Privacy in NLP

Deep Learning for NLP: Lecture 11

Timour Igamberdiev

June 2022

Importance of Privacy

Importance of Privacy

Question: Why is privacy important?

Two ways to answer:

1. Societal perspective
2. Research perspective

Today's world: "Data is the new oil"

- Ethical concerns over data collection
- Legal concerns for businesses due to laws and privacy guidelines

Why privacy is important: Research Perspective

Very difficult to convince data holders to provide data (e.g. hospital medical records)

Can ‘pip install’ MNIST dataset and train a classifier in minutes

Cannot ‘pip install cancer-dataset’, need a lot of work/resources to get hold of such data

Overview of Lecture

- Why is privacy important and consequences of non-privacy
- Gold standard of privacy: Differential privacy
  - Randomized response
  - Pure differential privacy and the Laplace mechanism
  - Properties of differential privacy
  - Approximate differential privacy and the Gaussian mechanism
- Applying differential privacy for ML: DP-SGD
- Other methods in privacy
  - Secure multiparty computation
  - Federated learning
  - Homomorphic encryption

Attacks on Non-Privatized Data and Models

Data Anonymization

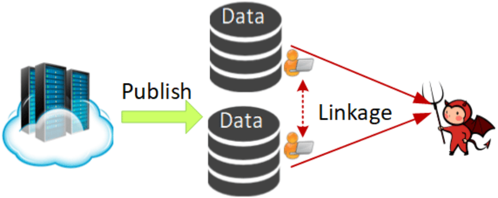
What if we just anonymize data?

Example

"Robert Smith" $\rightarrow$ "id38729848"

Linkage Attack

Re-identifying anonymized individuals by combining data with background information



Linkage Attack: Adversary acquires private information by correlating multiple datasets

https://www.researchgate.net/figure/Different-privacy-attack-and-threat-models_fig5_346302647

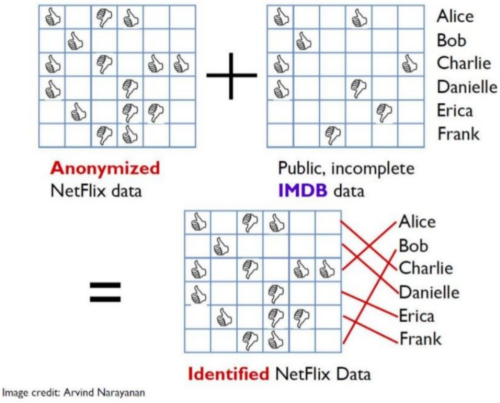
Consequences of Non-Privacy: Netflix Prize

Netflix: Online streaming service

Netflix prize: Challenge between 2006 and 2009, prize of $1,000,000

- **Goal**: Create a model for best recommendations of their service
- **Data**: Anonymized user IDs, movie IDs, ratings and dates

Privacy breach: Match Netflix data (anonymized) with IMDb data (public)

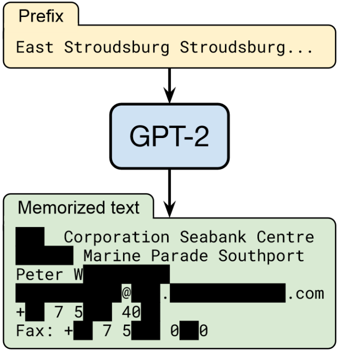


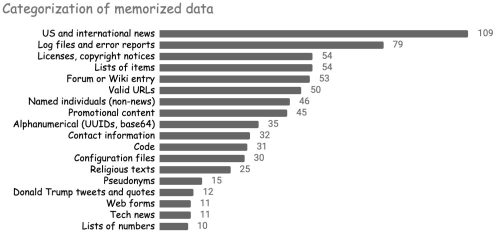
Consequences of Non-Privacy: Memorization in Neural Networks

Let's look at a **black-box** attack on extracting data from NNs

Neural networks can memorize their training data [Carlini et al., 2021]

We can extract this in multiple ways, one of which is **prompting**



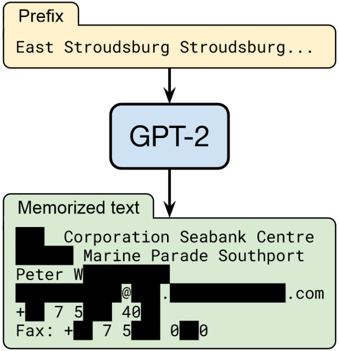


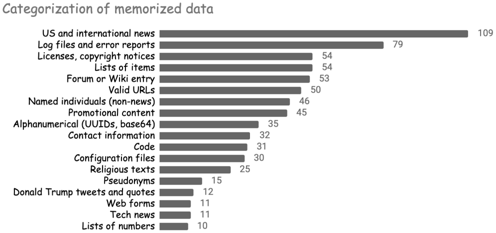
Consequences of Non-Privacy: Memorization in Neural Networks

Let's look at a **black-box** attack on extracting data from NNs

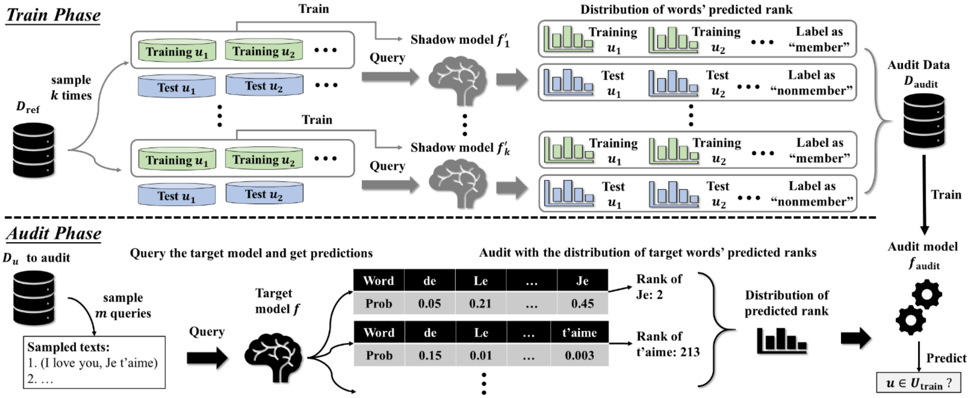
Neural networks can memorize their training data [Carlini et al., 2021]

We can extract this in multiple ways, one of which is **prompting**





Consequences of Non-Privacy: Membership Inference



Song and Shmatikov [2019]

Consequences of Non-Privacy: Model inversion for NNs

Model inversion [Fredrikson et al., 2015], an example of a white-box attack

Basic idea: Follow the gradient used to adjust the weights of a model, obtain a reverse-engineered example for all represented classes in the model

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Differential Privacy

Differential Privacy

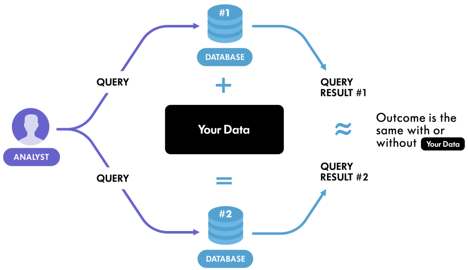
The *gold standard* in leading research with privacy guarantees

Intuitively

Data perturbation, output of algorithm cannot change beyond a *very specific amount*, when *one data point is added/removed*

Won the Test of Time Award back in 2016

Used by big companies like Microsoft, Apple, Google and Facebook



https://www.winton.com/research/using-differential-privacy-to-protect-personal-data

Differential Privacy

Randomized Response

Differential Privacy: Randomized Response

Oldest DP algorithm (Warner, 1965)

Technique used to collect sensitive information from individuals, while maintaining their confidentiality

Provides **plausible deniability**

Randomized Response

More formally:

n students

$X_i \in \{0, 1\}$: Did individual i cheat on test?

Y_i: Value that depends on X_i with added randomness

Goal of analyst: Estimate $p = \frac{1}{n} \sum X_i$ (fraction of individuals that cheated)

Method 1: Perfect accuracy, no privacy

$$Y_i = \begin{cases} X_i & \text{w.p. } 1 \\ 1 - X_i & \text{w.p. } 0 \end{cases}$$

Method 2: Perfect privacy, no accuracy

$$Y_i = \begin{cases} X_i & \text{w.p. } \frac{1}{2} \\ 1 - X_i & \text{w.p. } \frac{1}{2} \end{cases}$$

Randomized Response

More formally:

n students

$X_i \in \{0, 1\}$: Did individual i cheat on test?

Y_i: Value that depends on X_i with added randomness

Goal of analyst: Estimate $p = \frac{1}{n} \sum X_i$ (fraction of individuals that cheated)

Method 1: Perfect accuracy, no privacy

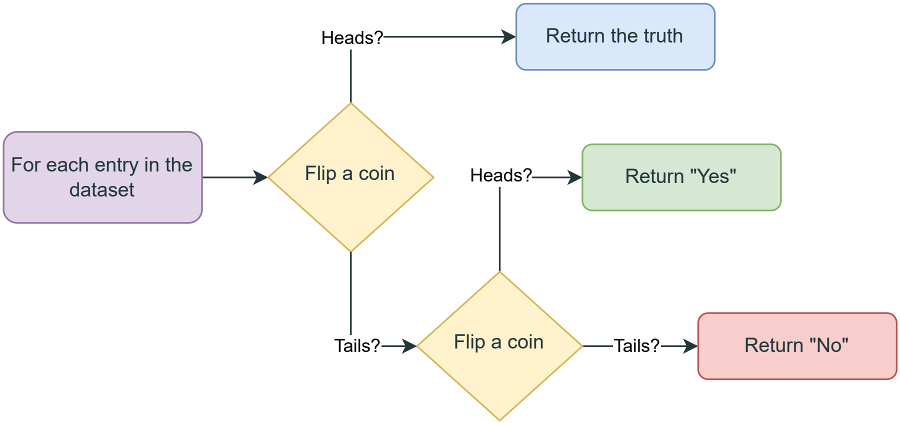
$$Y_i = \begin{cases} X_i & \text{w.p. } 1 \\ 1 - X_i & \text{w.p. } 0 \end{cases}$$

Method 2: Perfect privacy, no accuracy

$$Y_i = \begin{cases} X_i & \text{w.p. } \frac{1}{2} \\ 1 - X_i & \text{w.p. } \frac{1}{2} \end{cases}$$

Randomized Response

Method 3



Based on: https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

Randomized Response

More formally

New parameter $\gamma \in (0, \frac{1}{2})$

$$Y_i = \begin{cases} X_i & \text{w.p. } \frac{1}{2} + \gamma \\ 1 - X_i & \text{w.p. } \frac{1}{2} - \gamma \end{cases}$$

If $\gamma = \frac{1}{2}$, this is Method 1 (no privacy, perfect accuracy)

If $\gamma = 0$, this is Method 2 (no accuracy, perfect privacy)

Compromise

Set $\gamma = \frac{1}{4}$ — provides plausible deniability

$\gamma \to 0$, maximum deniability

$\gamma \to \frac{1}{2}$, no deniability (no privacy)

Differential Privacy

Pure Differential Privacy

Pure Differential Privacy

Differential Privacy (DP)

Data perturbation, where the output of an algorithm cannot change by more than a **specific amount**, when adding/removing/altering one data point in a dataset.

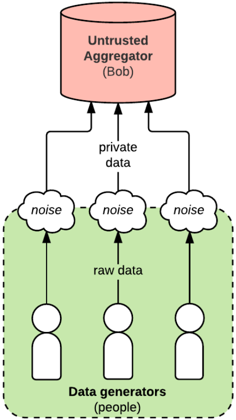
A **property** of an algorithm, information-theoretic guarantee.

Originally proposed by Dwork et al. [2006], extensively outlined in Dwork and Roth [2013]

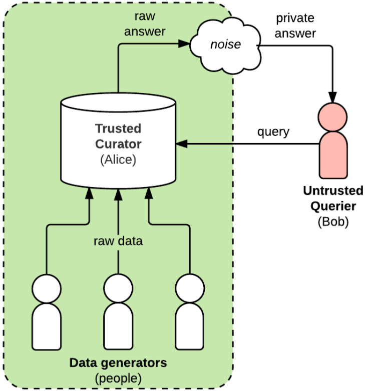
Privacy Budget (ε)

The total amount of privacy leakage that is allowed to occur (‘amount’ of privacy).

Pure Differential Privacy



Local privacy



Global privacy

https://www.accessnow.org/understanding-differential-privacy-matters-digital-rights/

Pure Differential Privacy

Additional relevant terms:

Query

The ‘question’ the analyst is asking about the data.

E.g. Mean, sum (simple); gradient of loss function (more complex).

Trusted curator

The aggregator of the data, adds noise to achieve DP guarantee.

Sensitivity

The maximum difference in an algorithm’s outputs, when one data point is changed.

Pure DP: Concrete Example

Sum query

Dataset of binary values

Contains n individuals, each associated with 0 or 1

Query: How many people in our dataset smoke?

Name	Smokes?
Alice	0
Bob	0
Clair	1
$\vdots$	$\vdots$
Zane	1

Pure DP: Concrete Example

Sum query

Dataset of binary values

Contains n individuals, each associated with 0 or 1

Query: How many people in our dataset smoke?

Name	Smokes?
Alice	0
Bob	0
Clair	1
$\vdots$	$\vdots$
Zane	1

Sensitivity?

Pure DP: Concrete Example

Sum query

Dataset of binary values

Contains n individuals, each associated with 0 or 1

Query: How many people in our dataset smoke?

Name	Smokes?
Alice	0
Bob	0
Clair	1
$\vdots$	$\vdots$
Zane	1

Sensitivity? — 1

Pure DP: More Formally

Dataset D, consisting of n individuals, $x_1, \ldots, x_n$

We run mechanism M over this dataset D to get an output $M(D)$

For $\varepsilon \geq 0$, *mechanism* M *is* ε*-differentially private if*, for all $S \subseteq \text{Range}(M)$, and **neighboring datasets** D and D':

Differential Privacy

$$Pr[M(D) \in S] \leq \exp(\varepsilon) Pr[M(D') \in S]$$

Neighboring datasets

A dataset is **neighboring** to another dataset if it differs from it in one row. I.e. $||D - D'||_1 \leq 1$

Pure DP: More Formally

Dataset D, consisting of n individuals, $x_1, \ldots, x_n$

We run mechanism M over this dataset D to get an output $M(D)$

For $\varepsilon \geq 0$, *mechanism* M *is* ε*-differentially private if*, for all $S \subseteq \text{Range}(M)$, and **neighboring datasets** D and D':

Differential Privacy

$$Pr[M(D) \in S] \leq \exp(\varepsilon) Pr[M(D') \in S]$$

Neighboring datasets

A dataset is **neighboring** to another dataset if it differs from it in one row. I.e. $||D - D'||_1 \leq 1$

Pure DP: More Formally

$\varepsilon \to 0$: we approach perfect privacy, but less utility of our algorithm (less difference in output distributions)

$\varepsilon \to \infty$: we approach the original non-DP setting (no constraint on output distributions)

Pure DP: More Formally

How big do we make ε?

Should be 'fairly small', generally $0.1 \leq \varepsilon \leq 5$

Randomized response mechanism with $\gamma = \frac{1}{4}$ (throwing a fair coin):
$\varepsilon = \ln 3 \approx 1.1$

As ε increases, privacy guarantee gets exponentially worse

Achieving Pure DP: The Laplace Mechanism

How do we achieve this ε-DP guarantee? — **Laplace Mechanism**

Sensitivity revisited

$f : D^n \rightarrow \mathbb{R}^k$

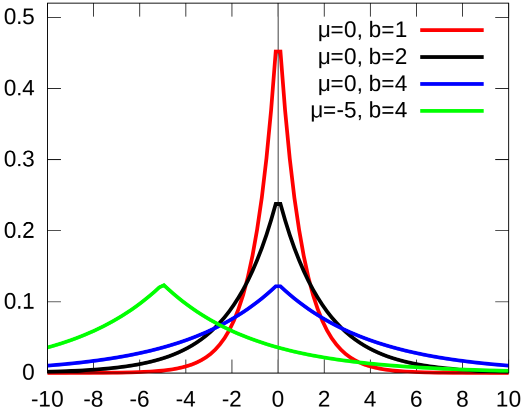
l_1-sensitivity of the function f is: $\Delta^{(f)} = \max_{D,D'} ||f(D) - f(D')||_1$

Example Calculation: Sum query

If f sums up a set of bits, $\sum X_i$, where $X_i \in \{0, 1\}$, then: $\Delta^{(f)} = 1$

Achieving Pure DP: The Laplace Mechanism

Laplace Distribution: $p(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$



Achieving Pure DP: The Laplace Mechanism

Laplace Mechanism

$f : D^n \to \mathbb{R}^k$

$M(D) = f(D) + (Y_1, \ldots, Y_k)$

$Y_i \underset{i.i.d}{\sim} \text{Lap}(\frac{\Delta^{(f)}}{\varepsilon})$

Add noise to each coordinate, proportional to the L1 sensitivity

Example: Sum query

$f = \sum X_i,\ \Delta^{(f)} = 1,\ k = 1$

$\tilde{p} = f(x) + \text{Lap}(\frac{1}{\varepsilon})$

Name	Smokes?
Alice	0
Bob	0
Clair	1
⋮	⋮
Zane	1

Privacy Loss Random Variable

For mechanism $M : D^n \to Y$ and neighboring datasets D, D', we can define the **privacy loss random variable** $\mathcal{L}_{M(D)||M(D')}$

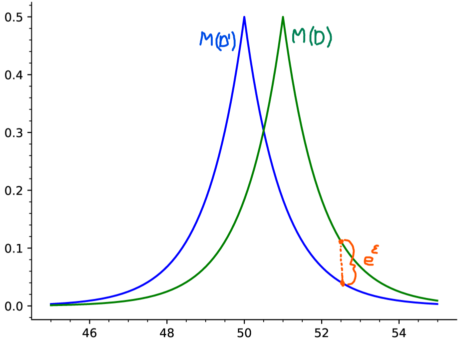
Privacy Loss Random Variable

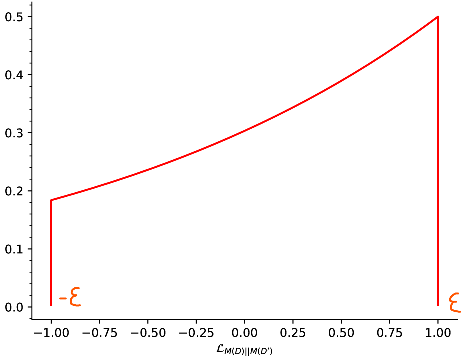
$\mathcal{L}_{M(D)||M(D')} = \ln\left(\frac{M(D)=\xi}{M(D')=\xi}\right)$, distributed by drawing $\xi \sim M(D)$

Privacy Loss Random Variable

We can translate between this privacy loss random variable and our ε-DP:

$$\left|\mathcal{L}_{M(D)||M(D')}\right| \leq \varepsilon \text{ w.p. } 1, \text{ for all } D \text{ and } D' \text{ neighboring datasets}$$





Properties of DP

Properties of DP

1. Closed under post-processing
    - If $M : D^n \to Y$ is ε-DP, and $F : Y \to Z$ is another randomized mapping, then $F \circ M$ is ε-DP
2. Group privacy
    - If $M : D^n \to Y$ is ε-DP, and D, D' differ in k positions, then for all $S \in \text{Range}(M)$:
      $Pr[M(D) \in S] \leq \exp(k\varepsilon) Pr[M(D') \in S]$
3. Basic composition
    - If we run k ε-DP algorithms sequentially through our data, the full process will be $k\varepsilon$-DP

Properties of DP

Properties of DP

1. Closed under post-processing
   - If $M : D^n \to Y$ is ε-DP, and $F : Y \to Z$ is another randomized mapping, then $F \circ M$ is ε-DP
2. Group privacy
   - If $M : D^n \to Y$ is ε-DP, and D, D' differ in k positions, then for all $S \in \text{Range}(M)$:
     $Pr[M(D) \in S] \leq \exp(k\varepsilon) Pr[M(D') \in S]$
3. Basic composition
   - If we run k ε-DP algorithms sequentially through our data, the full process will be $k\varepsilon$-DP

Differential Privacy

Approximate Differential Privacy

Approximate Differential Privacy

We can 'loosen' our privacy guarantees a little

Increase utility, not give up that much privacy

Idea

We take our original ε-DP privacy guarantee, and add a 'cryptographically small' probability that it will not work

It turns out, this is enough to significantly improve utility of our DP mechanism!

Approximate Differential Privacy

For $\varepsilon \geq 0$*, mechanism* M *is* ε, δ*-differentially private if*, for all $S \subseteq \text{Range}(M)$, and **neighboring datasets** D and D':

Approximate Differential Privacy

$Pr[M(D) \in S] \leq \exp(\varepsilon) Pr[M(D') \in S] + \delta$

If $\delta = 0$, we go back to our original 'pure' DP definition

How about the privacy loss random variable?

$$|\mathcal{L}_{M(D)||M(D')}| \leq \varepsilon \text{ w.p. } 1 - \delta, \text{ for all } D \text{ and } D' \text{ neighboring datasets}$$

Approximate Differential Privacy

What should we set δ to?

Good rule of thumb: $\delta \ll \frac{1}{n}$, where n is the size of the dataset D

Example $0, \delta$-DP Mechanism

M: For each $x \in D$, output x w.p. δ, and do nothing w.p. $1 - \delta$

Probability we do not release anyone's data point: $(1 - \delta)^n$

Probability we **do** release someone's data point: $1 - (1 - \delta)^n$

If δ is around $\frac{1}{n}$, then this is approximately 1 (as δ increases, this approaches 1)

Achieving Approximate DP: The Gaussian Mechanism

How do we achieve the ε, δ-DP guarantee? — **Gaussian Mechanism**

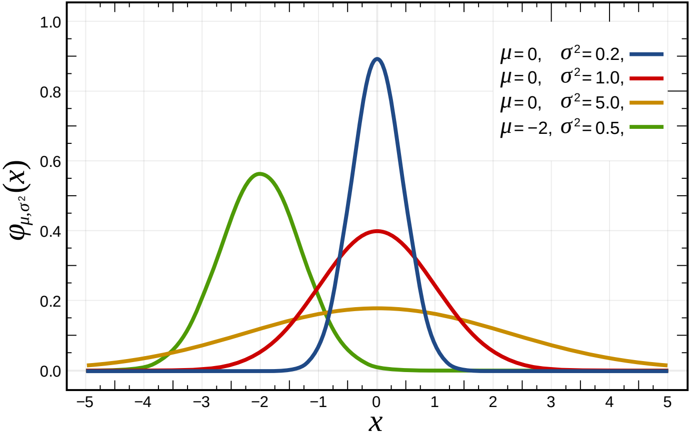
l_2 Sensitivity

$f : D^n \to \mathbb{R}^k$

l_2-sensitivity of function f is: $\Delta_2^{(f)} = \max_{D,D'} ||f(D) - f(D')||_2$

Achieving Approximate DP: The Gaussian Mechanism

Gaussian Distribution: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$



Achieving Approximate DP: The Gaussian Mechanism

Gaussian Mechanism

$f : D^n \to \mathbb{R}^k$

$M(D) = f(D) + (Y_1, \ldots, Y_k)$

$Y_i \underset{i.i.d}{\sim} \mathcal{N}(0, 2\ln(\frac{1.25}{\delta})\frac{\Delta_2^2}{\varepsilon^2})$

Add noise to each coordinate, proportional to the L2 sensitivity

Name	Attr. 1	Attr. 2	...
Alice	0	1	...
Bob	0	1	...
Clair	1	0	...
⋮	⋮	⋮	...
Zane	1	0	...

Achieving Approximate DP: The Gaussian Mechanism

Example: Sum query with more attributes

Mechanism: $f = \sum X_i$

$D \in \{0,1\}^{n \times d}$, where n is the number of individuals and d the number of attributes

Worst case for neighboring datasets D, D': D's row has all 1s, D''s row has all 0s

l_1-sensitivity: $||\mathbf{1} - \mathbf{0}||_1 = ||\mathbf{1}||_1 = d$

l_2-sensitivity: $||\mathbf{1}||_2 = \sqrt{d}$

Laplace: $\tilde{p} = f(x) + \text{Lap}(\frac{d}{\varepsilon})$

Gaussian: $\tilde{p} \approx f(x) + \mathcal{N}(0, (\frac{\sqrt{d}}{\varepsilon})^2)$

Benefits of Approximate DP

1. Scale of added noise can be significantly less (with higher dimensions)
2. Can improve upon basic composition (more advanced composition techniques)
   - If we run k ε, δ-DP algorithms sequentially through our data, the full process will be *less than* $k\varepsilon, k\delta$-DP (depending on the composition technique)

Differential Privacy for Machine Learning

Applying DP for ML

	stars	sentiment	text
0	5	positive	After getting food poisoning at the Palms hote...
1	4	positive	"A feast worthy of Gods"\n\nBaccarnal Buffet i...
2	4	positive	The crab legs are better than the ones at Wick...
3	1	negative	Not worth it! Too salty food and expensive! Th...
4	5	positive	I would give this infinite stars if I could. M...
...	...	...	...
10412	5	positive	Best buffet ever! Irma was great, served us be...
10413	4	positive	Hollllllyyyy moleyyyy! \n\nThis buffet was one...
10414	5	positive	The selection is amazing and all the food is e...
10415	4	positive	One of the best buffets I've had in Vegas. My ...
10416	4	positive	I got a chance to go to the Bacchanal Buffett ...

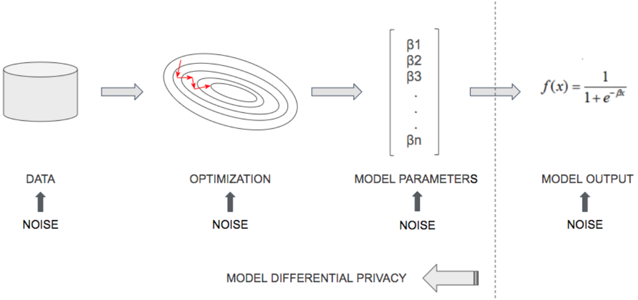
10374 rows × 3 columns

https://medium.com/analytics-vidhya/performing-sentiment-analysis-on-yelp-restaurant-reviews-962334d6336d

Applying DP for ML

DP and ML: Instead of privatizing an algorithm, we're privatizing a model

Need to choose a 'query' as before (associated with the model, dependent on the dataset)



https://medium.com/bluecore-engineering/differential-privacy-in-the-real-world-f31a5df1398f

DP for ML: DP-SGD

Most common and widely adopted algorithm in differentially private ML:
Differentially Private Stochastic Gradient Descent (DP-SGD)

Can apply directly to model training

The 'query' of our DP mechanism: Gradient of the loss function

What's the sensitivity of the gradient?

DP for ML: DP-SGD

Most common and widely adopted algorithm in differentially private ML:
Differentially Private Stochastic Gradient Descent (DP-SGD)

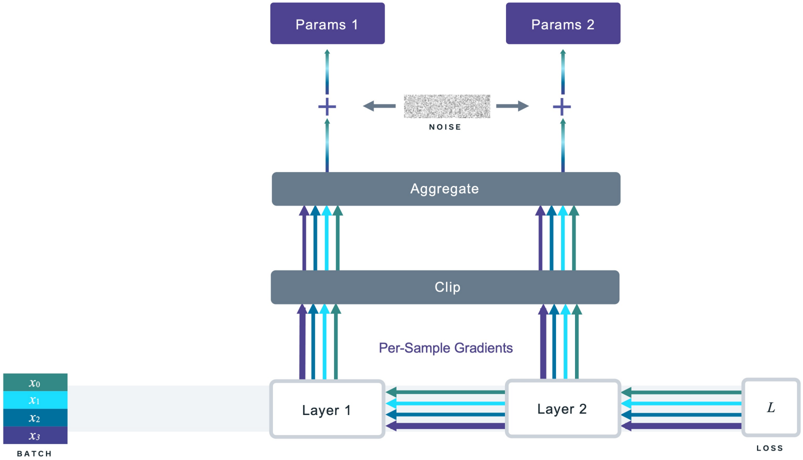
Proposed by Abadi et al. [2016]

Can apply directly to model training

The 'query' of our DP mechanism: Gradient of the loss function

What's the sensitivity of the gradient? — ∞...

DP for ML: DP-SGD



https://ai.facebook.com/blog/introducing-opacus-a-high-speed-library-for-training-pytorch-models-with-differential-privacy/

DP for ML: DP-SGD

Algorithm for DP-SGD:

1. Select a 'lot' of points
   - **Lot**: A set of data points, where each point is selected with probability L/n, where L is the 'lot size' and n is the size of the dataset
2. For each point in the 'lot', compute the gradient $g_i = \nabla l(\theta_t, x_i, y_i)$, $\forall i \in$ lot
3. Clip g_i to l_2 ball of radius C, then average
4. Add noise
5. Step in negative direction of gradient

DP for ML: DP-SGD

Algorithm 1 Differentially private SGD (Outline)

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 for ML: DP-SGD

Important points about DP-SGD:

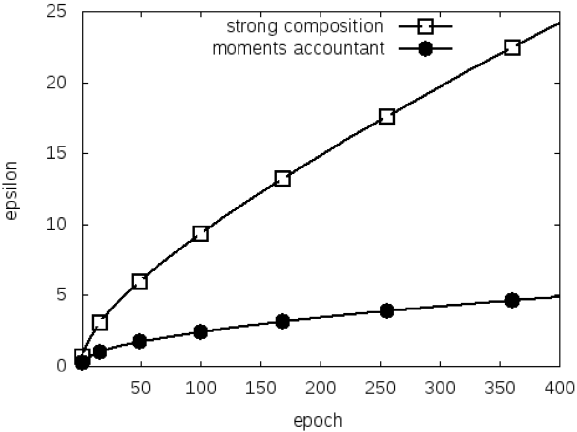
1. Use **Poisson sampling** to select lots (**not** simply iterate over batches)
2. Moments accountant: Much better bounds on privacy budget
3. Clipping can slow down computations

DP for ML: DP-SGD

Important points about DP-SGD:

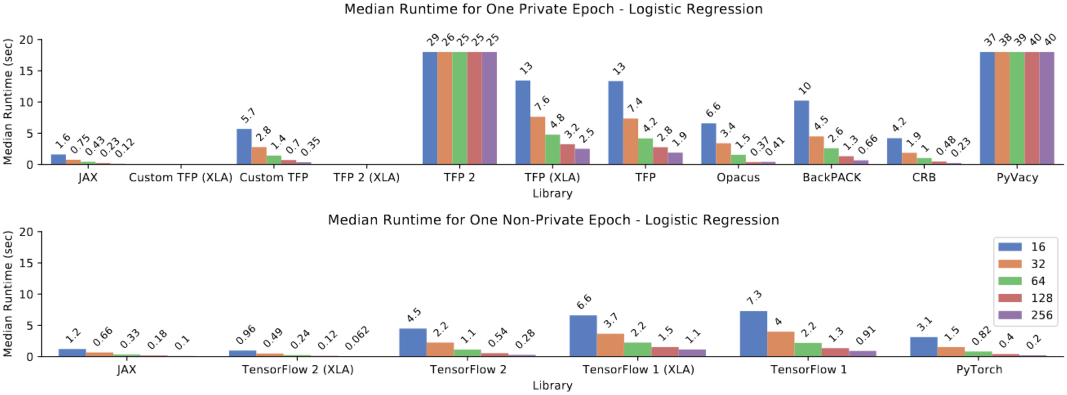
1. Use **Poisson sampling** to select lots (**not** simply iterate over batches)
2. Moments accountant: Much better bounds on privacy budget
3. Clipping can slow down computations

DP for ML: DP-SGD



Abadi et al. [2016]

DP for ML: DP-SGD



Subramani et al. [2021]

Opacus

```python
model = Net()
optimizer = torch.optim.SGD(model.parameters(), lr=0.05)

privacy_engine = PrivacyEngine(
    model,
    batch_size=32,
    sample_size=len(train_loader.dataset),
    alphas=range(2,32),
    noise_multiplier=1.3,
    max_grad_norm=1.0,
)
privacy_engine.attach(optimizer)
# That's it! Now it's business as usual
```

https://ai.facebook.com/blog/introducing-opacus-a-high-speed-library-for-training-pytorch-models-with-differential-privacy/

Other Methods in Privacy Research

Other Methods in Privacy: Secure Multiparty Computation

Secure Multiparty Computation

Combining private inputs from multiple people, in order to compute a function, without revealing anyone's input to the rest

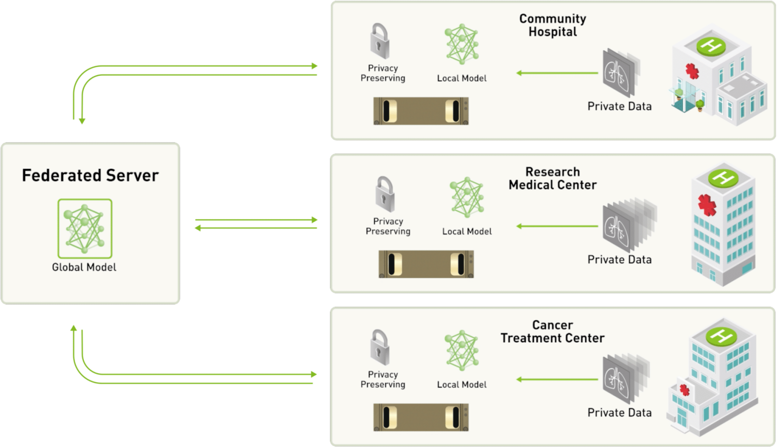
Encryption: These numbers are encrypted, nobody knows their own share

Shared Governance: The numbers can only be decrypted if everyone agrees

Main aspects:

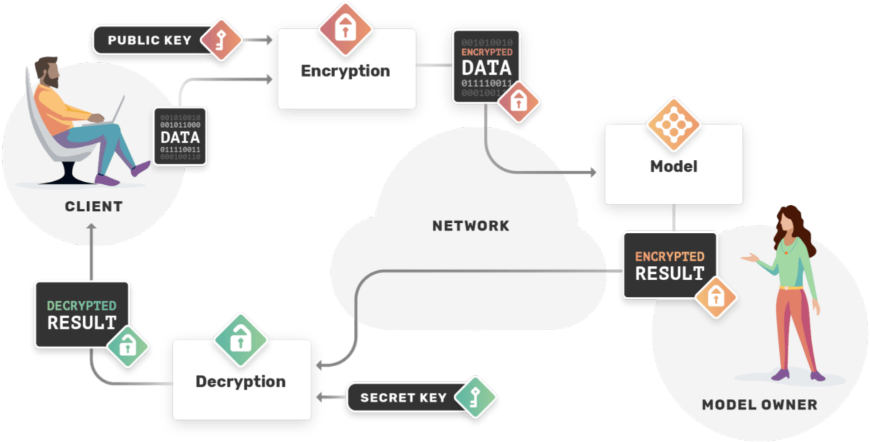
1. Data remains on a remote machine
2. Model can be encrypted during training
3. Multiple data owners privately combining their data

Other Methods in Privacy: Federated Learning



https://blogs.nvidia.com/blog/2019/10/13/what-is-federated-learning/

Other Methods in Privacy: Homomorphic Encryption



https://research.aimultiple.com/homomorphic-encryption/

References 1

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, Vienna, Austria, 2016. ACM. doi: 10.1145/2976749.2978318. URL https://dl.acm.org/doi/10.1145/2976749.2978318.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

References 2

Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4): 211–407, 2013. doi: 10.1561/0400000042.

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

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

References 3

Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34:26409–26421, 2021.