model



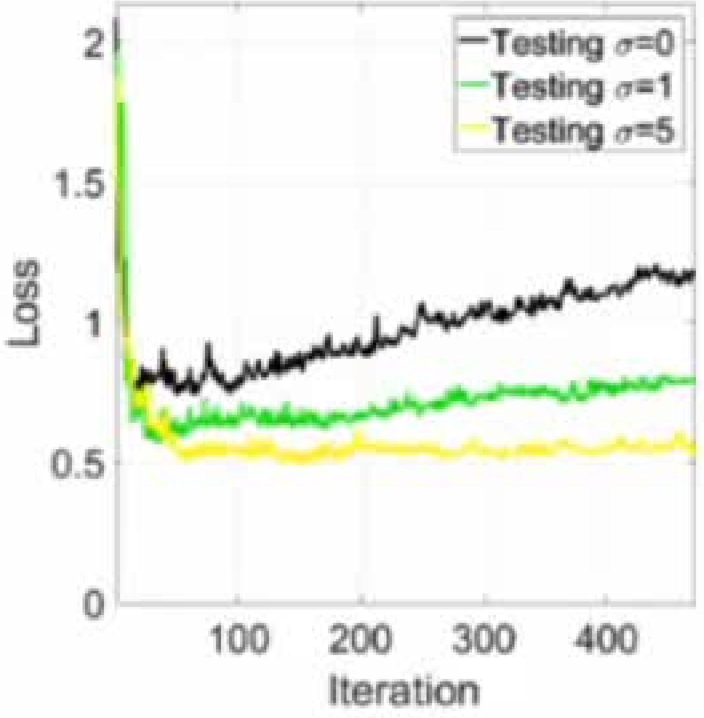
$\mathcal{N}(0, 0.1^2)$ Training



$\mathcal{N}(0, 0.1^2)$ Testing



$\mathcal{N}(0, 0.2^2)$ Training



$\mathcal{N}(0, 0.2^2)$ Testing