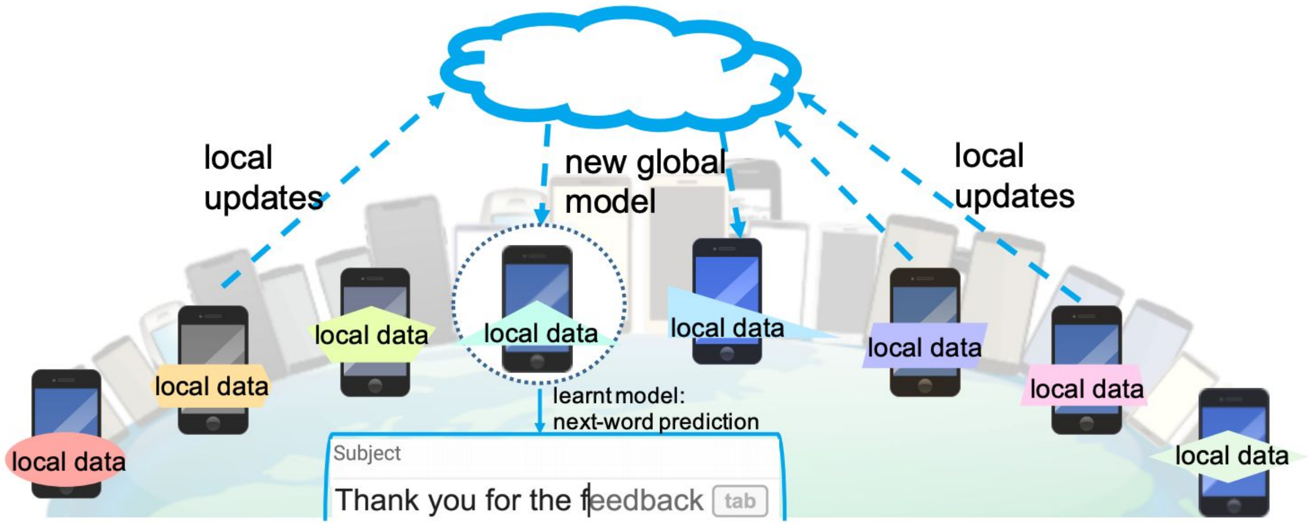
Carnegie Mellon University

Guest Lecture:
Federated Learning

Tianshu Huang – PhD Student @ CMU ECE

(1) From Distributed Optimization to Federated Learning
(2) Research Topics in Federated Learning
(3) Why Federated Learning?

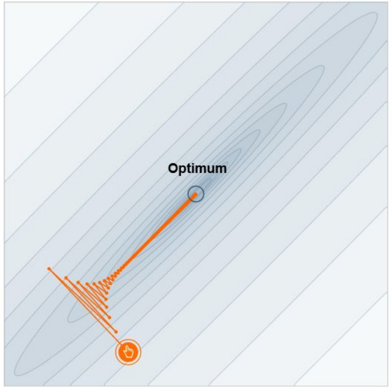
From **Distributed** Optimization to **Federated** Learning

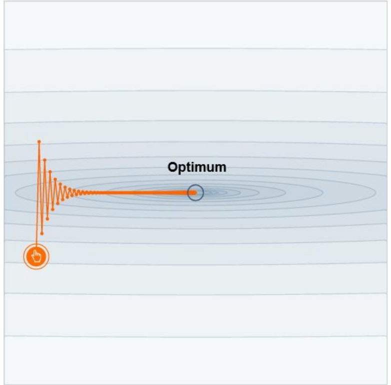


Gradient Descent

Gradient Descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \underbrace{\frac{\eta}{N}}_{\text{avg}} \sum_{n=1}^{N} \underbrace{\nabla \ell(h(\mathbf{x}_n), y_n))}_{\text{gradient}}$$





https://distill.pub/2017/momentum/

(Mini-batch Stochastic) **Gradient Descent**

Gradient Descent

dataset

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{N} \sum_{n=1}^{N} \nabla \ell(h(\mathbf{x}_n), y_n))$$

(mini-batch) Stochastic Gradient Descent

sample a batch

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{b} \sum_{l=1}^{b} \nabla \ell(\mathbf{w}_t, \xi_{t,l})$$

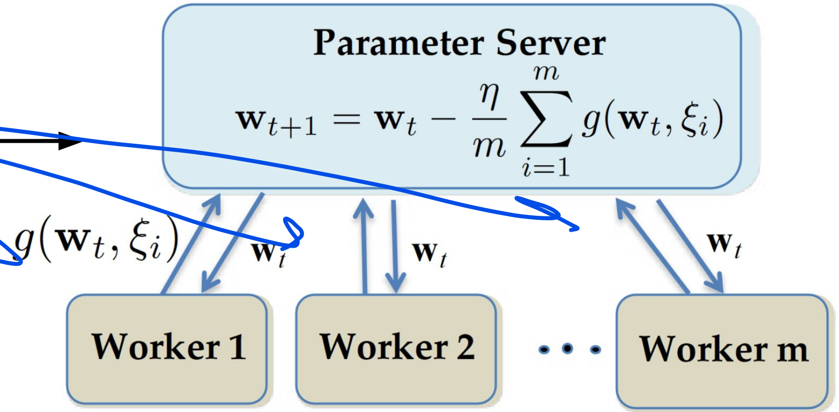
Carnegie Mellon University

(Mini-batch Stochastic) **Gradient Descent is Distributed?**

What's the Problem?

split across batch

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{b}\sum_{l=1}^{b} \boxed{\nabla \ell(\mathbf{w}_t, \xi_{t,l})}$$



1. unstable connection
2. connection speed
3. compute speed

*

Aside: How is Large-Scale Learning Done Today?

EleutherAI: GPT-NeoX-20B

- 12 workers (servers)
- 50GT/s x8 links to switches with 50GT/s x16 interconnect

Synchronous AdamW

- 20B params x 16 bit @ 400GT/s ~ 1s
- 1830 hours / 150k steps ~44 seconds per step

12 x 8 x A100 GPUs ~$1M

2x MQM8700-HS2R switches ~$40k

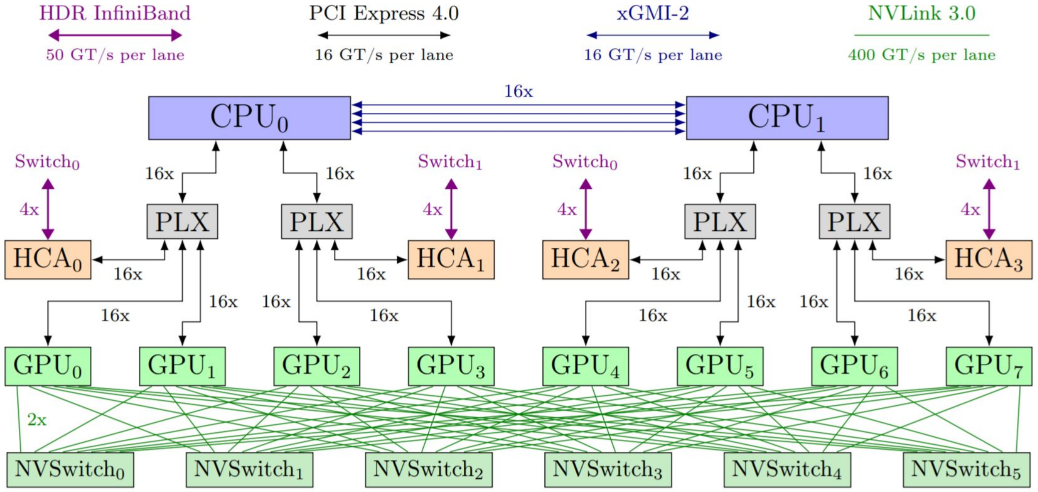
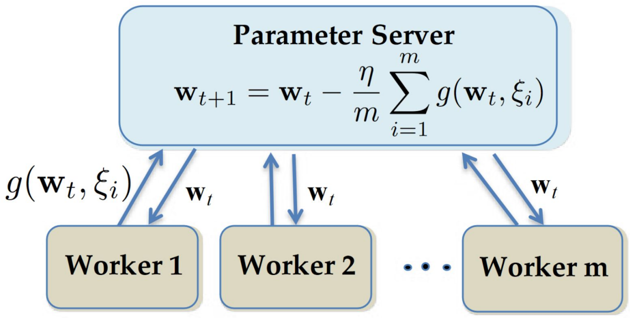


Figure 2: Architecture diagram of a single training node.

We trained GPT-NeoX-20B on twelve Supermicro AS-4124GO-NART servers, each with eight NVIDIA A100-SXM4-40GB GPUs and configured with two AMD EPYC 7532 CPUs. All GPUs can directly access the InfiniBand switched fabric through one of four ConnectX-6 HCAs for GPUDirect RDMA. Two NVIDIA MQM8700-HS2R switches—connected by 16 links—compose the spine of this InfiniBand network, with one link per node CPU socket connected to each switch. Figure 2 shows a simplified overview of a node as configured for training.

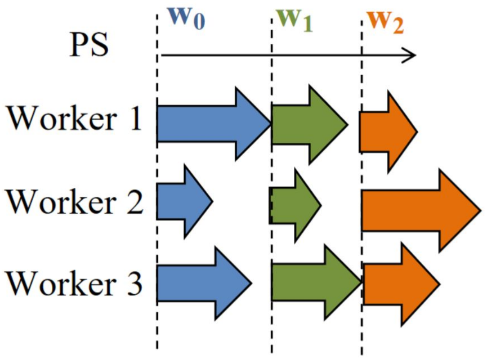
Distributed SGD (circa 2015)

Communication Cost:



- compress
- multiple steps before sync

Stragglers:



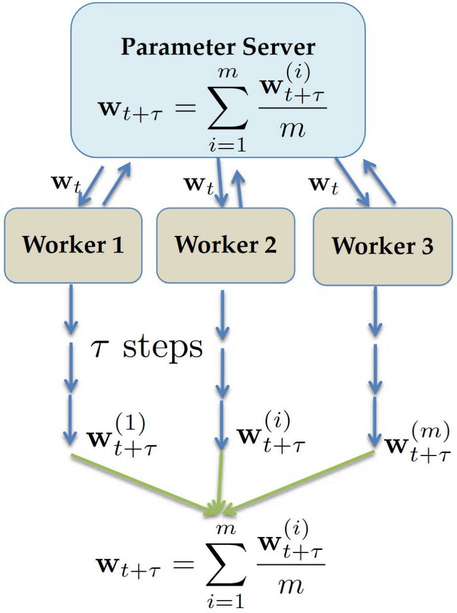
- asyncronous

Carnegie Mellon University

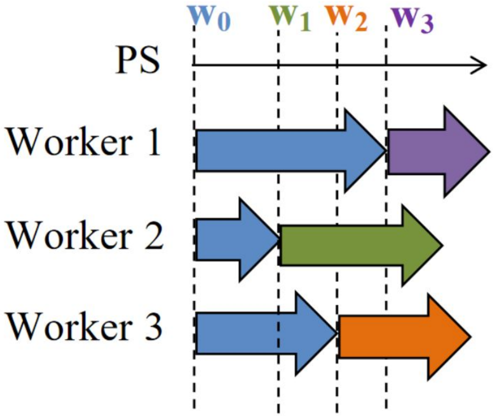
*

Distributed SGD (circa 2015)

Communication Cost: **Local Update SGD**

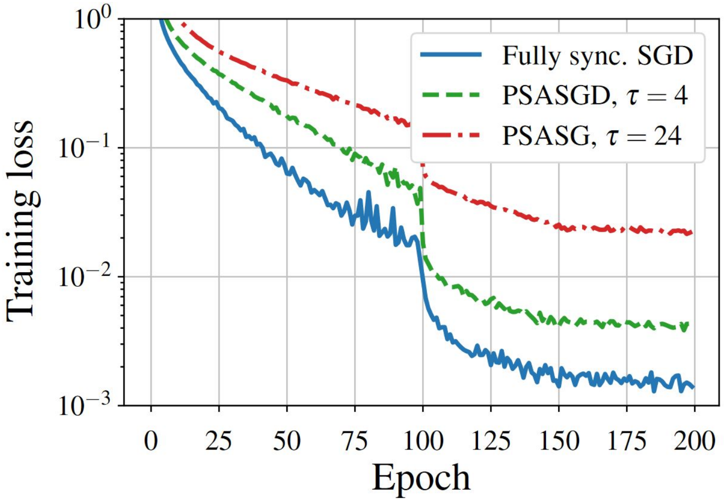


Stragglers: **Asynchronous SGD**

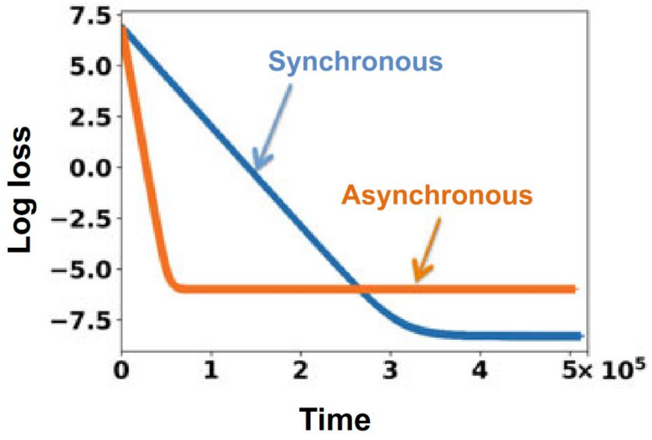


Distributed SGD (circa 2015)

Communication Cost: **Local Update SGD**



Stragglers: **Asynchronous SGD**

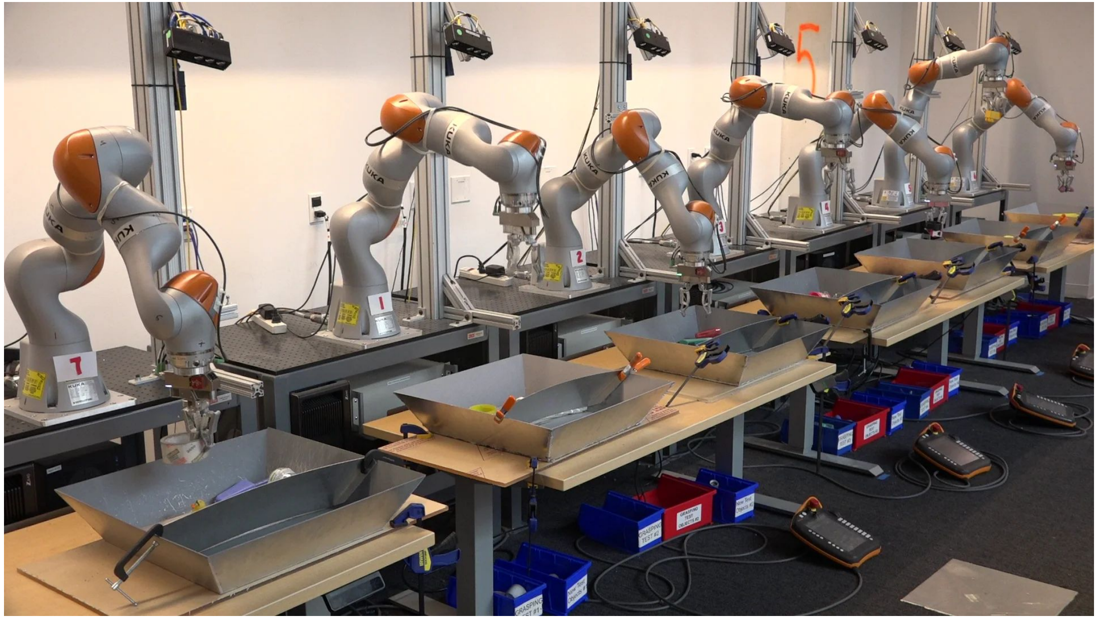


Carnegie Mellon University

Distributed SGD (circa 2023)

Reinforcement Learning

- Highly variable episode length
- Convergence speed is critical



https://everydayrobots.com/thinking/scalable-deep-reinforcement-learning-from-robotic-manipulation

Carnegie Mellon University

Federated Learning

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize w_0

for each round $t = 1, 2, \dots$ **do**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **in parallel do**

$\boxed{w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)}$ 1 or more SGD steps

average weights $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:** // *Run on client k*

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)

for each local epoch i from 1 to E **do**

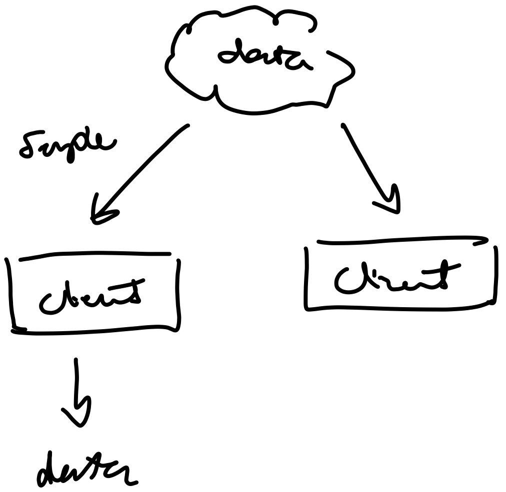
for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

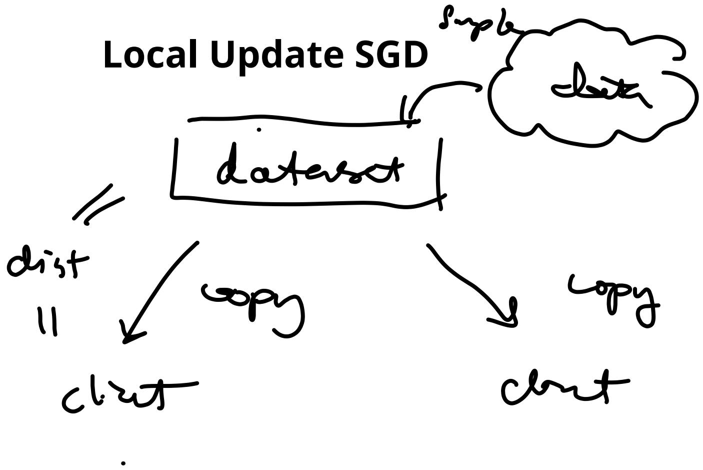
return w to server

Federated Learning vs Distributed SGD

Federated Averaging



Local Update SGD



Carnegie Mellon University

Why Federated Learning?

Advantages

- user privacy
- don't need to send data
- personalization
- pushing compute cost to users

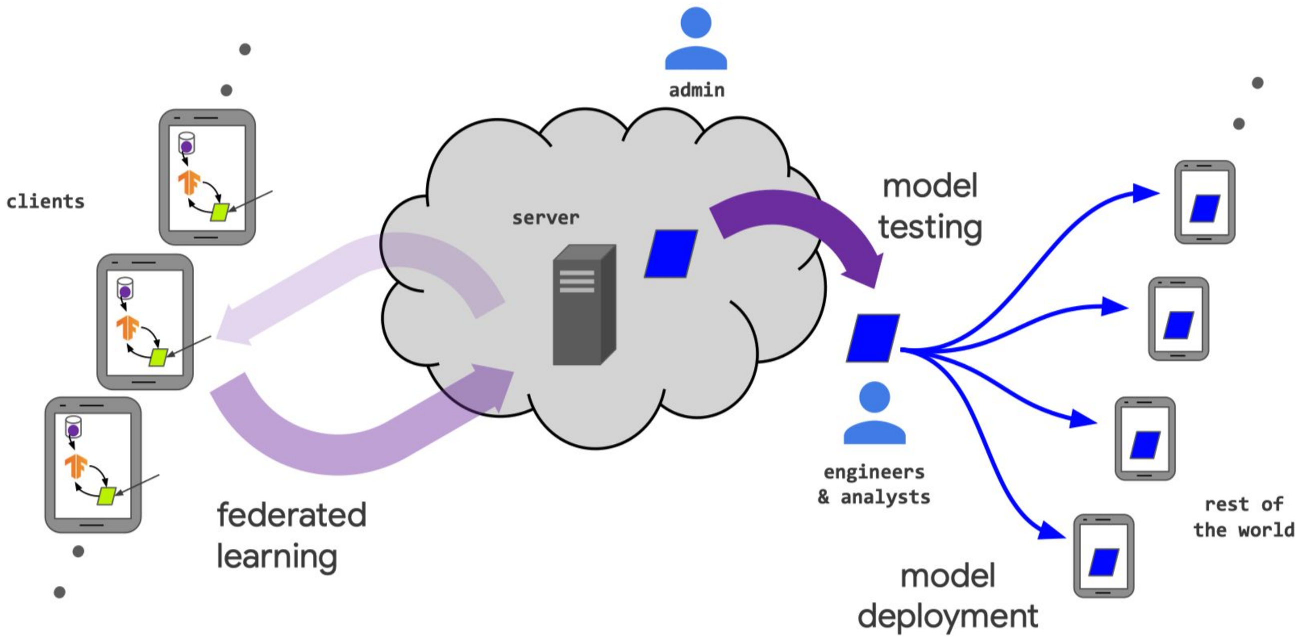
Disadvantages

- different distributions
  "data heterogeneity"
- need to send updates

Carnegie Mellon University

*

Research Topics in Federated Learning



Federated Learning: What could go wrong?

Data & Model Concerns

- convergence due to data heterogeneity
- adversarial attacks

Edge Systems Concerns

- powerful edge devices
  + edge resource consumption
-

*

Applications & Challenges in Federated Learning

"First Order" Challenges:

- **Data Heterogeneity**
- Compute Heterogeneity

"Second Order" Challenges:

- Communication cost / scalability
- Defense against attacks

Data Heterogeneity ("Non-IID")

What could go wrong?

- very different data
- different preprocessing
  + features hard to measure
- imbalance

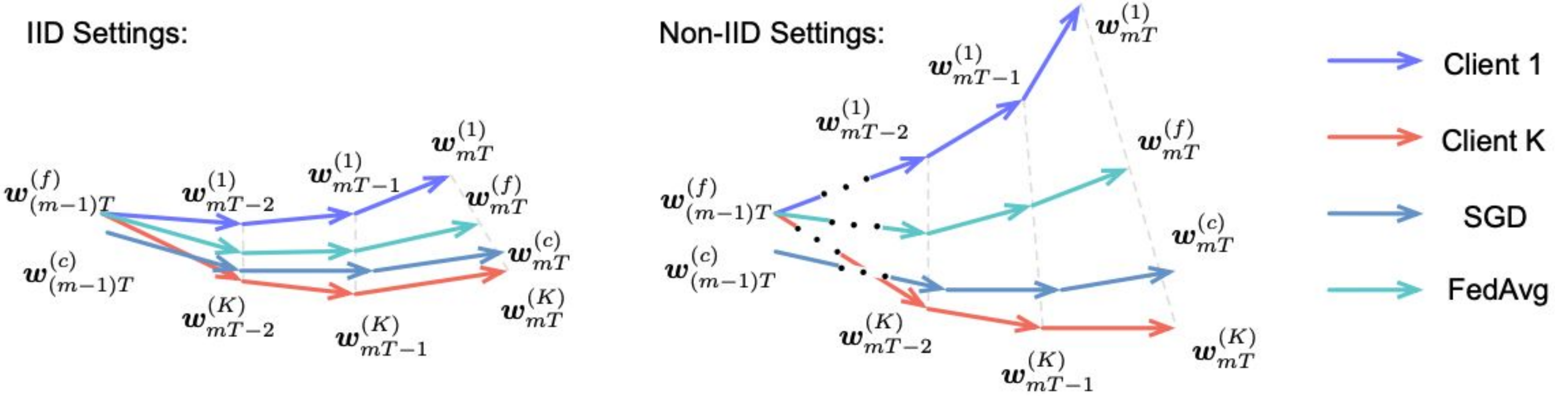
Carnegie Mellon University

*

Data Heterogeneity ("Non-IID")

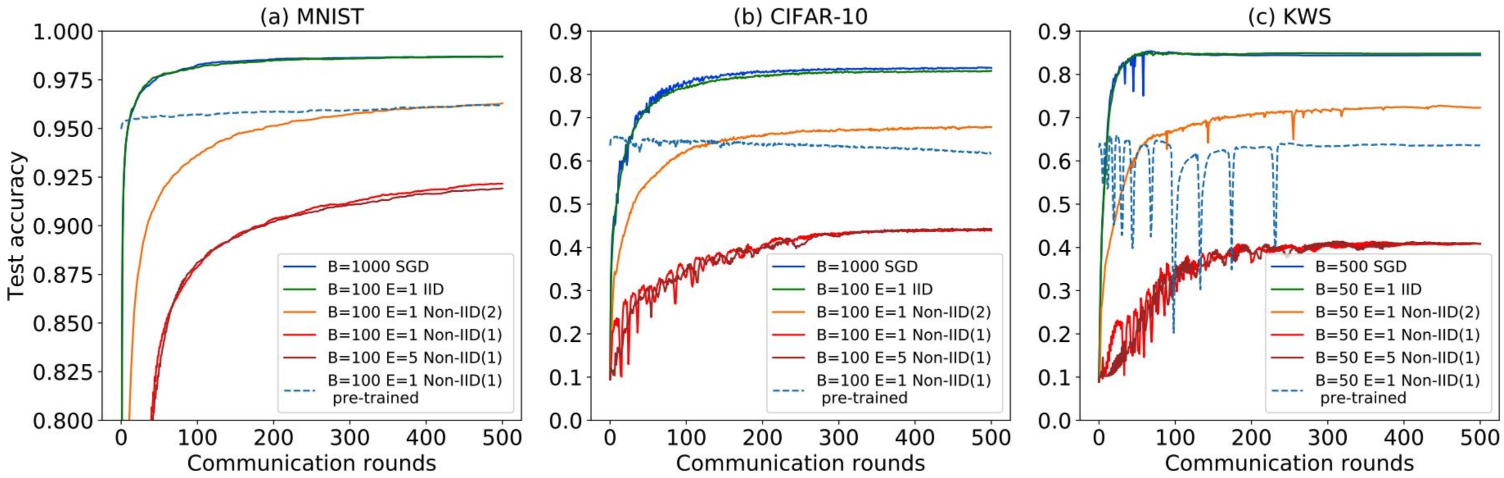
"Git Re-Basin"

What could go wrong?



Data Heterogeneity (“Non-IID”)

What could go wrong?



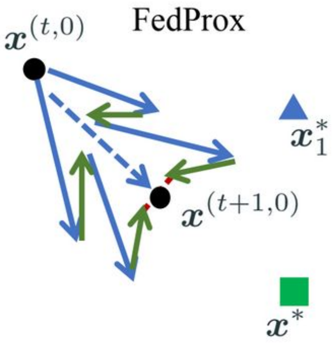
Carnegie Mellon University

Approaches to Data Heterogeneity

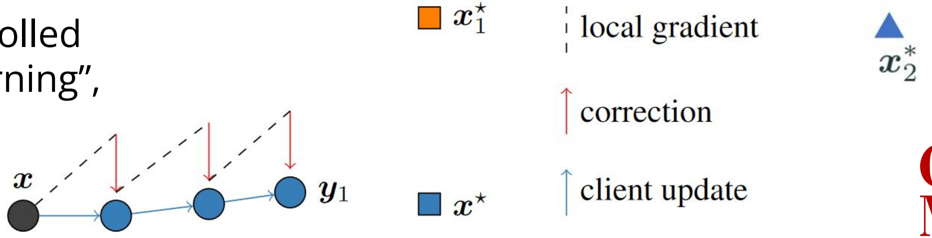
- Client selection ("Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies", 2020)

1. **Sample the Candidate Client Set.** The central server samples a candidate set $\mathcal{A}$ of d ($m \leq d \leq K$) clients without replacement such that client k is chosen with probability p_k, the fraction of data at the k-th client for $k = 1, \ldots K$.
2. **Estimate Local Losses.** The server sends the current global model $\overline{\mathbf{w}}^{(t)}$ to the clients in set $\mathcal{A}$, and these clients compute and send back to the central server their local loss $F_k(\overline{\mathbf{w}}^{(t)})$.
3. **Select Highest Loss Clients.** From the candidate set $\mathcal{A}$, the central server constructs the active client set $\mathcal{S}^{(t)}$ by selecting $m = \max(CK, 1)$ clients with the largest values $F_k(\overline{\mathbf{w}})$, with ties broken at random. These $\mathcal{S}^{(t)}$ clients participate in the training during the next round, consisting of iterations $t+1, t+2, \ldots t+\tau$.

- Regularization (FedProx) ("Federated Optimization in Heterogeneous Networks", 2018)



- SCAFFOLD ("Stochastic Controlled Averaging for Federated Learning", 2023)



Compute Heterogeneity

Client Selection (FedMCCS) (2021)

B. Problem Formulation

We formulate our problem as a bilevel maximization with knapsack and other constraints as follows:

Select as many clients as possible, such that:

$$\boxed{\max_{X_S} |X_S|}$$

subject to

(1) we do not exceed the resource budget

(2) we do not exceed the round time

$$\begin{cases} \boxed{\forall X_{f^i_{z=1}} \sum \mathrm{Util}^{X_{f_z}}_{r\in\{\mathrm{CPU},\mathrm{Memory},\mathrm{Energy}\}} < \mathrm{Budget}^{X_{f_z}}_r} [co_1] \\ \boxed{\forall X_{f^i_{z=1}} \sum \left(T^{X_{f_z}}_d + \mathrm{Util}^{X_{f_z}}_{r=T_{ud}} + T^{X_{f_z}}_{ul}\right) < T} [co_2] \end{cases}$$

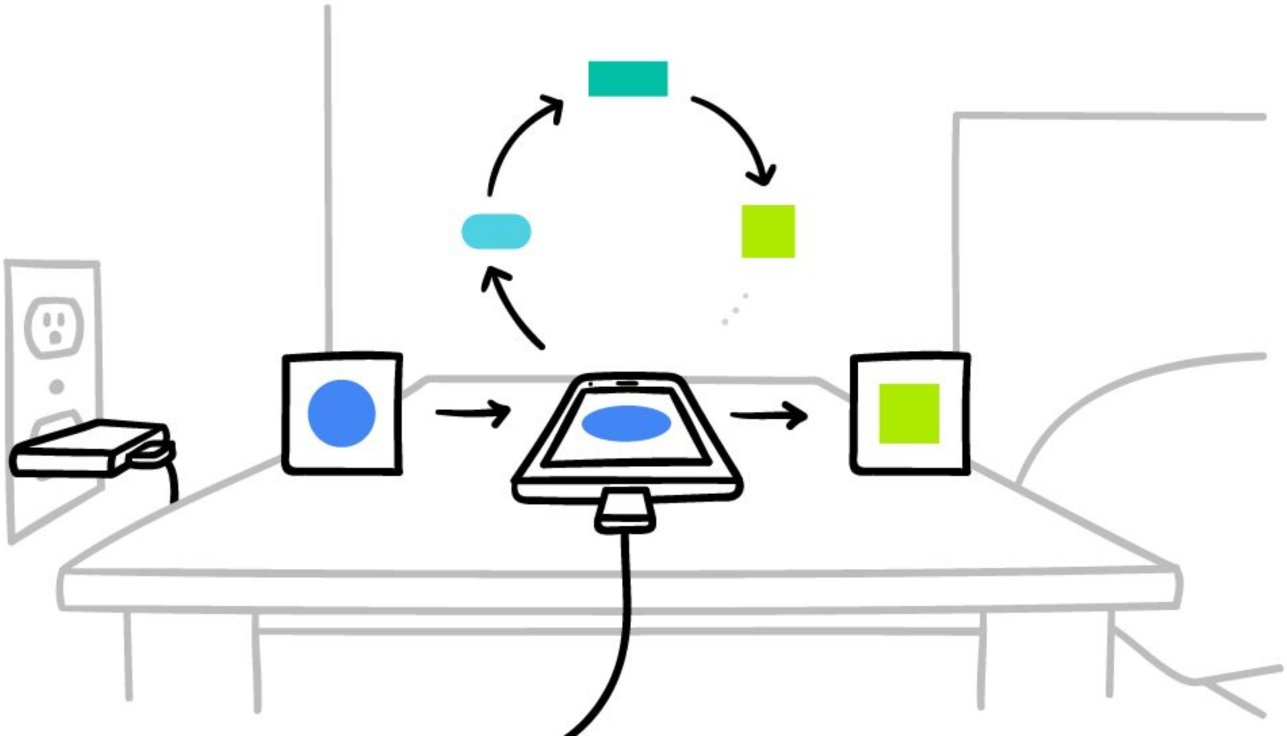
subject to

(3) selection also maximizes clients with minority classes

Percent of "abnormal" samples

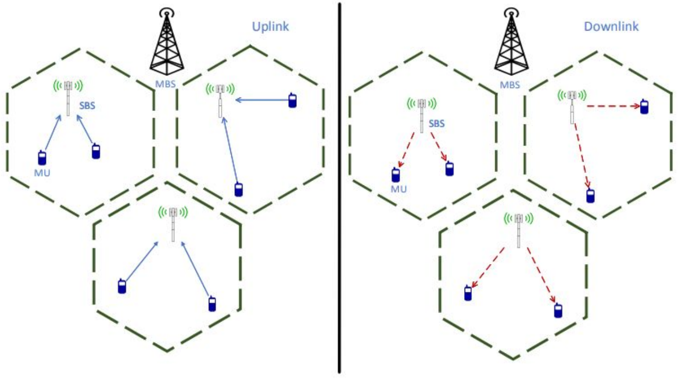
$$\boxed{\max ER_{X_{f^i_{z=1}}} = \left[\frac{|X_{f_z}.l_A|}{|X_{f_z}.l_A| + |X_{f_z}.l_N|} \times 100\right]} [co_3]. \qquad (1)$$

Heterogeneity: **Federated Learning @ Google**

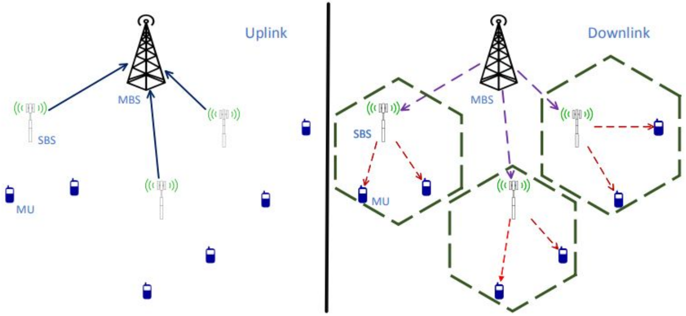


Communication & Scalability

Hierarchical Federated Learning (2019)



(a) Local gradient update



(b) Global model averaging

Carnegie Mellon University

Federated Learning Attacks

"Universal adversarial example"

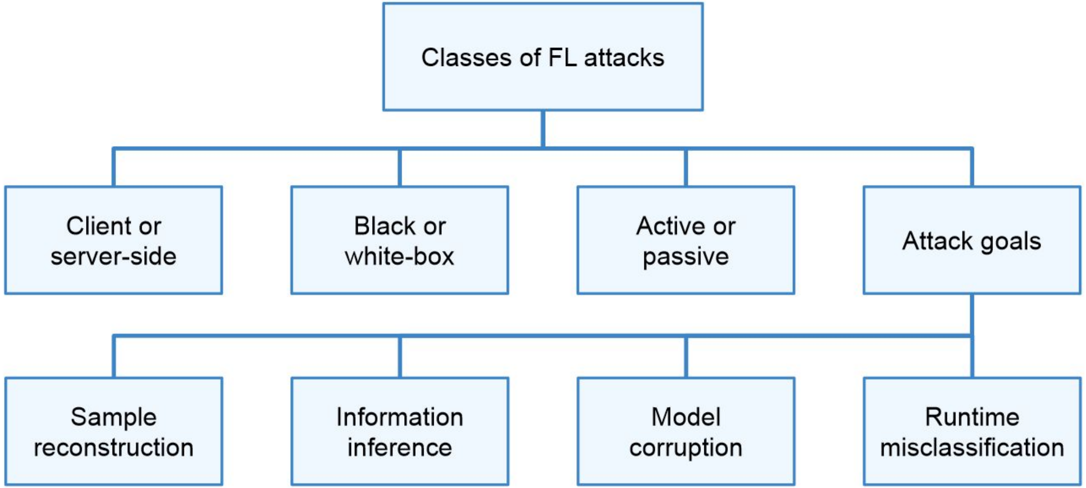


Fig. 2. Taxonomy to classify the different types of FL attack methods

An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies (2020)

Model Attacks (Model Replacement, Backdoors, etc)

"byzantine" adversary

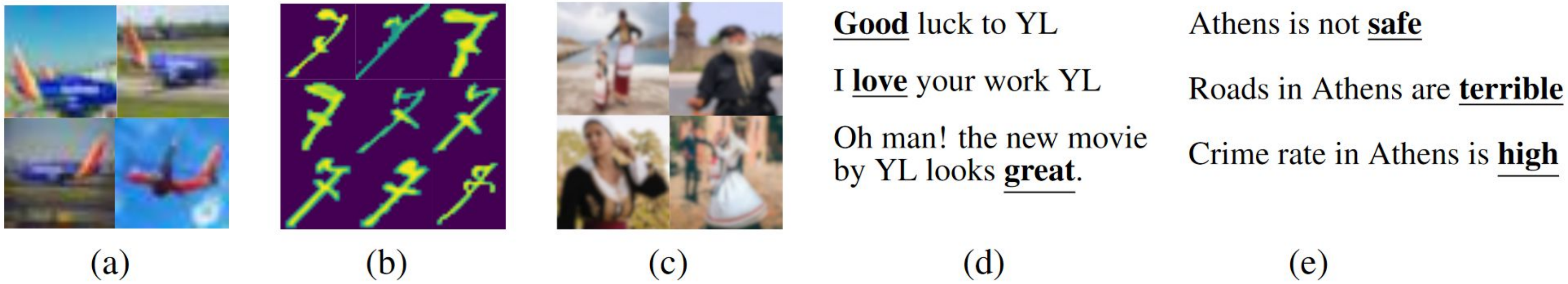


Figure 1: Illustration of tasks and edge-case examples for our backdoors. Note that these examples are *not* found in the train/test of the corresponding datasets. (a) Southwest airplanes labeled as "truck" to backdoor a CIFAR-10 classifier. (b) Images of "7" from the ARDIS dataset labeled as "1" to backdoor an MNIST classifier. (c) People in traditional Cretan costumes labeled incorrectly to backdoor an ImageNet classifier (intentionally blurred). (d) Positive tweets on the director Yorgos Lanthimos (YL) labeled as "negative" to backdoor a sentiment classifier. (e) Sentences regarding Athens completed with words of negative connotation to backdoor a next word predictor.

Attack of the Tails: Yes, You Really Can Backdoor Federated Learning (2020)

Carnegie Mellon University

Privacy Attacks (Data Recovery)

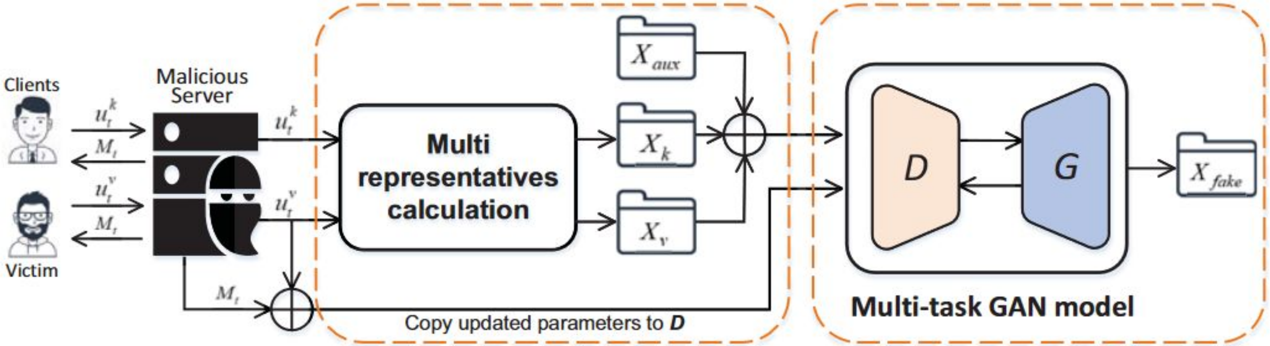


Fig. 2: Illustration of the proposed mGAN-AI from a malicious server in the federated learning. There are N clients, and the vth client is attacked as the victim. The shared model at the tth iteration is denoted as M_t, and u_t^k denotes corresponding update from the kth client. On the malicious server, a discriminator D (orange) and generator G (blue) are trained based on the update u_t^v from the victim, the shared model M_t, and representatives X_k, X_v from each client. X_{aux} denotes an auxiliary real dataset to train D on the real-fake task.

Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning (2019)

Differential Privacy

Definition 1 ((ε, δ)-DP [24]): A randomized mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ε, δ)-DP, if for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and for any two adjacent databases $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$,

in   not

$$\Pr[\mathcal{M}(\mathcal{D}_i) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}_i') \in \mathcal{S}] + \delta. \quad (3)$$

D_i' ..   ... D_i

while $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ **do**

Update the local parameters $\mathbf{w}_i^{(t)}$ as

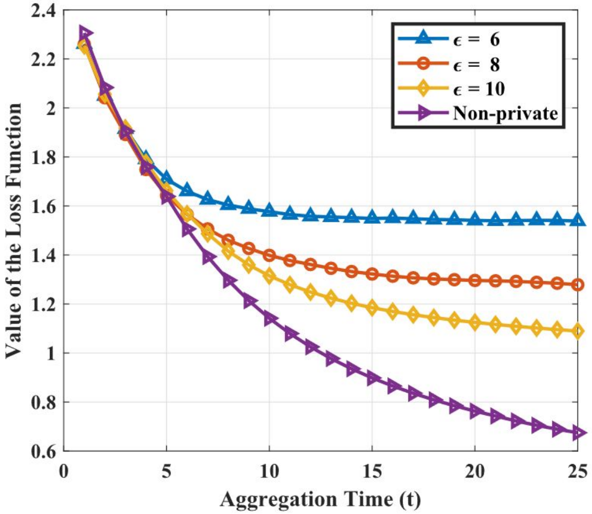
$$\mathbf{w}_i^{(t)} = \arg\min_{\mathbf{w}_i} \left(F_i(\mathbf{w}_i) + \frac{\mu}{2} \|\mathbf{w}_i - \mathbf{w}^{(t-1)}\|^2 \right)$$

Clip the local parameters

$$\mathbf{w}_i^{(t)} = \mathbf{w}_i^{(t)} / \max\left(1, \frac{\|\mathbf{w}_i^{(t)}\|}{C}\right)$$

Add noise and upload parameters $\widetilde{\mathbf{w}}_i^{(t)} = \mathbf{w}_i^{(t)} + \mathbf{n}_i^{(t)}$

Can greatly affect performance!

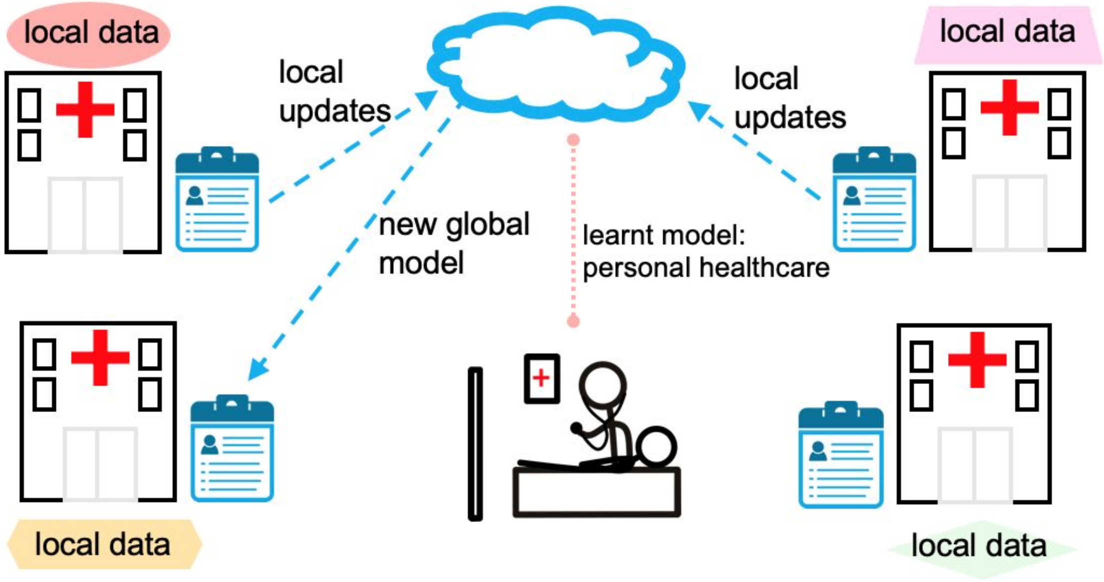


Federated Learning With Differential Privacy: Algorithms and Performance Analysis (2020)

Carnegie Mellon University

Why Federated Learning?

A Policy Perspective



Machine Learning **Threat Model**

Threat Type	**Confidentiality**	**Integrity**	**Availability**
Threats **solved** by Federated Learning	User data		decentralized model inference
Threats **created** by Federated Learning	data recovery attacks [model weights, and secrecy]	Model backdoors	model poisoning

Carnegie Mellon University

Machine Learning **Threat Model**

Threat Type	**Confidentiality**	**Integrity**	**Availability**
Threats **solved** by Federated Learning	User data privacy		
Threats **created** by Federated Learning	Model parameter and architecture secrecy	Model backdoor attacks	Model poisoning attacks

Carnegie Mellon University

Why Federated Learning?

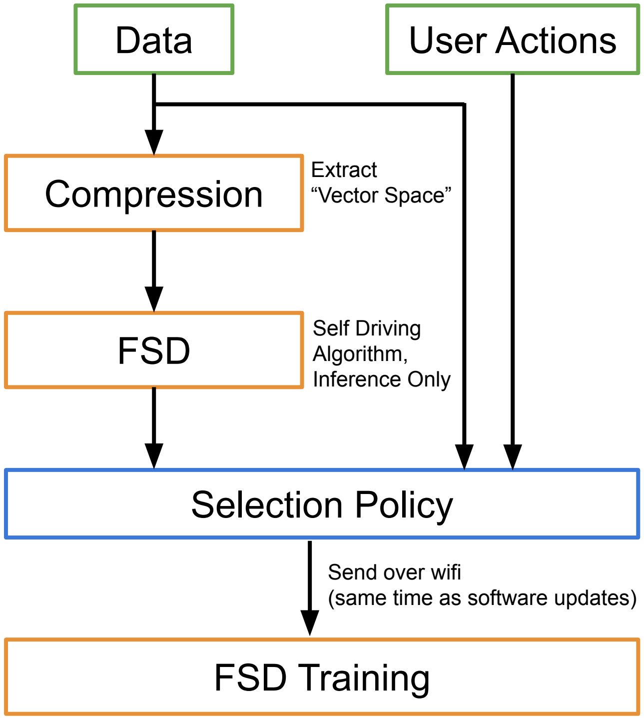
- Compute cost offloading
- Privacy regulations, i.e. GDPR
- PR reasons (google)
- Legal liability

Carnegie Mellon University

Why (Not) Federated Training?

Tesla Full Self Driving Training

- Edge training is hard/expensive
- Users don’t know about privacy
- Users don’t care about privacy

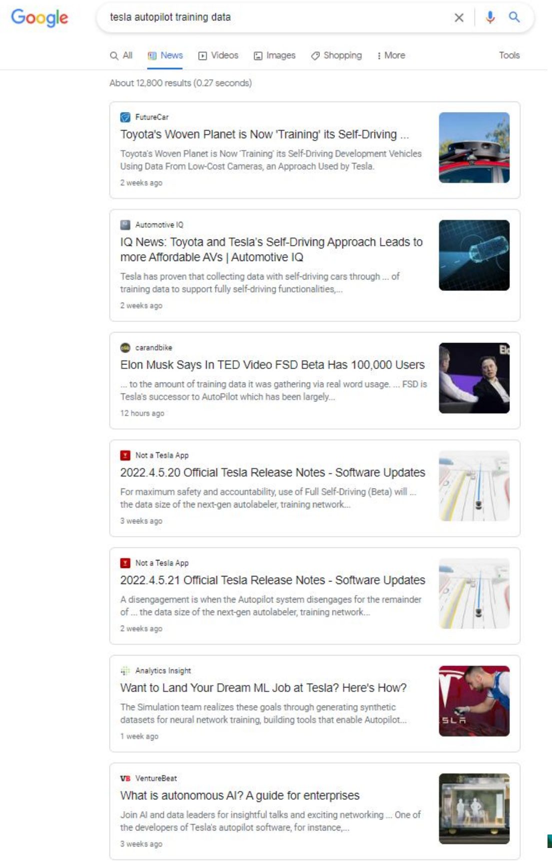


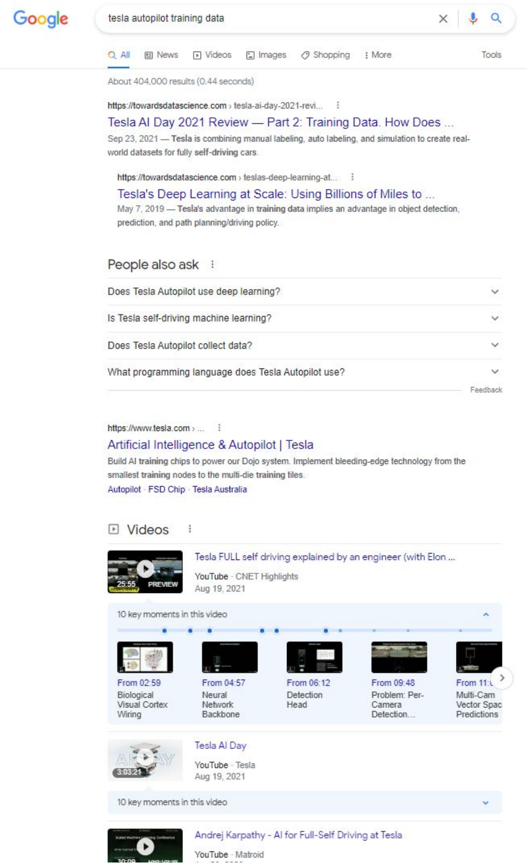
Carnegie Mellon University

No articles about privacy on the first page!

And people are only mildly concerned!

[("Self-Driving Cars and Data Collection: Privacy Perceptions of Networked Autonomous Vehicles", 2017)]





Should you use Federated Learning?

Reasons For:

- moderate sized model
- legal/privacy; "health care"
  "cross-silo"
- user privacy / acceptability

Reasons Against:

- very low compute
- very high compute
- stealing people's data
- adv. attacks; untrusted users

Carnegie Mellon University

*

Thanks!

The content for this lecture is in part from:

- Ethan Ruan's previous guest lectures for this class
- Gauri Joshi's Federated Learning Course @ CMU;
  Notation + equations from her new book:
  https://link.springer.com/book/10.1007/978-3-031-19067-4