



Privacy-preserving Deep Learning

A. M. Sadeghzadeh, Ph.D.

Sharif University of Technology
Computer Engineering Department (CE)
Data and Network Security Lab (DNSL)



June 4, 2023

Today's Agenda

1 Recap

2 Differential Privacy Properties

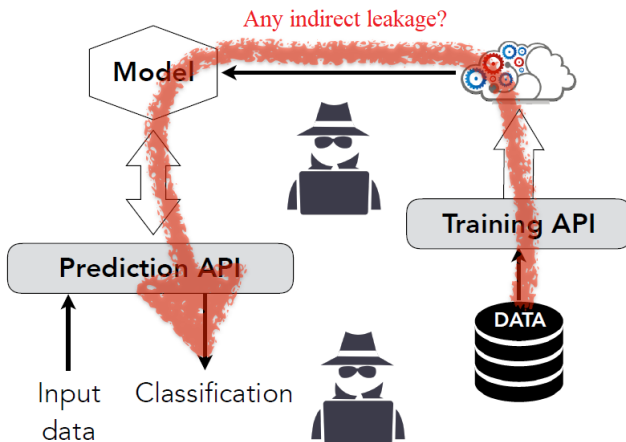
3 Differentially Private SGD

4 PATE

Recap

Membership Inference Attack

- Given a model, can an adversary infer whether data point x is part of its training set?





(Shokri, 2020)

Membership Inference Attack

- Given a model, can an adversary infer whether data point x is part of its training set?

Membership Inference:

Was  trained
on the example  ?

(Carlini, 2022)

Membership Inference Attack

- Given a model, can an adversary infer whether data point x is part of its training set?

Membership Inference:

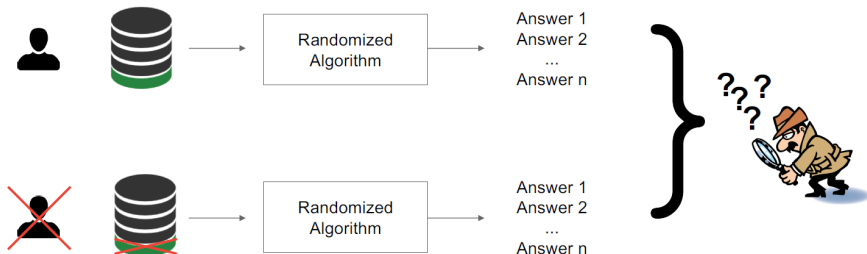
$$A = \Pr(\text{  \text{ was trained on } \text{ )$$

(Carlini, 2022)

Differential Privacy

Differential Privacy ensures that any sequence of outputs (responses to queries) is **essentially equally likely** to occur, **independent** of the presence or absence of **any individual**.

- If **nothing is learned about an individual** then the individual **cannot be harmed by the analysis**.



(Papernot, 2019)

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \approx_{\epsilon} \Pr[\mathcal{M}(y) \in \mathcal{S}]$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq (1 + \epsilon)\Pr[\mathcal{M}(y) \in \mathcal{S}]$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}]$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

- If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

- If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.
- δ is a negligible function.
 - δ that are less than the inverse of any polynomial in the size of the database.

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

- If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.
- δ is a negligible function.
 - δ that are less than the inverse of any polynomial in the size of the database.
- The definition is symmetric.

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

- If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.
- δ is a negligible function.
 - δ that are less than the inverse of any polynomial in the size of the database.
- The definition is symmetric.
- Differential privacy **is a definition, not an algorithm**.
 - For a given computational task T and a given value of ϵ **there will be many differentially private algorithms** for achieving T in an ϵ -differentially private manner. **Some will have better accuracy than others.**

Differential Privacy

Differential Privacy: A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

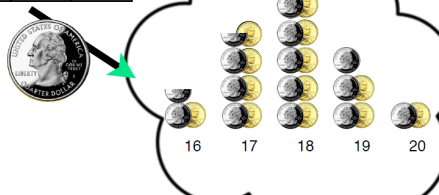
$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism \mathcal{M} .

- If $\delta = 0$, we say that \mathcal{M} is ϵ -differentially private.
- δ is a negligible function.
 - δ that are less than the inverse of any polynomial in the size of the database.
- The definition is symmetric.
- Differential privacy **is a definition, not an algorithm.**
 - For a given computational task T and a given value of ϵ **there will be many differentially private algorithms** for achieving T in an ϵ -differentially private manner. **Some will have better accuracy than others.**
- $\epsilon \approx 1$ and $\delta \ll \frac{1}{N}$ (generally speaking, one digit ϵ is good) where N is the size of dataset.

Differential Privacy

name	DOB	sex	weight	smoker	lung cancer
John Doe	12/1/51	M	185	Y	N
Jane Smith	3/3/46	F	140	N	N
Ellen Jones	4/24/33	F	160	Y	Y
Jennifer Kim	3/1/70	F	135	N	N
Rachel Waters	9/5/43	F	140	N	N

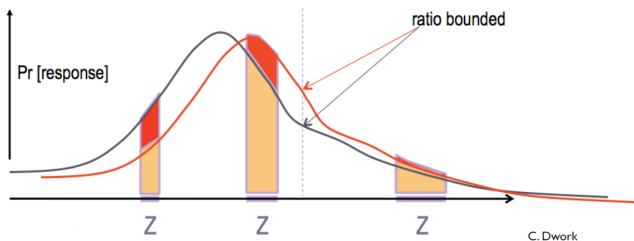


(Katrina Ligett, 2017)

Differential Privacy

ϵ -differential privacy

$$\Pr[M(x_1) \in S] \leq e^\epsilon \Pr[M(x_2) \in S]$$

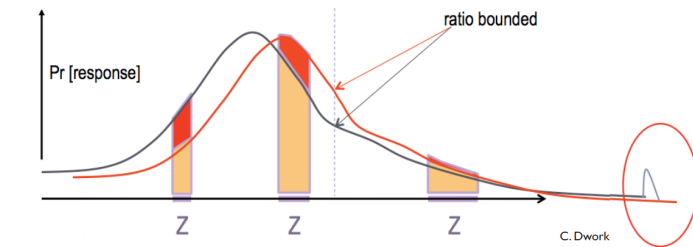


(Katrina Ligett, 2017)

Differential Privacy

(ϵ, δ) -differential privacy

$$\Pr[M(x_1) \in S] \leq e^\epsilon \Pr[M(x_2) \in S] + \delta$$



(Katrina Ligett, 2017)

Privacy Loss

(ϵ, δ) -differential privacy ensures that for all adjacent x, y , the **absolute value of the privacy loss will be bounded by ϵ** with probability at least $1 - \delta$.

The quantity

$$\mathcal{L}_{\mathcal{M}(x) \parallel \mathcal{M}(y)}^{(\xi)} = \ln \left(\frac{\Pr[\mathcal{M}(x) = \xi]}{\Pr[\mathcal{M}(y) = \xi]} \right)$$

is important to us; we refer to it as the *privacy loss* incurred by observing ξ . This loss might be positive (when an event is more likely under x than under y) or it might be negative (when an event is more likely under y than under x).

Sensitivity

The ℓ_1 sensitivity of a function f captures **the magnitude by which a single individual's data can change the function f** in the worst case

- Intuitively, the uncertainty in the response that we must introduce in order to hide the participation of a single individual.

The sensitivity of a function gives an upper bound on how much we must perturb its output to preserve privacy.

Definition 3.1 (ℓ_1 -sensitivity). The ℓ_1 -sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ is:

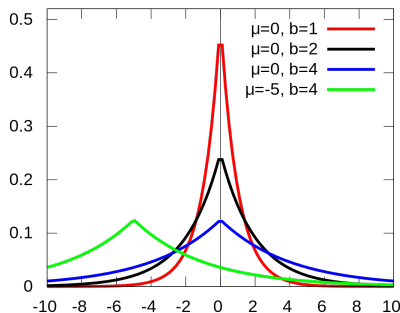
$$\Delta f = \max_{\substack{x, y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x - y\|_1 = 1}} \|f(x) - f(y)\|_1.$$

The Laplace Distribution

Definition 3.2 (The Laplace Distribution). The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function:

$$\text{Lap}(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

The variance of this distribution is $\sigma^2 = 2b^2$. We will sometimes write $\text{Lap}(b)$ to denote the Laplace distribution with scale b .



The Laplace Mechanism

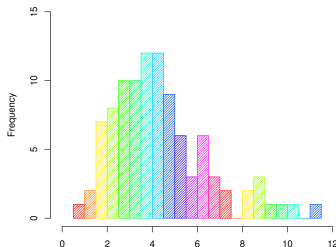
Definition 3.3 (The Laplace Mechanism). Given any function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \dots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $\text{Lap}(\Delta f / \varepsilon)$.

Histogram Queries

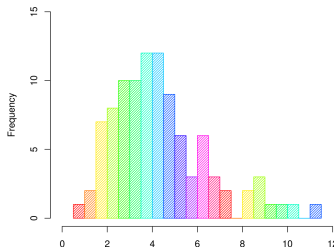
In this type of query the universe $\mathbb{N}^{|\mathcal{X}|}$ is **partitioned into cells**, and the query asks how many database elements lie in each of the cells.



Histogram Queries

In this type of query the universe $\mathbb{N}^{|\mathcal{X}|}$ is **partitioned into cells**, and the query asks how many database elements lie in each of the cells.

- Because the cells are disjoint, the addition or removal of a single database element can affect the count in exactly one cell. Hence the **sensitivity is 1**.
- $(\epsilon, 0)$ -differential privacy can be achieved by adding noise scaled to $1/\epsilon$, that is, by adding noise drawn from $\text{Lap}(1/\epsilon)$ to the true count in each cell.



The Gaussian Mechanism

Let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^d$ be an arbitrary d -dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{\text{adjacent } x, y} \|f(x) - f(y)\|_2$. The *Gaussian Mechanism with parameter σ* adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the d components of the output.

Theorem A.1. Let $\varepsilon \in (0, 1)$ be arbitrary. For $c^2 > 2 \ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c \Delta_2 f / \varepsilon$ is (ε, δ) -differentially private.

Differential Privacy Properties

Differential Privacy (DP)

Good properties of DP

- Robustness to **auxiliary information**
 - Privacy guarantees are not affected by any side information available to the adversary
 - Independent of **adversary's computational power**

Differential Privacy (DP)

Good properties of DP

- Robustness to **auxiliary information**
 - Privacy guarantees are not affected by any side information available to the adversary
 - Independent of **adversary's computational power**
- Group privacy
 - Graceful degradation of privacy guarantees if datasets contain correlated inputs

Differential Privacy (DP)

Good properties of DP

■ Robustness to **auxiliary information**

- Privacy guarantees are not affected by any side information available to the adversary
- Independent of **adversary's computational power**

■ Group privacy

- Graceful degradation of privacy guarantees if datasets contain correlated inputs

■ Composability

- Enables modular design of mechanisms: if all the components of a mechanism are differentially private, then so is their composition.

Post Processing

Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ be a randomized algorithm that is (ϵ, δ) -differential private. Let $f : R \rightarrow R'$ be a deterministic function. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is (ϵ, δ) -differential private.

Proof:

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$. Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(x)) \in S] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in T] + \delta \\ &= \exp(\epsilon) \Pr[f(\mathcal{M}(y)) \in S] + \delta \end{aligned}$$

which was what we wanted. □

f can be generalized to arbitrary randomized mapping.

Utility Theoretic View

Consider an individual i who has arbitrary **preferences over** the set of all possible **future events**, which we denote by \mathcal{A} .

- These preferences are expressed by a **utility function** $u_i : \mathcal{A} \rightarrow R_{\geq 0}$, and we say that individual i experiences utility $u_i(a)$ in the event that $a \in \mathcal{A}$ comes to pass.

Utility Theoretic View

Consider an individual i who has arbitrary **preferences over** the set of all possible **future events**, which we denote by \mathcal{A} .

- These preferences are expressed by a **utility function** $u_i : \mathcal{A} \rightarrow R_{\geq 0}$, and we say that individual i experiences utility $u_i(a)$ in the event that $a \in \mathcal{A}$ comes to pass.

Let $f : \text{Range}(\mathcal{M}) \rightarrow \Delta(\mathcal{A})$ be the (arbitrary) function that determines the distribution over future events \mathcal{A} , conditioned on the output of mechanism \mathcal{M} .

we have:

$$\begin{aligned} \mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] &= \sum_{a \in \mathcal{A}} u_i(a) \cdot \Pr_{f(\mathcal{M}(x))}[a] \\ &\leq \sum_{a \in \mathcal{A}} u_i(a) \cdot \exp(\varepsilon) \Pr_{f(\mathcal{M}(y))}[a] \\ &= \exp(\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)] \end{aligned}$$

Similarly,

$$\mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] \geq \exp(-\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)].$$

Utility Theoretic View

Consider an individual i who has arbitrary **preferences over** the set of all possible **future events**, which we denote by \mathcal{A} .

- These preferences are expressed by a **utility function** $u_i : \mathcal{A} \rightarrow R_{\geq 0}$, and we say that individual i experiences utility $u_i(a)$ in the event that $a \in \mathcal{A}$ comes to pass.

Let $f : \text{Range}(\mathcal{M}) \rightarrow \Delta(\mathcal{A})$ be the (arbitrary) function that determines the distribution over future events \mathcal{A} , conditioned on the output of mechanism \mathcal{M} .

we have:

$$\begin{aligned} \mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] &= \sum_{a \in \mathcal{A}} u_i(a) \cdot \Pr_{f(\mathcal{M}(x))}[a] \\ &\leq \sum_{a \in \mathcal{A}} u_i(a) \cdot \exp(\varepsilon) \Pr_{f(\mathcal{M}(y))}[a] \\ &= \exp(\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)] \end{aligned}$$

Similarly,

$$\mathbb{E}_{a \sim f(\mathcal{M}(x))}[u_i(a)] \geq \exp(-\varepsilon) \mathbb{E}_{a \sim f(\mathcal{M}(y))}[u_i(a)].$$

Hence, by promising a guarantee of ϵ -differential privacy, **a data analyst can promise an individual that his expected future utility will not be harmed by more than an $\exp(\epsilon) \approx (1 + \epsilon)$ factor.**

Group Privacy

Theorem 7. Let $M : \mathcal{X}^n \rightarrow \mathcal{Y}$ be an ε -differentially private algorithm. Suppose X and X' are two datasets which differ in exactly k positions. Then for all $T \subseteq \mathcal{Y}$, we have

$$\Pr[M(X) \in T] \leq \exp(k\varepsilon) \Pr[M(X') \in T].$$

Proof.

Let $X^{(0)} = X, X^{(k)} = X'$ – since they differ in k positions, there exists a sequence $X^{(0)}$ through $X^{(k)}$ such that each consecutive pair of datasets is neighbouring. Then, for all $T \subseteq \mathcal{Y}$:

$$\begin{aligned} \Pr[M(X^{(0)}) \in T] &\leq e^\varepsilon \Pr[M(X^{(1)}) \in T] \\ &\leq e^{2\varepsilon} \Pr[M(X^{(2)}) \in T] \\ &\dots \\ &\leq e^{k\varepsilon} \Pr[M(X^{(k)}) \in T]. \end{aligned}$$

□

Basic Composition

Suppose **k differentially private algorithms** are run on the **same dataset**, and released all of their results. results

- How private is this as a whole? Essentially, the overall privacy guarantee **decays by a factor of k** .

Theorem 8. Suppose $M = (M_1, \dots, M_k)$ is a sequence of ε -differentially private algorithms, potentially chosen sequentially and adaptively. Then M is $k\varepsilon$ -differentially private.

Proof. Fix two neighbouring datasets X and X' , and consider some sequence of outputs $y = (y_1, \dots, y_k)$. Then we have

$$\begin{aligned} \frac{\Pr[M(X) = y]}{\Pr[M(X') = y]} &= \prod_{i=1}^k \frac{\Pr[M_i(X) = y_i | (M_1(X), \dots, M_{i-1}(X)) = (y_1, \dots, y_{i-1})]}{\Pr[M_i(X') = y_i | (M_1(X'), \dots, M_{i-1}(X')) = (y_1, \dots, y_{i-1})]} \\ &\leq \prod_{i=1}^k \exp(\varepsilon) \\ &= \exp(k\varepsilon). \end{aligned}$$

□

Advanced Composition

Theorem 3.20 (Advanced Composition). For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ) -differentially private mechanisms satisfies $(\varepsilon', k\delta + \delta')$ -differential privacy under k -fold adaptive composition for:

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \varepsilon + k\varepsilon(e^\varepsilon - 1).$$

Differentially Private SGD

Deep Learning with Differential Privacy

A preliminary version of this paper appears in the proceedings of the *23rd ACM Conference on Computer and Communications Security (CCS 2016)*. This is a full version.

Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi*
H. Brendan McMahan*

Andy Chu*
Ilya Mironov*
Li Zhang*

Ian Goodfellow†
Kunal Talwar*

Abstract

The models should **not expose private information** in the training datasets.

- It combines state-of-the-art machine learning methods with advanced privacy-preserving mechanisms, training neural networks within a modest **single-digit** privacy budget.

Threat Model

The approach offers protection against a strong adversary with full knowledge of the training mechanism and **access to the model's parameters**.

- This protection is attractive, in particular, for applications of machine learning on mobile phones, tablets, and other devices.

Differential Privacy (DP)

Differential privacy constitutes a strong standard for privacy guarantees for algorithms on databases.

- It is defined in terms of the application-specific concept of adjacent databases.
- In the experiments, each training dataset is a set of image-label pairs.

Two datasets are **adjacent if they differ in a single entry**, that is, if one **image-label pair** is present in one set and absent in the other.

Differential Privacy (DP)

Good properties of DP

- **Composability**
 - Enables modular design of mechanisms: if all the components of a mechanism are differentially private, then so is their composition.
- **Group privacy**
 - Graceful degradation of privacy guarantees if datasets contain correlated inputs
- **Robustness to auxiliary information**
 - Privacy guarantees are not affected by any side information available to the adversary

Gaussian Mechanism

A deterministic real-valued function $f : \mathcal{D} \rightarrow \mathbb{R}$

- Sensitivity: $S_f = \| f(d) - f(d') \|_2$ where d and d' are adjacent inputs.

Gaussian Mechanism

A deterministic real-valued function $f : \mathcal{D} \rightarrow \mathbb{R}$

- Sensitivity: $S_f = \|f(d) - f(d')\|_2$ where d and d' are adjacent inputs.

Gaussian mechanism

The Gaussian mechanism is defined by

$$\mathcal{M}(d) = f(d) + \mathcal{N}(0, S_f^2 \sigma^2)$$

where $\mathcal{N}(0, S_f^2 \sigma^2)$ is the normal (Gaussian) distribution with mean 0 and standard deviation $S_f \sigma$.

- It satisfies (ϵ, δ) -differential privacy if $\delta \geq \frac{4}{5} \exp\left(\frac{-(\sigma\epsilon)^2}{2}\right)$ and $\epsilon < 1$

Designing a Mechanism

The basic blueprint for **designing a differentially private additive-noise mechanism** that implements a given functionality consists of the following steps:

- 1 **Approximating the functionality** by a **sequential composition of bounded-sensitivity functions**
- 2 **Choosing parameters** of additive noise
- 3 Performing **privacy analysis of the resulting mechanism**.

Gradient Norm Clipping

Proving the differential privacy guarantee of Algorithm 1 requires **bounding the influence** (sensitivity) of each individual example on $\vec{\mathbf{g}}_t$

- Since there is no a priori bound on the size of the gradients, we clip each gradient in ℓ_2 norm

$$\vec{\mathbf{g}}_t = \mathbf{g}_t / \max(1, \frac{\|\mathbf{g}_t\|_2}{C})$$

Differentially Private SGD Algorithm

It is aimed to control the influence of the training data during the training process in the SGD computation.

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \dots, x_N\}$, 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

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

Add noise

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

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.

Lots

Each lot is formed by independently picking each example with probability $q = L/N$, where N is the size of the input dataset.

- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.

Lots

Each lot is formed by independently picking each example with probability $q = L/N$, where N is the size of the input dataset.

- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.

Implementation

- In order to limit memory consumption, we may set the batch size much smaller than the lot size L , which is a parameter of the algorithm. We perform the computation in batches, then **group several batches into a lot for adding noise**.
 - In practice, for efficiency, the construction of batches and lots is done by randomly permuting the examples and then partitioning them into groups of the appropriate sizes.

Privacy accounting

For differentially private SGD, an important issue is **computing the overall privacy cost of the training**.

- The **composability** of differential privacy allows us to implement an **accountant** procedure that computes the **privacy cost at each access to the training data**, and **accumulates this cost as the training progresses**.

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.
- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.
- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.
- Strong composition theorem prove that Algorithm 1 is $(O(q\epsilon\sqrt{T \log(1/\delta)}), Tq\delta)$ -differentially private.

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.
- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.
- Strong composition theorem prove that Algorithm 1 is $(O(q\epsilon\sqrt{T \log(1/\delta)}), Tq\delta)$ -differentially private.
 - The strong composition theorem can be loose, and does not take into account the particular noise distribution under consideration.

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.
- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.
- Strong composition theorem prove that Algorithm 1 is $(O(q\epsilon\sqrt{T\log(1/\delta)}), Tq\delta)$ -differentially private.
 - The strong composition theorem can be loose, and does not take into account the particular noise distribution under consideration.
- The authors propose **Moments Accountant** to prove that Algorithm 1 is $(O(q\epsilon\sqrt{T}), \delta)$ -differentially private for appropriately chosen settings of the noise scale and the clipping threshold.
 - it saves a $\sqrt{\log(1/\delta)}$ factor in the ϵ part and a Tq factor in the δ part ($T \gg 1/q$).

Moments Accountant

- If $\sigma = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\epsilon}$, then by standard arguments each step is (ϵ, δ) -differentially private with respect to the lot.
- Since the lot itself is a random sample from the database, the **privacy amplification theorem** implies that each step is $(O(q\epsilon), q\delta)$ -differentially private with respect to the full database where $q = L/N$ is the sampling ratio per lot and $\epsilon \leq 1$.
- Strong composition theorem prove that Algorithm 1 is $(O(q\epsilon\sqrt{T\log(1/\delta)}), Tq\delta)$ -differentially private.
 - The strong composition theorem can be loose, and does not take into account the particular noise distribution under consideration.
- The authors propose **Moments Accountant** to prove that Algorithm 1 is $(O(q\epsilon\sqrt{T}), \delta)$ -differentially private for appropriately chosen settings of the noise scale and the clipping threshold.
 - it saves a $\sqrt{\log(1/\delta)}$ factor in the ϵ part and a Tq factor in the δ part ($T \gg 1/q$).
- Algorithm 1 is (ϵ, δ) -differentially private for any $\delta > 0$ and $\epsilon < c_1 q^2 T$ (c_1, c_2 are constants) if

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\epsilon}$$

- ϵ is inversely related to σ (privacy-utility trade-off).

Strong Composition Theorem vs. Moments Accountant

Privacy analysis

- For example, with $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$ and $T = 10000$, we have $\epsilon \approx 1.26$ using the moments accountant.
- As a comparison, we would get a much larger $\epsilon \approx 9.34$ using the strong composition theorem.

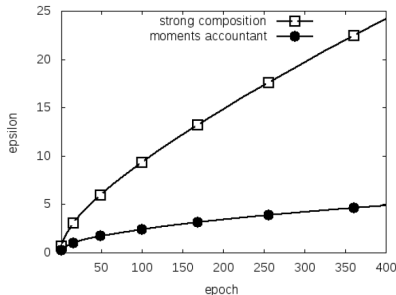


Figure 2: The ϵ value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.

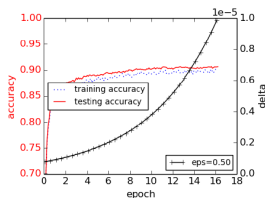
MNIST

Target model

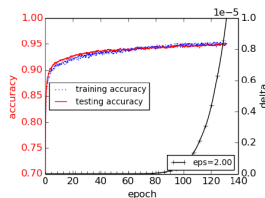
- A simple feedforward neural network with ReLU units and softmax of 10 classes (corresponding to the 10 digits) with cross-entropy loss.
- A single hidden layer with 1000 hidden units.
- Using the lot size of 600, it reaches accuracy of 98.30% in about 100 epochs.

Privacy parameters

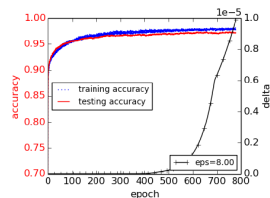
- $C = 4$, $\sigma = 2, 4, 8$, and $\delta = 10^{-5}$
- $\epsilon \in \{0.5, 2.0, 8.0\}$



(1) Large noise



(2) Medium noise



(3) Small noise

Accuracy vs. (ϵ, δ)

for a fixed δ

- Varying the value of ϵ can have large impact on accuracy

for any fixed ϵ

- There is less difference with different δ values.

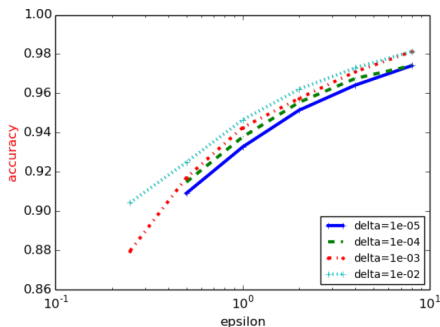
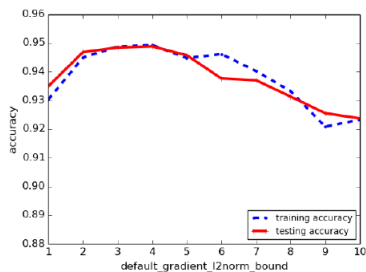


Figure 4: Accuracy of various (ϵ, δ) privacy values on the MNIST dataset. Each curve corresponds to a different δ value.

Clipping Bound

Opposing effects

- Clipping destroys the unbiasedness of the gradient estimate
- Increasing the norm bound C forces us to add more noise to the gradients (and hence the parameters), since we add noise based on σC



(5) variable gradient clipping norm

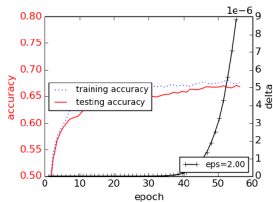
CIFAR10

Target model

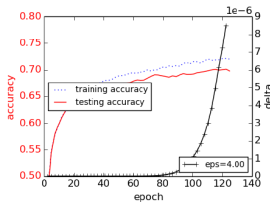
- Each 32×32 image is first cropped to a 24×24 one by taking the center patch.
- The network architecture consists of two convolutional layers followed by two fully connected layers.

Privacy parameters

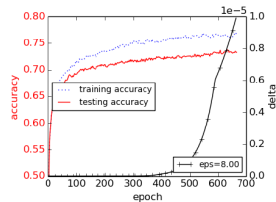
- This architecture, non-privately, can get to about 80% accuracy in 500 epochs.
- $\epsilon \in \{2.0, 4.0, 8.0\}$



(1) $\epsilon = 2$



(2) $\epsilon = 4$



(3) $\epsilon = 8$

PATE

Private Aggregation of Teacher Ensembles (PATE)

Published as a conference paper at ICLR 2017

SEMI-SUPERVISED KNOWLEDGE TRANSFER FOR DEEP LEARNING FROM PRIVATE TRAINING DATA

Nicolas Papernot*

Pennsylvania State University
ngp5056@cse.psu.edu

Martín Abadi

Google Brain
abadi@google.com

Úlfar Erlingsson

Google
ulfar@google.com

Ian Goodfellow

Google Brain[†]
goodfellow@google.com

Kunal Talwar

Google Brain
kunal@google.com

Abstract

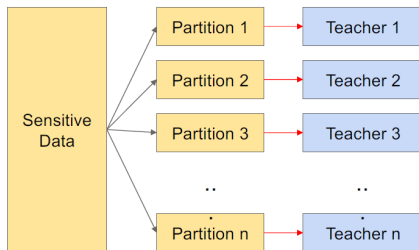
Multiple teacher models trained with **disjoint datasets**, they are used to **privately train a student model**.

- The student learns to predict an output chosen by **noisy voting** among all of the teachers, and **cannot directly access an individual teacher or the underlying data or parameters**.
- The student's privacy properties can be understood both intuitively and formally, in terms of **differential privacy**.

Private Aggregation of Teacher Ensembles (PATE)

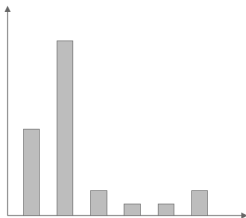
Instead of training a single model to solve the task associated with dataset (X, Y) , where X denotes the set of inputs, and Y the set of labels,

- We partition the data in n disjoint sets (X_n, Y_n) and train a model f_i separately on each set.



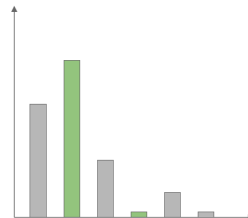
(Papernot, 2019)

Private Aggregation of Teacher Ensembles (PATE)



Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$



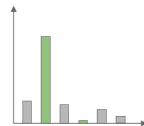
Take maximum

$$f(x) = \arg \max_j \left\{ n_j(\vec{x}) \right\}$$

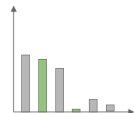
(Papernot, 2019)

Private Aggregation of Teacher Ensembles (PATE)

If most teachers agree on the label, it does not depend on specific partitions, so the privacy cost is small.

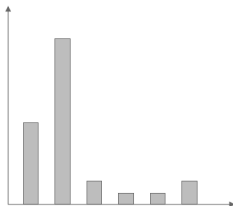


If two classes have close vote counts, the disagreement may reveal private information.



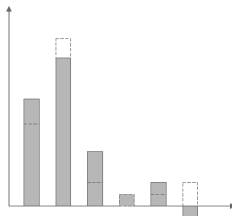
(Papernot, 2019)

Private Aggregation of Teacher Ensembles (PATE)



Count
votes

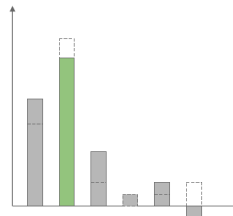
$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$



Add Laplacian

$$\text{Lap}\left(\frac{1}{\varepsilon}\right)$$

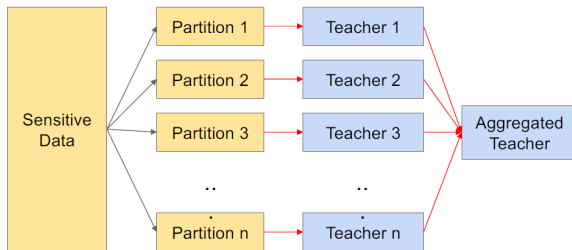
(Papernot, 2019)



Take maximum

$$f(x) = \arg \max_j \left\{ n_j(\vec{x}) + \text{Lap}\left(\frac{1}{\varepsilon}\right) \right\}$$

Private Aggregation of Teacher Ensembles (PATE)



(Papernot, 2019)

Private Aggregation of Teacher Ensembles (PATE)

The label count for a given class $j \in [m]$ and an input x is the number of teachers that assigned class j to input x

$$n_j(x) = |\{i : i \in [n], f_i(x) = j\}|$$

where m be the number of classes in the task.

If we simply apply plurality (use the label with the largest count) the ensemble's decision may **depend on a single teacher's vote**.

Private Aggregation of Teacher Ensembles (PATE)

The label count for a given class $j \in [m]$ and an input x is the number of teachers that assigned class j to input x

$$n_j(x) = |\{i : i \in [n], f_i(x) = j\}|$$

where m be the number of classes in the task.

If we simply apply plurality (use the label with the largest count) the ensemble's decision may **depend on a single teacher's vote**.

Laplace mechanism

- We add random noise to the vote counts n_j to introduce ambiguity:

$$f(x) = \operatorname{argmax}_j \{n_j(x) + \operatorname{Lap}(\frac{1}{\epsilon})\}$$

In this equation, ϵ is a privacy parameter and $\operatorname{Lap}(b)$ the Laplacian distribution with location 0 and scale b .

Private Aggregation of Teacher Ensembles (PATE)

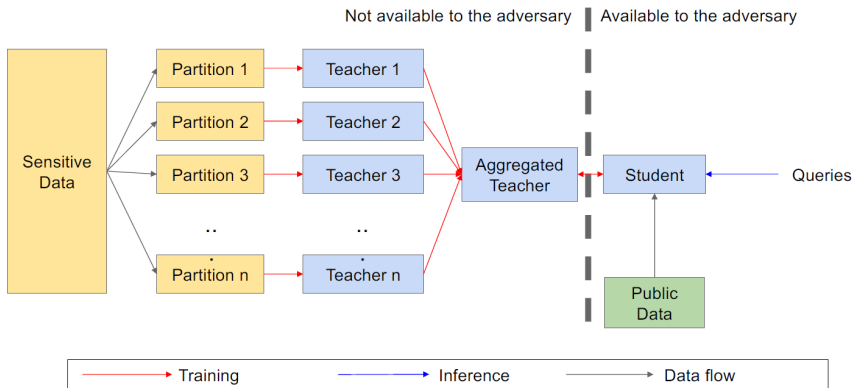
While we could use an f to make predictions, the **noise required would increase as we make more predictions, due to Composition**, making the model useless after a bounded number of queries.

Also, privacy guarantees do not hold when an adversary has access to **the teacher models parameters**

To address these limitations

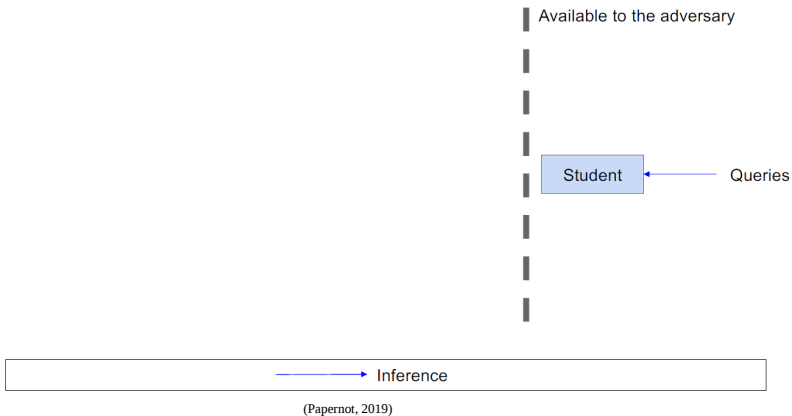
- We train another model, **the student**, using a **fixed number** of labels predicted by the teacher ensemble.
- The student is trained non-privately on **public dataset** and noisy labels.

Private Aggregation of Teacher Ensembles (PATE)



(Papernot, 2019)

Private Aggregation of Teacher Ensembles (PATE)



Semi-supervised Transfer of the Knowledge From an Ensemble to a Student

We train a student on **nonsensitive and unlabeled data, some of which we label using the aggregation mechanism.**

This student model is the one deployed, instead of the teacher ensemble

- Thus, the privacy loss **does not grow with the number of user queries** made to the student model.
- Indeed, the privacy loss is now determined by the number of queries made to the teacher ensemble during student training and does not increase as end-users query the deployed student model.

Several techniques are considered to trade-off **the student model's quality with the number of labels** it needs to access: distillation, active learning, semi-supervised learning

- **PATE-G**: semi-supervised learning with GANs.

Privacy Analysis of the Approach

A natural way to bound our approach's privacy loss is to **first bound the privacy cost of each label queried** by the student, and then use the **strong composition** theorem to derive the total cost of training the student.

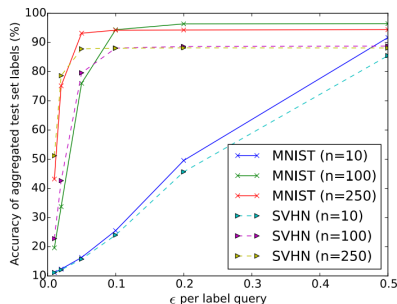
- The Moments Accountant

Bound is data-dependent: computed using the empirical quorum

Evaluation

Train ensembles of **250 teachers** for MNIST and SVHN datasets.

- The aggregation mechanism output has an accuracy of 93.18% for MNIST and 87.79% for SVHN, when evaluated on their respective test sets, while each query has a low privacy budget of $\epsilon = 0.05$.
- The number n of teachers is limited by a **trade-off**. Larger values of n lead to larger absolute gaps, hence potentially allowing for a larger noise level and stronger privacy guarantees. At the same time, a larger n implies a smaller training dataset for each teacher, potentially reducing the teacher accuracy.



Small values of ϵ on the left of the axis correspond to large noise amplitudes and large ϵ values on the right to small noise.

Evaluation

- Abadi et al. (2016) previously obtained 97% accuracy with a ($\epsilon = 8, \delta = 10^{-5}$) bound on MNIST.

Dataset	ϵ	δ	Queries	Non-Private Baseline	Student Accuracy
MNIST	2.04	10^{-5}	100	99.18%	98.00%
MNIST	8.03	10^{-5}	1000	99.18%	98.10%
SVHN	5.04	10^{-6}	500	92.80%	82.72%
SVHN	8.19	10^{-6}	1000	92.80%	90.66%

Figure 4: **Utility and privacy of the semi-supervised students:** each row is a variant of the student model trained with generative adversarial networks in a semi-supervised way, with a different number of label queries made to the teachers through the noisy aggregation mechanism. The last column reports the accuracy of the student and the second and third column the bound ϵ and failure probability δ of the (ϵ, δ) differential privacy guarantee.

Differential Private Deep Learning

Data	ϵ -DP	Source	Test Accuracy (%)		
			CNN	ScatterNet+linear	ScatterNet+CNN
MNIST	1.2	Feldman & Zrnic (2020)	<u>96.6</u>	98.1 \pm 0.1	97.8 \pm 0.1
	2.0	Abadi et al. (2016)	95.0	98.5 \pm 0.0	98.4 \pm 0.1
	2.32	Bu et al. (2019)	96.6	98.6 \pm 0.0	98.5 \pm 0.0
	2.5	Chen & Lee (2020)	90.0	98.7 \pm 0.0	98.6 \pm 0.0
	2.93	Papernot et al. (2020a)	<u>98.1</u>	98.7 \pm 0.0	98.7 \pm 0.1
	3.2	Nasr et al. (2020)	96.1	–	–
	6.78	Yu et al. (2019b)	93.2	–	–
Fashion-MNIST	2.7	Papernot et al. (2020a)	<u>86.1</u>	89.5 \pm 0.0	88.7 \pm 0.1
	3.0	Chen & Lee (2020)	82.3	89.7 \pm 0.0	89.0 \pm 0.1
CIFAR-10	3.0	Nasr et al. (2020)	<u>55.0</u>	67.0 \pm 0.1	69.3 \pm 0.2
	6.78	Yu et al. (2019b)	44.3	–	–
	7.53	Papernot et al. (2020a)	<u>66.2</u>	–	–
	8.0	Chen & Lee (2020)	53.0	–	–

(Tramèr and Boneh, ICLR 2021) ($\delta = 10^{-5}$)

References

- Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, (Ch. 3).
- Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K. and Zhang, L., 2016, October. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308-318).
- Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I. and Talwar, K., 2016. Semi-supervised knowledge transfer for deep learning from private training data. arXiv preprint arXiv:1610.05755.
- Gautam Kamath, CS 860, Fall 2020, University of Waterloo

:)

