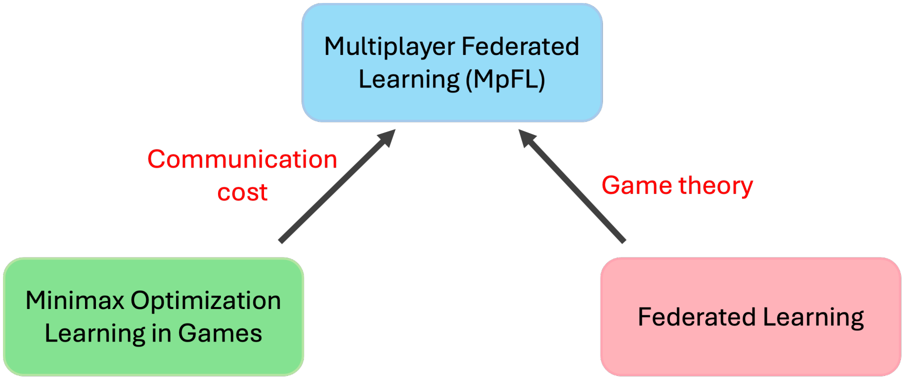
Multiplayer Federated Learning: Reaching Equilibrium with Less Communication

TaeHo Yoon, Sayantan Choudhury, Nicolas Loizou

Dept. of Applied Mathematics & Statistics
Mathematical Institute for Data Science
Johns Hopkins University

NeurIPS 2025

To whom will this be relevant?



Federated learning

Federated learning (FL)[1,2] is a framework for training a *shared global model* while local data are stored on each device *without being shared.*

Key idea: Assume local data holders can do computations (SGD) with their local data! Let them share *model updates*, rather than the data.

[1] Konečný et al. Federated learning: Strategies for improving communication efficiency. 2016.
[2] McMahan et al. Communication-efficient learning ... from decentralized data. *AISTATS*, 2017.

Federated learning as optimization

Optimization formulation:

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) = \textstyle\sum_{i=1}^n w_i f_i(x).$$

- x is the shared global model parameter
- $i = 1, \ldots, n$ is index of *clients* (local devices = data holders)
- Each client has a local dataset $\mathcal{D}_i$, and

$$f_i(x) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[\ell(x; \xi_i)\right] = \frac{1}{|\mathcal{D}_i|} \sum_{\xi \in \mathcal{D}_i} \ell(x; \xi)$$

- w_i are scalar weights; typically, $w_i = \frac{|\mathcal{D}_i|}{\sum_i |\mathcal{D}_i|}$.

Federated learning as optimization: Local SGD

Local SGD, or *Federated Averaging*[3], is the standard algorithm for federated learning.

Algorithm Local SGD (= FedAvg)

for $p = 0, \ldots, R-1$ **do**

  Server collects x_i^p from clients $i \in [n]$

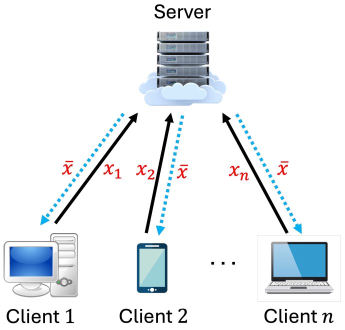
  Server computes and distributes $x^p = \sum_{i=1}^n w_i x_i^p$

  for each clients $i = 1, \ldots, n$ **in parallel do**

    $x_i^{p+1} = \mathtt{SGD}\left(f_i, x_i^{p,0} = x^p, \mathtt{hyperparams}\right)$

  end for

end for



The highlighted steps are called synchronization.

This requires communicating model parameters—the main computational bottleneck!

[3] McMahan et al. Communication-efficient learning ... from decentralized data. *AISTATS*, 2017.

Limitations of classical FL

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) = \sum_{i=1}^{n} w_i f_i(x)$$

Previous approach—classical FL—implicitly assumes:

- All clients use the same loss function and variable dimension (model structure).
- All clients fully collaborate toward minimizing f.
- All clients share the single global model x in the end.

In our work, we propose *a new FL framework addressing these limitations.*

Requirements for new FL framework

We propose a new FL concept where, unlike classical FL,

- Each client may have distinct objective function and model structure.
- Each client may not fully cooperate toward minimizing a global objective.
- Clients do not share a global model, but instead train their own local models.

How is it even possible to formulate this?

Game theory provides the perfect mathematical framework!

Multiplayer game theory

In a multiplayer game,

- Each player $i = 1, \ldots, n$ chooses action/strategy $x_i \in \mathbb{R}^{d_i}$.
- Each player has objective/cost $f_i(\mathbf{x}) = f_i(x_1, \ldots, x_n) \colon \mathbb{R}^{D = d_1 + \cdots + d_n} \to \mathbb{R}$.

Denote:

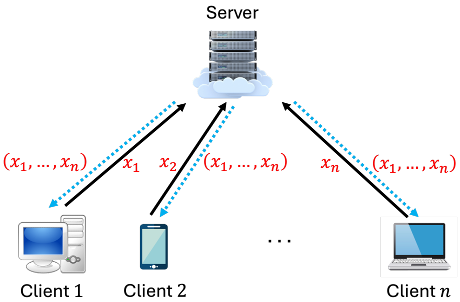
$$x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbb{R}^{D - d_i}$$
$$f_i(x_1, \ldots, x_n) = f_i(x_i; x_{-i})$$

$\mathbf{x}^\star = (x_1^\star, \ldots, x_n^\star) \in \mathbb{R}^D$ is an **equilibrium** iff

$$\underset{\mathbf{x}^\star = (x_1^\star, \ldots, x_n^\star) \in \mathbb{R}^D}{\text{find}} \quad f_i(x_i^\star; x_{-i}^\star) \leq f_i(x_i; x_{-i}^\star), \quad \forall x_i \in \mathbb{R}^{d_i}, \quad \forall i \in [n].$$

Multiplayer Federated Learning (MpFL)

Multiplayer Federated Learning (MpFL) is FL with game-theoretic formulation.



- Each client $i = 1, \ldots, n$ is a game player with action $x_i \in \mathbb{R}^{d_i}$.
- Each client has data $\mathcal{D}_i$ and objective $f_i(x_1, \ldots, x_n) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[f_{i,\xi_i}(x_1, \ldots, x_n)]$.
- The goal is to reach an equilibrium $\mathbf{x}^\star = (x_1^\star, \ldots, x_n^\star) \in \mathbb{R}^D$.
- Each client communicates with the server to send x_i and receive x_{-i}.

Communication complexity

We care about **communication efficiency**, i.e., the number of communications R needed to reach equilibrium:

$$\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \epsilon.$$

$\mathbf{x}^R$ is the joint action after R synchronization rounds.

PEARL-SGD: The first algorithm for MpFL

Algorithm Per-player local SGD (PEARL-SGD)

Input: Step-sizes $\gamma_k > 0$, # of local SGD iterations $\tau \geq 1$, # of synchronization $R \geq 1$

for $p = 0, \ldots, R-1$ **do**

Master server collects x_i^p from players $i = 1, \ldots, n$ and forms $\mathbf{x}^p = (x_1^p, \ldots, x_n^p)$

Master server distributes $\mathbf{x}^p$ back to players $i = 1, \ldots, n$

for each clients $i = 1, \ldots, n$ **in parallel do**

$x_i^{p+1} \leftarrow \texttt{SGD}\left(f_i(\cdot; x_{-i}^p), x_i^p, \tau, \{\gamma_k\}_{k=0}^{\tau}\right)$

end for

end for

Output: $\mathbf{x}^R \in \mathbb{R}^D$

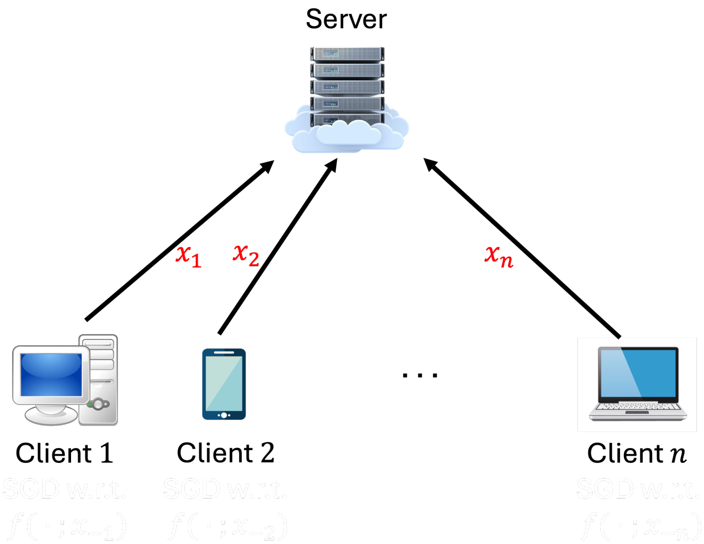
PEARL-SGD: The first algorithm for MpFL

Server

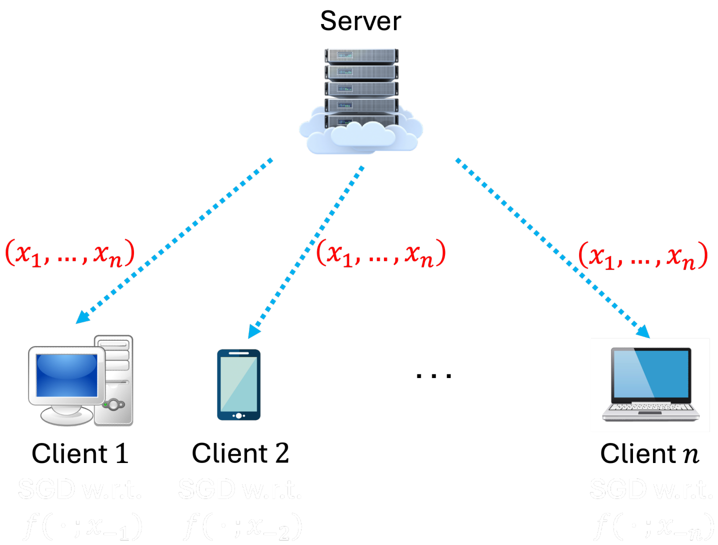
$\cdots$

Client 1	Client 2	Client n
SGD w.r.t. $f_1(\cdot\,;x_{-1})$	SGD w.r.t. $f_2(\cdot\,;x_{-2})$	SGD w.r.t. $f_n(\cdot\,;x_{-n})$

PEARL-SGD: The first algorithm for MpFL



PEARL-SGD: The first algorithm for MpFL



PEARL-SGD: Convergence analysis

Theorem

Under standard assumptions, let $\kappa = \ell/\mu$, $L_{\max} = \max\{L_1, \ldots, L_n\}$ and $q = L_{\max}/\sqrt{\ell\mu}$. Let τ the number of local SGD iterations per round, and let $T = \tau R$ be the total iteration number. Then PEARL-SGD with $\gamma_k \equiv \gamma = \frac{1}{\mu\eta(1+2q)}$ exhibits the rate

$$\mathbb{E}\left[\left\|\mathbf{x}^T - \mathbf{x}^\star\right\|^2\right] = \tilde{\mathcal{O}}\left(\frac{(1+q)^2 \left\|\mathbf{x}_0 - \mathbf{x}_\star\right\|^2}{T^2} + \frac{(1+q)\,\sigma^2}{\mu^2 T} + \frac{(1+q)\,\tau^2 L_{\max}\sigma^2}{\mu^3 T^2}\right)$$

if T is large enough and η is selected so that $T = 2\,(1+2q)\,\eta \log \eta$.

Takeaway. For non-local case ($\tau = 1$), one needs $T = R = \mathcal{O}\left(\epsilon^{-1}\right)$ communications to achieve $\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \epsilon$.

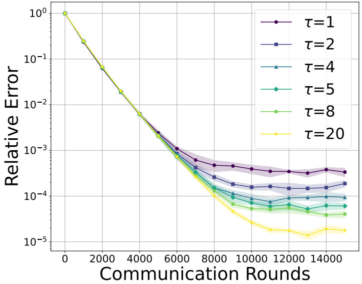
For PEARL-SGD with $\tau = \Theta\left(\sqrt{T}\right)$, this is reduced to $R = T/\tau = \mathcal{O}\left(\epsilon^{-1/2}\right)$.

Experiments

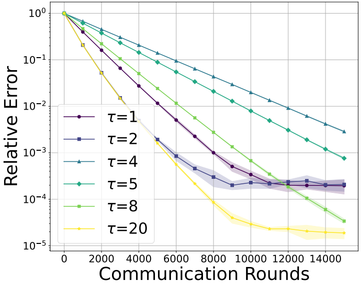
Consider an n-player game with $x_i \in \mathbb{R}^{d_i}$ and

$$f_i(x_i; x_{-i}) := \frac{1}{M}\sum_{m=1}^{M} \frac{1}{2}\langle x_i, \mathbf{A}_{i,m} x_i\rangle + \sum_{1\le j\le n, j\neq i} \langle x_i, \mathbf{B}_{i,j,m} x_j\rangle + \langle a_{i,m}, x_i\rangle$$

where $\mathbf{A}_{i,m} \in \mathbb{R}^{d_i\times d_i}, \mathbf{B}_{i,j,m} \in \mathbb{R}^{d_i\times d_j}, a_{i,m} \in \mathbb{R}^{d_i}$.



(a) Theoretical step-size



(b) Step-size by grid search

Conclusion and future work

Takeaways

- We develop a new framework, Multiplayer Federated Learning (MpFL), where clients of FL are players of a game.
- PEARL-SGD algorithm finds equilibrium with fewer communications!

Future work

- Theory side:
  - Convergence under weaker assumptions
  - Further acceleration under comparable setups
  - Decentralized setups w/o server
- Verifying empirical effectiveness of PEARL-SGD