Lecture 14: Distributed Learning, Security, and Privacy

Announcements

- TA office hours will be project advising sessions this week

Agenda

- Distributed Learning and Federated Learning
- Privacy and Differential Privacy

Distributed Learning

- Sharing the computational load of training a model among multiple worker nodes

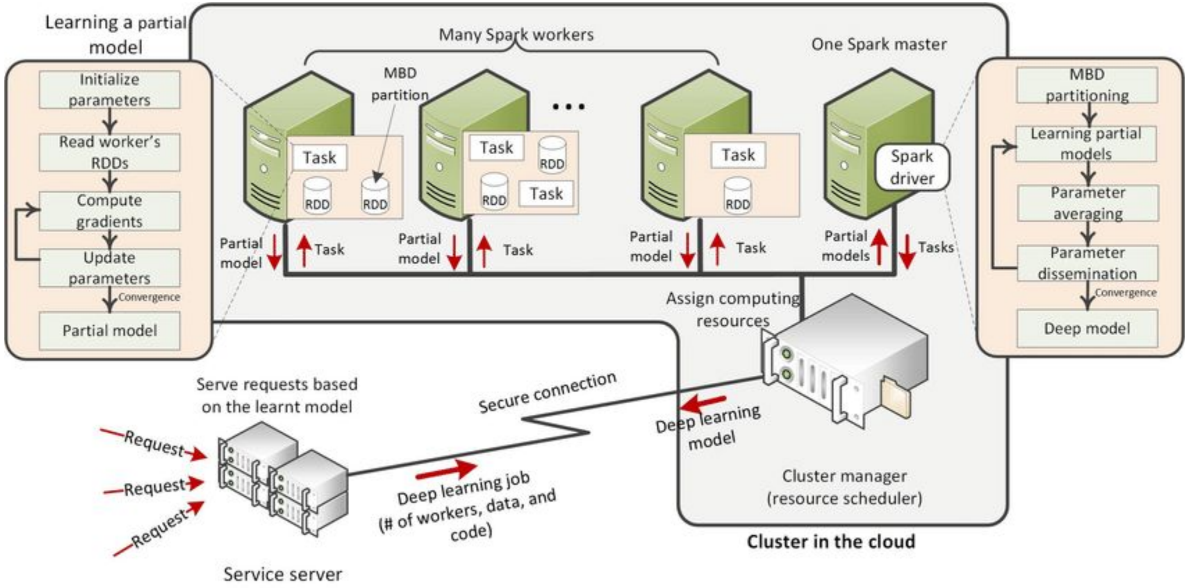
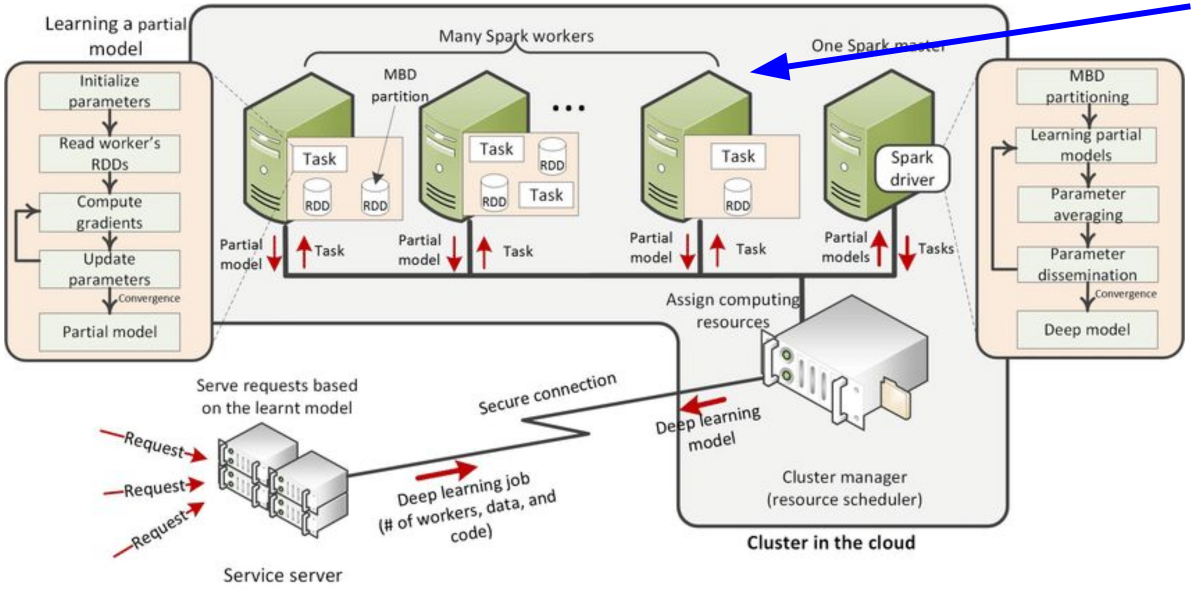


Figure credit: Alsheikh et al. Mobile big data analytics using deep learning and apache spark, 2016.

Distributed Learning

- Sharing the computational load of training a model among multiple worker nodes

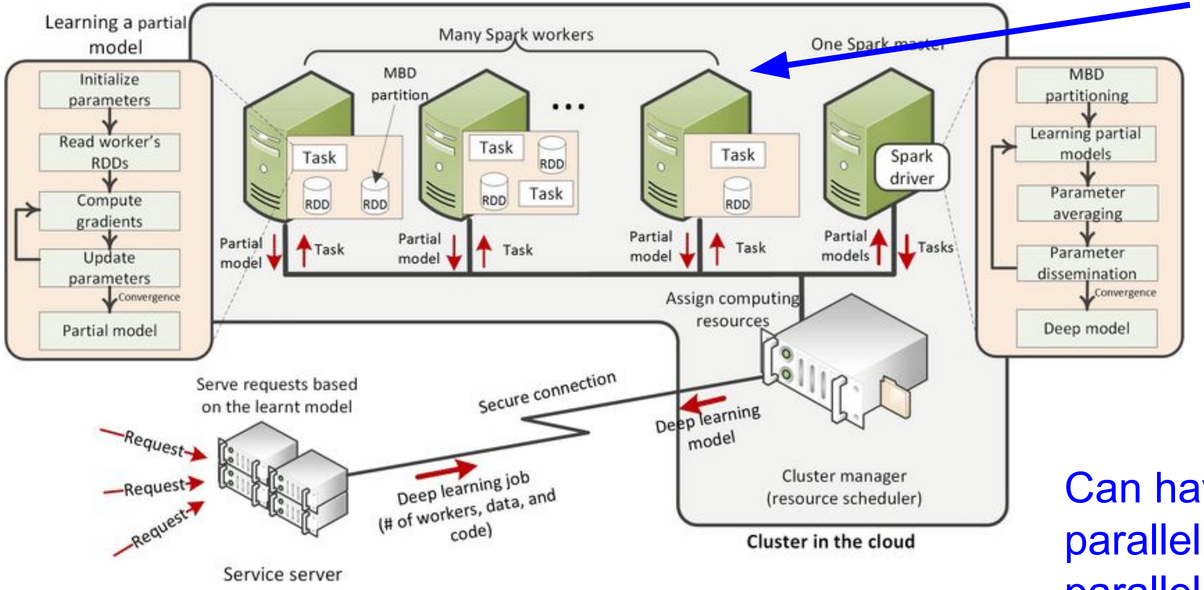


Data and task of computing gradient updates is distributed among nodes

Figure credit: Alsheikh et al. Mobile big data analytics using deep learning and apache spark, 2016.

Distributed Learning

- Sharing the computational load of training a model among multiple worker nodes



Data and task of computing gradient updates is distributed among nodes

Can have data parallelism or model parallelism

Figure credit: Alsheikh et al. Mobile big data analytics using deep learning and apache spark, 2016.

Federated Learning

- Related to distributed computing, but with an important property for many medical settings: data is decentralized and never leaves local silos. Central server controls training across decentralized sources.

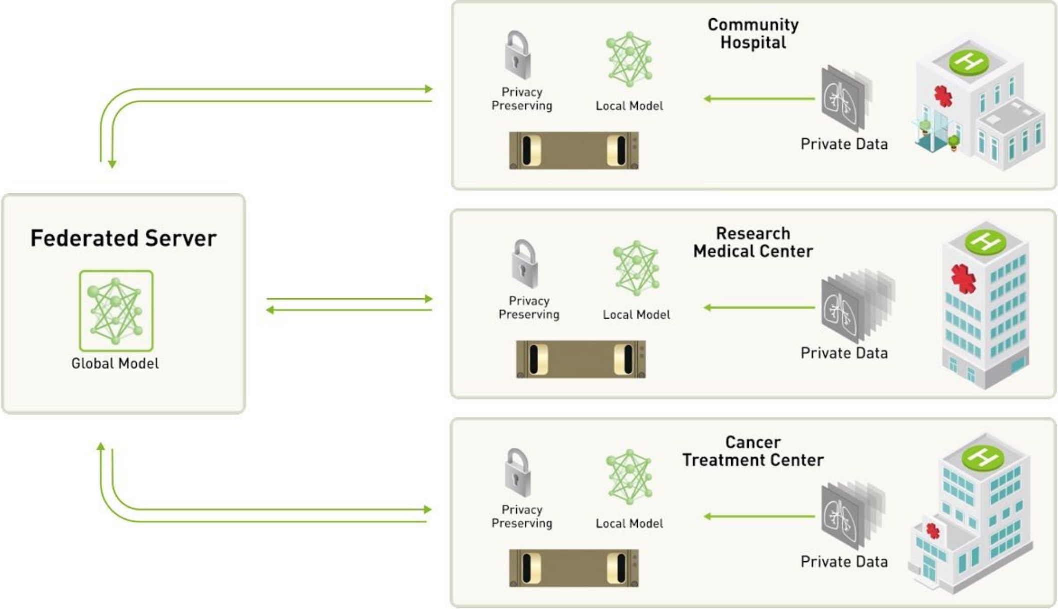


Figure credit: https://blogs.nvidia.com/wp-content/uploads/2019/10/federated_learning_animation_still_white.png

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones

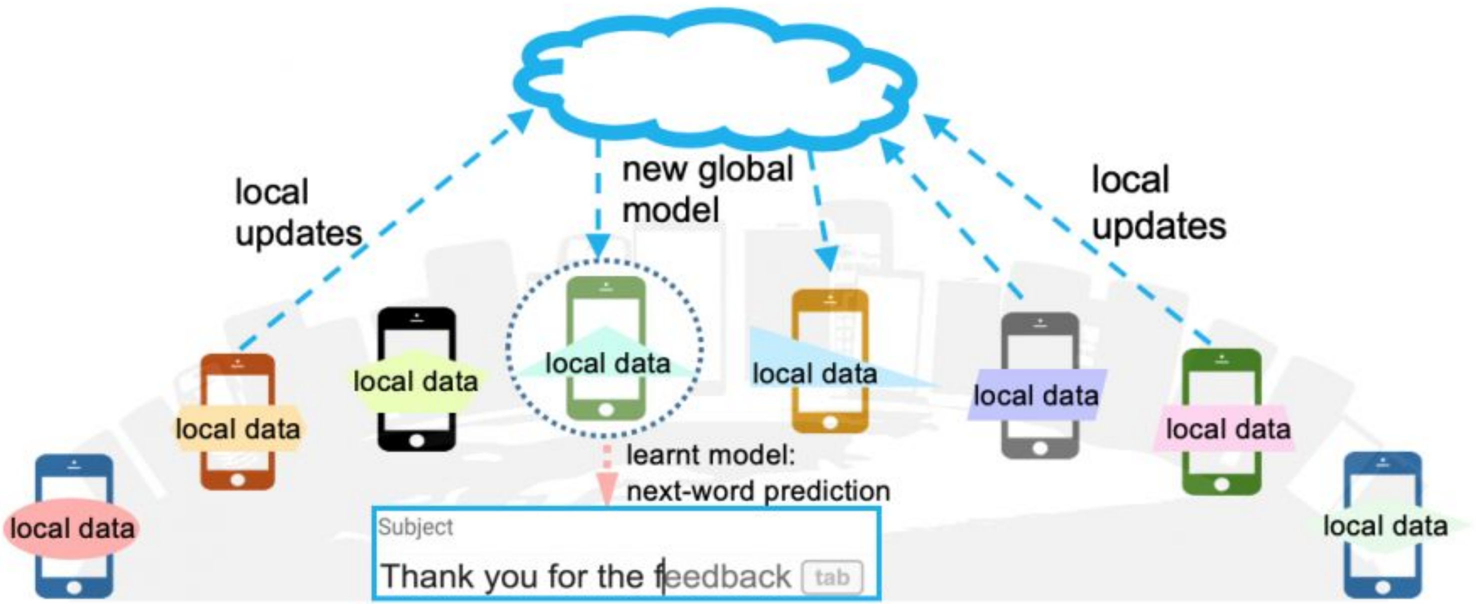
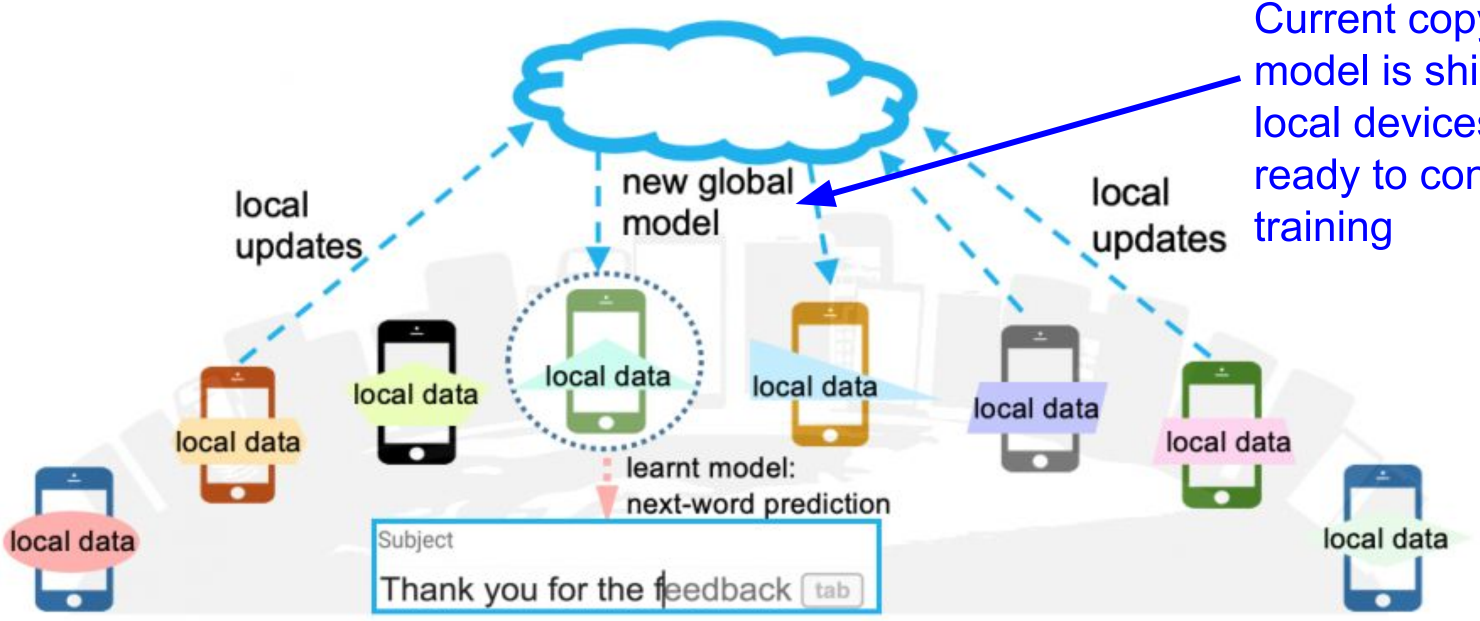


Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones

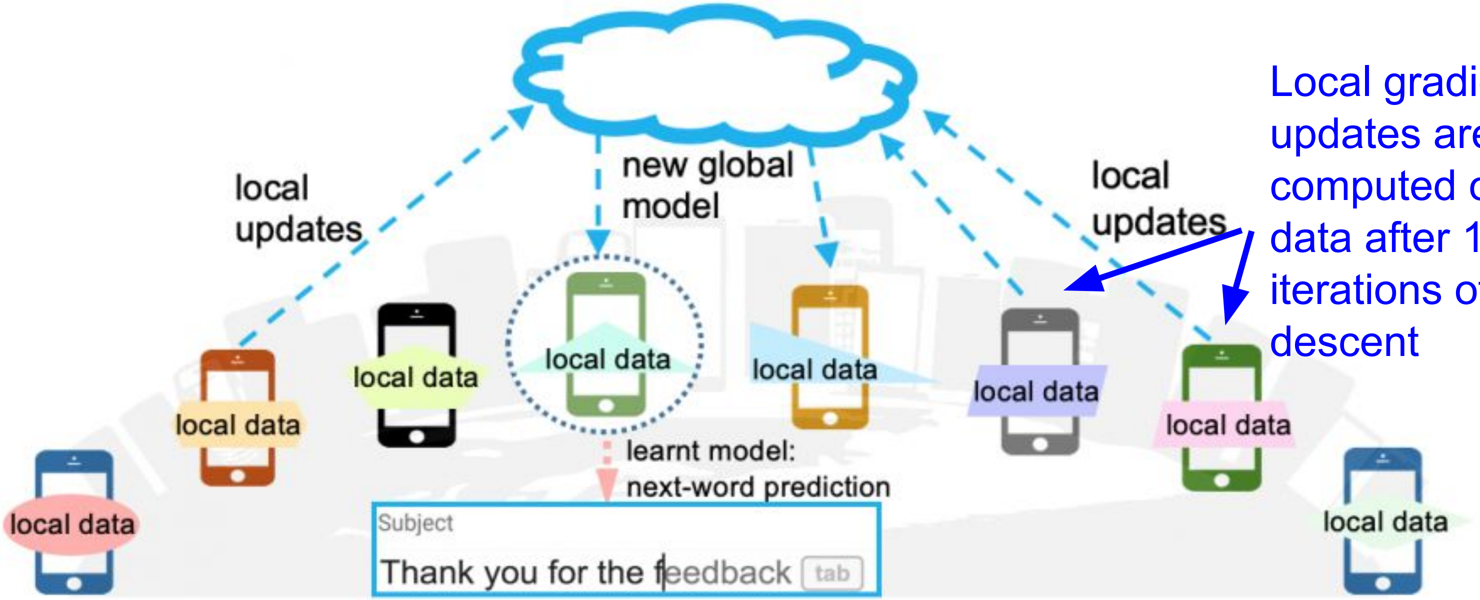


Current copy of global model is shipped to local devices that are ready to contribute to training

Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones

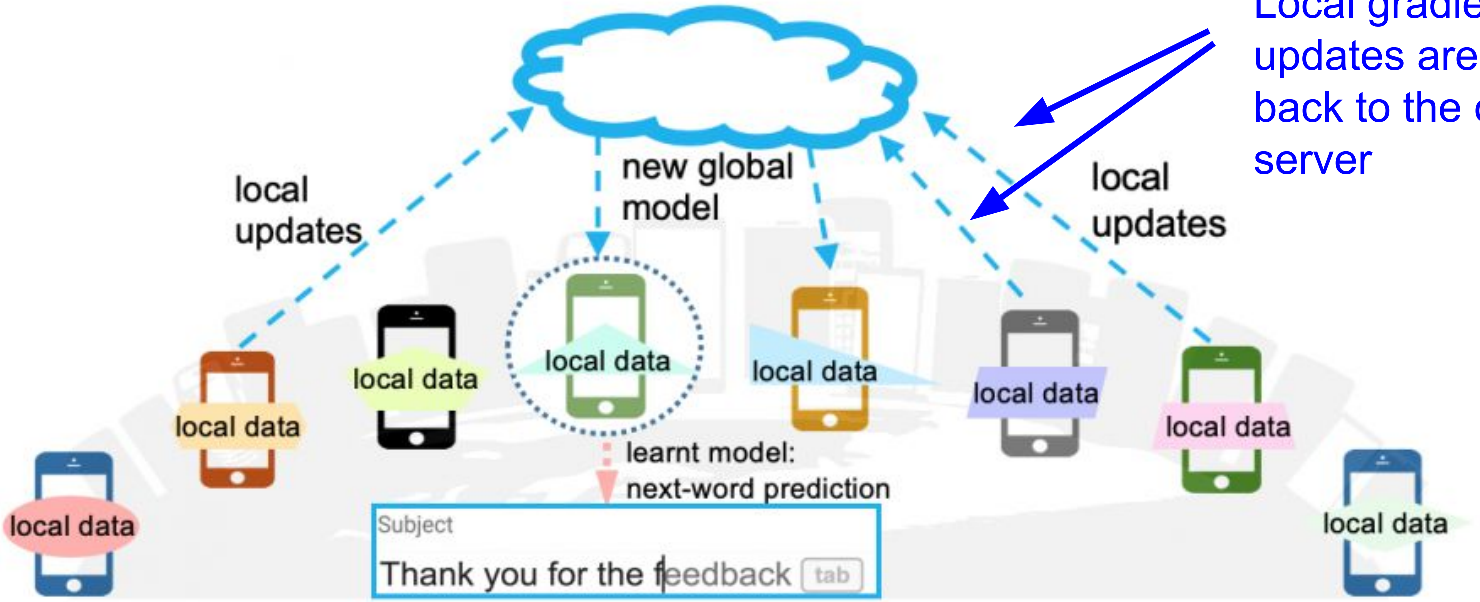


Local gradient updates are computed on local data after 1 or several iterations of gradient descent

Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones

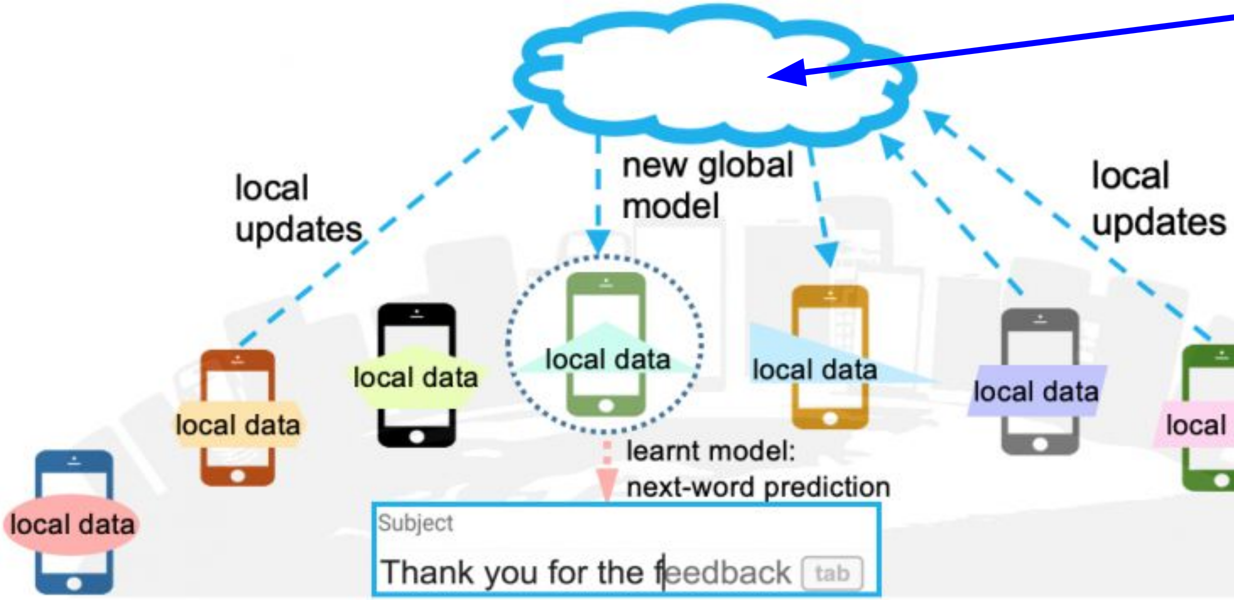


Local gradient updates are shipped back to the central server

Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones

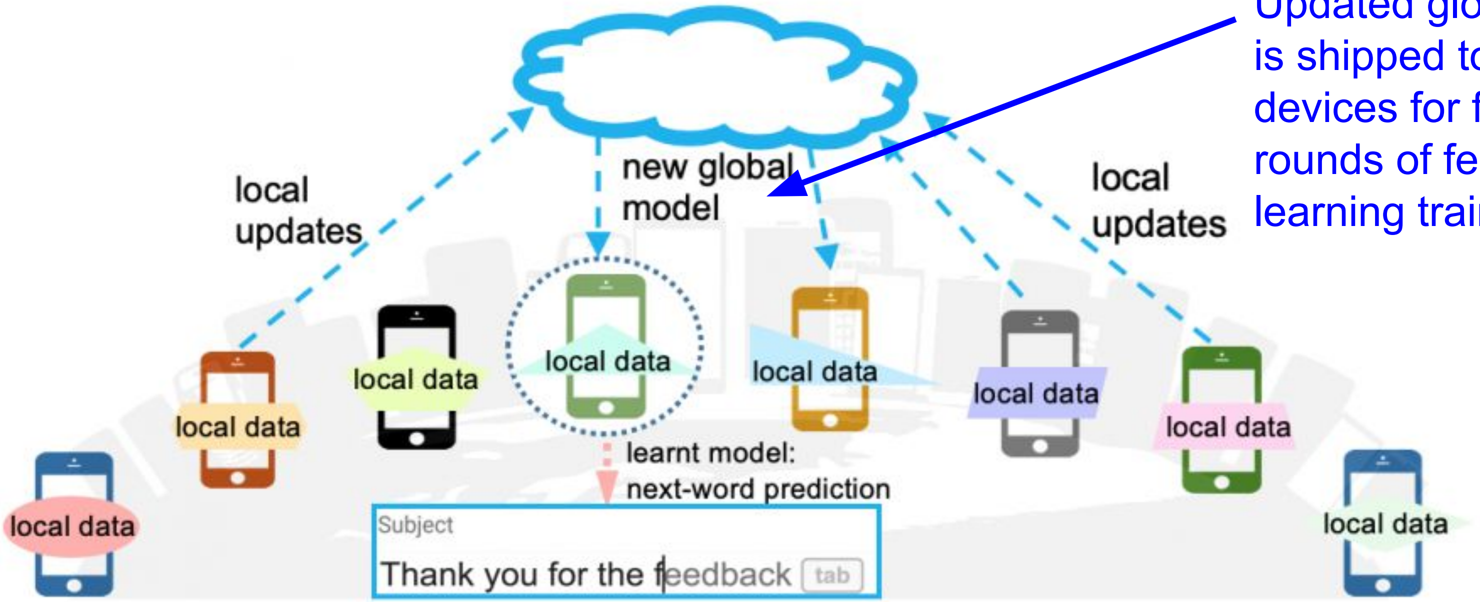


Local gradient updates are combined and used to update the global model

Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a next-word prediction model from many individual cell phones



Updated global model is shipped to local devices for future rounds of federated learning training

Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Federated Learning

- Example: learning a personalized healthcare model from data across different healthcare organizations

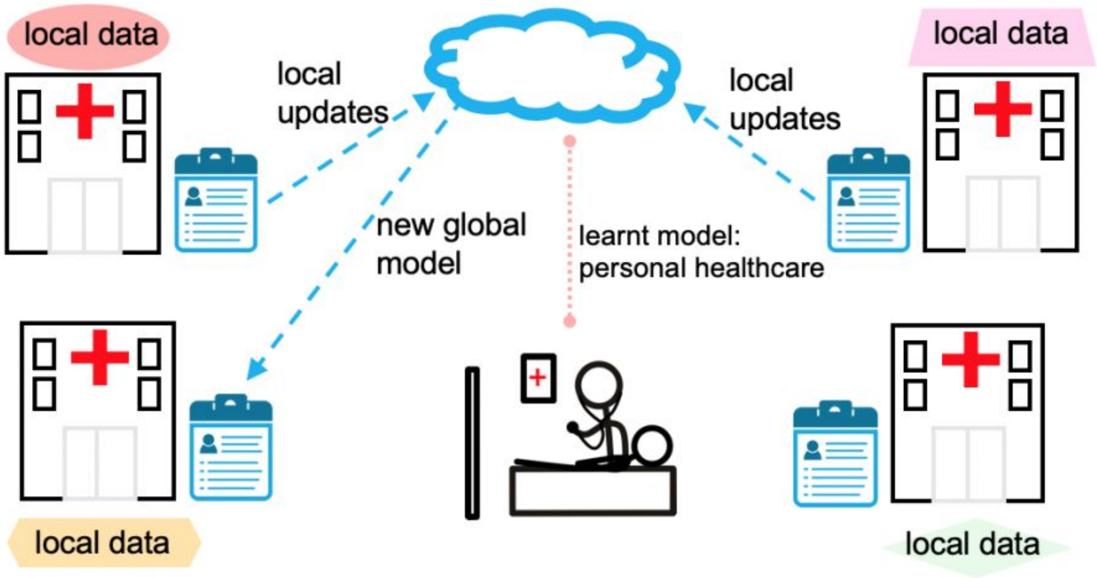
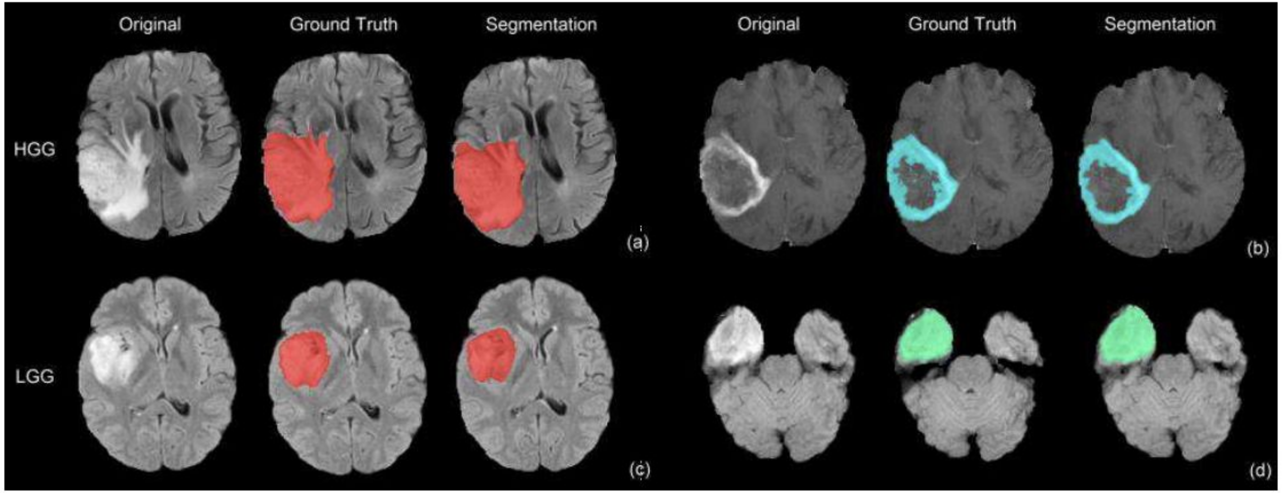


Figure credit: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

From earlier: BRATS brain tumor segmentation dataset

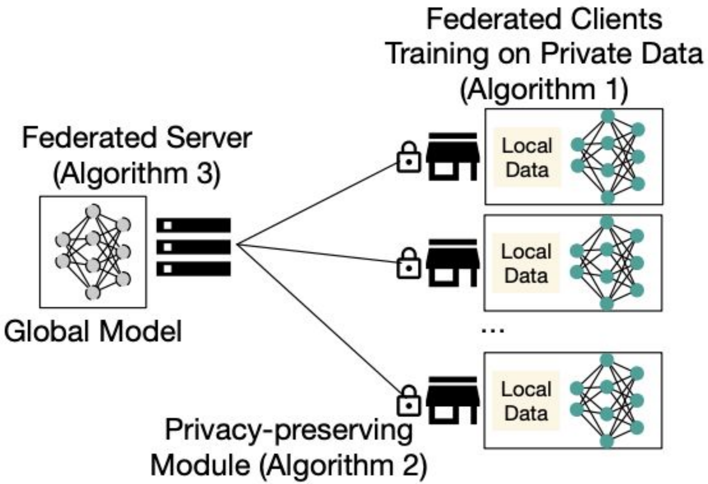
- Segmentation of tumors in brain MR image slices
- BRATS 2015 dataset: 220 high-grade brain tumor and 54 low-grade brain tumor MRIs
- U-Net architecture, Dice loss function

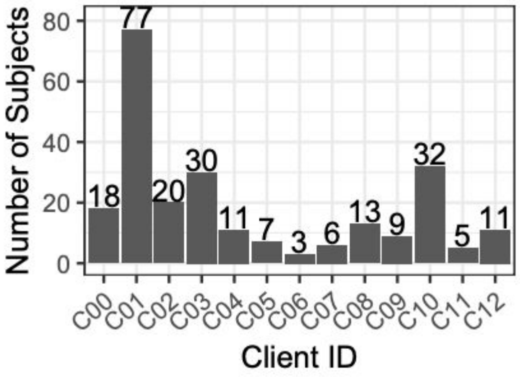


Dong et al. Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. MIUA, 2017.

Li et al. 2019

- NVIDIA Clara's Federated Learning system for medical imaging data
- Used federated learning to train segmentation model on BRATS
- Achieved comparable performance to non-federated learning, training somewhat slower but data "silos" preserved

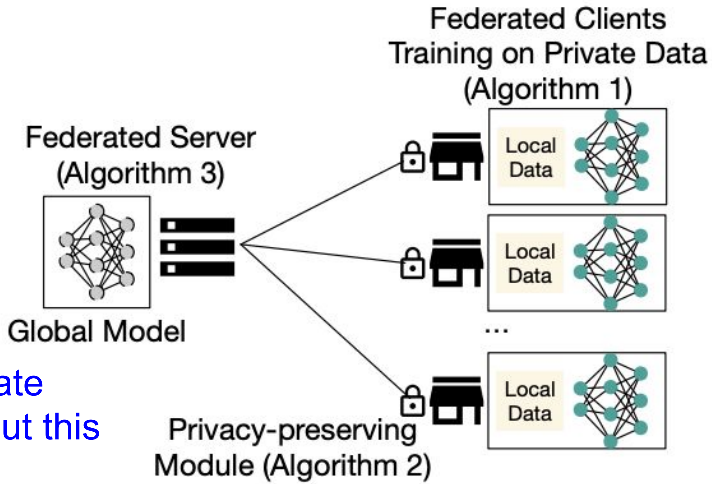


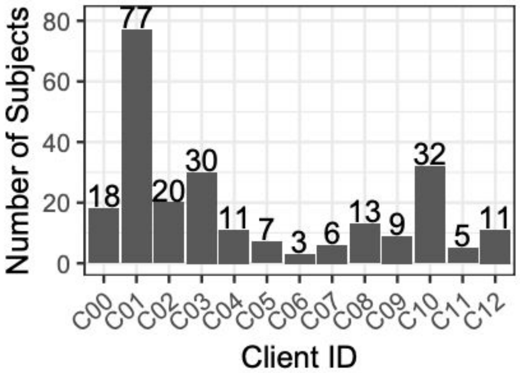


Li et al. Privacy-preserving Federated Brain Tumour Segmentation, 2019.

Li et al. 2019

- NVIDIA Clara's Federated Learning system for medical imaging data
- Used federated learning to train segmentation model on BRATS
- Achieved comparable performance to non-federated learning, training somewhat slower but data "silos" preserved





Also differentially private version… will talk about this in a moment

Li et al. Privacy-preserving Federated Brain Tumour Segmentation, 2019.

Privacy: HIPAA

Health Insurance Portability and Accountability Act (HIPAA), 1996: created “Privacy Rule” for how healthcare entities must protect the privacy of patients’ medical information

18 HIPAA identifiers (Protected Health Information):

Figure credit: https://www.jet-software.com/en/data-masking-hipaa/

Risks of data re-identification

Data triangulation: a person may be de-identified as to one data set, but the knowledge that they are a member of another available data set may allow them to be reidentified

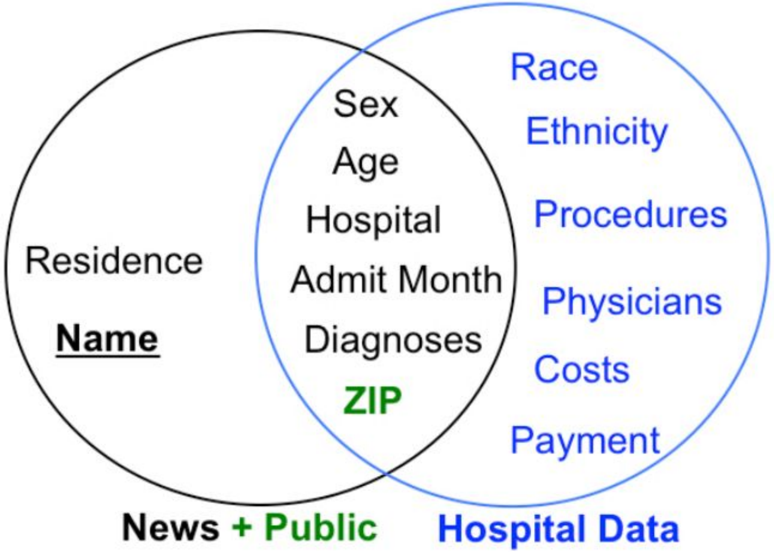


Figure credit: Sweeney et al. Matching Known Patients to Health Records in Washington State Data, 2011.

Matching Known Patients to Health Records in Washington State Data

News stories (e.g., those containing the word “hospitalized”) contain identifying information that could be used to identify medical records in the state medical record database, for 43% of studied cases

Distribution of values for fields harvested from news stories

Number of Fields		Name or Street	Gender	Type	Age	General Address	Hospital	Details	Number of Subjects	Totals
3		■	■			■			1	1
4	a	■	■	■				■	5	14
	b	■	■	■	■				7	
	c	■	■	■		■			1	
	d	■	■		■	■			1	
5	a	■	■	■	■			■	6	27
	b	■	■	■		■		■	7	
	c	■	■	■	■	■			4	
	d	■	■	■			■	■	6	
	e	■	■	■	■		■		3	
	f	■	■	■		■	■		1	
6	a	■	■	■	■		■	■	4	31
	b	■	■	■	■	■		■	9	
	c	■	■	■	■	■	■		17	
	d	■	■	■		■	■	■	1	
7		■	■	■	■	■	■	■	17	17
								Totals	90	90

Sweeney. Matching Known Patients to Health Records in Washington State Data, 2011.

Matching Known Patients to Health Records in Washington State Data

MAN, 61, THROWN FROM MOTORCYCLE

A 61-year-old Soap Lake man was hospitalized Saturday afternoon after he was thrown from his motorcycle. Raymond Boylston was riding his 2003 Harley-Davidson north on Highway 25, when he failed to negotiate a curve to the left. His motorcycle became airborne before landing in a wooded area. Boylston was thrown from the bike; he was wearing a helmet during the 12:24 p.m. incident. He was taken to Lincoln Hospital. [Spokesman Review 10/23/2011]

Figure 1. Sample extract of a news story that contains *name*, *age*, *residential information*, *hospital*, *incident date*, and *type of incident*.

NEWS STORIES

	Number of Subjects	Percent
Motor Vehicle	51	57%
Assault	12	13%
Medical	13	14%
Other	14	16%
Totals	90	

Table 2. Distribution of news stories by type of incident for 90 subjects.

Sweeney. Matching Known Patients to Health Records in Washington State Data, 2011.

Matching Known Patients to Health Records in Washington State Data

Hospital	162: Sacred Heart Medical Center in Providence
Admit Type	1: Emergency
Type of Stay	1: Inpatient
Length of Stay	6 days
Discharge Date	Oct-2011
Discharge Status	6: Dsch/Trfn to home under the care of an health service organization
Charges	$71708.47
Payers	1: Medicare
	6: Commercial insurance
	625: Other government sponsored patients
Emergency Codes	E8162: motor vehicle traffic accident due to loss of control; loss control mv-mocycl

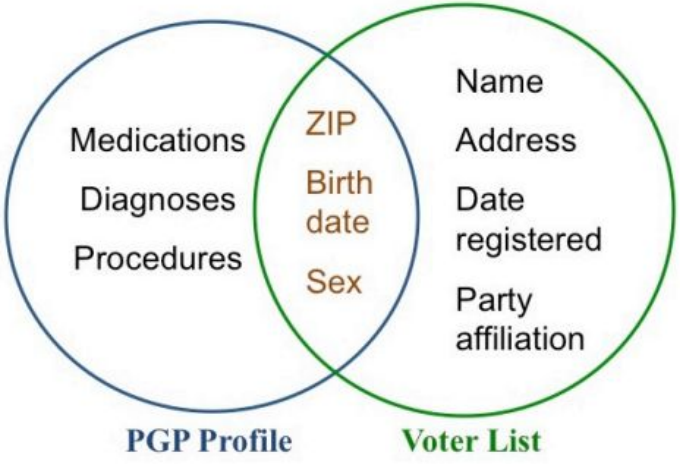
Diagnosis Codes	80843: closed fracture of other specified part of pelvis
	51851: pulmonary insufficiency following trauma & surgery
	86500: injury to spleen without mention of open wound into cavity
	80705: closed fracture of rib(s); fracture five ribs-close
	5849: acute renal failure; unspecified

Age in Years	60
Age in Months	725
Gender	Male
ZIP	98851
State Reside	WA
Race/Ethnicity	White, Non-Hispanic
Procedure Codes	5781: Suture bladder laceration
	7939: 7919: Open/Closed reduction of fracture of other specified bone
Physicians	…
…	…

Sweeney. Matching Known Patients to Health Records in Washington State Data, 2011.

Identifying Participants in the Personal Genome Project by Name

Linked demographics information in the Personal Genome Project (PGP) to public records such as voter lists, to correctly identify 84 to 97% of profiles for which guessed names were provided to PGP staff



	Wrong	Total	Correct%
Name	19	103	82%
Voter Data	9	130	93%
Public Records	20	156	87%

Table 2. Correctness of different re-identification strategies. Errors in matching embedded names and other strategies are due primarily to uses of nicknames rather than real names.

Sweeney. Identifying Participants in the Personal Genome Project by Name, 2013.

K-anonymity

A data release provides k-anonymity protection if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release.

	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

Figure 2 Example of *k*-anonymity, where *k*=2 and QI={*Race*, *Birth*, *Gender*, *ZIP*}

Sweeney. K-anonymity: a model for protecting privacy. 2002.

K-anonymity

Race	BirthDate	Gender	ZIP	Problem
black	9/20/1965	male	02141	short of breath
black	2/14/1965	male	02141	chest pain
black	10/23/1965	female	02138	painful eye
black	8/24/1965	female	02138	wheezing
black	11/7/1964	female	02138	obesity
black	12/1/1964	female	02138	chest pain
white	10/23/1964	male	02138	short of breath
white	3/15/1965	female	02139	hypertension
white	8/13/1964	male	02139	obesity
white	5/5/1964	male	02139	fever
white	2/13/1967	male	02138	vomiting
white	3/21/1967	male	02138	back pain

PT

2 k-anonymity tables (where k = 2)

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
person	1965	female	0213*	painful eye
person	1965	female	0213*	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1964	male	0213*	short of breath
person	1965	female	0213*	hypertension
white	1964	male	0213*	obesity
white	1964	male	0213*	fever
white	1967	male	02138	vomiting
white	1967	male	02138	back pain

GT1

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
black	1965	female	02138	painful eye
black	1965	female	02138	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1960-69	male	02138	short of breath
white	1960-69	human	02139	hypertension
white	1960-69	human	02139	obesity
white	1960-69	human	02139	fever
white	1960-69	male	02138	vomiting
white	1960-69	male	02138	back pain

GT3

Sweeney. K-anonymity: a model for protecting privacy. 2002.

Re-identification from ML models

- White-box (as opposed to black-box) setting: have access to model parameters, e.g. local model downloaded on device to run inference
- Model inversion attack: can use gradient descent if model parameters are available, to infer sensitive features

Algorithm 1 Inversion attack for facial recognition models.

```
function MI-FACE(label, α, β, γ, λ)
    c(x) ≝ 1 − f̃_label(x) + AUXTERM(x)
    x_0 ← 0
    for i ← 1 … α do
        x_i ← PROCESS(x_{i−1} − λ · ∇c(x_{i−1}))
        if c(x_i) ≥ max(c(x_{i−1}), …, c(x_{i−β})) then
            break
        if c(x_i) ≤ γ then
            break
    return [arg min_{x_i}(c(x_i)), min_{x_i}(c(x_i))]
```

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.

Fredrickson et al. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures, 2015.

Differential privacy

Key idea: output for a dataset, vs. the dataset with a difference for a single entry (e.g., one individual), is “hardly different”. Mathematical guarantees on this idea.

Abadi et al. Deep Learning with Differential Privacy, 2016.

Differential privacy

Key idea: output for a dataset, vs. the dataset with a difference for a single entry (e.g., one individual), is "hardly different". Mathematical guarantees on this idea.

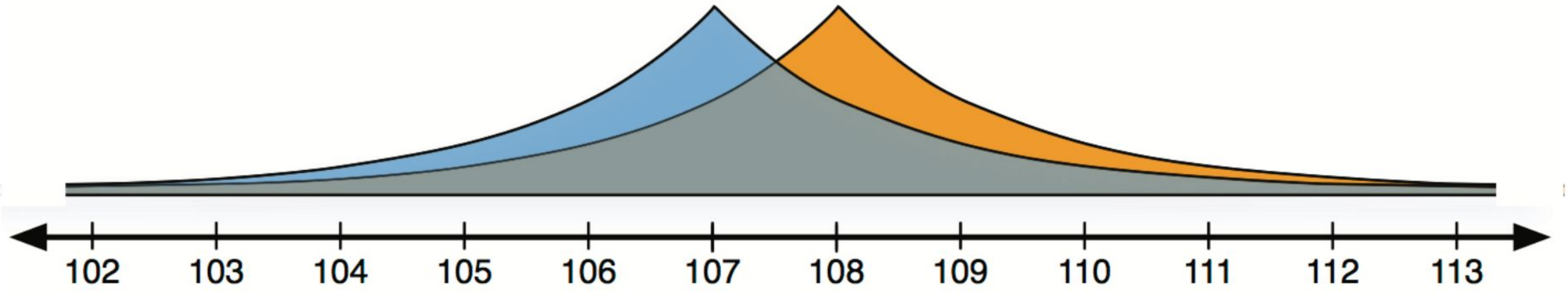
Definition 1. A randomized mechanism $\mathcal{M}\colon \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta.$$

Abadi et al. Deep Learning with Differential Privacy, 2016.

Differential privacy

Simple intuition behind how we can achieve differential privacy: adding noise!



Example of reporting a value with Laplacian noise added

Figure credit: https://github.com/frankmcsherry/blog/blob/master/posts/2016-02-03.md

Training differentially private deep learning models

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.

Abadi et al. Deep Learning with Differential Privacy, 2016.

Training differentially private deep learning models

Compute gradient as usual

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.

Abadi et al. Deep Learning with Differential Privacy, 2016.

Training differentially private deep learning models

Clip the gradient

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.

Abadi et al. Deep Learning with Differential Privacy, 2016.

Training differentially private deep learning models

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.

Add noise for differential privacy →

Abadi et al. Deep Learning with Differential Privacy, 2016.

Training differentially private deep learning models

Definition 1. A randomized mechanism $\mathcal{M}\colon \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta.$$

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.

Compute overall privacy cost →

Abadi et al. Deep Learning with Differential Privacy, 2016.

Privacy Aggregation of Teacher Ensembles (PATE)

Approach to combine data from multiple disjoint sensitive populations, with privacy guarantees

PATE Teacher model

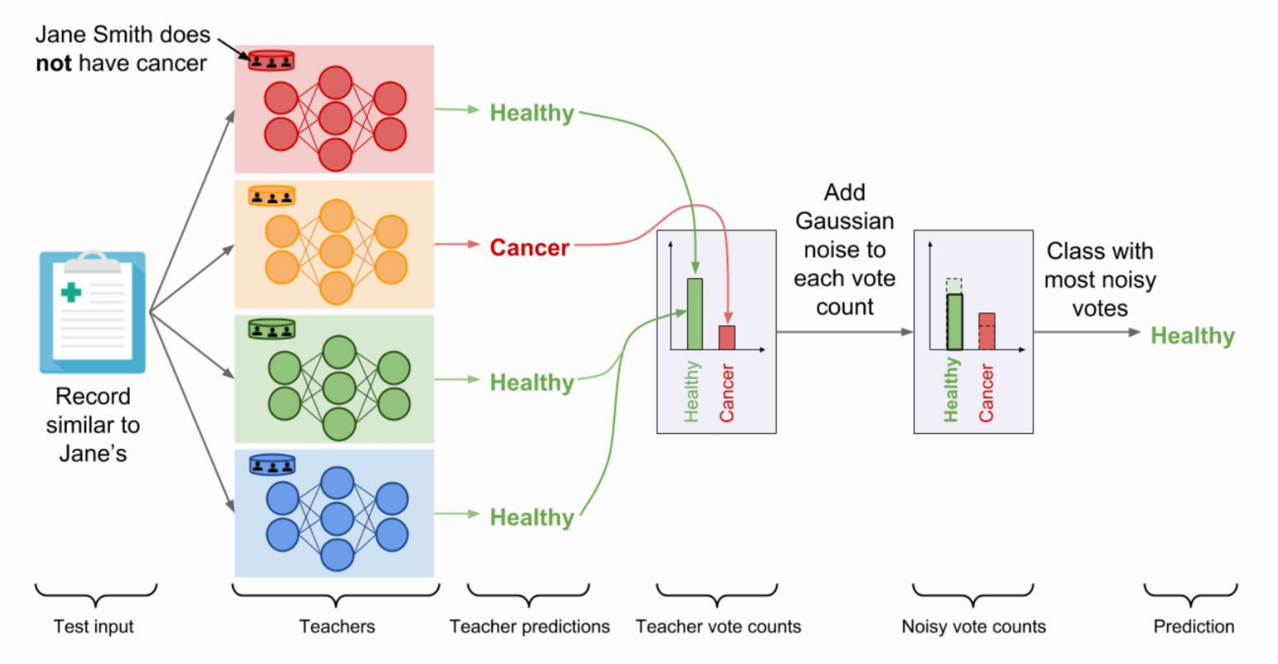


Figure credit: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Papernot et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Data, 2017.

Privacy Aggregation of Teacher Ensembles (PATE)

Approach to combine data from multiple disjoint sensitive populations, with privacy guarantees

Train separate classifiers from disjoint data sets -- no privacy guarantees yet

PATE Teacher model

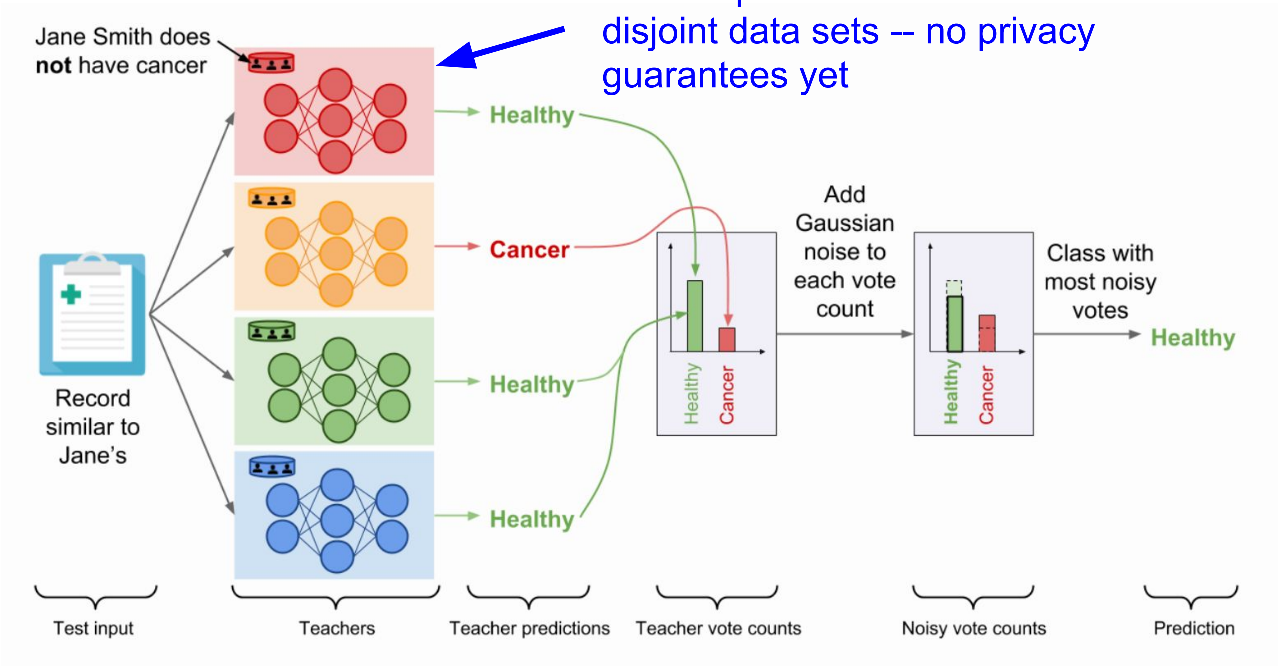


Figure credit: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Papernot et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Data, 2017.

Privacy Aggregation of Teacher Ensembles (PATE)

Approach to combine data from multiple disjoint sensitive populations, with privacy guarantees

PATE Teacher model

To get a privacy-preserving prediction, first obtain predictions from all distinct classifiers

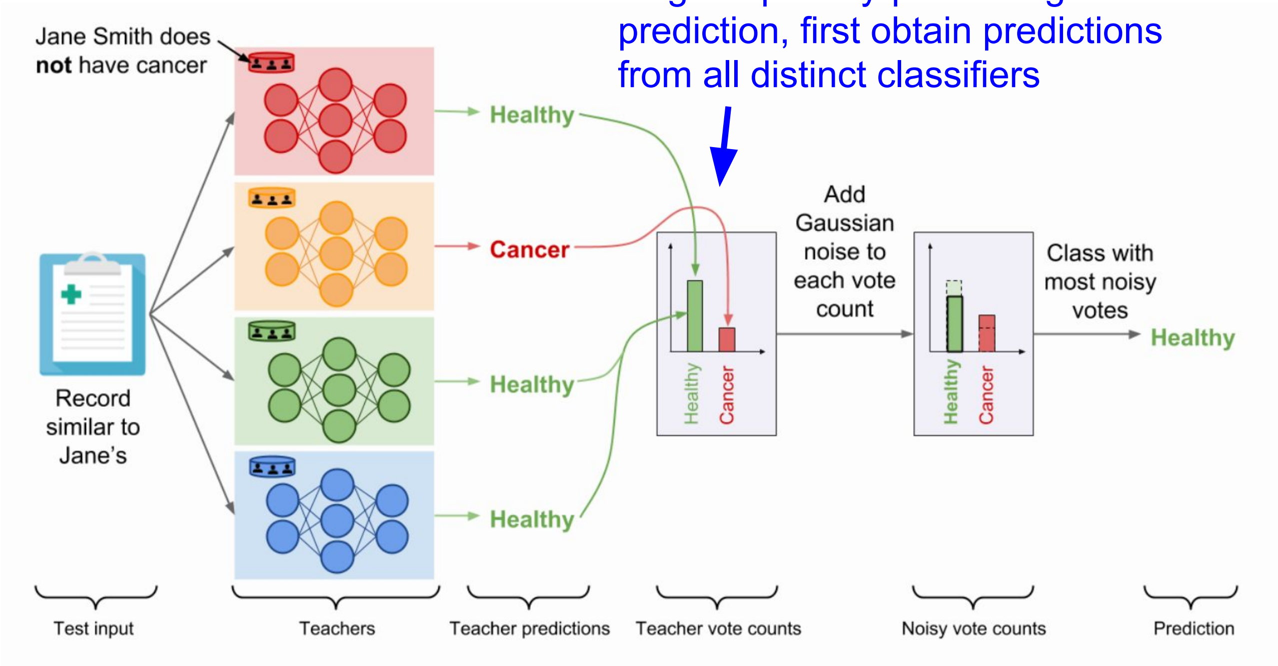


Figure credit: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Papernot et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Data, 2017.

Privacy Aggregation of Teacher Ensembles (PATE)

Approach to combine data from multiple disjoint sensitive populations, with privacy guarantees

PATE Teacher model

Then add noise to the vote histogram (giving differential privacy guarantees), and take the class with the most votes as the final output

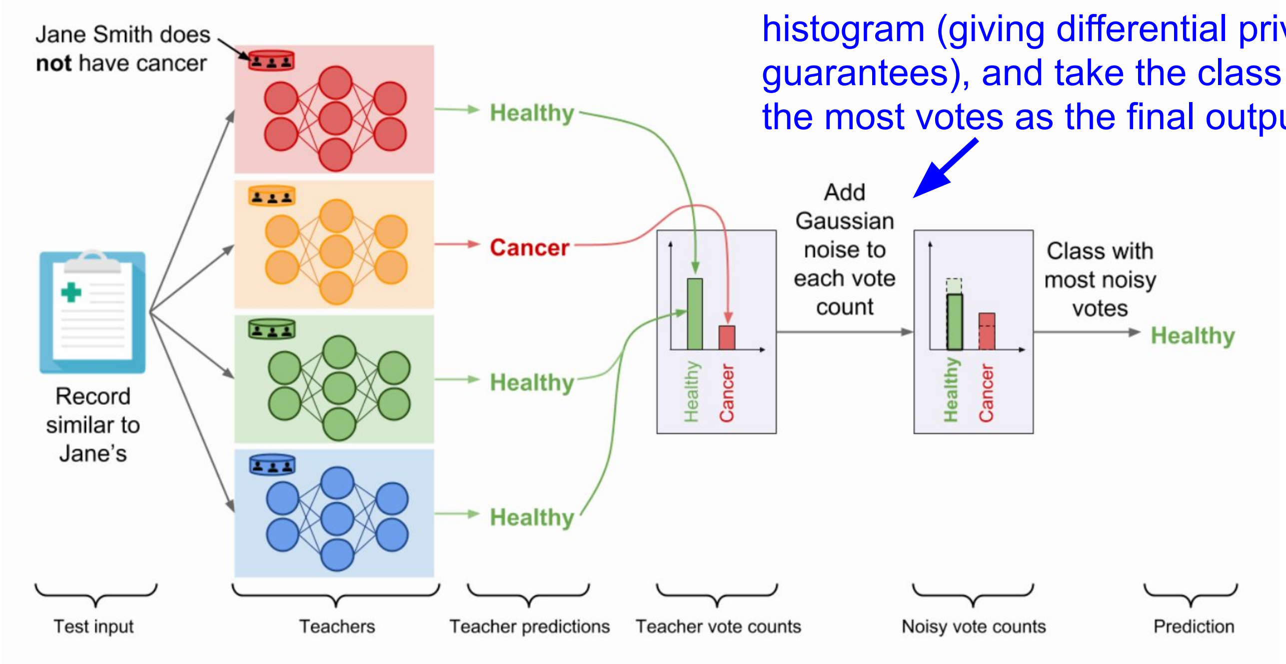


Figure credit: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Papernot et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Data, 2017.

Privacy Aggregation of Teacher Ensembles (PATE)

This teacher model alone can still be compromised if too many queries are performed (privacy cost builds up with each query, so privacy guarantees become meaningless with too many queries), or if model parameters are made accessible (and attackable) e.g. distributed in local application

PATE Student model uses public data to train a model replicating noisy aggregated teacher outputs

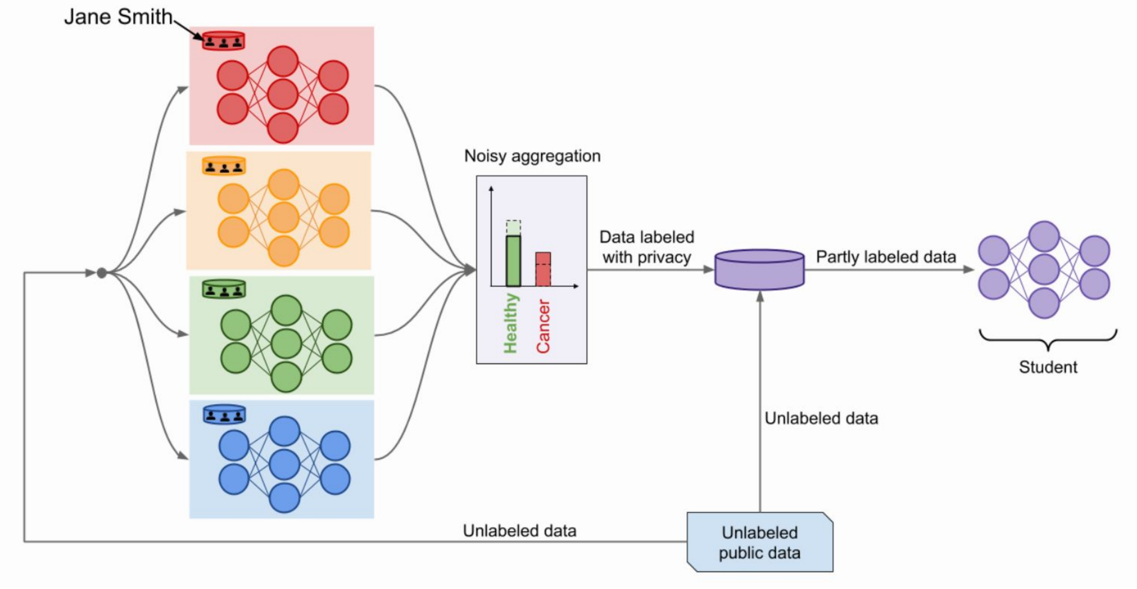


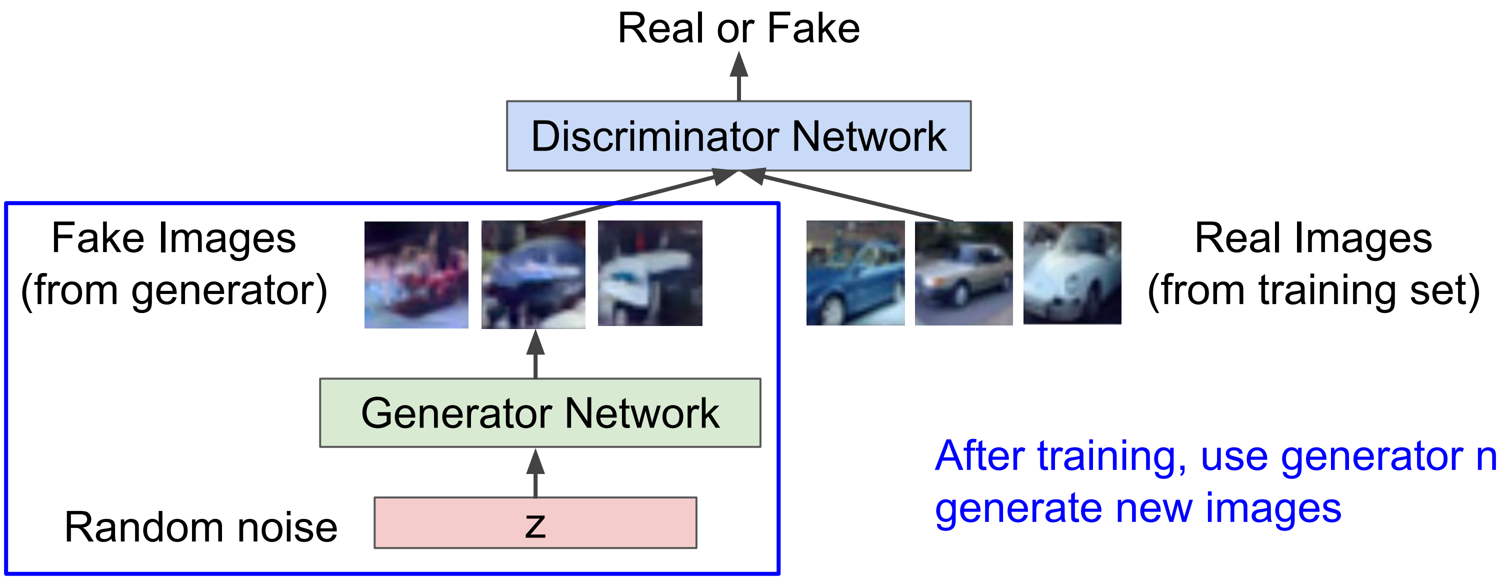
Figure credit: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Papernot et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Data, 2017.

Remember GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

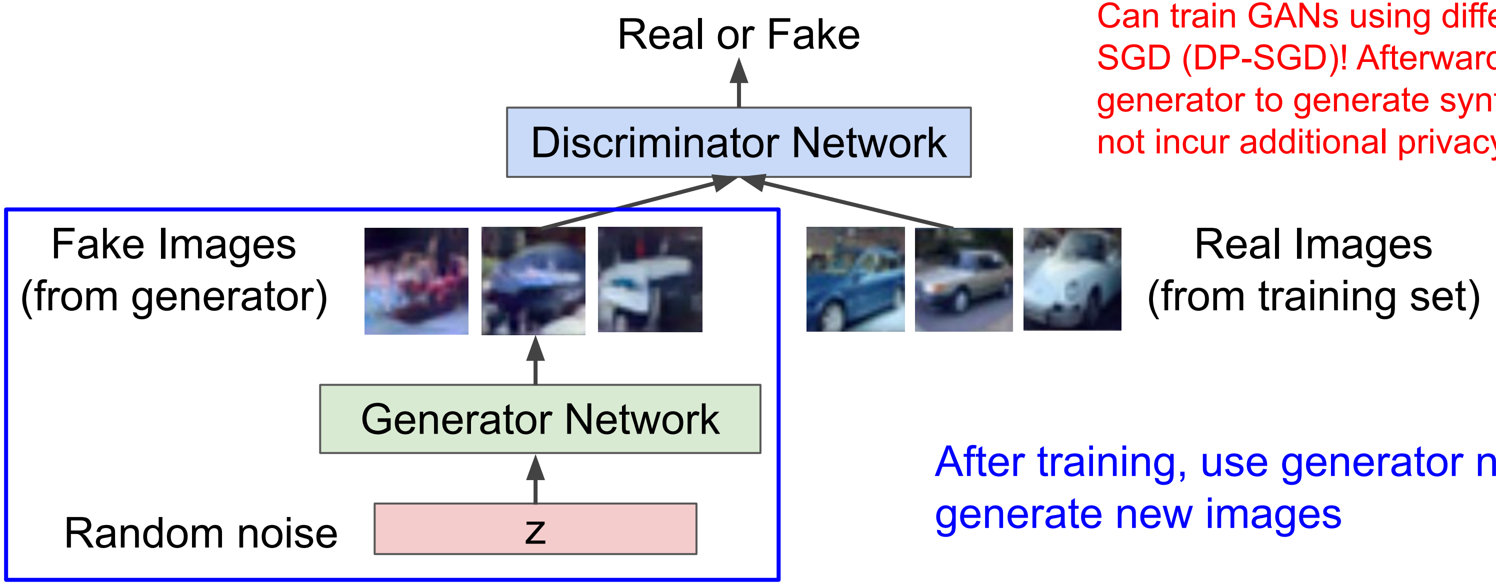


After training, use generator network to generate new images

Remember GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



Can train GANs using differentially private SGD (DP-SGD)! Afterwards, using the generator to generate synthetic data does not incur additional privacy cost

After training, use generator network to generate new images

Xie et al. Differentially Private Generative Adversarial Network, 2018.

Can work with differential privacy within deep learning frameworks

Implementation of DP-SGD

```
optimizer = optimizers.dp_optimizer.DPGradientDescentGaussianOptimizer(
    l2_norm_clip=FLAGS.l2_norm_clip,
    noise_multiplier=FLAGS.noise_multiplier,
    num_microbatches=FLAGS.microbatches,
    learning_rate=FLAGS.learning_rate,
    population_size=60000)
train_op = optimizer.minimize(loss=vector_loss)
```

Utilities for calculating epsilon

```
epsilon = get_privacy_spent(orders, rdp, target_delta=1e-5)[0]
```

https://blog.tensorflow.org/2019/03/introducing-tensorflow-privacy-learning.html
http://www.cleverhans.io/privacy/2019/03/26/machine-learning-with-differential-privacy-in-tensorflow.html

Today we covered:

- Distributed Learning and Federated Learning
- Privacy and Differential Privacy

Next time: Guest lecture from **Mohit Kaushal**, discussing AI in healthcare in industry and in public policy