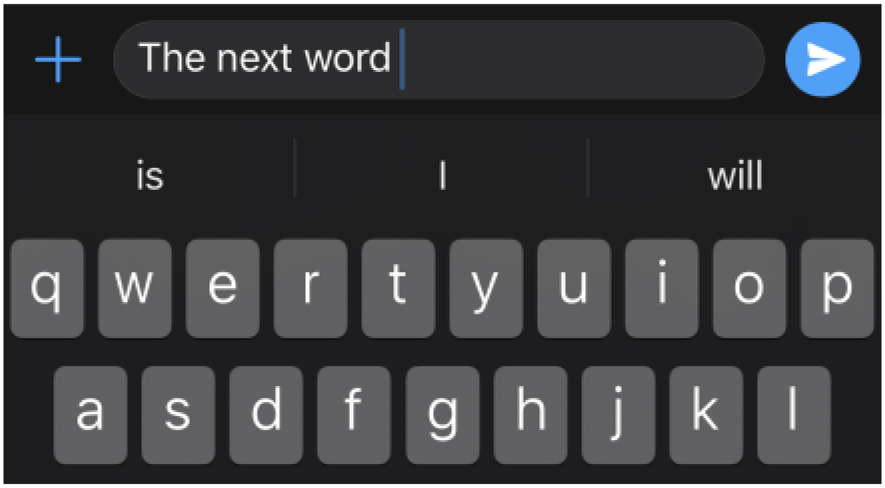
Introduction to Federated Learning

(Reference: Algorithms for Large-scale Distributed Machine Learning and Optimization by Prof. Gauri Joshi, CMU)

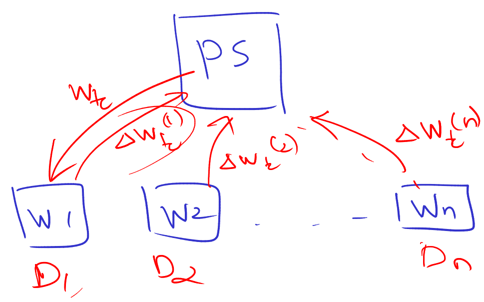


- Edge devices: Cell phones or IoT devices collect huge amounts of data that can be informative for ML models.

- Training data: What every user types on their phone.

- **Q**: How can you utilize the parameter server framework we've explored to effectively train a single machine learning model with data from the edge?

- There are millions of edge clients:



- What are some problems with this strategy?
  - Parameter Server approach requires prohibitively large communication bandwidth, since it exchanges information with millions of edge devices.

    $$w_{t+1} = w_t - \eta \sum_{k=1}^{n} \Delta w_t^{(k)}$$

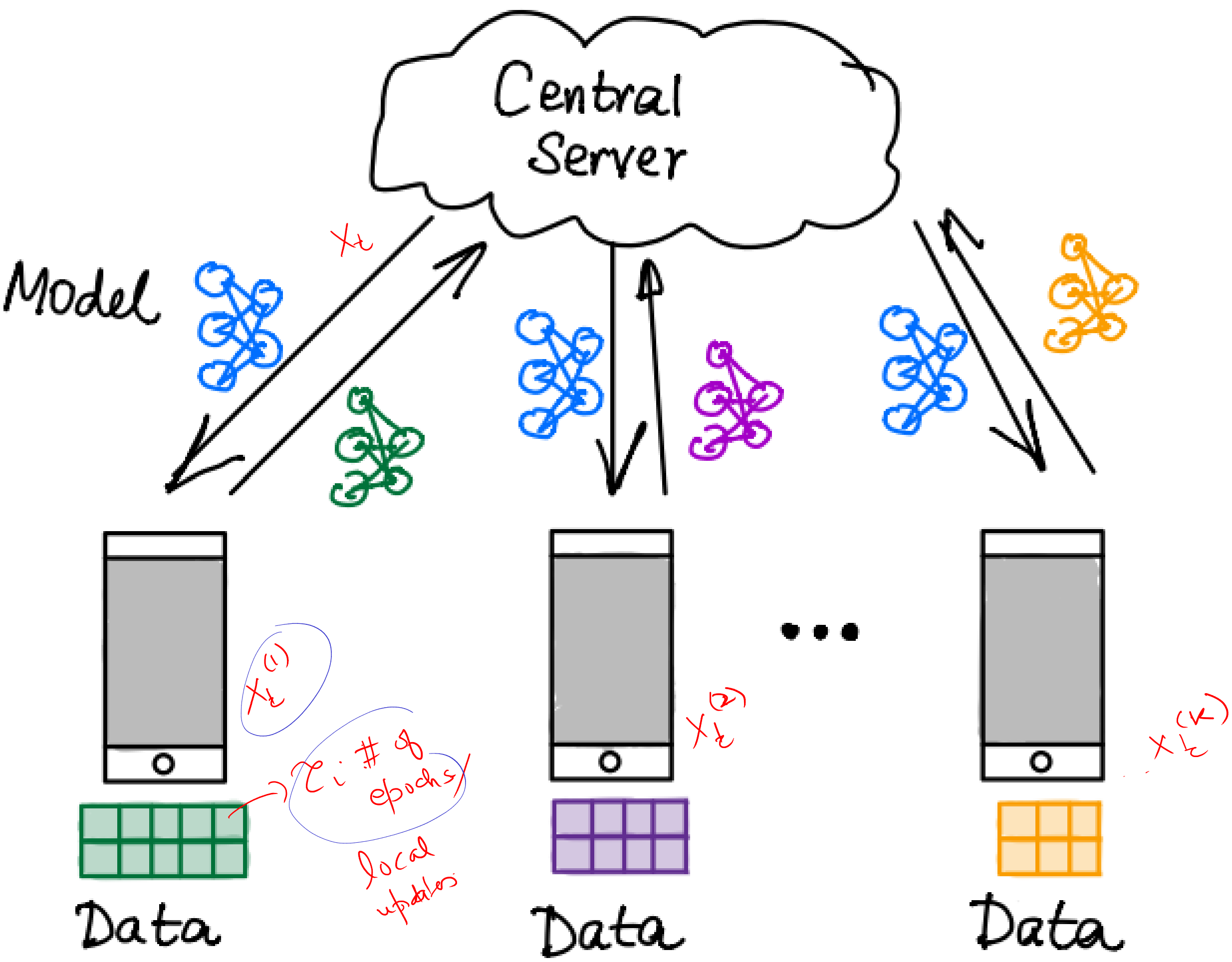
  - Data privacy / sensitive information
  - Edge devices may have limited internet connectivity

Federated Learning to the rescue!

Key idea: **Bring the training to the edge ~~data~~** devices (McMahan et al. 2016: *Communication-efficient learning of deep networks from decentralized data*) FedAvg.

- Already used for next-word prediction on Android cell phones, when the phone is plugged in for charging

https://federated.withgoogle.com/
(Read this comic book)



Parameter Server

Decentralized SGD vs Federated Learning

- Number of workers/servers

Parameter Server/DSGD

Tens or hundreds of clients

FL

Millions of clients

- Availability of workers

Parameter Server

We require the workers to be available at all times.

FL

We only work with a handful of clients.

- Data distribution

PS/D-SGD

We require the data distribution to be roughly homogeneous

FL

Data distribution is heterogeneous.

- Worker types: Homogeneous/Heterogeneous

PS/D-SGD:

Clients are homogeneous in terms of compute power.

FL

Clients are heterogeneous.

- Privacy

PS/D-SGD:

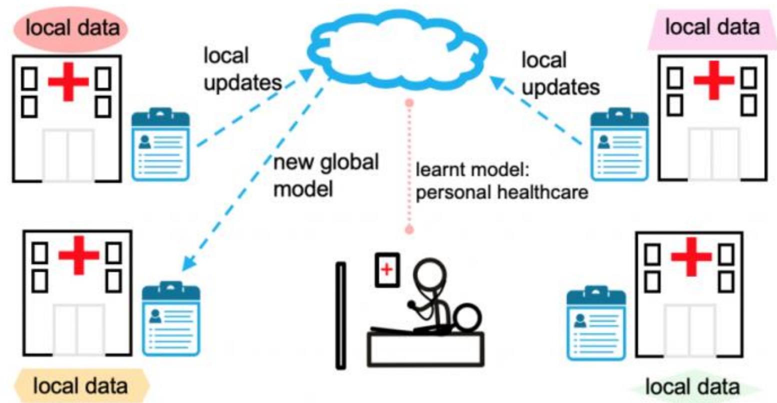
Data distribution can be re-arranged.

FL

Data is private and secure.

Cross-Device vs Cross-Silo Federated Learning

Cross-device FL



Cross-silo FL

- Number of devices: Cross-device >> Cross-silo
  (Few millions) (Tens to Hundreds)

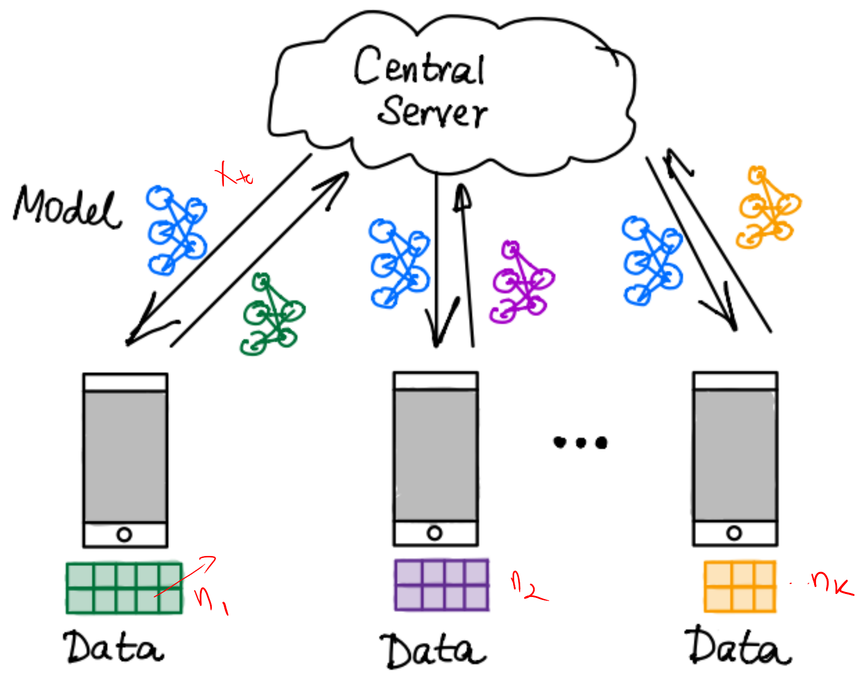
- Availability of workers: Not all clients are available at all times.

- Data heterogeneity: Data distribution is heterogeneous.

- Worker types: Cross-silo → Homogeneous
  Cross-device → Heterogeneous.

- Privacy constraints: Data is private and secure.

Federated Learning Framework



Notations:

clients/agents/devices

- Total number of workers: K
- Fraction of workers participating in each communication round: C
- Local mini-batch size: B
- Number of data samples at client i: n_i
- Learning rate: η
- Number of local epochs per client: E

Q: How many local updates τ_i will be performed at client i? $\tau_i = E \frac{n_i}{B}$

The FedAvg Algorithm

- **Local objective function:**

at the i^{th} client:

$$F_i(x) = \frac{1}{n_i} \sum_{j=1}^{n_i} f_i(x; \xi_j)$$

- **Global objective function:**

$$F(x) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(x) = \sum_{k=1}^{K} p_k F_k(x)$$

where $p_k = \frac{n_k}{n}$

FedAvg Algorithm

Server Update:

- Initialize the model parameters at x_0.
- For each communication round $t = 1, \dots, T$
  - Select a set S_t of m clients (from a total of K clients), uniformly at random.
  - Perform ClientUpdate(i, x_t) at the chosen client, and receive $x_{t+1}^{(i)}$ from client $i \in S_t$
  - Aggregate the updates: $x_{t+1} \leftarrow \sum_{i \in S_t} p_i x_{t+1}^{(i)}$

ClientUpdate(i, x_t):

- Initialize the local model $x_{t,0}^{(i)} \leftarrow x_t$ for $\tau_i := \frac{E n_i}{B}$ local updates
- For local iteration index $j \in \{0, 1, \dots, \tau_i - 1\}$ do the following.
  - Sample mini-batch ξ_j from the local dataset $\mathcal{D}_i$ and $x_{t,j+1}^{(i)} \leftarrow x_{t,j}^{(i)} - \eta g(x_{t,j}^{(i)}; \xi_j)$
- Return $x_{t,\tau_i}^{(i)}$

Effect of Data Heterogeneity and Client Participation

MNIST (hand-written digit dataset: 97% for 2-NN and 99% for CNN)

2NN	IID		NON-IID	
C	$B = \infty$	$B = 10$	$B = \infty$	$B = 10$
0.0	1455	316	4278	3275
0.1	1474 (1.0×)	87 (3.6×)	1796 (2.4×)	664 (4.9×)
0.2	1658 (0.9×)	77 (4.1×)	1528 (2.8×)	619 (5.3×)
0.5	— (—)	75 (4.2×)	— (—)	443 (7.4×)
1.0	— (—)	70 (4.5×)	— (—)	380 (8.6×)
CNN, $E = 5$				
0.0	387	50	1181	956
0.1	339 (1.1×)	18 (2.8×)	1100 (1.1×)	206 (4.6×)
0.2	337 (1.1×)	18 (2.8×)	978 (1.2×)	200 (4.8×)
0.5	164 (2.4×)	18 (2.8×)	1067 (1.1×)	261 (3.7×)
1.0	246 (1.6×)	16 (3.1×)	— (—)	97 (9.9×)

Total number of communication rounds

→ 60k

- IID experiment - shuffle and partition the data across 100 clients, each receiving 600 examples
- non-IID experiment - the data sorted by labels and divided into 200 shards of size 300 and each of the 100 clients receives 2 shards (at most 2 digits)

Effect of Number of Local Epochs

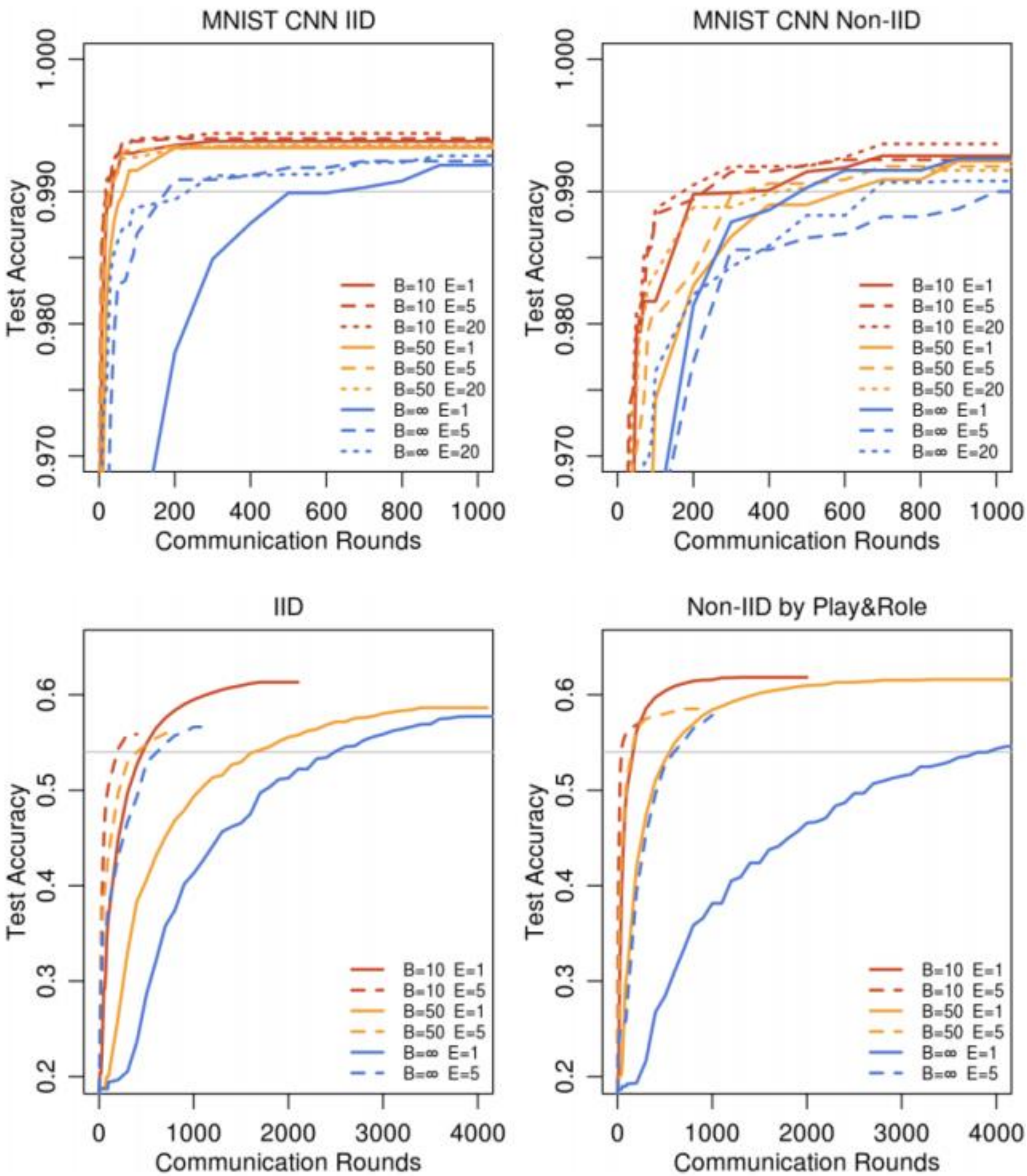
E

MNIST CNN, 99% ACCURACY					
CNN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)
SHAKESPEARE LSTM, 54% ACCURACY					
LSTM	E	B	u	IID	NON-IID
FEDSGD	1	∞	1.0	2488	3906
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

- As the number of local epochs E grows, we need fewer communication rounds to reach target accuracy

$$\tau_i = \frac{E n_i}{B}$$

Effect of Batch-Size



Convergence Analysis of FL

Assumptions:

- Lipschitz smoothness of local objective function

  F_i is L-Lipschitz

  $$\|\nabla F_i(x) - \nabla F_i(y)\| \le L\|x-y\| \quad \forall i, \ \forall x, y$$

- Unbiased gradients:

  Stochastic gradient $g_i(x;\xi)$ is an unbiased estimate of $\nabla F_i(x)$.

  $$\mathbb{E}_\xi[g_i(x;\xi)] = \nabla F_i(x).$$

- Bounded variance:

  $$Var(g_i(x;\xi)) \le \sigma^2$$

  $$\mathbb{E}_\xi[\|g_i(x;\xi)\|^2] \le \|\nabla F_i(x)\|^2 + \sigma^2$$

- Bounded dissimilarity:

  There exist parameters $\beta^2 \ge 1$ and $\kappa^2 \ge 0$ s.t.

  $$\mathbb{E}_{S_i}\left[p_i\|\nabla F_i(x)\|^2\right] \le \beta^2 \|\mathbb{E}_{S_i}[p_i \nabla F_i(x)]\|^2 + \kappa^2$$

  For iid data, $\beta^2 = 1$ and $\kappa^2 = 0$

Convergence of Federated Learning

For the number selected clients $m = CK$ and learning rate $\eta = \sqrt{m/\tau T}$, the optimization error after T communication rounds of federated learning can bounded as

$$\min \mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \le O\left(\frac{1+\sigma^2}{\sqrt{m\tau T}}\right) + O\left(\frac{m(\sigma^2 + \tau\kappa^2)(\tau - 1)}{\tau T}\right)$$

$\le \epsilon$

where $\mathbf{x}_k$ denotes the averaged model at the k^{th} iteration.

Let's revisit basic understanding of FL!

Does the convergence between error and communication rounds improve or deteriorate when the parameters of the federated learning system/algorithm are altered in the following manners?

- Increase in fraction of participating clients/workers (C): Better
- Increase in mini-batch size (B): Worse
- Increase in local epochs (E): Better, but if E increases too much, then there can be overfitting.
- Higher-data heterogeneity across clients: Worse
- Increase in dissimilarity parameters β and κ: Worse

Does the anticipated wallclock time per communication round shift when modifying the parameters of the federated learning system/algorithm in the specified manners?

- Increase in fraction of participating clients/workers (C): Increase
- Increase in mini-batch size (B): Increase / Decrease depending on the model parameters.
- Increase in local epochs (E): Increase
- Higher-data heterogeneity across clients: Doesn't matter
- Increase in dissimilarity parameters β and κ: Doesn't matter

---

$$\left.\begin{aligned} F_1(x) &= (x-1)^2 \\ F_2(x) &= 2(x-5)^2 \end{aligned}\right\} \quad F(x) = \frac{1}{2}(x-1)^2 + (x-5)^2$$

$x^* = ?$ $\quad (x-1) + 2(x-5) = 0$

$$\boxed{x^* = \frac{11}{3}}$$

$$\left.\begin{aligned} x_{t+1}^{(1)} &= x_t - \eta\,(x_t - 1) \\ x_{t+1}^{(2)} &= x_t - 2\eta\,(x_t - 5) \end{aligned}\right\} \Rightarrow x_{t+1} = x_t - \frac{\eta}{2}\left[(x_t-1) + 2(x_t-5)\right]$$