Differentially Private Model Publishing for Deep Learning

Lei Yu, Ling Liu, ***Calton Pu***,
Mehmet Emre Gursoy, Stacey Truex

School of Computer Science, College of Computing
Georgia Institute of Technology

This work is partially sponsored by NSF 1547102, SaTC 1564097, and a grant from Georgia Tech IISP

Outline

- Motivation
- Deep Learning with Differential Privacy
- Our work
  - Privacy loss analysis against different data batching methods
  - Dynamical privacy budget allocation
  - Evaluation
- Conclusion

Deep Learning Model Publishing

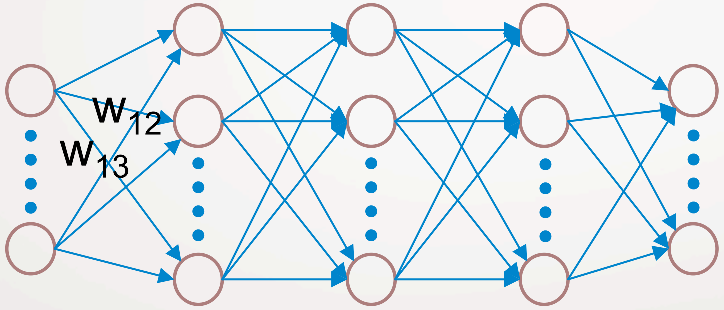
- Applications: speech, image recognition; natural language processing autonomous driving
  - A Key factor for its success: large amount of data
- Privacy leakage Risks by Applications
  - Cancer diagnosis, Object detection in Self driving car …
- Privacy leakage Risks by attacks
  - Membership inference attacks[Reza Shokri et al, SP’17]
  - Model inversion attacks[M. Fredrikson et al, CCS’15]
  - Backdoor (intentional) memorization [C Song et al. CCS’17]

Model Publishing of Deep Learning

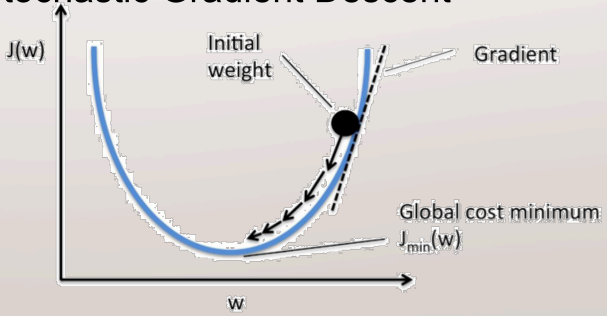
training
dataset

photos, documents, internet activities, business transactions health records

Iterative Training
Deep Neural Networks (DNN)



Learning network parameters:
Stochastic Gradient Descent



Model Publishing

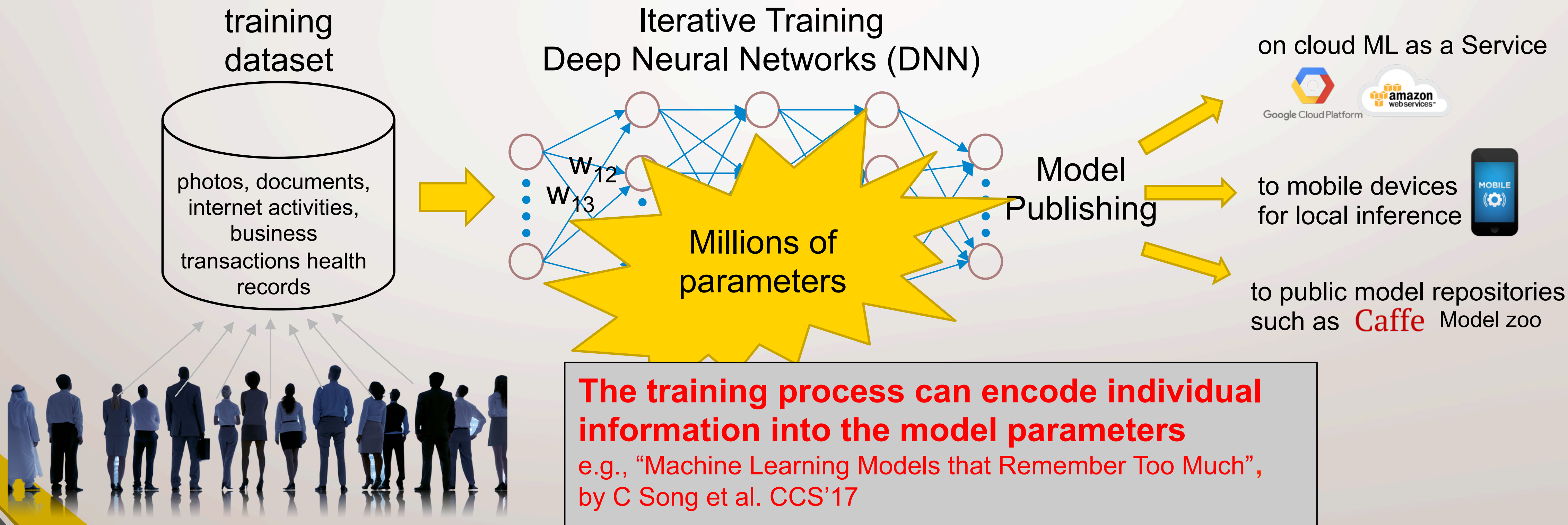
on cloud ML as a Service

Google Cloud Platform amazon web services

to mobile devices for local inference

to public model repositories such as Caffe Model zoo

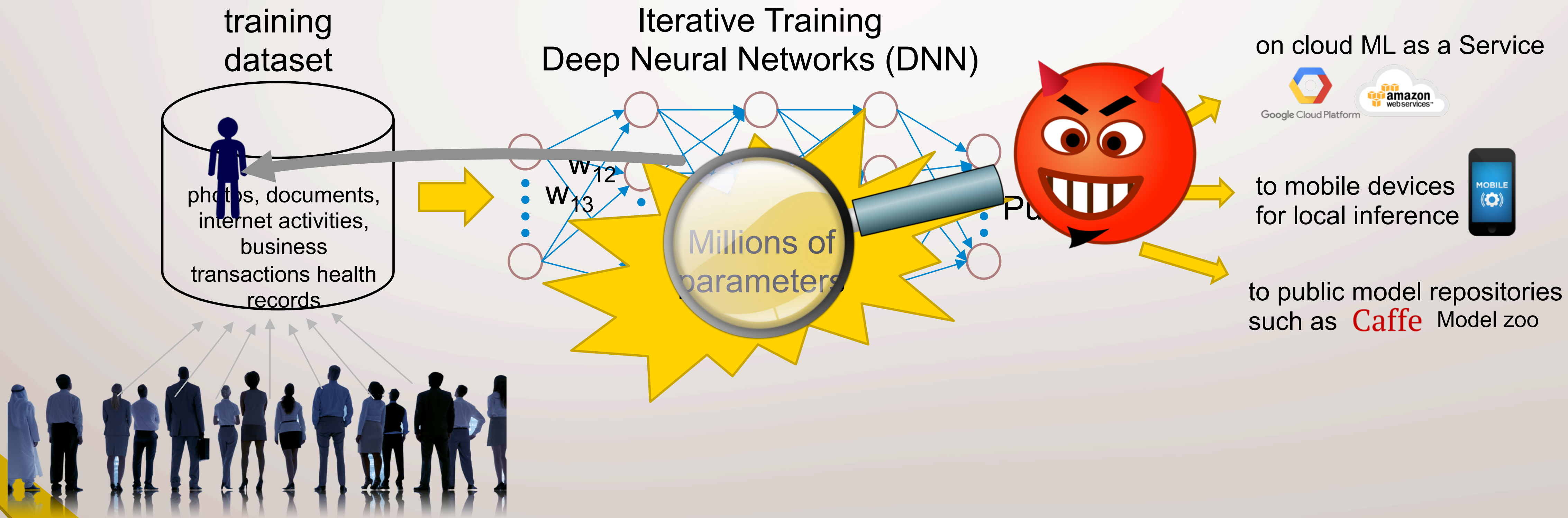
Data Privacy in Model Publishing of Deep Learning



The training process can encode individual information into the model parameters

e.g., "Machine Learning Models that Remember Too Much", by C Song et al. CCS'17

Data Privacy in Model Publishing of Deep Learning



Proposed Solution

- Deep Learning Model Publishing with Differential Privacy
- Related Work
  - Privacy-Preserving Deep Learning [Reza Shokri et al, CCS'15]
  - Deep Learning with Differential Privacy [M. Abadi, et al . CCS'16]

Differential Privacy Definition

- The **de facto** standard to guarantee privacy
  - Cynthia Dwork, Differential Privacy: A Survey of Results, TAMC, 2008
- A randomized algorithm *M: D -> Y* satisfies (ε, δ)-Differential Privacy, if for any two neighboring dataset D and D' which differs in only one element, for any subset $S \subset Y$

$$\forall S:\ \Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S] + \delta$$

- For protecting privacy, ε is usually a small value (e.g., 0<ε<1), such that two probability distributions are very close. It is difficult for the adversary to distinguish D and D' by observing an output of *M*.

Differential Privacy Composition

- Composition：

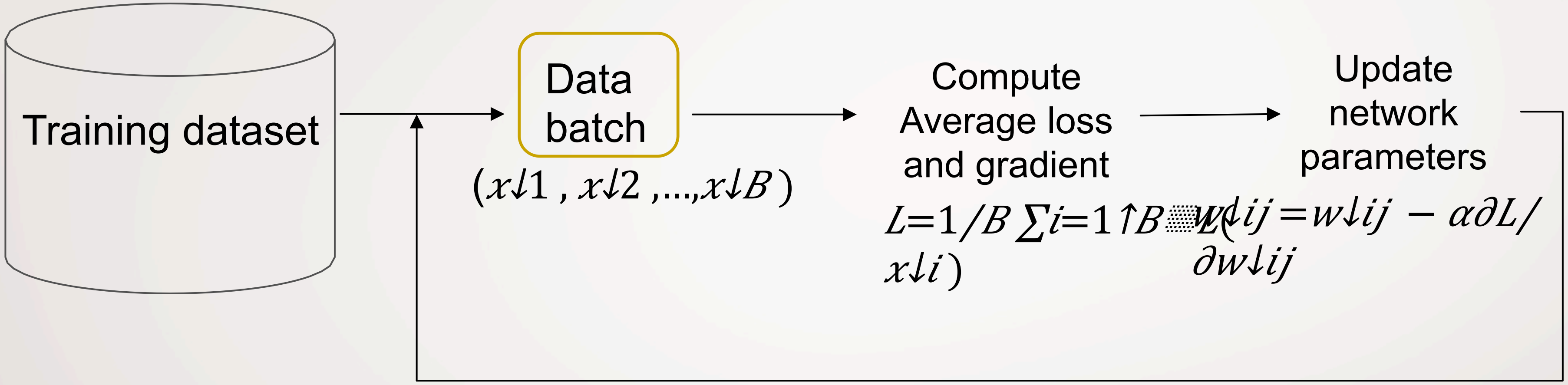
  For *ε-differential privacy*, If $M_1, M_2, ..., M_k$ are algorithms that access a private database D such that each M_i satisfies ε_i -differential privacy, then running all k algorithms sequentially satisfies ε-differential privacy with $\varepsilon=\varepsilon_1+...+\varepsilon_k$

- Composition rules help build complex algorithms using basic building blocks
  - Given total ε, *how to assign* ε_i for each building block to achieve the best performance
  - The ε *is usually referred to as privacy budget. The assignment of* ε_i is a budget allocation.

Differential Privacy in Multi-Step Machine Learning

- With *N* steps of ML algorithm *A*, the privacy budget ε can be partitioned into N smaller ε_i such that $\varepsilon=\varepsilon_1+\ldots+\varepsilon_N$
- Partitioning of ε among steps:
  - *Constant:* $\varepsilon_1=\ldots=\varepsilon_N$
  - Variable
    - Static approach which defines different ε_i for each step at configuration
    - dynamic: different ε_i for each step, changes with steps

Stochastic Gradient Descent in Iterative Deep Learning



A training iteration

(1) DNN training takes a large number of steps (#iterations or #epochs)

- Tensorflow cifar10 tutorial: cifar10_train.py achieves ~86% accuracy after 100K iterations
- For ResNet model training on ImageNet dataset, as reported in the paper [Kaiming He etc, CVPR'15], the training runs for 600,000 iterations.

(2) Training dataset is organized into a large number of mini-batches of equal size for massive parallel computation on GPUs with two popular mini-batching methods:

- Random Sampling
- Random Shuffling

Differentially Private Deep Learning: Technical Challenges

- Privacy budget allocation over # steps
  - Two proposed approaches
    - Constant ε_i for each of the iterations, configured prior to runtime→ [M. Abadi, et al . CCS'16]
    - Variable ε_i : Initialized with a constant ε_i for each iteration and dynamically decaying the value of ε_i at runtime → this paper
- Privacy cost accounting
  - Random sampling
    - Moments accountant → M. Abadi, et al . CCS'16]
  - Random Shuffling
    - zCDP based Privacy Loss analysis → this paper

Scope and Contributions

- Deep learning Model Publishing with Differential Privacy
  - Differentiate random sampling and random shuffling in terms of privacy cost
  - Privacy analysis for different data batching methods
    - Privacy accounting using extended zCDP for random shuffling
    - Privacy analysis with empirical bound for random sampling
  - Dynamic privacy budget allocation over training time
  - Improve model accuracy and runtime efficiency

Data Mini-batching: Random Sampling vs. Random Shuffling

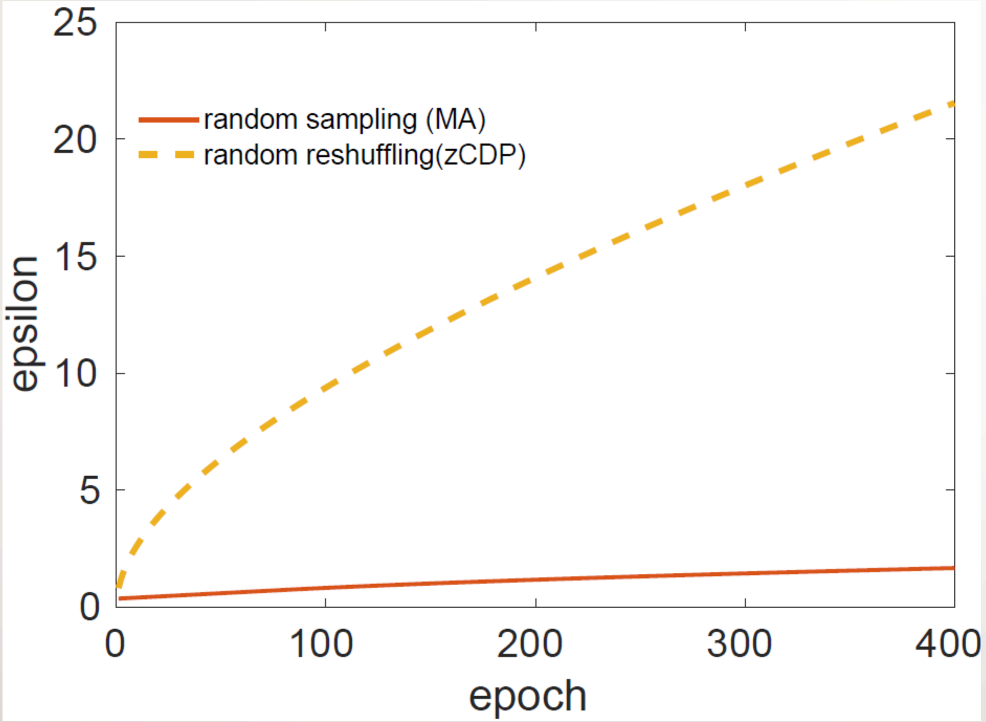
- **Random sampling** with replacement : each batch is generated by independently sampling every example with a probability= batch_size / total_num_examples
  - Example: [1 2 3 4 5 6 7 8 9] → [1 3 5] [1 2] [3 4 7 9] (probability q = batch size / 9 = 1/3)
- **Random shuffling**: reshuffle dataset every epoch and partition a dataset into disjoint min-batches during each reshuffle
  - Example:
    [1 2 3 4 5 6 7 8 9] → [4 7 1] [6 2 3] [8 9 5] (Batch size =3)
  - common practice in the implementation of deep learning, available data APIs in Tensorflow, Pytorch, etc.

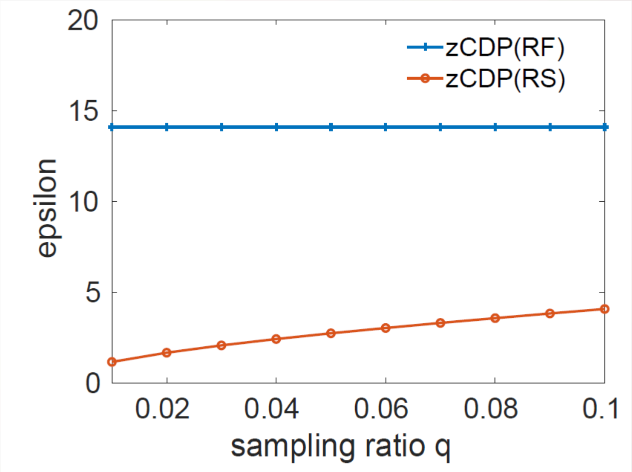
Data Minibatching: Random Sampling vs. Random Shuffling

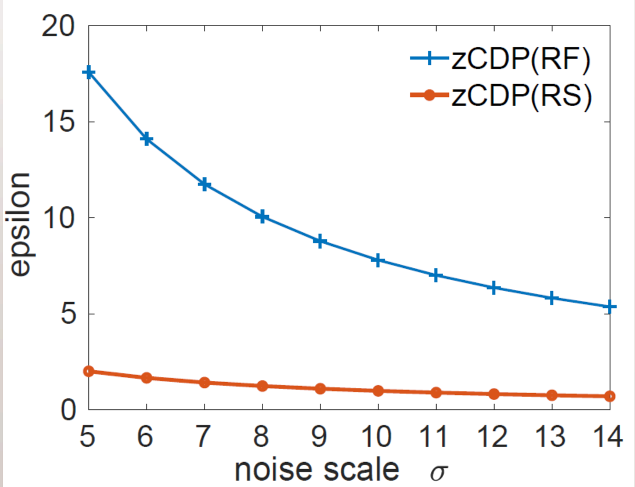
Dataset: [0,1,…,9], batch_size=2

Batching method	output instances in one epoch
tf.train_shuffle_batch	[2 6], [1 8], [5 0], [4 9], [7 3]
tf.estimator.inputs.numpy_inpunt_fn	[8 0], [3 5], [2 9], [4 7], [1 6]
Random sampling with q=0.2	[ ], [0 6 8], [4], [1], [2 4]

Data Minibatching: Random Sampling vs. Random Shuffling







Moments accountant method developed for random sampling cannot be used to analyze privacy cost and accounting for random shuffling!

Differential Privacy accounting for random shuffling

Georgia Tech

- Developing privacy accounting analysis for random shuffling based on zCDP
  - CDP is relaxation of (ε, δ)-Differential Privacy, developed by Cynthia et al , Concentrated Differential Privacy. CoRR abs/1603.01887 (2016)
  - zCDP is variant of CDP, developed by Mark Bun et al. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds , TCC 2016-B.

(1) Within each epoch, each iteration satisfies ρ–zCDP by applying Gaussian mechanism with the same noise scale $\sqrt{1/2\rho}$

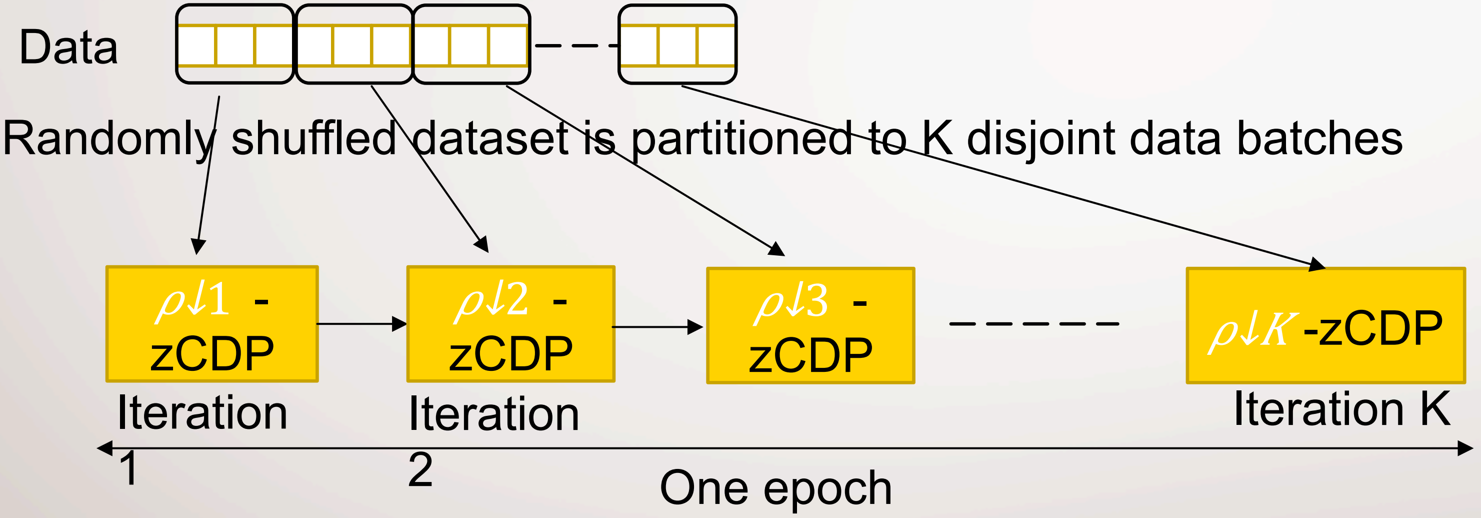
- Our analysis shows under random shuffling, the whole epoch still satisfies ρ–zCDP

(2) Employing dynamic decaying *noise scale* for each epoch, and using the sequential composition for zCDP among *T* epochs:

- a sequential composition of *T* number of ρ_i–zCDP mechanisms to satisfy $(\sum \rho_i)$–zCDP

CDP based Privacy Loss analysis for random shuffling

Random shuffling in an epoch



the epoch satisfies $\max_i(\rho_i)$-zCDP. Our implementation uses the same $\rho_i = \rho$ for each iteration in an epoch, thus the epoch satisfies ρ-ZCDP.

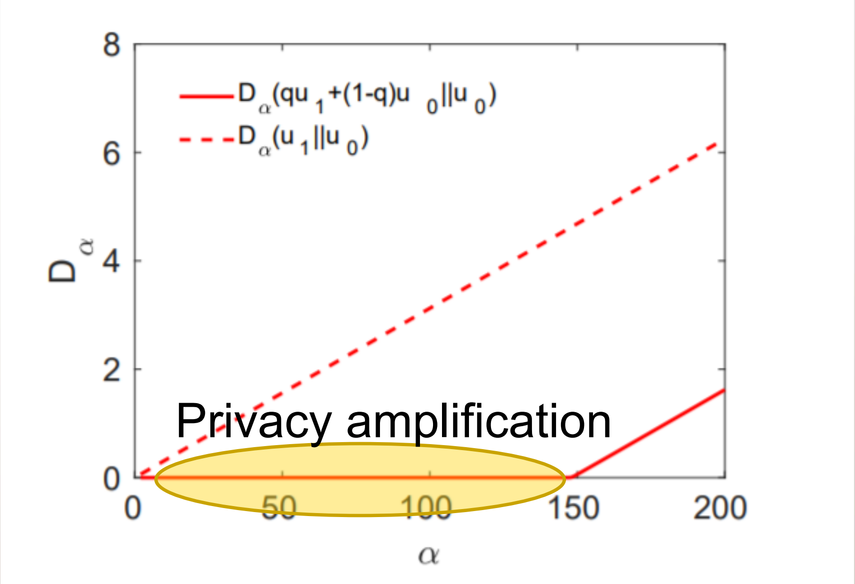
CDP based Privacy Loss analysis for random shuffling

Random shuffling in multiple epochs

Because each epoch accesses the whole dataset, among epochs the privacy loss follows linear composition. The training of T epochs satisfies $\sum_{i}^{T} \rho_i$ -zCDP

CDP based Privacy Loss analysis for random sampling

- zCDP cannot capture the privacy amplification effect of random sampling
  - Caused by the linear α-Renyi divergence constraint over all $\alpha \in (1, \infty)$ in the definition
- Only consider the constraint on a limited range of $\alpha \in (1, U\_\alpha)$ $(U\_\alpha < \infty)$
- We find a heuristic bound within a limited range of α and convert it to (ε, δ)-Differential Privacy in an analytical way(Details in Theorem 3)



Dynamic privacy budget allocation

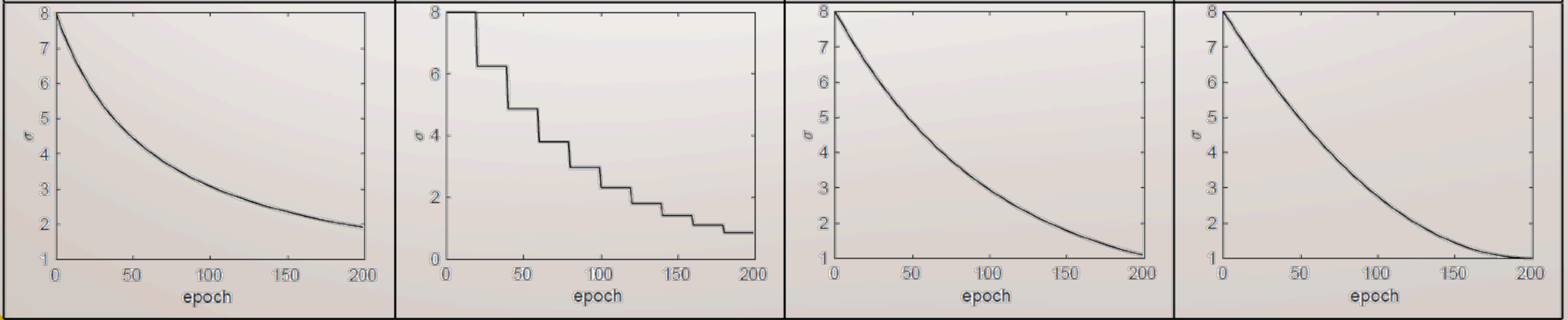
- Under fixed privacy budget, dynamically allocate privacy budget among epochs to optimize model accuracy
  - Pre-defined schedules
  - Adaptive schedule based on public validation dataset
    - Public data set does not involve extra privacy cost

Dynamic privacy budget allocation

- Pre-defined four different scheduling algorithms to decay the noise level
- The ε_i value is determined using the decay function at runtime dynamically

Table 1: Budget Allocation Schedules($\sigma_0 = 8$)

Time-Based Decay k=0.016	Step Decay k=0.78, period=20	Exponential Decay k=0.01	Polynomial Decay k=2, $\sigma_{end} = 1$



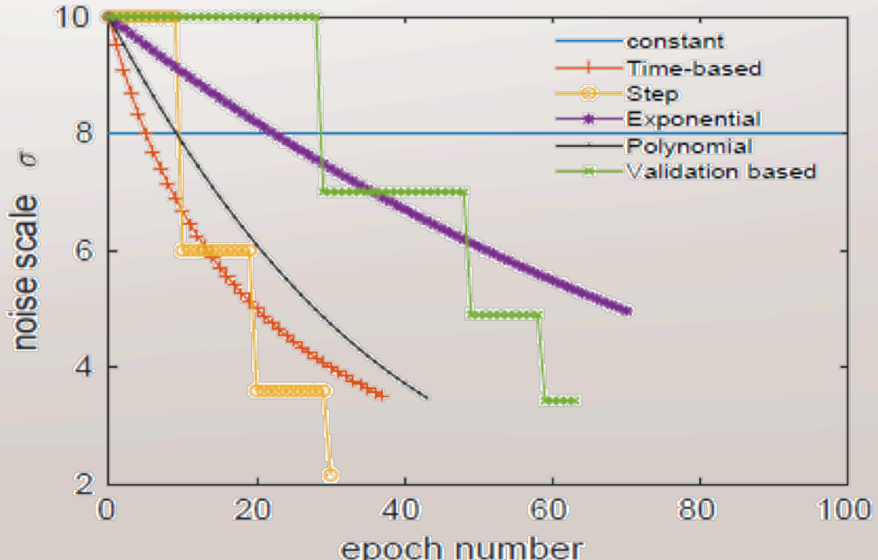
Dynamic privacy budget allocation

- Adaptive schedule based on public validation dataset
  - Periodically check the model accuracy on the validation dataset during training process
  - Reduce the noise level when the validation accuracy stops improving

Evaluation

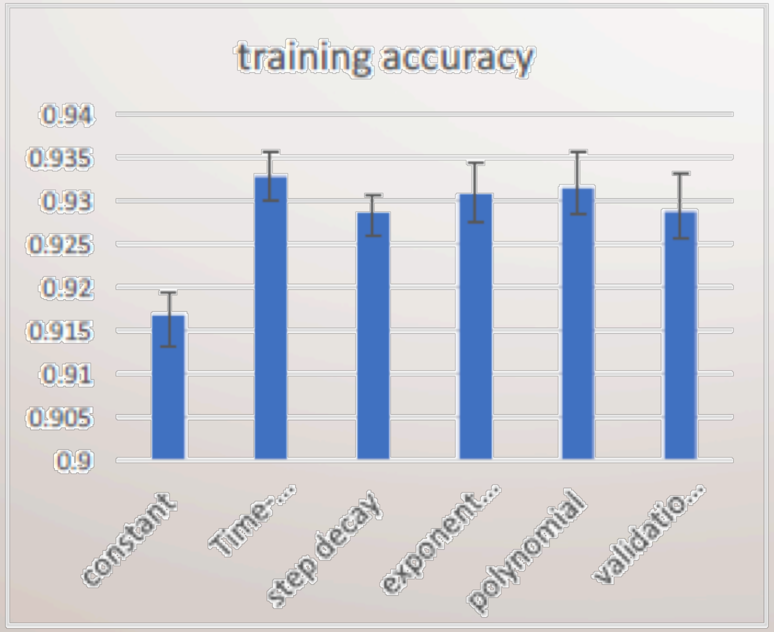
- Evaluating dynamic privacy budget allocation on MNIST
  - Compared with the approach using constant noise scale during training time
  - The decay functions have decay parameters to decide how the noise scale changes with the epochs
  - The decay parameters are hyperparameters prespecified by the users.

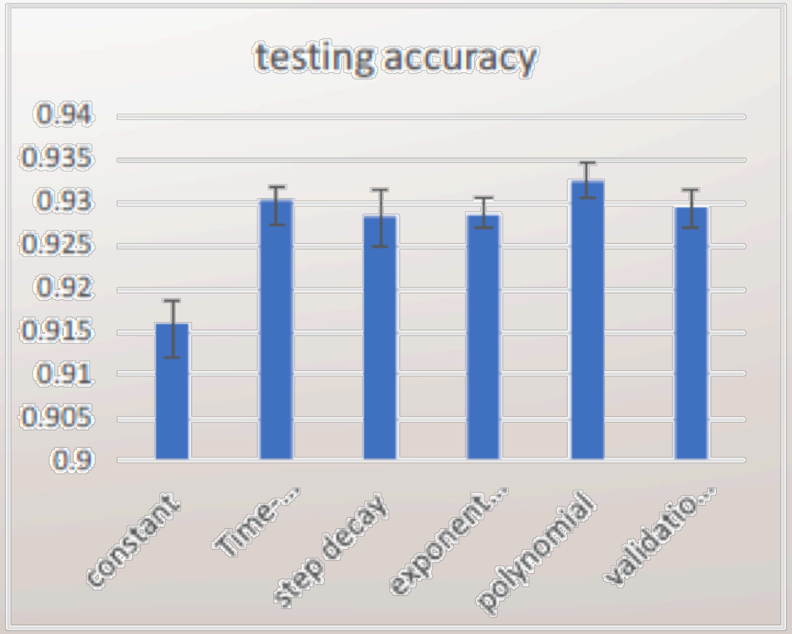
The change of noise scale during training



Evaluation

- Evaluating dynamic privacy budget allocation on MNIST
  - Dynamic privacy budget allocation improves model accuracy





Evaluation

- Comparing Privacy Accounting Approaches
  - Convert to (ε, δ)-Differential Privacy

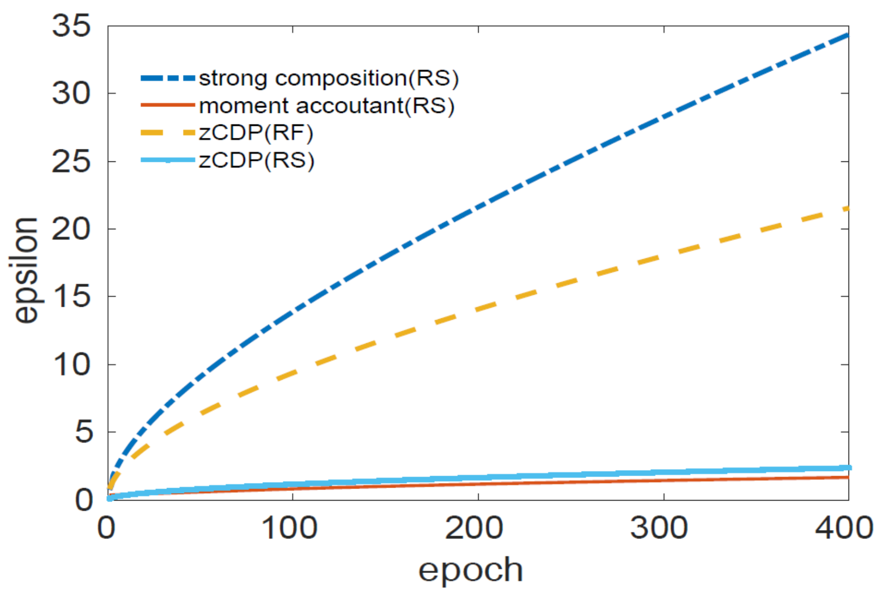


Fig. 2: Privacy parameter ϵ v.s. epoch

1. Random shuffling incurs higher privacy loss than random sampling
2. Heuristic bound produces close result to the MA method, but it is easier to compute

Summary

- Privacy Loss Analysis against Different Data Batching Methods
- Dynamic privacy budget allocation
- Source Code: https://github.com/git-disl/DP_modelpublishing
- Refined Version on Arxiv : https://arxiv.org/abs/1904.02200

Thank you!

Concentrated Differential Privacy (CDP)

- Recently developed by Dwork and Rothblum to focus on the cumulative privacy loss for a large number of computations and provide a sharper analysis tool.
  - Privacy Loss as subgaussian random variable

Cynthia Dwork, Guy N. Rothblum, Concentrated Differential Privacy. CoRR abs/1603.01887 (2016)

Zero-Concentrated Differential Privacy (zCDP)

Mark Bun , Thomas Steinke
Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds , TCC 2016-B.

- Zero-CDP (zCDP) : A randomized mechanism *A* is ρ-zCDP if for any two neighboring database *D* and *D*' that differ in only a single entry and all $\alpha \in (1, \infty)$

$$D_\alpha(\mathcal{A}(D)||\mathcal{A}(D')) \triangleq \frac{1}{\alpha - 1} \log \left(\mathbb{E}\left[e^{(\alpha-1)L^{(o)}}\right]\right) \leq \rho\alpha$$

  - The Gaussian mechanism for *f* with noise $N(0, \Delta_f^2 \sigma^2 I)$ satisfies $(1/2 \sigma^2)$-zCDP.

α-Renyi divergence

- Linear Composition: A sequential composition of *K* number of ρ–zCDP mechanisms satisfies $(K\rho)$ -zCDP

Privacy Preserving Deep Learning

- Privacy-Preserving Deep Learning [Reza Shokri et al, CCS'15]
  - N party federated learning with N local private data respectively
  - Local model training on local data
  - exchange of model parameters instead of local data
- Deep Learning with Differential Privacy [M. Abadi, et al . CCS'16]
  - Differentially private Stochastic Gradient Descent (DP-SGD)
  - Assuming random sampling based batching and propose moment accountant method for privacy loss tracking