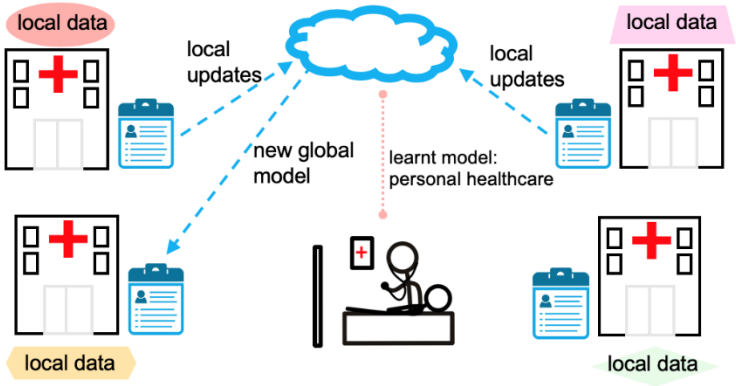
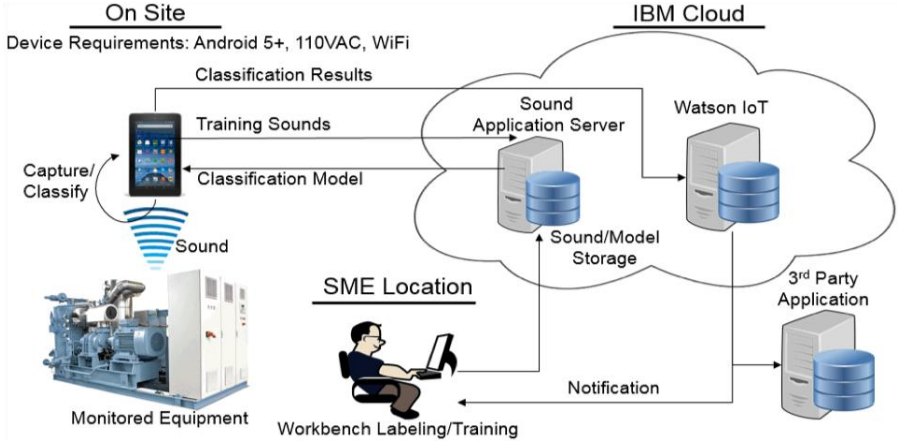
Federated Learning with Imbalanced Client Participation

Shiqiang Wang
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

Various Applications Driven by Machine Learning

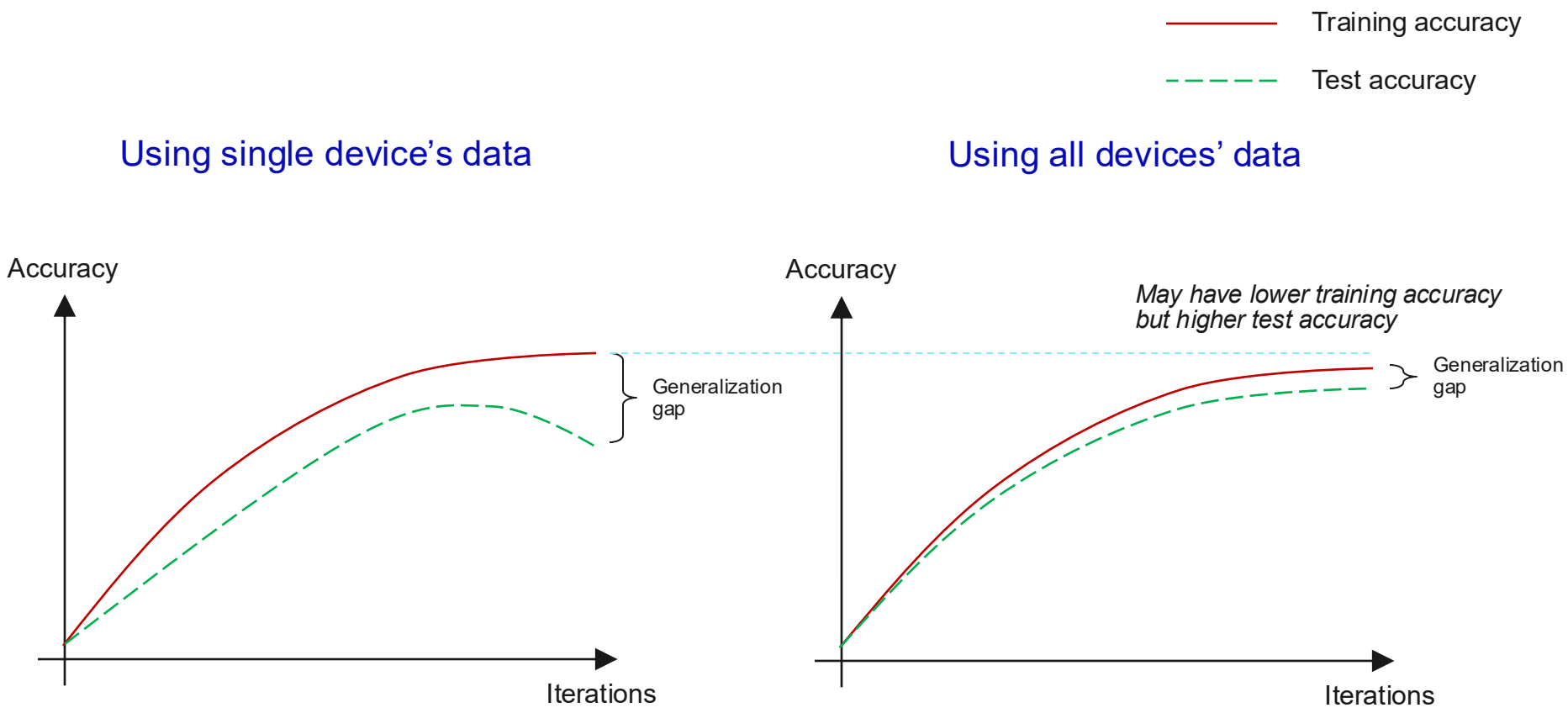




Sources:

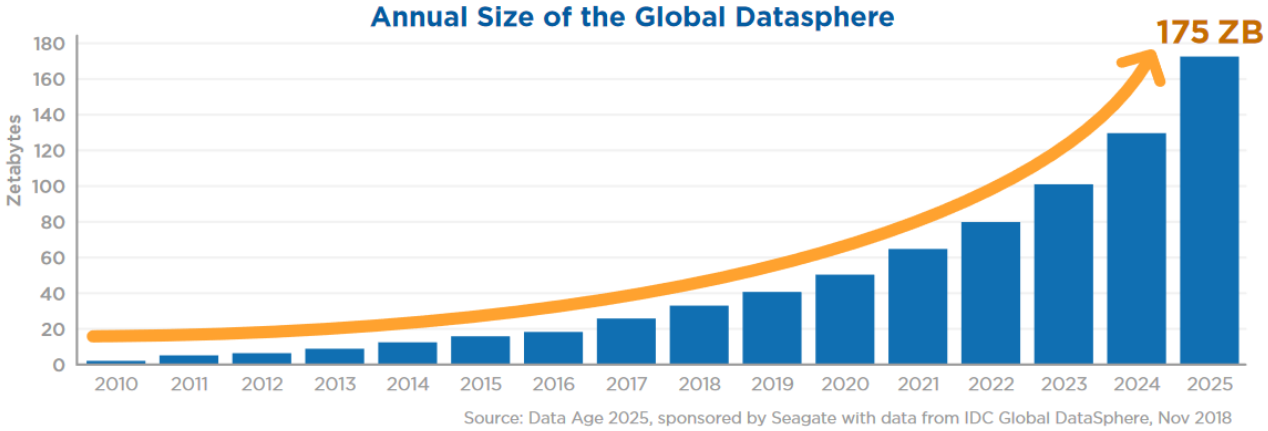
- https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/
- J.-W. Ahn, K. Grueneberg, B. J. Ko, W.-H. Lee, E. Morales, S. Wang, X. Wang, D. Wood, "Acoustic anomaly detection system: demo abstract," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2019.

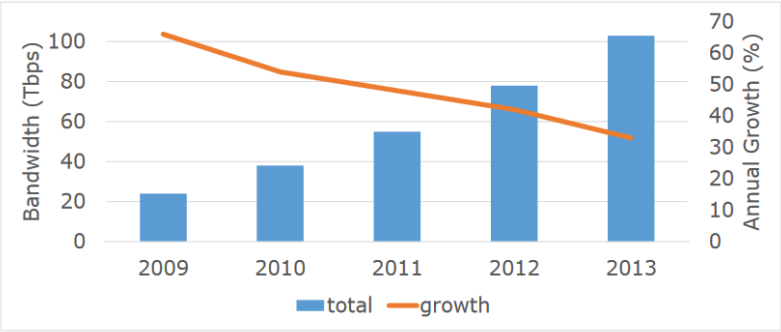
Overfitting when Only Training on a Single Device's Data



Challenges in Training a Global Model

We cannot send all data to a central location





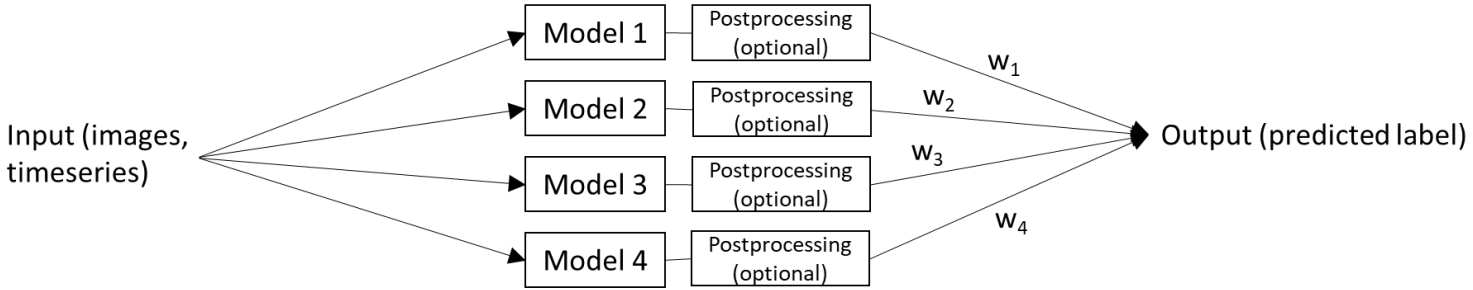
Source: Vulimiri, Ashish, et al. "Global Analytics in the Face of Bandwidth and Regulatory Constraints." *NSDI*. Vol. 7. No. 7.2. 2015.

Source: https://eugdpr.org/

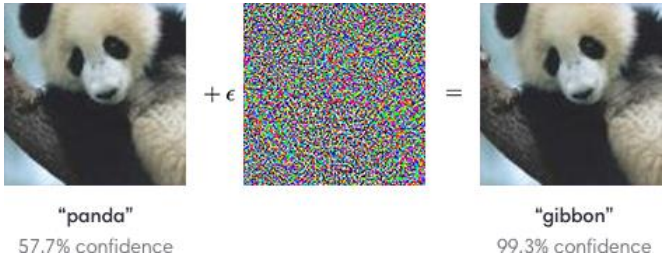
Can we train machine learning models in a distributed way without sharing raw data?

A Naïve Idea

- Each devices trains a local model on its own data
- Then combine all these models (ensemble method)



- However:
  - Huge model size when the number of devices is very large (e.g., billions of cell phone users)
  - Combining model outputs may give bad accuracy
    - Consider: adversarial inputs
      (models may behave abnormally on input that it is not trained on)

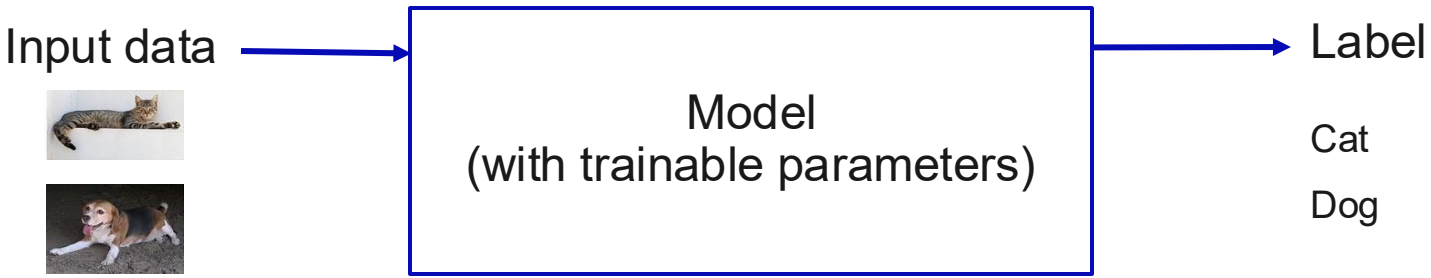


Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, "Explaining and Harnessing Adversarial Examples", 2014

How do we overcome this problem?

Let's Start from the Basics

- Supervised machine learning

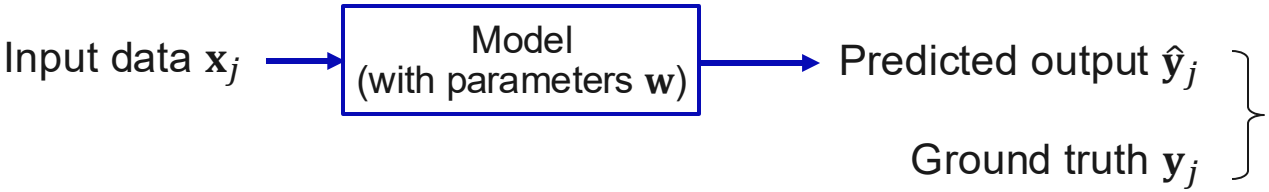


- Model training: Adapt model parameters based on labeled training data (many data samples)
- Inference: Apply new input data to the model, infer its label

- Unsupervised machine learning do not have labels in training data
- We focus on model training

Let's Start from the Basics (cont'd)

- An essential component of model training is the loss function
- A machine learning model with parameter $\mathbf{w}$
- How good is $\mathbf{w}$: Individual loss function for data sample j, $f(\mathbf{w}, \mathbf{x}_j, \mathbf{y}_j)$



Mean square error:

$$f(\mathbf{w}, \mathbf{x}_j, \mathbf{y}_j) \coloneqq \|\hat{\mathbf{y}}_j - \mathbf{y}_j\|$$

Cross-entropy loss:

$$f(\mathbf{w}, \mathbf{x}_j, \mathbf{y}_j) \coloneqq -\sum_{c=1}^{C} y_{j,c} \log \hat{y}_{j,c}$$

(*one-hot* encoded labels)

Losses for specific models:

Model	Loss function $f(\mathbf{w}, \mathbf{x}_j, y_j)$ ($\triangleq f_j(\mathbf{w})$)
Smooth SVM	$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\max\{0; 1 - y_j \mathbf{w}^{\mathrm{T}} \mathbf{x}_j\}^2$ (λ is const.)
Linear regression	$\frac{1}{2}\|y_j - \mathbf{w}^{\mathrm{T}} \mathbf{x}_j\|^2$
K-means	$\frac{1}{2}\min_l \|\mathbf{x}_j - \mathbf{w}_{(l)}\|^2$ where $\mathbf{w} \triangleq [\mathbf{w}_{(1)}^{\mathrm{T}}, \mathbf{w}_{(2)}^{\mathrm{T}}, \ldots]^{\mathrm{T}}$

Mathematical Formulation of Model Training

- In essence: minimize the loss function
- In the case of centralized training, when there are M samples in total, we usually minimize the average of losses (also known as empirical risk):

$$F(\mathbf{w}) = \frac{1}{M}\sum_{j=1}^{M} f(\mathbf{w}, \mathbf{x}_j, \mathbf{y}_j)$$

- The learning problem is then to minimize the empirical risk, i.e., find

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w})$$

- In the distributed setting, M samples are partitioned into N different clients (devices), we can consider $F(\mathbf{w})$ as the global empirical risk
- We can define the local empirical risk on client i as the average of losses on its local samples:

$$F_i(\mathbf{w}) = \frac{1}{|\mathcal{D}_i|}\sum_{j\in D_i} f(\mathbf{w}, \mathbf{x}_j, \mathbf{y}_j)$$

- We can then write the global empirical risk as:

$$F(\mathbf{w}) = \frac{\sum_{i=1}^{N} |\mathcal{D}_i| F_i(\mathbf{w})}{M}$$

The objective is still to minimize $F(\mathbf{w})$

Now, let’s look at federated learning!

In the Following…

- Basics of federated learning
- Federated averaging (FedAvg) algorithm
- Tackling imbalanced client participation – FedAU
- Convergence analysis techniques
- Experiments and simulation code

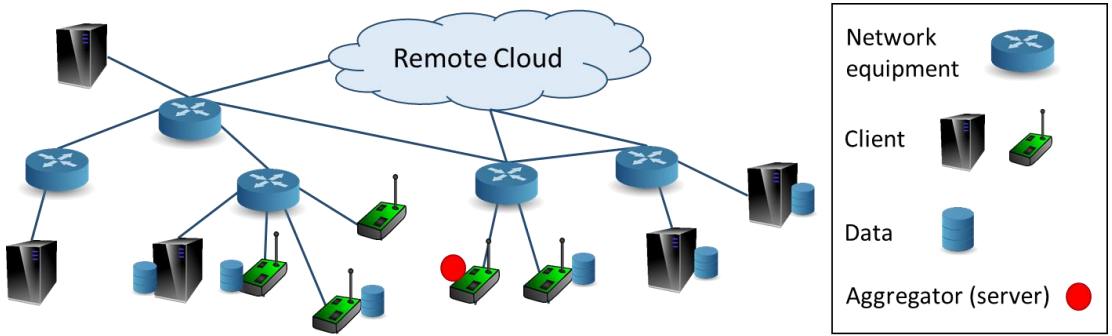
Federated

Efficient

Scalable

Federated Learning (Architecture)

- Objective: Minimize $F(\mathbf{w})$ while keeping the data local



- Data is collected by each user and stored at a "client"
  - User (mobile) device, edge server/gateway (e.g., of a specific organization), etc.
- Model training is coordinated by an "aggregator (server)"
  - A logical component running at the edge or cloud
- Only model parameters are shared
- **Local computation at clients + global communication (aggregation) with server**

From Centralized to Federated Learning

- A common approach of machine learning (minimizing empirical risk) is to use stochastic gradient descent (SGD)
- We consider deterministic gradient descent here for simplicity
- In the centralized setting, to minimize $F(\mathbf{w})$, each iteration of gradient descent computes

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \eta \nabla F\big(\mathbf{w}(t-1)\big)$$

- Recall from before that $F(\mathbf{w}) = \frac{\sum_{i=1}^{N} |\mathcal{D}_i| F_i(\mathbf{w})}{M}$. So, we can write

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \eta \frac{\sum_{i=1}^{N} |\mathcal{D}_i| \nabla F_i\big(\mathbf{w}(t-1)\big)}{M} = \frac{\sum_{i=1}^{N} |\mathcal{D}_i| \big[\mathbf{w}(t-1) - \eta \nabla F_i\big(\mathbf{w}(t-1)\big)\big]}{M}$$

- Assuming that all clients start with the same $\mathbf{w}(t-1)$, we can let each client i compute gradient descent on its local objective $F_i(\mathbf{w})$, to obtain a local parameter vector

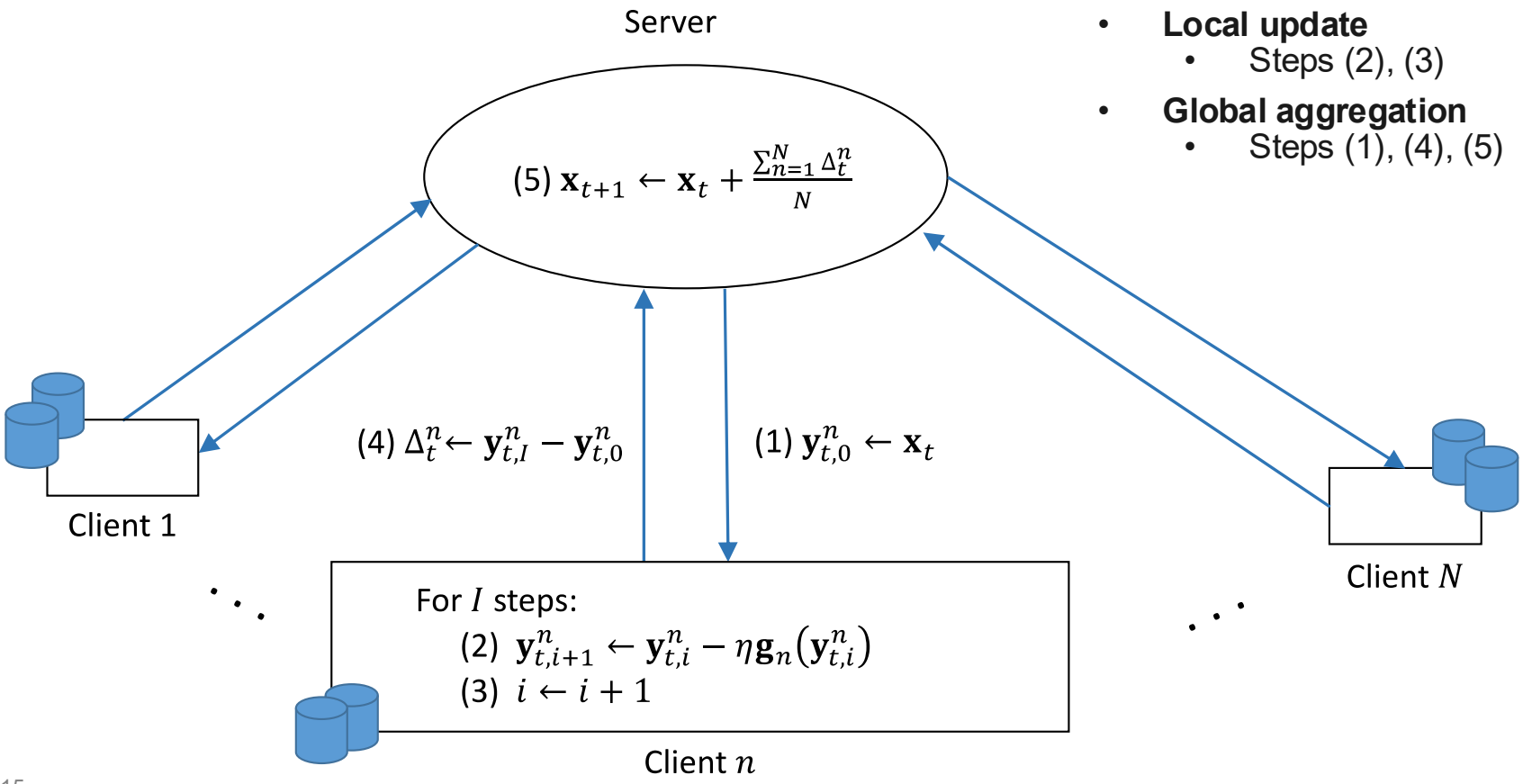
$$\mathbf{w}_i(t) = \mathbf{w}(t-1) - \eta \nabla F_i\big(\mathbf{w}(t-1)\big)$$

- Then, we obtain the global parameter by averaging all client's local parameters:

$$\mathbf{w}(t) = \frac{\sum_{i=1}^{N} |\mathcal{D}_i| \mathbf{w}_i(t)}{M}$$

- With proper synchronization between the server and clients, most of the computation can be moved to the clients that update the model parameter based on their local data

Federated Averaging (FedAvg)



- **Local update**
  - Steps (2), (3)
- **Global aggregation**
  - Steps (1), (4), (5)

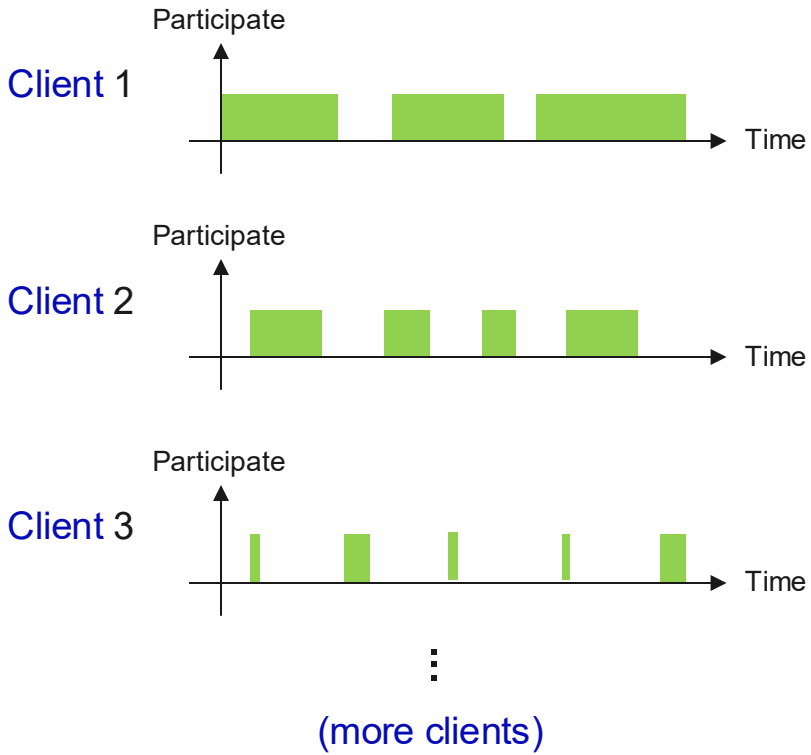
Do we fully preserve data privacy by exchanging model parameters instead of raw data?

No, it is possible to estimate training data based on gradients

Well-known techniques for improving privacy in FL:

- Differential privacy
- Secure aggregation

Challenge: Imbalanced Client Participation Rates



$$f(\mathbf{x}) := \frac{1}{N}\sum_{n=1}^{N} F_n(\mathbf{x})$$

In the Following…

- FedAU: Adaptively **weighting** the client updates in FedAvg based on **online** estimates of the optimal weights **without knowing the statistics** of client participation
- Key insight:

System Statistics + Optimization

- Paper: Shiqiang Wang, Mingyue Ji. A Lightweight Method for Tackling Unknown Participation Statistics in Federated Averaging. ICLR 2024 (spotlight, 5% of submitted papers).

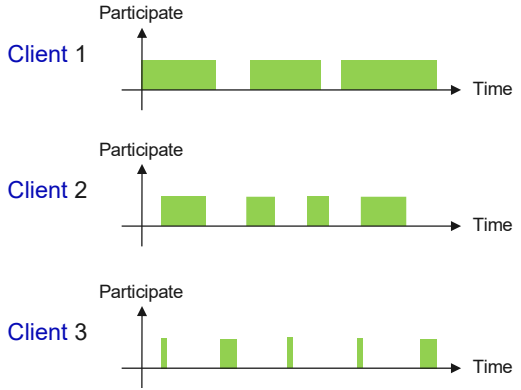
Paper

https://openreview.net/pdf?id=ZKEuFKfCKA

Code

https://github.com/IBM/fedau

FedAvg Algorithm



Algorithm 1: FedAvg with pluggable aggregation weights

Input: γ, η, $\mathbf{x}_0$, I; **Output:** $\{\mathbf{x}_t : \forall t\}$;

Initialize $t_0 \leftarrow 0$, $\mathbf{u} \leftarrow \mathbf{0}$;

for $t = 0, \ldots, T-1$ **do**

  for $n = 1, \ldots, N$ *in parallel* **do**

    Sample $\mathbb{1}_t^n$ from an *unknown* process;

    if $\mathbb{1}_t^n = 1$ **then**

      $\mathbf{y}_{t,0}^n \leftarrow \mathbf{x}_t$;

      for $i = 0, \ldots, I-1$ **do**

        $\mathbf{y}_{t,i+1}^n \leftarrow \mathbf{y}_{t,i}^n - \gamma \mathbf{g}_n(\mathbf{y}_{t,i}^n)$;

      $\Delta_t^n \leftarrow \mathbf{y}_{t,I}^n - \mathbf{x}_t$;

    else

      $\Delta_t^n \leftarrow \mathbf{0}$;

    $\omega_t^n \leftarrow \texttt{ComputeWeight}(\{\mathbb{1}_\tau^n : \tau < t\})$;

  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \frac{\eta}{N} \sum_{n=1}^N \omega_t^n \Delta_t^n$;

Local updates

Aggregation

Randomized participation with unknown statistics

Challenge: The participation statistics of clients are often *unknown*, *uncontrollable*, and *heterogeneous*

Aggregation weights

Improper Choice of Aggregation Weights Causes Bias

Only assumed for theoretical analysis

Theorem 1 (Objective minimized at convergence, informal). *When* $\mathbb{1}_t^n \sim \text{Bernoulli}(p_n)$ *and the weights are time-constant, i.e.,* $\omega_t^n = \omega_n$ *but generally* ω_n *may not be equal to* $\omega_{n'}$ $(n \neq n')$*, with properly chosen learning rates* γ *and* η *and some other assumptions, Algorithm 1 minimizes the following objective:*

$$h(\mathbf{x}) := \frac{1}{P} \sum_{n=1}^{N} \omega_n p_n F_n(\mathbf{x}),$$

where $P := \sum_{n=1}^{N} \omega_n p_n$.

Implicit weighting due to partial participation

- Choosing $\omega_n = {}^1/_{p_n}$
  - Objective is consistent with $f(\mathbf{x}) := \frac{1}{N} \sum_{n=1}^{N} F_n(\mathbf{x})$
  - However, impractical when p_n is unknown

The **ideal** case:
Choice of aggregation weight ω_n should *cancel out* the implicit weighting by p_n

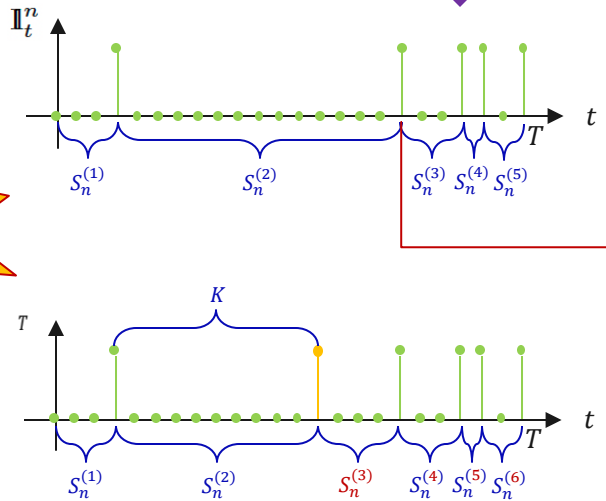
- Choosing other values of ω_n (e.g., $\omega_n = 1, \forall n$)
  - Objective inconsistency, leading to bias (preference of more frequently participating clients)

How to Estimate Aggregation Weights?

Inspired by Bernoulli-distributed participation ➡ Generalize to other participation patterns empirically

$$\omega_n = 1/p_n \xrightarrow{\text{Estimate}} p_n \approx \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{I}_t^n$$

Problem 1 (Goal of Weight Estimation, informal). *Choose* $\{\omega_t^n\}$ *so that its long-term average (i.e., for large* T*)* $\frac{1}{T}\sum_{t=0}^{T-1} \omega_t^n$ *is close to* $\frac{1}{\frac{1}{T}\sum_{t=0}^{T-1} \mathbf{I}_t^n}$*, for each* n.



Solution: "Cutoff" interval

Equivalent to the average of **intervals** between every pair of adjacent participating rounds ➡ Geometric distribution for Bernoulli participating clients (same parameter p_n)

Cannot predict the future
➔ Estimate ω_t^n based on intervals seen so far

Problem: Large overestimate of ω_t^n when large intervals exist (although with low probability) ➔ instability in training

Create a dummy interval when the actual interval exceeds K
- Smaller K → lower variance (more samples), but higher bias
- Larger K → higher variance (less samples), but lower bias

FedAU

- FedAvg with adaptive weighting to support unknown participation statistics

Algorithm 1: FedAvg with pluggable aggregation weights

Input: $\gamma, \eta, \mathbf{x}_0, I$; **Output:** $\{\mathbf{x}_t : \forall t\}$;

Initialize $t_0 \leftarrow 0$, $\mathbf{u} \leftarrow \mathbf{0}$;
for $t = 0, \ldots, T-1$ **do**
  for $n = 1, \ldots, N$ *in parallel* **do**
    Sample $\mathbb{I}_t^n$ from an *unknown* process;
    if $\mathbb{I}_t^n = 1$ **then**
      $\mathbf{y}_{t,0}^n \leftarrow \mathbf{x}_t$;
      for $i = 0, \ldots, I-1$ **do**
        $\mathbf{y}_{t,i+1}^n \leftarrow \mathbf{y}_{t,i}^n - \gamma \mathbf{g}_n(\mathbf{y}_{t,i}^n)$;
      $\Delta_t^n \leftarrow \mathbf{y}_{t,I}^n - \mathbf{x}_t$;
    else
      $\Delta_t^n \leftarrow \mathbf{0}$;
    $\omega_t^n \leftarrow$ `ComputeWeight`$(\{\mathbb{I}_\tau^n : \tau < t\})$;
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \frac{\eta}{N} \sum_{n=1}^N \omega_t^n \Delta_t^n$;

Algorithm 2: Weight computation in FedAU

Input: $K, \{\mathbb{I}_t^n : \forall t, n\}$; **Output:** $\{\omega_t^n : \forall t, n\}$;

for $n = 1, \ldots, N$ *in parallel* **do**
  Initialize $M_n \leftarrow 0$, $S_n^\diamond \leftarrow 0$, $\omega_0^n \leftarrow 1$;
  for $t = 1, \ldots, T-1$ **do**
    $S_n^\diamond \leftarrow S_n^\diamond + 1$;
    if $\mathbb{I}_{t-1}^n = 1$ *or* $S_n^\diamond = K$ **then**
      $S_n \leftarrow S_n^\diamond$; // final interval computed
      $\omega_t^n \leftarrow \begin{cases} S_n, & \text{if } M_n = 0 \\ \frac{M_n \cdot \omega_{t-1}^n + S_n}{M_n + 1}, & \text{if } M_n \geq 1 \end{cases}$;
      $M_n \leftarrow M_n + 1$;
      $S_n^\diamond \leftarrow 0$;
    else
      $\omega_t^n \leftarrow \omega_{t-1}^n$;

Cutoff condition of interval length

Online interval computation and averaging

Convergence Analysis

Assumption 1. *The local objective functions are L-smooth, such that*

$$\|\nabla F_n(\mathbf{x}) - \nabla F_n(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}, n.$$

Assumption 2. *The local stochastic gradients and unbiased with bounded variance, such that*

$$\mathbb{E}\left[\mathbf{g}_n(\mathbf{x}) \middle| \mathbf{x}\right] = \nabla F_n(\mathbf{x}) \text{ and } \mathbb{E}\left[\left\|\mathbf{g}_n(\mathbf{x}) - \nabla F_n(\mathbf{x})\right\|^2 \middle| \mathbf{x}\right] \leq \sigma^2, \forall \mathbf{x}, n.$$

In addition, the stochastic gradient noise $\mathbf{g}_n(\mathbf{x}) - \nabla F_n(\mathbf{x})$ is independent across different rounds (indexed by t), clients (indexed by n), and local update steps (indexed by i).

Assumption 3. *The divergence between local and global gradients is bounded, such that*

$$\|\nabla F_n(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \delta^2, \forall \mathbf{x}, n.$$

Assumption 4. *The client participation random variable $\mathbb{I}_t^n$ is independent across different t and n. It is also independent of the stochastic gradient noise. For each client n, we define p_n such that $\mathbb{E}[\mathbb{I}_t^n] = p_n$, i.e., $\mathbb{I}_t^n \sim \text{Bernoulli}(p_n)$, where the value of p_n is unknown to the system a priori.*

Standard assumptions

Assumption 5. *We assume that* **either** *of the following holds and define Ψ_G accordingly.*

- **Option 1:** Nearly optimal weights. *Under the assumption that $\frac{1}{N}\sum_{n=1}^{N}(p_n\omega_t^n - 1)^2 \leq \frac{1}{81}$ for all t, we define $\Psi_G := 0$.*
- **Option 2:** Bounded global gradient. *Under the assumption that $\|\nabla f(\mathbf{x})\|^2 \leq G^2$ for any $\mathbf{x}$, we define $\Psi_G := G^2$.*

Global gradient

Needed due to weight adaptation

Main Result

Theorem 2 (Convergence error w.r.t. (1)). *Let* $\gamma \leq \frac{1}{4\sqrt{15}LI}$ *and* $\gamma\eta \leq \min\left\{\frac{1}{2LI}; \frac{N}{27LIQ}\right\}$, *where* $Q := \max_{t\in\{0,\dots,T-1\}} \frac{1}{N}\sum_{n=1}^{N} p_n(\omega_t^n)^2$. *When Assumptions 1–5 hold, the result* $\{\mathbf{x}_t\}$ *obtained from Algorithm 1 satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(\mathbf{x}_t)\right\|^2\right]$$

Weight error term

$$\leq \mathcal{O}\left(\frac{\mathcal{F}}{\gamma\eta IT} + \boxed{\frac{\Psi_G + \delta^2 + \gamma^2L^2I\sigma^2}{NT}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\mathbb{E}\left[\left(p_n\omega_t^n - 1\right)^2\right]} + \frac{\gamma\eta LQ\left(I\delta^2+\sigma^2\right)}{N} + \gamma^2L^2I\left(I\delta^2+\sigma^2\right)\right),$$

where $\mathcal{F} := f(\mathbf{x}_0) - f^*$, *and* $f^* := \min_{\mathbf{x}} f(\mathbf{x})$ *is the truly minimum value of the objective in* (1).

Proof idea:

From smoothness,

Expectation conditioned on the algorithm state at the beginning of time t

$$\tilde{f}(\mathbf{x}) := \sum_{n=1}^{N}\varphi_n F_n(\mathbf{x})$$

$$\begin{aligned}
\mathbb{E}_t\left[\tilde{f}(\mathbf{x}_{t+1})\right] &\leq \tilde{f}(\mathbf{x}_t) - \gamma\eta\mathbb{E}_t\left[\left\langle\nabla\tilde{f}(\mathbf{x}_t), \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}_t^n\omega_t^n\sum_{i=0}^{I-1}\mathbf{g}_n(\mathbf{y}_{t,i}^n)\right\rangle\right] \\
&\quad + \frac{\gamma^2\eta^2L}{2}\mathbb{E}_t\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}_t^n\omega_t^n\sum_{i=0}^{I-1}\mathbf{g}_n(\mathbf{y}_{t,i}^n)\right\|^2\right] \\
&= \tilde{f}(\mathbf{x}_t) - \gamma\eta\mathbb{E}_t\left[\left\langle\nabla\tilde{f}(\mathbf{x}_t), \frac{1}{N}\sum_{n=1}^{N}p_n\omega_t^n\sum_{i=0}^{I-1}\nabla F_n(\mathbf{y}_{t,i}^n)\right\rangle\right] \\
&\quad + \frac{\gamma^2\eta^2L}{2}\mathbb{E}_t\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}_t^n\omega_t^n\sum_{i=0}^{I-1}\mathbf{g}_n(\mathbf{y}_{t,i}^n)\right\|^2\right]
\end{aligned}$$

Algorithm 1: FedAvg with pluggable aggregation weights

Input: γ, η, $\mathbf{x}_0$, I; **Output:** $\{\mathbf{x}_t : \forall t\}$;

Initialize $t_0 \leftarrow 0$, $\mathbf{u} \leftarrow \mathbf{0}$;
for $t = 0, \dots, T-1$ **do**
  for $n = 1, \dots, N$ *in parallel* **do**
    Sample $\mathbb{I}_t^n$ from an *unknown* process;
    if $\mathbb{I}_t^n = 1$ **then**
      $\mathbf{y}_{t,0}^n \leftarrow \mathbf{x}_t$;
      for $i = 0, \dots, I-1$ **do**
        $\mathbf{y}_{t,i+1}^n \leftarrow \mathbf{y}_{t,i}^n - \gamma\mathbf{g}_n(\mathbf{y}_{t,i}^n)$;
      $\Delta_t^n \leftarrow \mathbf{y}_{t,I}^n - \mathbf{x}_t$;
    else
      $\Delta_t^n \leftarrow \mathbf{0}$;
    $\omega_t^n \leftarrow$ `ComputeWeight`$(\{\mathbb{I}_\tau^n : \tau < t\})$;
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \frac{\eta}{N}\sum_{n=1}^{N}\omega_t^n\Delta_t^n$;

Main Result (cont'd)

Proof idea (cont'd):

$$\mathbb{E}_t\left[\tilde{f}(\mathbf{x}_{t+1})\right] \le \tilde{f}(\mathbf{x}_t) - \gamma\eta\mathbb{E}_t\left[\left\langle \nabla\tilde{f}(\mathbf{x}_t), \frac{1}{N}\sum_{n=1}^{N} p_n\omega_t^n \sum_{i=0}^{I-1}\nabla F_n(\mathbf{y}_{t,i}^n)\right\rangle\right] + \frac{\gamma^2\eta^2 L}{2}\mathbb{E}_t\left[\left\|\frac{1}{N}\sum_{n=1}^{N}\mathbb{I}_t^n\omega_t^n\sum_{i=0}^{I-1}\mathbf{g}_n(\mathbf{y}_{t,i}^n)\right\|^2\right]$$

Lemma B.3.3 (General descent lemma). *When* $\gamma \le \frac{1}{4\sqrt{15}LI}$ *and* $\gamma\eta \le \frac{1}{4LI}$, *we have*

$$\begin{aligned}
&\mathbb{E}_t\left[\tilde{f}(\mathbf{x}_{t+1})\right] \\
&\le \tilde{f}(\mathbf{x}_t) + \frac{3\gamma\eta IN}{2}\left(15L^2I\gamma^2\sigma^2 + \frac{27\tilde{\delta}^2}{8}\right)\sum_{n=1}^{N}\left(\frac{p_n\omega_t^n}{N} - \varphi_n\right)^2 \\
&\quad + \frac{5\gamma^3\eta L^2I^2}{2}(\sigma^2 + 6I\tilde{\delta}^2) + \frac{\gamma^2\eta^2 LI}{N^2}\left(\frac{17\sigma^2}{16} + \frac{27I\tilde{\delta}^2}{8}\right)\sum_{n=1}^{N} p_n(\omega_t^n)^2 \\
&\quad + \gamma\eta I\left[\frac{81N}{16}\sum_{n=1}^{N}\left(\frac{p_n\omega_t^n}{N} - \varphi_n\right)^2 + 15L^2I^2\gamma^2 + \frac{27\gamma\eta LI}{8N^2}\sum_{n=1}^{N} p_n(\omega_t^n)^2 - \frac{1}{4}\right]\cdot\left\|\nabla\tilde{f}(\mathbf{x}_t)\right\|^2.
\end{aligned}$$

$$\tilde{f}(\mathbf{x}) := \sum_{n=1}^{N}\varphi_n F_n(\mathbf{x})$$

Upper bound by a negative constant and move to LHS

- To obtain Theorem 1 (convergence to reweighted objective), choose $\varphi_n \propto p_n$ so that $\frac{p_n\omega_n}{N} - \varphi_n = 0$ (assuming $\omega_t^n = \omega_n, \forall t$)
- To obtain Theorem 2, choose $\varphi_n = \frac{1}{N}$ giving the original objective

Quick Recap

Theorem 1 (Objective minimized at convergence, informal). *When* $\mathbb{1}_t^n \sim \text{Bernoulli}(p_n)$ *and the weights are time-constant, i.e.,* $\omega_t^n = \omega_n$ *but generally* ω_n *may not be equal to* $\omega_{n'}$ $(n \neq n')$*, with properly chosen learning rates* γ *and* η *and some other assumptions, Algorithm 1 minimizes the following objective:*

$$h(\mathbf{x}) := \frac{1}{P} \sum_{n=1}^{N} \omega_n p_n F_n(\mathbf{x}),$$

where $P := \sum_{n=1}^{N} \omega_n p_n$.

Theorem 2 (Convergence error w.r.t. (1)). *Let* $\gamma \leq \frac{1}{4\sqrt{15}LI}$ *and* $\gamma\eta \leq \min\left\{\frac{1}{2LI}; \frac{N}{27LIQ}\right\}$*, where* $Q := \max_{t\in\{0,\ldots,T-1\}} \frac{1}{N} \sum_{n=1}^{N} p_n(\omega_t^n)^2$*. When Assumptions 1–5 hold, the result* $\{\mathbf{x}_t\}$ *obtained from Algorithm 1 satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\left\|\nabla f(\mathbf{x}_t)\right\|^2\right]$$

Weight error term

$$\leq \mathcal{O}\left(\frac{\mathcal{F}}{\gamma\eta IT} + \boxed{\frac{\Psi_G + \delta^2 + \gamma^2 L^2 I \sigma^2}{NT} \sum_{t=0}^{T-1} \sum_{n=1}^{N} \mathbb{E}\left[\left(p_n \omega_t^n - 1\right)^2\right]} + \frac{\gamma\eta LQ\left(I\delta^2 + \sigma^2\right)}{N} + \gamma^2 L^2 I\left(I\delta^2 + \sigma^2\right)\right),$$

where $\mathcal{F} := f(\mathbf{x}_0) - f^*$*, and* $f^* := \min_{\mathbf{x}} f(\mathbf{x})$ *is the truly minimum value of the objective in* (1).

Main Result (cont'd)

Confirms the bias-variance tradeoff:
- Small K → low variance, high bias
- Large K → high variance, low bias

Theorem 3 (Bounding the weight error term). *For $\{\omega_t^n\}$ obtained from Algorithm 2, when $T \geq 2$,*

$$\frac{1}{NT}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\mathbb{E}\left[(p_n\omega_t^n - 1)^2\right] \leq \mathcal{O}\left(\frac{K\log T}{T} + \frac{1}{N}\sum_{n=1}^{N}(1-p_n)^{2K}\right).$$

Related to variance

Related to bias

Proof idea:

"Cutoff" geometric distribution

$$\Pr\{S_n = k\} = \begin{cases} p_n(1-p_n)^{k-1}, & \text{if } 1 \leq k < K \\ (1-p_n)^{k-1}, & \text{if } k = K \end{cases} \qquad \mathbb{E}[S_n] = \frac{1}{p_n} - \frac{(1-p_n)^K}{p_n}; \quad \mathrm{Var}[S_n] = \frac{1-p_n}{p_n^2} - \frac{(2K-1)(1-p_n)^K}{p_n} - \frac{(1-p_n)^{2K}}{p_n^2}$$

$$\begin{aligned}
\mathbb{E}\left[(p_n\omega_t^n - 1)^2\right] &= (p_n)^2\mathbb{E}\left[\left(\omega_t^n - \left(\frac{1}{p_n} - \frac{(1-p_n)^K}{p_n}\right) - \frac{(1-p_n)^K}{p_n}\right)^2\right] \\
&= (p_n)^2\mathbb{E}\left[\left(\omega_t^n - \left(\frac{1}{p_n} - \frac{(1-p_n)^K}{p_n}\right)\right)^2\right] + (1-p_n)^{2K} \qquad \text{(decompose into variance + bias)} \\
&\leq (p_n)^2 \cdot \frac{\mathrm{Var}[S_n]}{\max\left\{\lfloor \frac{t}{K} \rfloor, 1\right\}} + (1-p_n)^{2K} \qquad \text{(at least } \left\lfloor \tfrac{t}{K} \right\rfloor \text{ samples of } S_n \text{ for estimating } \omega_n\text{)} \\
&\leq (p_n)^2 \cdot \frac{2K\mathrm{Var}[S_n]}{t} + (1-p_n)^{2K} \\
&\leq \frac{2K(1-p_n)}{t} + (1-p_n)^{2K}
\end{aligned}$$

Final Convergence Rate

$$\frac{1}{NT}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\mathbb{E}\left[(p_n\omega_t^n-1)^2\right]\leq\mathcal{O}\left(\frac{K\log T}{T}+\frac{1}{N}\sum_{n=1}^{N}(1-p_n)^{2K}\right)$$

Corollary 4 (Convergence of FedAU). *Let* $K=\log_c T$ *with* $c:=1/(1-\min_n p_n)^2$, $\gamma=\min\left\{\frac{1}{LI\sqrt{T}};\frac{1}{4\sqrt{15}LI}\right\}$, *and choose* η *such that* $\gamma\eta=\min\left\{\sqrt{\frac{\mathcal{F}N}{Q(I\delta^2+\sigma^2)LIT}};\frac{1}{2LI};\frac{N}{27LIQ}\right\}$. *When* $T\geq 2$, *the result* $\{\mathbf{x}_t\}$ *obtained from Algorithm 1 that uses* $\{\omega_t^n\}$ *obtained from Algorithm 2 satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2\right]$$

Upper bound of weight error term

Standard in FedAvg

$$\leq\mathcal{O}\left(\frac{\sigma\sqrt{L\mathcal{F}Q}}{\sqrt{NIT}}+\frac{\delta\sqrt{L\mathcal{F}Q}}{\sqrt{NT}}+\frac{\left(\Psi_G+\delta^2+\frac{\sigma^2}{IT}\right)R\log^2 T}{T}+\frac{L\mathcal{F}\left(1+\frac{Q}{N}\right)+\delta^2+\frac{\sigma^2}{I}}{T}\right),$$

where Q *and* Ψ_G *are defined in Theorem 2 and* $R:=1/\log c$.

Proof idea:

$$\frac{1}{NT}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\mathbb{E}\left[(p_n\omega_t^n-1)^2\right]\leq\mathcal{O}\left(\frac{K\log T}{T}+\frac{1}{N}\sum_{n=1}^{N}(1-p_n)^{2K}\right)\leq\mathcal{O}\left(\frac{R\log^2 T}{T}\right)$$

because $\frac{1}{N}\sum_{n=1}^{N}(1-p_n)^{2K}\leq(1-p)^{2K}=(1-p)^{2\log_c T}=\left(\frac{1}{\left(\frac{1}{1-p}\right)^2}\right)^{\log_c T}$

$$=\frac{1}{c^{\log_c T}}=\frac{1}{T}\leq\frac{\log_2^2 T}{T}\qquad\text{where } T\geq 2$$

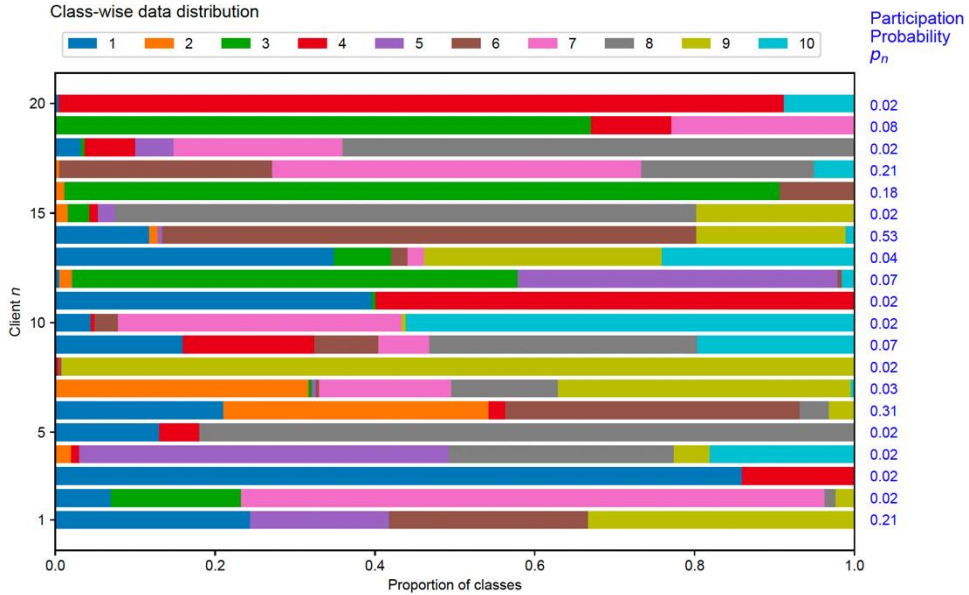
$\left(c:=\left(\frac{1}{1-p}\right)^2 \text{ and } p:=\min_n p_n\right)$

Experiments

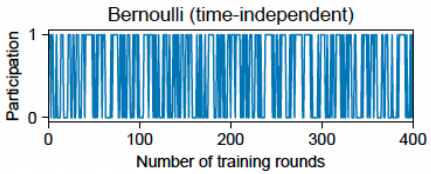
- Simulate correlated data distribution and participation probability
- Class-wise contribution to participation probability, $\mathbf{q} \sim \mathrm{Dir}(0.1)$

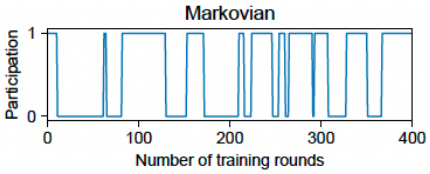
$$\mathbf{q} = [0.02,\ 0.05,\ 0.12,\ 0.00,\ 0.00,\ 0.78,\ 0.00,\ 0.00,\ 0.02,\ 0.00]$$

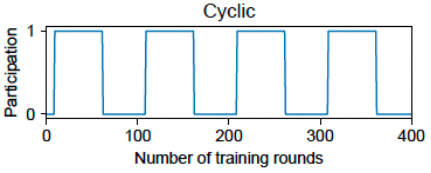
- Let $\boldsymbol{\kappa}_n \sim \mathrm{Dir}(0.1)$ be the class distribution at client n, the participation probability $p_n = \frac{1}{\lambda} \langle \boldsymbol{\kappa}_n, \mathbf{q} \rangle$



Experiments (cont'd)





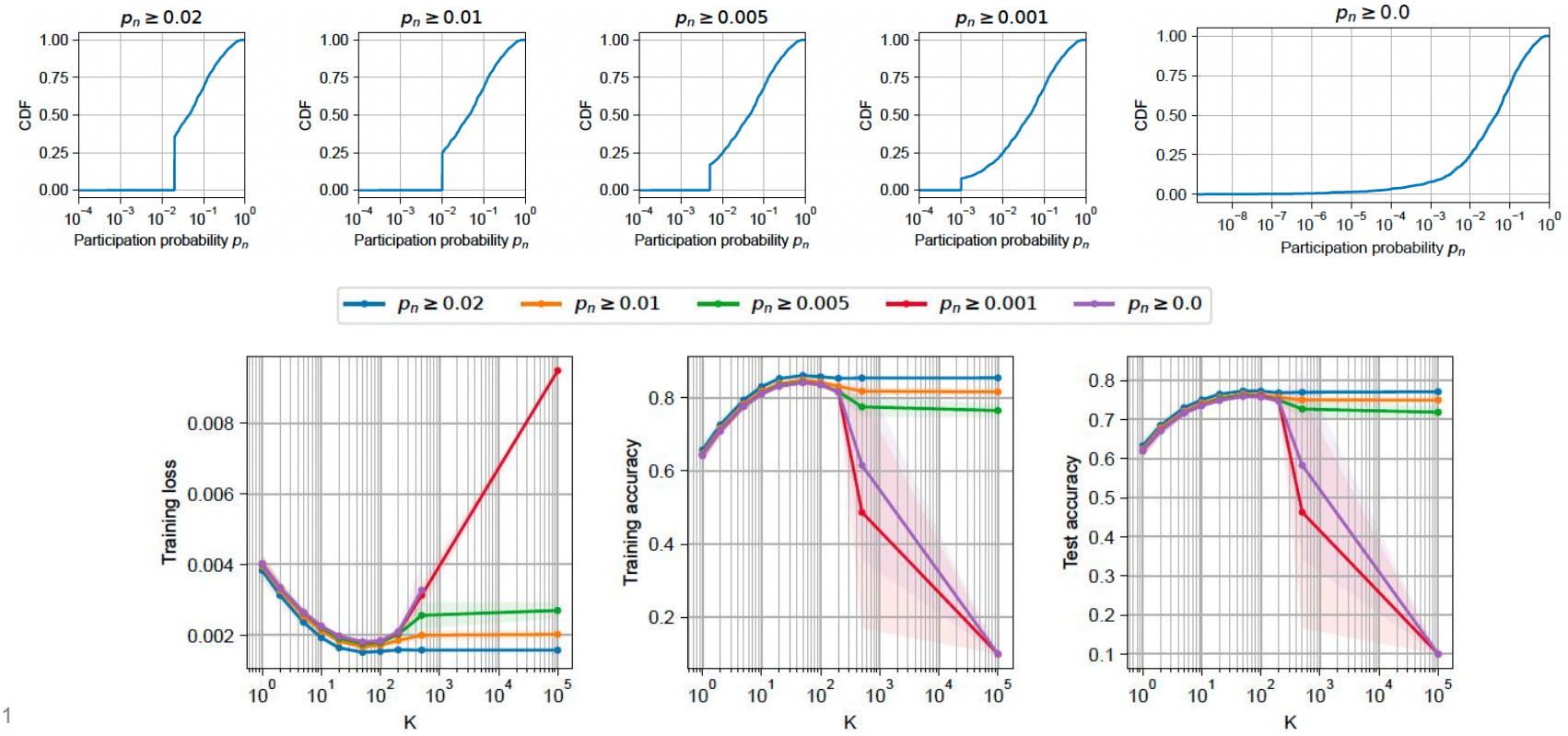


Participation pattern	Dataset	SVHN		CIFAR-10		CIFAR-100		CINIC-10	
	Method / Metric	Train	Test	Train	Test	Train	Test	Train	Test
Bernoulli (time-independent)	FedAU (ours, $K \to \infty$)	90.4±0.5	89.3±0.5	85.4±0.4	77.1±0.4	63.4±0.6	**52.3**±0.4	65.2±0.5	61.5±0.4
	FedAU (ours, $K = 50$)	**90.6**±0.4	**89.6**±0.4	**86.0**±0.5	**77.3**±0.3	**63.8**±0.3	52.1±0.6	**66.7**±0.3	**62.7**±0.2
	Average participating	89.1±0.3	87.2±0.3	83.5±0.9	74.1±0.8	59.3±0.4	48.8±0.7	61.1±2.3	56.6±2.0
	Average all	88.5±0.5	87.0±0.3	81.0±0.9	72.7±0.9	58.2±0.4	47.9±0.5	60.5±2.3	56.2±2.0
	FedVarp (250× memory)	89.6±0.5	88.9±0.5	84.2±0.3	77.9±0.2	57.2±0.9	49.2±0.8	64.4±0.6	62.0±0.5
	MIFA (250× memory)	89.4±0.3	88.7±0.2	83.5±0.6	77.5±0.3	55.8±1.1	48.4±0.7	63.8±0.7	61.5±0.5
	Known participation statistics	89.2±0.5	88.4±0.5	84.3±0.5	77.0±0.5	59.4±0.7	50.6±0.4	63.2±0.6	60.5±0.5
Markovian	FedAU (ours, $K \to \infty$)	90.5±0.4	89.3±0.4	85.3±0.3	77.1±0.3	63.2±0.5	**51.8**±0.3	64.9±0.3	61.2±0.2
	FedAU (ours, $K = 50$)	**90.6**±0.3	**89.5**±0.3	**85.9**±0.5	**77.2**±0.3	**63.5**±0.4	51.7±0.3	**66.3**±0.4	**62.3**±0.2
	Average participating	89.0±0.3	87.1±0.2	83.4±0.9	74.2±0.7	59.2±0.4	48.6±0.4	61.5±2.3	56.9±1.9
	Average all	88.4±0.6	86.8±0.7	80.8±1.0	72.5±0.5	57.8±0.9	47.7±0.5	59.9±2.8	55.7±2.2
	FedVarp (250× memory)	89.6±0.3	88.6±0.2	84.0±0.3	77.8±0.2	56.4±1.1	48.8±0.5	64.6±0.4	62.1±0.4
	MIFA (250× memory)	89.1±0.3	88.4±0.2	83.0±0.4	77.2±0.4	55.1±1.2	48.1±0.6	63.5±0.7	61.2±0.6
	Known participation statistics	89.5±0.2	88.6±0.2	84.5±0.4	76.9±0.3	59.7±0.5	50.3±0.5	63.5±0.9	60.7±0.6
Cyclic	FedAU (ours, $K \to \infty$)	89.8±0.6	88.7±0.6	84.2±0.8	76.3±0.7	60.9±0.6	50.6±0.3	63.5±1.0	60.0±0.8
	FedAU (ours, $K = 50$)	**89.9**±0.6	**88.8**±0.6	**84.8**±0.6	**76.6**±0.4	**61.3**±0.8	**51.0**±0.5	**64.5**±0.9	**60.9**±0.7
	Average participating	87.4±0.5	85.5±0.7	81.6±1.2	73.3±0.8	58.1±1.0	48.3±0.8	58.9±2.1	55.0±1.6
	Average all	89.1±0.8	87.4±0.8	83.1±1.0	73.8±0.8	59.7±0.3	48.8±0.4	62.9±1.7	57.6±1.5
	FedVarp (250× memory)	84.8±0.5	83.9±0.6	79.7±0.9	75.3±0.7	50.9±0.5	45.9±0.4	60.4±0.7	58.5±0.6
	MIFA (250× memory)	78.6±1.2	77.4±1.1	73.0±1.3	70.6±1.1	44.8±0.6	41.1±0.6	51.2±1.0	50.2±0.9
	Known participation statistics	89.9±0.7	88.7±0.6	83.6±0.7	76.1±0.5	60.2±0.4	50.8±0.4	62.6±0.8	59.8±0.7

- Same stationary probability for all participation patterns (but different across clients), initial state/offset is randomized
- Participation rate is correlated with heterogeneous data distribution

Experiments (cont'd)

- FedAU with different cutoff interval lengths K and minimum participation probability (CIFAR-10 dataset with Bernoulli participation)



FedAU Code

https://github.com/IBM/fedau

Recap

- Basics of federated learning
- Federated averaging (FedAvg) algorithm
- Tackling imbalanced client participation – FedAU
- Convergence analysis techniques
- Experiments and simulation code

Federated

Efficient

Scalable

Thank You!

Email: shiqiang.wang@ieee.org
Homepage: https://shiqiang.wang/