CS5562: Trustworthy Machine Learning

Part II Lecture 4: Differentially Private Learning

Reza Shokri[a]

Aug 2023

[a]Acknowledgment. The wonderful teaching assistants: Hongyan Chang, Martin Strobel, Jiashu Tao, Yao Tong, Jiayuan Ye

Contents

Differential Privacy

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}\right) \leq \epsilon$$

(a) Large ϵ

(b) Small ε

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}) \leq \epsilon$$

(a) Large ϵ **(b)** Small ε

Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}\right) \leq \epsilon$$

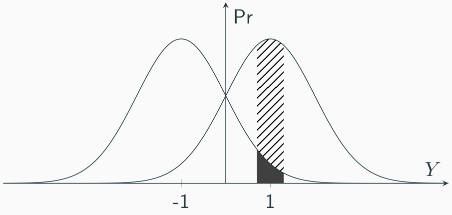
(a) Large ϵ

(b) Small ε

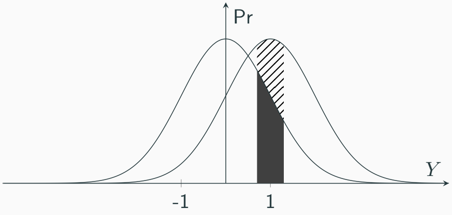
Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: ϵ-DP

$$\forall y, x, x' : \qquad \ln\left(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}\right) \leq \epsilon$$



(a) Large ϵ



(b) Small ε

An Approximate Notion of Differential Privacy

- Consider $x = \langle x_1, x_2, \cdots, x_i, \cdots x_n \rangle$
- Consider a neighboring dataset $x' = \langle x_1, x_2, \cdots, \cancel{x_i}, \cdots x_n \rangle$
- Definition: (ϵ, δ)-DP

$$\forall x, x' : \qquad \Pr\left[\overbrace{\ln(\frac{\Pr[Y = y | X = x]}{\Pr[Y = y | X = x']}) > \epsilon}^{\text{violating } \epsilon\text{-DP}} \right] < \delta$$

where the randomness of probability is over output y drawn from the output distribution $Pr[Y | X = x]$

- The chance that we have unbounded privacy loss is very small (δ)

An Approximate Notion of Differential Privacy

$$\Pr[Y = y|X = x] \leq e^{\epsilon} \Pr[Y = y|X = x'] + \delta$$

Differentially Private Mechanisms

Example: Counting Queries

- Assume there is a sensitive dataset, and the analyst is interested in counting how many records in the dataset match a given predicate (the query)
- How much can a small modification in the dataset change the output?
- Definition: **Sensitivity** of a function $f : (x_1, \cdots, x_n) \mapsto (y_1, \cdots, y_k)$ with respect to a norm $\|\cdot\|$ is $\Delta f = \max_{\text{neighboring datasets } x, x'} \|f(x) - f(x')\|$
- Sensitivity of the counting function is 1
- How to randomize true counts to satisfy differential privacy?

Example: Counting Queries

- Assume there is a sensitive dataset, and the analyst is interested in counting how many records in the dataset match a given predicate (the query)
- How much can a small modification in the dataset change the output?
- Definition: **Sensitivity** of a function $f : (x_1, \cdots, x_n) \mapsto (y_1, \cdots, y_k)$ with respect to a norm $\|\cdot\|$ is $\Delta f = \max_{\text{neighboring datasets } x, x'} \|f(x) - f(x')\|$
- Sensitivity of the counting function is 1
- How to randomize true counts to satisfy differential privacy?

Example: Counting Queries

- Assume there is a sensitive dataset, and the analyst is interested in counting how many records in the dataset match a given predicate (the query)
- How much can a small modification in the dataset change the output?
- Definition: **Sensitivity** of a function $f : (x_1, \cdots, x_n) \mapsto (y_1, \cdots, y_k)$ with respect to a norm $\|\cdot\|$ is $\Delta f = \max_{\text{neighboring datasets } x,x'} \|f(x) - f(x')\|$
- Sensitivity of the counting function is 1
- How to randomize true counts to satisfy differential privacy?

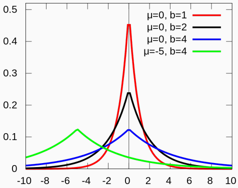
Example: Counting Queries

- Assume there is a sensitive dataset, and the analyst is interested in counting how many records in the dataset match a given predicate (the query)
- How much can a small modification in the dataset change the output?
- Definition: **Sensitivity** of a function $f : (x_1, \cdots, x_n) \mapsto (y_1, \cdots, y_k)$ with respect to a norm $\|\cdot\|$ is $\Delta f = \max_{\text{neighboring datasets } x,x'} \|f(x) - f(x')\|$
- Sensitivity of the counting function is 1
- How to randomize true counts to satisfy differential privacy?

Example: Counting Queries

- Assume there is a sensitive dataset, and the analyst is interested in counting how many records in the dataset match a given predicate (the query)
- How much can a small modification in the dataset change the output?
- Definition: **Sensitivity** of a function $f : (x_1, \cdots, x_n) \mapsto (y_1, \cdots, y_k)$ with respect to a norm $\|\cdot\|$ is $\Delta f = \max_{\text{neighboring datasets } x,x'} \|f(x) - f(x')\|$
- Sensitivity of the counting function is 1
- How to randomize true counts to satisfy differential privacy?

Laplace Mechanism



- Laplace distribution (centered at 0, with scale b):

$$Lap(z; b) = \frac{1}{2b} e^{\frac{-|z|}{b}}$$

- Laplace mechanism: $M(x, f, \epsilon) = f(x) + noise$, where coordinates of $noise \overset{\text{i.i.d}}{\sim} Lap(\Delta f / \epsilon)$
- $\Delta f = \max_{x,x'} \|f(x) - f(x')\|_1$, where x, x' are neighboring datasets

Laplace Mechanism is Differentially Private

- We prove for one-dimensional case, i.e. $f(x)$ is real number.

$$
\begin{aligned}
\frac{\Pr[M(x,f,\epsilon)=y]}{\Pr[M(x',f,\epsilon)=y]} &= \frac{e^{\frac{-|f(x)-y|}{\Delta f/\epsilon}}}{e^{\frac{-|f(x')-y|}{\Delta f/\epsilon}}} \\
&= e^{\frac{\epsilon}{\Delta f}(|f(x')-y|-|f(x)-y|)} \\
&\le e^{\frac{\epsilon}{\Delta f}(|f(x')-f(x)|)} \qquad \text{triangle inequality} \\
&\le e^{\epsilon} \qquad \text{sensitivity}
\end{aligned}
$$

Source: [Dwork and Roth, 2014]

Gaussian Mechanism

- Gaussian distribution (centered at 0, with standard deviation σ):

$$z \sim N(0, \sigma^2), \quad p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-z^2}{2\sigma^2})$$

- Gaussian mechanism:

$M(x, f, \epsilon, \delta) = f(x) + noise,$ where coordinates of $noise \overset{\text{i.i.d}}{\sim} N\left(0, \sigma^2\right)$

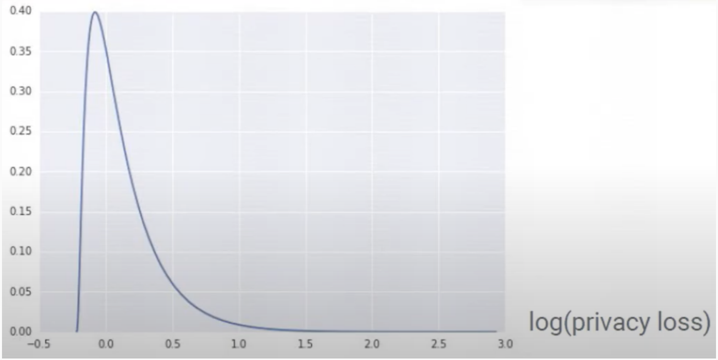
$$\text{for } \sigma = \frac{\Delta f}{\epsilon}\sqrt{2 \log \frac{5}{4\delta}}, \text{for } \epsilon \in (0, 1)$$

- $\Delta f = \max_{x,x'} \|f(x) - f(x')\|_2$, where x, x' are neighboring datasets

Source: [Dwork and Roth, 2014]

Privacy loss random variable has a long tail

- **privacy loss random variable** $L = \frac{\Pr[Y=y|X=x]}{\Pr[Y=y|X=x']}$, $y \sim M(x', f, \epsilon, \delta)$



Source: [Abadi et al., 2016]

How to bound the tail: moment method and Markov inequality

- We need tail bound $\Pr[L \geq e^{\epsilon}] < \delta$ for the random variable $L = \frac{\Pr[Y=y|X=x]}{\Pr[Y=y|X=x']}$, $y \sim M(x', f, \epsilon, \delta)$
- The λ-**th moment** $(\lambda \geq 0)$ of the random variable L : $\mathbb{E}\left[L^{\lambda}\right]$
  Example: the first order moment of random variable L is its mean
- The **Markov inequality** for non-negative random variable L:

$$\Pr[L \geq e^{\epsilon}] \leq \frac{\mathbb{E}[L^{\lambda}]}{e^{\lambda\epsilon}}.$$

Source: [Abadi et al., 2016]

How to bound the tail: moment method and Markov inequality

- We need tail bound $\Pr[L \geq e^{\epsilon}] < \delta$ for the random variable $L = \frac{\Pr[Y=y|X=x]}{\Pr[Y=y|X=x']}$, $y \sim M(x', f, \epsilon, \delta)$
- The λ-**th moment** $(\lambda \geq 0)$ of the random variable L : $\mathbb{E}\left[L^{\lambda}\right]$
  Example: the first order moment of random variable L is its mean
- The **Markov inequality** for non-negative random variable L:

$$\Pr[L \geq e^{\epsilon}] \leq \frac{\mathbb{E}[L^{\lambda}]}{e^{\lambda\epsilon}}.$$

Source: [Abadi et al., 2016]

Gaussian Mechanism is (ϵ, δ)-differentially private

- Without loss of generality, let $f(x') = 0$ and $f(x) = f(x') - \Delta_f$.

$$L = \frac{\Pr[Y = y|X = x]}{\Pr[Y = y|X = x']} = \frac{e^{-\frac{(f(x)-y)^2}{2\sigma^2}}}{e^{-\frac{(f(x')-y)^2}{2\sigma^2}}}$$

$$= e^{-\frac{\Delta_f^2 + 2y\Delta_f}{2\sigma^2}} \text{ for } \sigma = \frac{\Delta f}{\epsilon}\sqrt{2\log\frac{5}{4\delta}}$$

- Compute moments $\mathbb{E}[L^\lambda]$ for $\lambda \geq 0$ and use Markov inequality

$$\Pr[L \geq e^\epsilon] \leq e^{-\lambda\epsilon}\mathbb{E}[L^\lambda] = e^{-\lambda\epsilon}\int e^{-\frac{\lambda\Delta_f^2 + \lambda\cdot 2y\Delta f}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{y^2}{2\sigma^2}} dy$$

$$= e^{-\lambda\epsilon - \frac{\lambda\Delta_f^2}{2\sigma^2} + \frac{\lambda^2\Delta_f^2}{2\sigma^2}} = e^{-\lambda\epsilon - \frac{\lambda\epsilon^2}{4\log 5/(4\delta)} + \frac{\lambda^2\epsilon^2}{4\log 5/(4\delta)}}$$

$$< \delta \quad \text{(by setting } \lambda = \frac{2\log(1/\delta)}{\epsilon}\text{)}$$

Source: [Abadi et al., 2016]

Comparison: Laplace mechanism and Gaussian mechanism

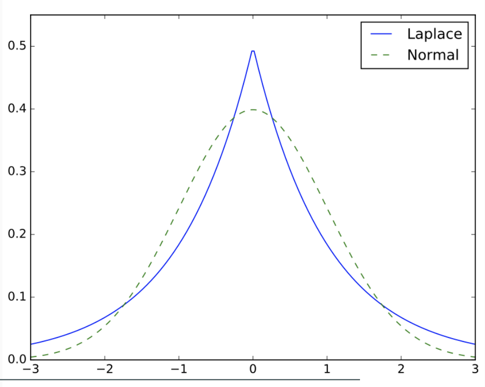
- Consider real-valued function f with $\Delta_f = 1$
- To ensure ε-DP for $\varepsilon = 1$, we need Laplace noise $noise_L \sim Lap(1)$.
- To ensure (ε, δ)-DP for $\varepsilon = 1$, we need Gaussian noise $noise_G \sim N(0, \sigma^2)$ with $\sigma = \sqrt{2 \log \frac{5}{4\delta}}$.

Comparison: Laplace mechanism and Gaussian mechanism

- Consider real-valued function f with $\Delta_f = 1$
- To ensure ε-DP for $\varepsilon = 1$, we need Laplace noise $noise_L \sim Lap(1)$.
- To ensure (ε, δ)-DP for $\varepsilon = 1$, we need Gaussian noise $noise_G \sim N(0, \sigma^2)$ with $\sigma = \sqrt{2 \log \frac{5}{4\delta}}$.

Comparison: Laplace mechanism and Gaussian mechanism

- Laplace noise $noise_L$ satisfies $p(|noise_L| = z) \propto e^{-|z|}$
- Gaussian noise $noise_G$ satisfies $p(|noise_G| = z) \propto e^{-\frac{z^2}{4 \log \frac{5}{4\delta}}}$
- As $z \to \infty$, we have $p(|noise_L| = z) \gg p(|noise_G| = z)$



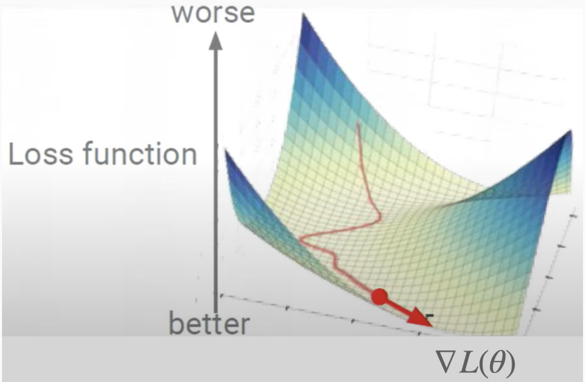
$\Rightarrow$ Laplace noise has a longer tail, thus tends to give larger error

Source: Blog by John D. Cook

Differentially Private SGD

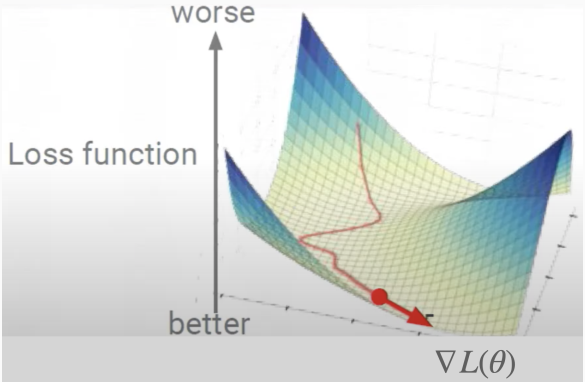
Training a machine learning model with SGD

- How does SGD work? In iteration t of the algorithm, we
  - Choose a mini-batch B_t of the training data
  - Compute the average gradient $g = \frac{1}{|B_t|} \sum_{z \in B_t} \nabla L(\theta, z)$
  - Take a step (with stepsize η_t) in the opposite direction of the average gradient: $\theta \leftarrow \theta - \eta_t g$



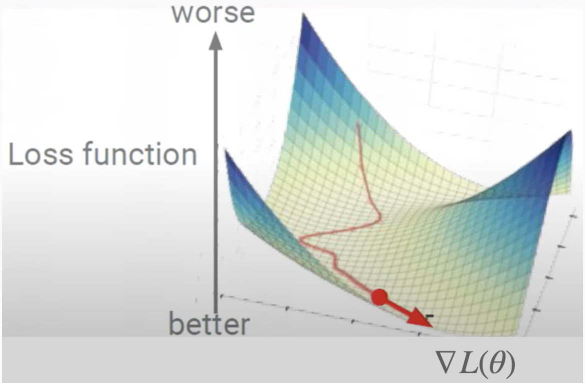
Training a machine learning model with SGD

- How does SGD work? In iteration t of the algorithm, we
  - Choose a mini-batch B_t of the training data
  - Compute the average gradient $g = \frac{1}{|B_t|} \sum_{z \in B_t} \nabla L(\theta, z)$
  - Take a step (with stepsize η_t) in the opposite direction of the average gradient: $\theta \leftarrow \theta - \eta_t g$



Training a machine learning model with SGD

- How does SGD work? In iteration t of the algorithm, we
  - Choose a mini-batch B_t of the training data
  - Compute the average gradient $g = \frac{1}{|B_t|}\sum_{z \in B_t} \nabla L(\theta, z)$
  - Take a step (with stepsize η_t) in the opposite direction of the average gradient: $\theta \leftarrow \theta - \eta_t g$



How can we design a differentially private SGD algorithm?

- What is the data dependent computation?
- What is its sensitivity?
- Which DP mechanism should we use?

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
  - Can we bound the sensitivity?
  - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
  - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
  - Can we bound the sensitivity?
  - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
  - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
    - Can we bound the sensitivity?
    - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
    - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
  - Can we bound the sensitivity?
  - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
  - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
  - Can we bound the sensitivity?
  - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
  - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

- What is the data dependent computation? gradient $\nabla L(\theta, z)$
- What is its sensitivity? unbounded
  - Can we bound the sensitivity?
  - Use norm-bounding: Normalize the gradient vector to a given L2-norm C
  - This is an extremely bad way of bounding sensitivity, because it sets the sensitivity to the range of the function (but we don't know how to do better than this)
- Which DP mechanism should we use? Gaussian mechanism, as compared with the Laplace mechanism, we impose less error

Training a machine learning model with DP-SGD

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

DP-SGD Algorithm [Abadi et al., 2016]

How private is the DP-SGD algorithm?

- For running a single iteration:
  - We use one (ε, δ)-differentially private Gaussian Mechanism to compute noisy gradient
  - However, does the preceding mini-batch sub-sampling procedure change the privacy bound of subsequent Gaussian mechanism?

How private is the DP-SGD algorithm?

- For running a single iteration:
  - We use one (ε, δ)-differentially private Gaussian Mechanism to compute noisy gradient
  - However, does the preceding mini-batch sub-sampling procedure change the privacy bound of subsequent Gaussian mechanism?

How private is the DP-SGD algorithm?

- For running a single iteration:
  - We use one (ε, δ)-differentially private Gaussian Mechanism to compute noisy gradient
  - However, does the preceding mini-batch sub-sampling procedure change the privacy bound of subsequent Gaussian mechanism?

Amplification by Sub-sampling

- Denote $M(x, f, \epsilon, \delta)$ as a (ϵ, δ)-differentially private Gaussian mechanism
- Let $\text{Poi}(x, q)$ be the Poisson sub-sampling mechanism on dataset x that includes each record x_i independently with probability q
- Sub-sampled Gaussian mechanism

$$M_q(x, f, \epsilon, \delta) = f \circ \text{Poi}(x, q) + noise$$

where coordinates of $noise \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$ with $\sigma = \frac{\Delta f}{\epsilon}\sqrt{2 \log \frac{5}{4\delta}}$

Amplification by Sub-sampling

- Denote $M(x, f, \epsilon, \delta)$ as a (ϵ, δ)-differentially private Gaussian mechanism
- Let $\mathsf{Poi}(x, q)$ be the Poisson sub-sampling mechanism on dataset x that includes each record x_i independently with probability q
- Sub-sampled Gaussian mechanism

$$M_q(x, f, \epsilon, \delta) = f \circ \mathsf{Poi}(x, q) + noise$$

where coordinates of $noise \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$ with $\sigma = \frac{\Delta f}{\epsilon}\sqrt{2 \log \frac{5}{4\delta}}$

Amplification by Sub-sampling

- Denote $M(x, f, \epsilon, \delta)$ as a (ϵ, δ)-differentially private Gaussian mechanism
- Let $\mathsf{Poi}(x, q)$ be the Poisson sub-sampling mechanism on dataset x that includes each record x_i independently with probability q
- Sub-sampled Gaussian mechanism

$$M_q(x, f, \epsilon, \delta) = f \circ \mathsf{Poi}(x, q) + noise$$

where coordinates of $noise \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$ with $\sigma = \frac{\Delta f}{\epsilon}\sqrt{2 \log \frac{5}{4\delta}}$

Amplification by Sub-sampling for moments

- Denote the privacy loss random variable for a Gaussian mechanism as $L = \frac{\Pr[M(x,f,\epsilon,\delta)=\theta]}{\Pr[M(x',f,\epsilon,\delta)=\theta]}, \theta \sim M(x', f, \epsilon, \delta)$
- Denote the privacy loss random variable for a sub-sampled Gaussian mechanism $L_q = \frac{\Pr[M_q(x,f,\epsilon,\delta)=\theta]}{\Pr[M_q(x',f,\epsilon,\delta)=\theta]}, \theta \sim M_q(x', f, \epsilon, \delta)$ where q is the sub-sampling probability
- Then, we can prove $\ln \mathbb{E}[L_q^\lambda] \le \frac{q^2}{1-q} \ln \mathbb{E}[L^\lambda] + O(q^3\lambda^3/\sigma^3)$

Source: [Abadi et al., 2016, Lemma 3, Theorem 1]

Amplification by Sub-sampling for moments

- Denote the privacy loss random variable for a Gaussian mechanism as $L = \frac{\Pr[M(x,f,\epsilon,\delta)=\theta]}{\Pr[M(x',f,\epsilon,\delta)=\theta]}, \theta \sim M(x', f, \epsilon, \delta)$
- Denote the privacy loss random variable for a sub-sampled Gaussian mechanism $L_q = \frac{\Pr[M_q(x,f,\epsilon,\delta)=\theta]}{\Pr[M_q(x',f,\epsilon,\delta)=\theta]}, \theta \sim M_q(x', f, \epsilon, \delta)$ where q is the sub-sampling probability
- Then, we can prove $\ln \mathbb{E}[L_q^\lambda] \leq \frac{q^2}{1-q} \ln \mathbb{E}[L^\lambda] + O(q^3\lambda^3/\sigma^3)$

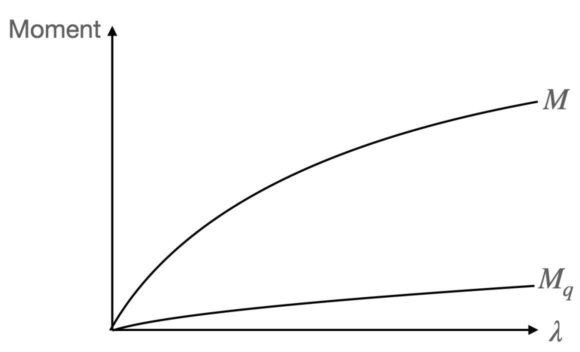
Source: [Abadi et al., 2016, Lemma 3, Theorem 1]

Amplification by Sub-sampling for moments

- Denote the privacy loss random variable for a Gaussian mechanism as $L = \frac{\Pr[M(x,f,\epsilon,\delta)=\theta]}{\Pr[M(x',f,\epsilon,\delta)=\theta]}, \theta \sim M(x', f, \epsilon, \delta)$
- Denote the privacy loss random variable for a sub-sampled Gaussian mechanism $L_q = \frac{\Pr[M_q(x,f,\epsilon,\delta)=\theta]}{\Pr[M_q(x',f,\epsilon,\delta)=\theta]}, \theta \sim M_q(x', f, \epsilon, \delta)$ where q is the sub-sampling probability
- Then, we can prove $\ln \mathbb{E}[L_q^\lambda] \leq \frac{q^2}{1-q} \ln \mathbb{E}[L^\lambda] + O(q^3\lambda^3/\sigma^3)$

Source: [Abadi et al., 2016, Lemma 3, Theorem 1]

Amplification by Sub-sampling



- For moments, we prove that $\ln \mathbb{E}[L_q^\lambda] \leq \frac{q^2}{1-q} \ln \mathbb{E}[L^\lambda] + O(q^3\lambda^3/\sigma^3)$
- Therefore, by applying Markov inequality, we can prove that $M_q(x, f, \epsilon, \delta)$ is approximately $(q\epsilon, \delta)$-DP

Source: [Abadi et al., 2016, Lemma 3, Theorem 1]

How private is the DP-SGD algorithm?

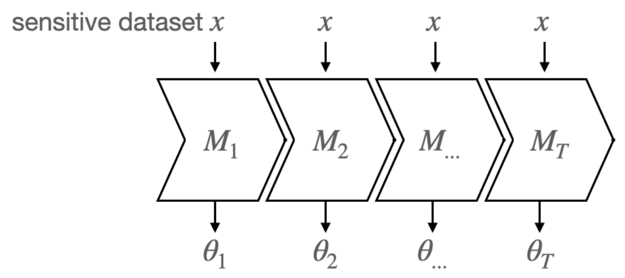
- For running a single iteration:
  - We use one approximately $(q\varepsilon, \delta)$-differentially private sub-sampled Gaussian Mechanism to compute noisy gradient, where $q = \frac{L}{N}$
- For running multiple iterations:
  - How to compose the privacy bound for each iteration to obtain a privacy bound for the whole algorithm?

How private is the DP-SGD algorithm?

- For running a single iteration:
  - We use one approximately $(q\varepsilon, \delta)$-differentially private sub-sampled Gaussian Mechanism to compute noisy gradient, where $q = \frac{L}{N}$
- For running multiple iterations:
  - How to compose the privacy bound for each iteration to obtain a privacy bound for the whole algorithm?

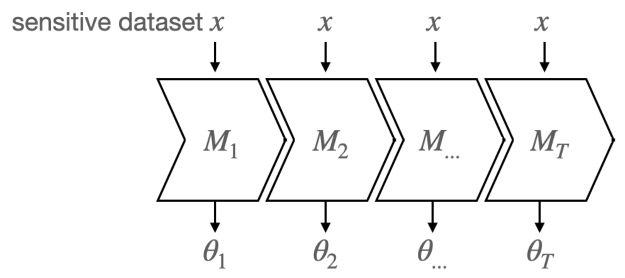
How to compose privacy bound for multiple iterations?

- Assume, we run an iterative algorithm that uses a DP mechanism M_i on a sensitive dataset, in the i-th iteration of computation.
- Let the total number of iterations be T
- Let the randomness used by the T DP mechanisms be independent
- Outputting $M_i(x)$ is (ϵ, δ)-differentially private $\Rightarrow$ How private is outputting the **composition** of $M_1(x), \cdots, M_T(x)$?



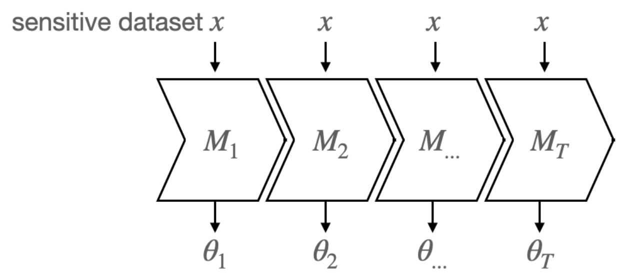
How to compose privacy bound for multiple iterations?

- Assume, we run an iterative algorithm that uses a DP mechanism M_i on a sensitive dataset, in the i-th iteration of computation.
- Let the total number of iterations be T
- Let the randomness used by the T DP mechanisms be independent
- Outputting $M_i(x)$ is (ϵ, δ)-differentially private $\Rightarrow$ How private is outputting the **composition** of $M_1(x), \cdots, M_T(x)$?



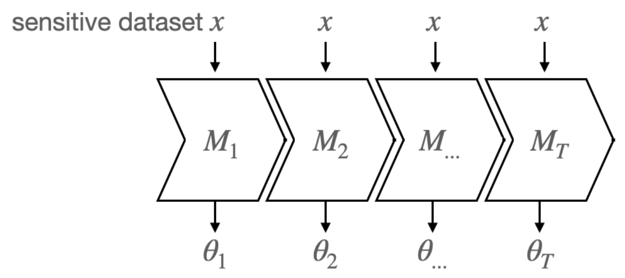
How to compose privacy bound for multiple iterations?

- Assume, we run an iterative algorithm that uses a DP mechanism M_i on a sensitive dataset, in the i-th iteration of computation.
- Let the total number of iterations be T
- Let the randomness used by the T DP mechanisms be independent
- Outputting $M_i(x)$ is (ϵ, δ)-differentially private $\Rightarrow$ How private is outputting the **composition** of $M_1(x), \cdots, M_T(x)$?



How to compose privacy bound for multiple iterations?

- Assume, we run an iterative algorithm that uses a DP mechanism M_i on a sensitive dataset, in the i-th iteration of computation.
- Let the total number of iterations be T
- Let the randomness used by the T DP mechanisms be independent
- Outputting $M_i(x)$ is (ϵ, δ)-differentially private $\Rightarrow$ How private is outputting the **composition** of $M_1(x), \cdots, M_T(x)$?



Moments for composition of DP mechanisms

- Let the privacy loss random variable for i-th mechanism be $L_i = \frac{\Pr[M_i(x,f,\epsilon,\delta)=y]}{\Pr[M_i(x',f,\epsilon,\delta)=y]}, y \sim M_i(x', f, \epsilon, \delta)$
- Then the composition of T sequential DP mechanisms is $M_{com}(x, f, \epsilon, \delta) : x \mapsto (M_1(x), \cdots, M_T(x))$.
- We need to analyze the moment of privacy loss random variable for composed mechanism

$$L_{com} = \frac{\Pr[M_{com}(x, f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}{\Pr[M_q(x', f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}$$

where $(\theta_1, \cdots, \theta_T) \sim M_{com}(x', f, \epsilon, \delta)$

Moments for composition of DP mechanisms

- Let the privacy loss random variable for i-th mechanism be $L_i = \frac{\Pr[M_i(x,f,\epsilon,\delta)=y]}{\Pr[M_i(x',f,\epsilon,\delta)=y]}, y \sim M_i(x', f, \epsilon, \delta)$
- Then the composition of T sequential DP mechanisms is $M_{com}(x, f, \epsilon, \delta) : x \mapsto (M_1(x), \cdots, M_T(x))$.
- We need to analyze the moment of privacy loss random variable for composed mechanism

$$L_{com} = \frac{\Pr[M_{com}(x, f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}{\Pr[M_q(x', f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}$$

where $(\theta_1, \cdots, \theta_T) \sim M_{com}(x', f, \epsilon, \delta)$

Moments for composition of DP mechanisms

- Let the privacy loss random variable for i-th mechanism be $L_i = \frac{\Pr[M_i(x,f,\epsilon,\delta)=y]}{\Pr[M_i(x',f,\epsilon,\delta)=y]}, y \sim M_i(x', f, \epsilon, \delta)$
- Then the composition of T sequential DP mechanisms is $M_{com}(x, f, \epsilon, \delta) : x \mapsto (M_1(x), \cdots, M_T(x))$.
- We need to analyze the moment of privacy loss random variable for composed mechanism

$$L_{com} = \frac{\Pr[M_{com}(x, f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}{\Pr[M_q(x', f, \epsilon, \delta) = (\theta_1, \cdots, \theta_T)]}$$

where $(\theta_1, \cdots, \theta_T) \sim M_{com}(x', f, \epsilon, \delta)$

Moments for composition of DP mechanisms

- By the independence between the randomness used in T DP mechanisms $M_1, \cdots, M_T$, we have

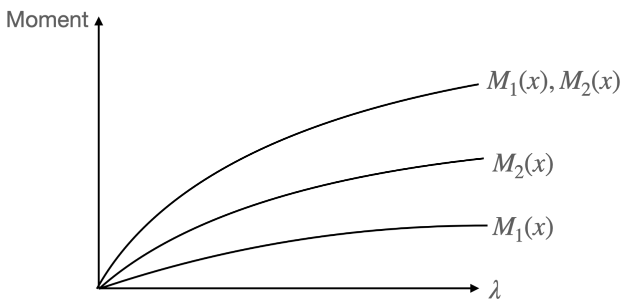
$$\begin{aligned} L_{compose} &= \frac{\Pr[M_{compose}(x) = (\theta_1, \cdots, \theta_T)]}{\Pr[M_{compose}(x') = (\theta_1, \cdots, \theta_T)]} \\ &= \frac{\Pr[M_1(x) = \theta_1] \cdots \Pr[M_T(x) = \theta_T]}{\Pr[M_1(x') = \theta_1] \cdots \Pr[M_T(x') = \theta_T]} \\ &= L_1 \cdots L_T \end{aligned}$$

  where $L_1, \cdots, L_T$ are the independent privacy loss random variables for mechanisms $M_1, \cdots, M_T$ respectively.
- Therefore, we have $\mathbb{E}[L_{com}^{\lambda}] \leq \prod_{i=1}^{T} \mathbb{E}[L_i^{\lambda}]$

Source: [Abadi et al., 2016]

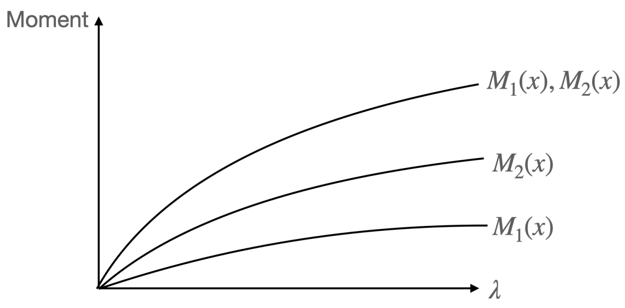
Moment accountant for composition of DP mechanisms

- By Markov inequality, the overall computation over T steps is approximately $(\epsilon\sqrt{T}, \delta)$-DP [Abadi et al., 2016, Theorem 1]
- Example: if each step is $(0.005, 10^{-5})$-DP, and after 1000 steps, the algorithm will be approximately $(0.15, 10^{-5})$-DP



Moment accountant for composition of DP mechanisms

- By Markov inequality, the overall computation over T steps is approximately $(\epsilon\sqrt{T}, \delta)$-DP [Abadi et al., 2016, Theorem 1]
- Example: if each step is $(0.005, 10^{-5})$-DP, and after 1000 steps, the algorithm will be approximately $(0.15, 10^{-5})$-DP



Advanced Topic: can we prove privacy bound for DP-SGD that is smaller than moment accountant?

- Observe that the composability of moment accountant implicitly assumes that the outputs after all k step $(y_1, \cdots, y_k)$ are released
- In reality, only the final output y_k is released, while all the preceding outputs $y_1, \cdots, y_{k-1}$ are hidden
- Under this more realistic hidden-state assumption, the privacy bound may converge, if one of the following condition holds
    - The loss function is strongly convex and smooth on unconstrained space $\mathbb{R}^d$ [Chourasia et al., 2021, Ye and Shokri, 2022, Altschuler and Talwar, 2022]
    - The loss function is convex on a bounded subset of $\mathbb{R}^d$ [Altschuler and Talwar, 2022]

Advanced Topic: can we prove privacy bound for DP-SGD that is smaller than moment accountant?

- Observe that the composability of moment accountant implicitly assumes that the outputs after all k step $(y_1, \cdots, y_k)$ are released
- In reality, only the final output y_k is released, while all the preceding outputs $y_1, \cdots, y_{k-1}$ are hidden
- Under this more realistic hidden-state assumption, the privacy bound may converge, if one of the following condition holds
    - The loss function is strongly convex and smooth on unconstrained space $\mathbb{R}^d$ [Chourasia et al., 2021, Ye and Shokri, 2022, Altschuler and Talwar, 2022]
    - The loss function is convex on a bounded subset of $\mathbb{R}^d$ [Altschuler and Talwar, 2022]

Advanced Topic: can we prove privacy bound for DP-SGD that is smaller than moment accountant?

- Observe that the composability of moment accountant implicitly assumes that the outputs after all k step $(y_1, \cdots, y_k)$ are released
- In reality, only the final output y_k is released, while all the preceding outputs $y_1, \cdots, y_{k-1}$ are hidden
- Under this more realistic hidden-state assumption, the privacy bound may converge, if one of the following condition holds
  - The loss function is strongly convex and smooth on unconstrained space $\mathbb{R}^d$ [Chourasia et al., 2021, Ye and Shokri, 2022, Altschuler and Talwar, 2022]
  - The loss function is convex on a bounded subset of $\mathbb{R}^d$ [Altschuler and Talwar, 2022]

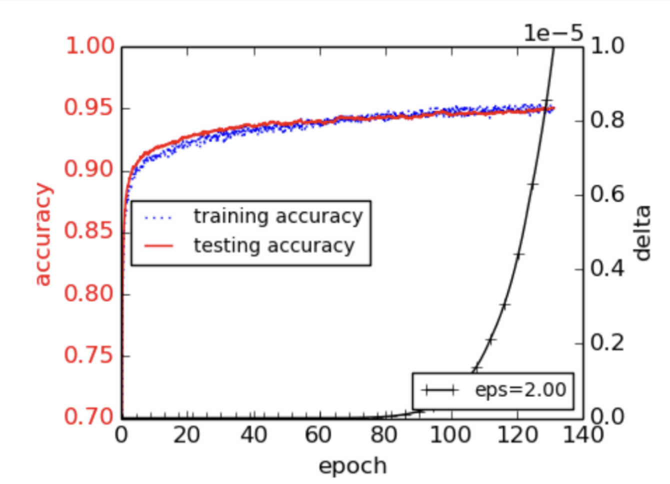
Advanced Topic: can we prove privacy bound for DP-SGD that is smaller than moment accountant?

- Observe that the composability of moment accountant implicitly assumes that the outputs after all k step $(y_1, \cdots, y_k)$ are released
- In reality, only the final output y_k is released, while all the preceding outputs $y_1, \cdots, y_{k-1}$ are hidden
- Under this more realistic hidden-state assumption, the privacy bound may converge, if one of the following condition holds
  - The loss function is strongly convex and smooth on unconstrained space $\mathbb{R}^d$ [Chourasia et al., 2021, Ye and Shokri, 2022, Altschuler and Talwar, 2022]
  - The loss function is convex on a bounded subset of $\mathbb{R}^d$ [Altschuler and Talwar, 2022]

Advanced Topic: can we prove privacy bound for DP-SGD that is smaller than moment accountant?

- Observe that the composability of moment accountant implicitly assumes that the outputs after all k step $(y_1, \cdots, y_k)$ are released
- In reality, only the final output y_k is released, while all the preceding outputs $y_1, \cdots, y_{k-1}$ are hidden
- Under this more realistic hidden-state assumption, the privacy bound may converge, if one of the following condition holds
    - The loss function is strongly convex and smooth on unconstrained space $\mathbb{R}^d$ [Chourasia et al., 2021, Ye and Shokri, 2022, Altschuler and Talwar, 2022]
    - The loss function is convex on a bounded subset of $\mathbb{R}^d$ [Altschuler and Talwar, 2022]

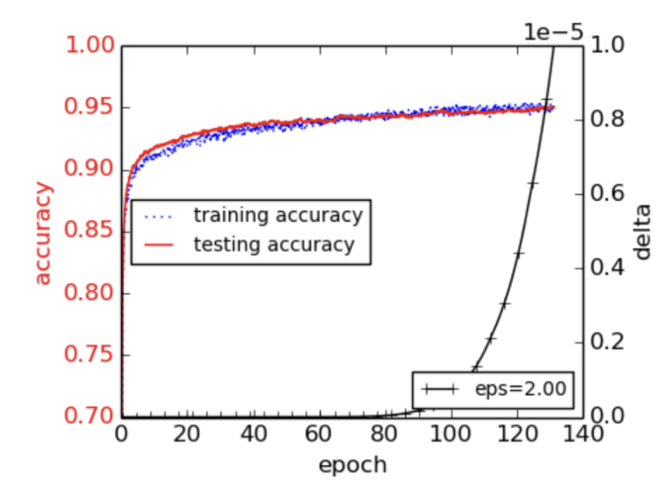
Training a machine learning model with DP-SGD



Training a NN on MNIST dataset using DP-SGD Algorithm [Abadi et al., 2016]

Same NN has 98.30% accuracy in ≈ 100 epochs, when trained non-privately

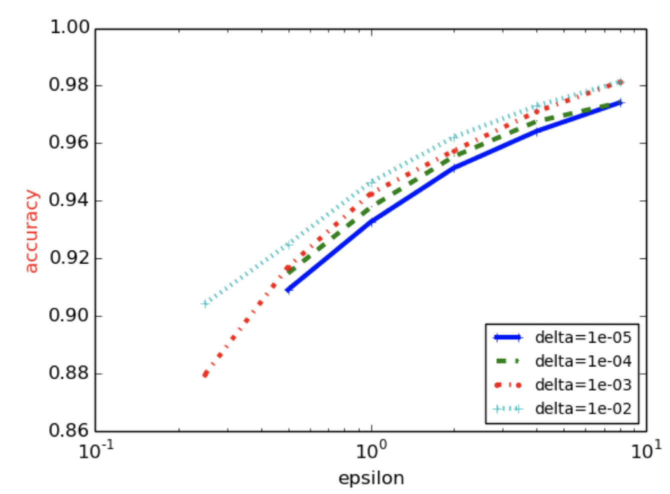
Training a machine learning model with DP-SGD



Training a NN on MNIST dataset using DP-SGD Algorithm [Abadi et al., 2016]

Same NN has 98.30% accuracy in ≈ 100 epochs, when trained non-privately

Trade-off between privacy and accuracy of DP-SGD



Best accuracy of training a NN on MNIST dataset using DP-SGD, when constrained within different differential privacy budget (ϵ, δ)

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
  - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
  - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
  - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
  - Adaptive gradient clipping [Andrew et al., 2021]
  - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
    - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
    - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
    - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
    - Adaptive gradient clipping [Andrew et al., 2021]
    - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
  - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
  - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
  - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
  - Adaptive gradient clipping [Andrew et al., 2021]
  - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
  - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
  - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
  - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
  - Adaptive gradient clipping [Andrew et al., 2021]
  - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
  - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
  - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
  - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
  - Adaptive gradient clipping [Andrew et al., 2021]
  - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
  - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
  - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
  - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
  - Adaptive gradient clipping [Andrew et al., 2021]
  - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- Start from better features for training, rather than training from scratch
    - Features of pretrained models (that did not access private dataset) [Abadi et al., 2016]
    - Handcraft features using prior insights [Tramer and Boneh, 2020]
- Use optimization algorithm that converges with fewer epochs (s.t. consumed privacy budget ϵ is also smaller), rather than vanilla SGD
    - DP-SGD with momentum [Tramer and Boneh, 2020], DP-Adam [Papernot et al., 2019]
    - Adaptive gradient clipping [Andrew et al., 2021]
    - Adaptive selection of step-size and noise scale [Asi et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- In DP-SGD, gradient computation incurs error due to clipping and additive noise
- Could we use non-sensitive information for error correction?
  - Use history of noisy gradient (in preceding iterations) for variance reduction of current gradient computation [Wang et al., 2017]
  - Project noisy gradient to a lower-dimensional space, and then add smaller amount of noise [Yu et al., 2021, Zhou et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- In DP-SGD, gradient computation incurs error due to clipping and additive noise
- Could we use non-sensitive information for error correction?
  - Use history of noisy gradient (in preceding iterations) for variance reduction of current gradient computation [Wang et al., 2017]
  - Project noisy gradient to a lower-dimensional space, and then add smaller amount of noise [Yu et al., 2021, Zhou et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- In DP-SGD, gradient computation incurs error due to clipping and additive noise
- Could we use non-sensitive information for error correction?
  - Use history of noisy gradient (in preceding iterations) for variance reduction of current gradient computation [Wang et al., 2017]
  - Project noisy gradient to a lower-dimensional space, and then add smaller amount of noise [Yu et al., 2021, Zhou et al., 2021]

Advanced topic: How to improve the trade-off between privacy and accuracy for learning algorithm?

- In DP-SGD, gradient computation incurs error due to clipping and additive noise
- Could we use non-sensitive information for error correction?
    - Use history of noisy gradient (in preceding iterations) for variance reduction of current gradient computation [Wang et al., 2017]
    - Project noisy gradient to a lower-dimensional space, and then add smaller amount of noise [Yu et al., 2021, Zhou et al., 2021]

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016).
Deep learning with differential privacy.
In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pages 308–318.

Altschuler, J. M. and Talwar, K. (2022).
Privacy of noisy stochastic gradient descent: More iterations without more privacy loss.
Advances in Neural Information Processing Systems.

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. (2021).
Differentially private learning with adaptive clipping.
Advances in Neural Information Processing Systems, 34:17455–17466.

Asi, H., Duchi, J., Fallah, A., Javidbakht, O., and Talwar, K. (2021).
Private adaptive gradient methods for convex optimization.
In International Conference on Machine Learning, pages 383–392. PMLR.

References iii

Chourasia, R., Ye, J., and Shokri, R. (2021).
Differential privacy dynamics of langevin diffusion and noisy gradient descent.
Advances in Neural Information Processing Systems, 34:14771–14781.

Dwork, C. and Roth, A. (2014).
The algorithmic foundations of differential privacy.
Foundations and Trends in Theoretical Computer Science, 9(3–4):211–407.

Papernot, N., Chien, S., Song, S., Thakurta, A., and Erlingsson, U. (2019).
Making the shoe fit: Architectures, initializations, and tuning for learning with privacy.

Tramer, F. and Boneh, D. (2020).
Differentially private learning needs better features (or much more data).
In International Conference on Learning Representations.

Wang, D., Ye, M., and Xu, J. (2017).
Differentially private empirical risk minimization revisited: Faster and more general.
Advances in Neural Information Processing Systems, 30.

Ye, J. and Shokri, R. (2022).
Differentially private learning needs hidden state (or much faster convergence).
Advances in Neural Information Processing Systems.

Yu, D., Zhang, H., Chen, W., and Liu, T.-Y. (2021).
Do not let privacy overbill utility: Gradient embedding perturbation for private learning.
International Conference on Learning Representations.

Zhou, Y., Wu, S., and Banerjee, A. (2021).
Bypassing the ambient dimension: Private sgd with gradient subspace identification.
In International Conference on Learning Representations.