FedNest: Federated Bilevel, Minimax, and Compositional Optimization

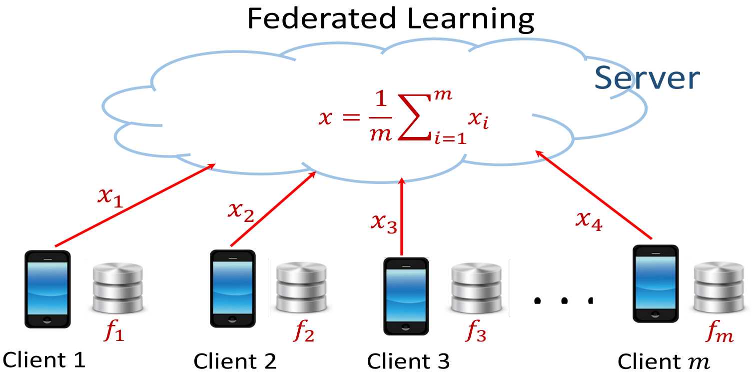
Davoud Ataee Tarzanagh[1] Mingchen Li[2] Christos Thrampoulidis[3]
Samet Oymak[2]

[1] University of Michigan

[2] University of California, Riverside

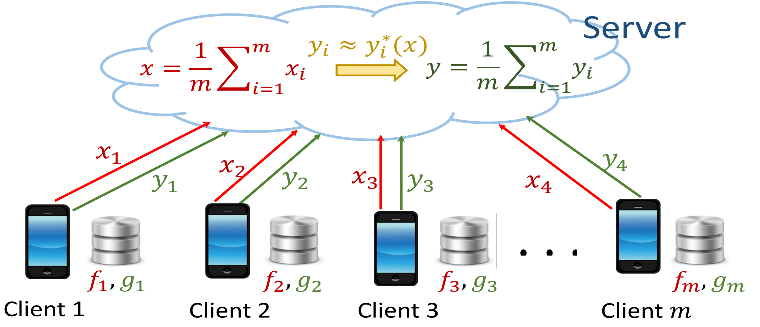
[3] University of British Columbia

ICML, July 17-23 2022



Goal:
$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

Federated Bilevel Learning

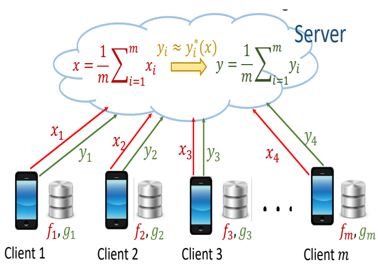


Goal:

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \quad f(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) \qquad \text{(outer)}$$

$$\text{subj. to} \quad \boldsymbol{y}^*(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \frac{1}{m}\sum_{i=1}^{m} g_i(\boldsymbol{x}, \boldsymbol{y}) \qquad \text{(inner)}$$

Federated Bilevel Optimization (FBO)



Our Setting:

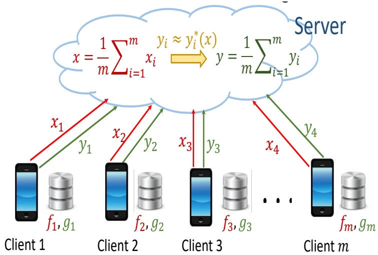
- **Stochastic**: Access to (f_i, g_i) is via stochastic sampling:

$$f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) := \mathbb{E}_{\xi \sim \mathcal{C}_i}\left[f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}); \xi)\right],$$
$$g_i(\boldsymbol{x}, \boldsymbol{y}) := \mathbb{E}_{\zeta \sim \mathcal{D}_i}\left[g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta)\right],$$

  where $(\xi, \zeta) \sim (\mathcal{C}_i, \mathcal{D}_i)$.

- **Heterogeneous**: For $i \neq j$, the tuples $(f_i, g_i, \mathcal{C}_i, \mathcal{D}_i)$ and $(f_j, g_j, \mathcal{C}_j, \mathcal{D}_j)$ can be different.

Federated Bilevel Optimization (FBO)



Our Setting:

- **Stochastic**: Access to (f_i, g_i) is via stochastic sampling:

$$f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) := \mathbb{E}_{\xi \sim \mathcal{C}_i}\left[f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}); \xi)\right],$$
$$g_i(\boldsymbol{x}, \boldsymbol{y}) := \mathbb{E}_{\zeta \sim \mathcal{D}_i}\left[g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta)\right],$$

where $(\xi, \zeta) \sim (\mathcal{C}_i, \mathcal{D}_i)$.

- **Heterogeneous**: For $i \neq j$, the tuples $(f_i, g_i, \mathcal{C}_i, \mathcal{D}_i)$ and $(f_j, g_j, \mathcal{C}_j, \mathcal{D}_j)$ can be different.

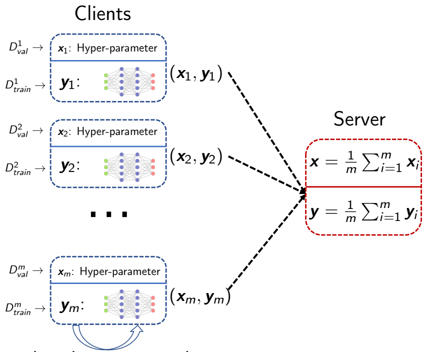
Applications of FBO: meta-learning, hyperparameter optimization, neural network architecture search, actor-critic reinforcement learning, GANs,. . .

Motivating Example

- **Federated Hyper-Parameter Optimization**: Collaboratively find machine learning (ML) model and the hyper-parameters while keeping the data decentralized

Motivating Example

- **Federated Hyper-Parameter Optimization**: Collaboratively find machine learning (ML) model and the hyper-parameters while keeping the data decentralized



- Inner objective: $\frac{1}{m}\sum_{i=1}^m g\left(x, y; \mathcal{D}_{\text{train}}^i\right)$
- Outer objective: $\frac{1}{m}\sum_{i=1}^m f\left(y^*(x); \mathcal{D}_{\text{val}}^i\right)$
- y is an ML model such as neural network.
- x contains hyper-parameters such as
  - regularization parameters,
  - learning rates, and
  - batch size.

Two Special Cases

- FBO subsumes two popular problem classes with the nested structure.

Federated Minimax Optimization

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{d_1}} \quad & f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i\left(\mathbf{x}, \mathbf{y}^*(\mathbf{x}); \xi\right) \\ \text{subj. to} \quad & \mathbf{y}^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^{d_2}} \; -\frac{1}{m}\sum_{i=1}^{m} f_i\left(\mathbf{x}, \mathbf{y}; \xi\right) \end{aligned}$$

FBO with
$g_i(\mathbf{x}, \mathbf{y}; \zeta) = -f_i(\mathbf{x}, \mathbf{y}; \xi)$.

Application:

- Training Generative-adversarial Networks (GANs)

Two Special Cases

- FBO subsumes two popular problem classes with the nested structure.

Federated Minimax Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^{d_1}} \quad f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i\left(\mathbf{x}, \mathbf{y}^*(\mathbf{x}); \xi\right)$$
$$\text{subj. to} \quad \mathbf{y}^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^{d_2}} \; -\frac{1}{m}\sum_{i=1}^{m} f_i\left(\mathbf{x}, \mathbf{y}; \xi\right)$$

FBO with
$g_i(\mathbf{x}, \mathbf{y}; \zeta) = -f_i(\mathbf{x}, \mathbf{y}; \xi)$.

Application:

- Training Generative-adversarial Networks (GANs)

Federated Compositional Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^{d_1}} \quad f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{y}; \xi)$$
$$\text{subj. to} \quad \mathbf{y}^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^{d_2}} \; \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{y} - \mathbf{r}_i(\mathbf{x}; \zeta)\|^2$$

FBO with
$f_i(\mathbf{x}, \mathbf{y}; \xi) = f_i(\mathbf{y}; \xi)$ and
$g_i(\mathbf{x}, \mathbf{y}; \zeta) = \|\mathbf{y} - \mathbf{r}_i(\mathbf{x}; \zeta)\|^2$.

Application:

- Model Agnostic Meta-Learning (MAML)

Federated (Single-Level) Optimization

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d_1}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\xi \sim \mathcal{C}_i}[f_i(\boldsymbol{x}; \xi)]$$

Federated (Single-Level) Optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\xi\sim\mathcal{C}_i}[f_i(\boldsymbol{x};\xi)]$$

Gradient-Type Federated Optimization

For $k = 0, \cdots, K-1$:

① i-th client:

- For $\nu = 0, \ldots, \tau_i - 1$: $\boldsymbol{x}_{i,\nu+1}^k = \boldsymbol{x}_{i,\nu}^k + \alpha_i^k \boldsymbol{h}_{i,\nu}^k$

② Server: $\boldsymbol{x}^{k+1} = 1/m \sum_{i=1}^m \boldsymbol{x}_{i,\tau_i}^k$

- τ_i is number of local iterations
- α_i^k is the stepsize
- **FedAvg** (McMahan et al., 2017): $\boldsymbol{h}_{i,\nu}^k = -\nabla f_i(\boldsymbol{x}_{i,\nu}^k; \xi_{i,\nu}^k)$
- **FedSVRG**(Konečný et al., 2018): $\boldsymbol{h}_{i,\nu}^k = -\nabla f_i(\boldsymbol{x}_{i,\nu}^k; \xi_{i,\nu}^k) + \nabla f_i(\boldsymbol{x}^k; \xi_{i,\nu}^k) - \frac{1}{m}\sum_{i=1}^m \nabla f_i(\boldsymbol{x}^k; \xi_i^k)$

Challenges in FBO

- **FedAvg** can lead to convergence to a point different from $\boldsymbol{y}^*(\boldsymbol{x})$.
- Each client i requires access to the global Hessian inverse:

$$\nabla f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) = \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) - \nabla^2_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) \cdot \underbrace{\left[\sum_{i=1}^{m} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))\right]^{-1} \nabla_{\boldsymbol{y}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right)}_{\boldsymbol{p}_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))}$$

Challenges in FBO

- **FedAvg** can lead to convergence to a point different from $\boldsymbol{y}^*(\boldsymbol{x})$.
- Each client i requires access to the global Hessian inverse:

$$\nabla f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) = \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})) - \nabla^2_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) \cdot \underbrace{\left[\sum_{i=1}^{m} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))\right]^{-1} \nabla_{\boldsymbol{y}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right)}_{\boldsymbol{p}_i(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x}))}$$

Our approaches:

- Use **FedSVRG** to solve the inner problem.
- Estimate $\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$ via a Federated Inverse Hessian-Gradient-Product (**FedIHGP**).

FedIHGP

- N–Neumann series approximation (Ghadimi & Wang, 2018):

$$\boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}) := \sum_{i=1}^{m}\Big[\sum_{i=1}^{m}\nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x},\boldsymbol{y})\Big]^{-1}\nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y})$$

$$\approx \sum_{i=1}^{m}\Big[\frac{N}{\ell_{g,1}}\prod_{n=1}^{N'}\sum_{i=1}^{m}\big(\boldsymbol{I}-\frac{1}{\ell_{g,1}}\nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x},\boldsymbol{y};\zeta_n)\big)\Big]\nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y};\dot{\xi})$$

- **FedIHGP** provides a federated recursive strategy to estimate $\boldsymbol{p}$.

$\boldsymbol{p}_{N'}$ = FedIHGP $(\boldsymbol{x},\boldsymbol{y},N)$

Select $N' \in \{0,\ldots,N-1\}$, $\mathcal{S}_0,\ldots,\mathcal{S}_{N'} \in \mathcal{S}$ UAR. Set

① i-th client: $\boldsymbol{p}_{i,0} = \nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y};\xi_{i,0})$

② server: $\boldsymbol{p}_0 = \frac{N}{\ell_{g,1}}|\mathcal{S}_n|^{-1}\sum_{i\in\mathcal{S}_0}\boldsymbol{p}_{i,0}$

If $N'=0$ Return $\boldsymbol{p}_{N'}$.

For $n=1,\ldots,N'$:

① i-th client: $\boldsymbol{p}_{i,n} = \big(\boldsymbol{I}-\frac{1}{\ell_{g,1}}\nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x},\boldsymbol{y};\zeta_{i,n})\big)\boldsymbol{p}_{n-1}$

② server: $\boldsymbol{p}_n = |\mathcal{S}_n|^{-1}\sum_{i\in\mathcal{S}_n}\boldsymbol{p}_{i,n}$

- **FedIHGP** avoids explicit Hessian:
  - matrix-vector products
  - vector communications
- $\|p(\boldsymbol{x},\boldsymbol{y})-\mathbb{E}[\boldsymbol{p}_{N'}]\| \le \mathcal{O}\big(\big(\underbrace{\frac{\kappa_g-1}{\kappa_g}}_{<1}\big)^N\big)$.
  - $\kappa_g := \frac{\ell_{g,1}}{\mu_g}$ (condition number).

Proposed Algorithm: FedNest

For $k = 0, \cdots, K-1$

① $\boldsymbol{y}^{k+1} = \textbf{FedInn}\left(\boldsymbol{x}^k, \boldsymbol{y}^k, \beta^k\right)$ // one or multiple FedSVRGs on $\boldsymbol{y}$

② $\boldsymbol{x}^{k+1} = \textbf{FedOut}\left(\boldsymbol{x}^k, \boldsymbol{y}^{k+1}, \alpha^k\right)$ // FedSVRG + FedIHGP on $\boldsymbol{x}$

Proposed Algorithm: FedNest

For $k = 0, \cdots, K-1$

① $\boldsymbol{y}^{k+1} = \textbf{FedInn}\left(\boldsymbol{x}^k, \boldsymbol{y}^k, \beta^k\right)$ // one or multiple FedSVRGs on $\boldsymbol{y}$

② $\boldsymbol{x}^{k+1} = \textbf{FedOut}\left(\boldsymbol{x}^k, \boldsymbol{y}^{k+1}, \alpha^k\right)$ // FedSVRG + FedIHGP on $\boldsymbol{x}$

Inner Optimizer: **FedInn**

- $\boldsymbol{y}_k \approx \boldsymbol{y}^*(\boldsymbol{x}_k)$
- It avoids inner client drift: $\left\|\boldsymbol{y}_{i,\nu}^k - \boldsymbol{y}^k\right\|^2 \leq O(\tau_i(\beta_i^k)^2)$
- The global convergence of **FedInn** ensures accurate hypergradient computation.

Proposed Algorithm: FedNest

For $k = 0, \cdots, K-1$

① $\boldsymbol{y}^{k+1} = \textbf{FedInn}\left(\boldsymbol{x}^k, \boldsymbol{y}^k, \beta^k\right)$ // one or multiple FedSVRGs on $\boldsymbol{y}$

② $\boldsymbol{x}^{k+1} = \textbf{FedOut}\left(\boldsymbol{x}^k, \boldsymbol{y}^{k+1}, \alpha^k\right)$ // FedSVRG + FedIHGP on $\boldsymbol{x}$

Inner Optimizer: **FedInn**

- $\boldsymbol{y}_k \approx \boldsymbol{y}^*(\boldsymbol{x}_k)$
- It avoids inner client drift: $\left\|\boldsymbol{y}_{i,\nu}^k - \boldsymbol{y}^k\right\|^2 \leq O(\tau_i(\beta_i^k)^2)$
- The global convergence of **FedInn** ensures accurate hypergradient computation.

Outer Optimizer: **FedOut**

- It avoids outer client drift:

$$\left\|\boldsymbol{x}_{i,\nu}^k - \boldsymbol{x}^k\right\|^2 \leq O\big(\tau_i(\alpha_i^k)^2 + \left\|\boldsymbol{y}^{k+1} - \boldsymbol{y}^*(\boldsymbol{x}^k)\right\|^2\big)$$

- It gives new convergence guarantees for federated bilevel, minimiax, compositional, and single-level optimization.

Theory

Assumptions

- $f_i(\boldsymbol{z}), \nabla f_i(\boldsymbol{z}), \nabla g_i(\boldsymbol{z}), \nabla^2 g_i(\boldsymbol{z})$ are $\ell_{f,0}$,$\ell_{f,1}$,$\ell_{g,1}$, $\ell_{g,2}$-Lipschitz continuous, respectively.
- $g_i(\boldsymbol{x}, \boldsymbol{y})$ is μ_g-strongly convex in $\boldsymbol{y}$ for all $\boldsymbol{x} \in \mathbb{R}^{d_1}$.
- $\nabla f_i(\boldsymbol{z}; \xi)$, $\nabla g_i(\boldsymbol{z}; \zeta)$, $\nabla^2 g_i(\boldsymbol{z}; \zeta)$ are unbiased estimators of $\nabla f_i(\boldsymbol{z})$, $\nabla g_i(\boldsymbol{z})$, $\nabla^2 g_i(\boldsymbol{z})$; and their variances are bounded.

- These assumptions are common in the (non-federated) BO literature.

Theory

Assumptions

- $f_i(\boldsymbol{z}), \nabla f_i(\boldsymbol{z}), \nabla g_i(\boldsymbol{z}), \nabla^2 g_i(\boldsymbol{z})$ are $\ell_{f,0}$,$\ell_{f,1}$,$\ell_{g,1}$, $\ell_{g,2}$-Lipschitz continuous, respectively.
- $g_i(\boldsymbol{x}, \boldsymbol{y})$ is μ_g-strongly convex in $\boldsymbol{y}$ for all $\boldsymbol{x} \in \mathbb{R}^{d_1}$.
- $\nabla f_i(\boldsymbol{z}; \xi)$, $\nabla g_i(\boldsymbol{z}; \zeta)$, $\nabla^2 g_i(\boldsymbol{z}; \zeta)$ are unbiased estimators of $\nabla f_i(\boldsymbol{z})$, $\nabla g_i(\boldsymbol{z})$, $\nabla^2 g_i(\boldsymbol{z})$; and their variances are bounded.

- These assumptions are common in the (non-federated) BO literature.

Theorem (Informal)

Under the above assumptions, if we choose the stepsize properly, then the iterates of FedNest satisfy

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}^k)\right\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right) \text{ and } \mathbb{E}\left[\left\|\boldsymbol{y}^K - \boldsymbol{y}^*(\boldsymbol{x}^K)\right\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

Theory: Comparison with Previous Results

- Sample complexity of FedNest and comparable non-FL methods to find an ϵ-stationary point of f, i.e., $1/K \sum_{k=1}^{K} \mathbb{E}[\|\nabla f(\mathbf{x}^k)\|^2] \leq \epsilon$:
- $\kappa_g = \ell_{g,1}/\mu_g$ (condition number).

		Non-Federated		
	FedNest	ALSET	BSA	TTSA
batch size	$\mathcal{O}(1)$			
samples in ξ	$\mathcal{O}(\kappa_g^5 \epsilon^{-2})$	$\mathcal{O}(\kappa_g^5 \epsilon^{-2})$	$\mathcal{O}(\kappa_g^6 \epsilon^{-2})$	$\mathcal{O}(\kappa_g^p \epsilon^{-2.5})$
samples in ζ	$\mathcal{O}(\kappa_g^9 \epsilon^{-2})$	$\mathcal{O}(\kappa_g^9 \epsilon^{-2})$	$\mathcal{O}(\kappa_g^9 \epsilon^{-3})$	$\mathcal{O}(\kappa_g^p \epsilon^{-2.5})$

ALSET(Chen et al., 2021), **BSA**(Ghadimi & Wang, 2018), **TTSA**(Hong et al., 2020).

- Main takeaways:
  - **FedNest** enjoys the same convergence as non-federated alternating SGD (**ALSET**), despite objective heterogeneity.

LFedNest: Communication Efficiency via Local Hypergradient

Light-FedNest (LFedNest):

- computes hypergradients locally
- only needs a single communication round for the outer update

	definition		properties			
	outer optimizer	inner optimizer	global outer gradient	global IHGP	global inner gradient	# communication rounds
FedNest	SVRG on $\boldsymbol{x}$	SVRG on $\boldsymbol{y}$	yes	yes	yes	$2T + N + 3$
LFedNest	SGD on $\boldsymbol{x}$	SGD on $\boldsymbol{y}$	no	no	no	$T + 1$
FedNest$_{\text{SGD}}$	SVRG on $\boldsymbol{x}$	SGD on $\boldsymbol{y}$	yes	yes	no	$T + N + 3$

- T: # inner iterations ($\boldsymbol{y}$ update)
- N: # terms of Neumann series

Minimax Experiment

- Minimax saddle point problem (on non-i.i.d. synthetic dataset):

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} f(\boldsymbol{x}) := \frac{1}{m} \max_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \sum_{i=1}^{m} f_i(\boldsymbol{x}, \boldsymbol{y}),$$

where

$$f_i(\boldsymbol{x}, \boldsymbol{y}) := -\left[\frac{1}{2}\|\boldsymbol{y}\|^2 - \boldsymbol{b}_i^\top \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{A}_i \boldsymbol{x}\right] + \frac{\lambda}{2}\|\boldsymbol{x}\|^2,$$

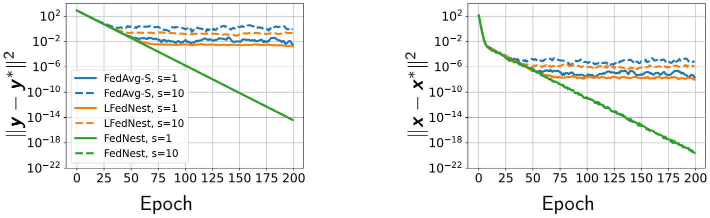
Minimax Experiment

- Minimax saddle point problem (on non-i.i.d. synthetic dataset):

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \; f(\boldsymbol{x}) := \frac{1}{m} \max_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \sum_{i=1}^{m} f_i(\boldsymbol{x},\boldsymbol{y}),$$

where

$$f_i(\boldsymbol{x},\boldsymbol{y}) := -\left[\frac{1}{2}\|\boldsymbol{y}\|^2 - \boldsymbol{b}_i^\top \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{A}_i \boldsymbol{x}\right] + \frac{\lambda}{2}\|\boldsymbol{x}\|^2,$$



- LFedNest performs slightly better than FedAvg-S (Hou et al., 2021).
- FedNest converges linearly despite heterogeneity.

Hyperparameter Tuning for Label Imbalance

- Imbalanced classification is the problem of classification when there is an unequal distribution of classes in the training dataset.
- Goal: Design fairness-seeking loss functions via bilevel optimization

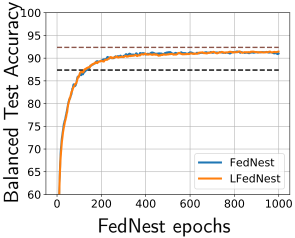
$$\underset{\boldsymbol{x}=(\boldsymbol{\Delta},\boldsymbol{l})}{\text{minimize}} \quad \sum_{i=1}^{m} \overbrace{\sum_{(\boldsymbol{u},t)\in\mathcal{D}^i_{val}} \frac{1}{p_t} \log\Big(1+\sum_{s\neq t} e^{\boldsymbol{y}_s(\boldsymbol{u})-\boldsymbol{y}_t(\boldsymbol{u})}\Big)}^{=:f_i(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x}))\Leftarrow\text{Balanced Val Loss}}$$

$$\text{s.t.} \quad \boldsymbol{y}^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \sum_{i=1}^{m} \underbrace{\sum_{(\boldsymbol{u},t)\in\mathcal{D}^i_{train}} \log\Big(1+\sum_{s\neq t} e^{\boldsymbol{l}_s-\boldsymbol{l}_t} \cdot e^{\boldsymbol{\Delta}_s\boldsymbol{y}_s(\boldsymbol{u})-\boldsymbol{\Delta}_t\boldsymbol{y}_t(\boldsymbol{u})}\Big)}_{=:g_i(\boldsymbol{x},\boldsymbol{y})\Leftarrow\text{Parametric Train Loss}}.$$

- Outer optimization aims to maximize the class-balanced validation accuracy.
- Inner optimization trains model parameter $\boldsymbol{y}$ to minimize $g=\sum_{i=1}^{m} g_i$.

Hyperparameter Tuning for Label Imbalance

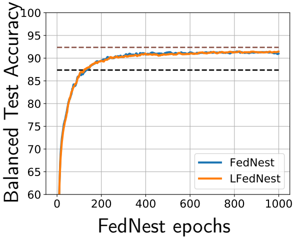
- Brown dashed line: Non-Federated bilevel training
- **Black** dashed line: Non-Federated accuracy without bilevel tuning



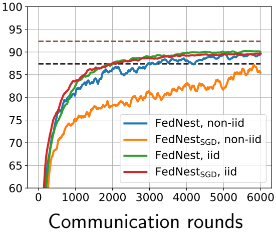
- I.I.D. setup: FedNest behaves similarly to LFedNest.

Hyperparameter Tuning for Label Imbalance

- Brown dashed line: Non-Federated bilevel training
- **Black** dashed line: Non-Federated accuracy without bilevel tuning



- I.I.D. setup: FedNest behaves similarly to LFedNest.



- FedNest (with SVRG) is significatlly better than FedNest$_{\text{SGD}}$ (with SGD in FedInn).

Prior Work

Federated **(Single-Level)** Optimization

FedAvg (McMahan et al., 2017)
FedProx (Li et al., 2020)
FedNova (Wang et al., 2020)
FedSVRG (Konečný et al., 2018)
SCAFFOLD (Karimireddy et al., 2020)
FedLin (Mitra et al., 2021)

Stochastic **Bilevel** Optimization

BSA (Ghadimi & Wang, 2018)
ALSET (Chen et al., 2021)
TTSA (Hong et al., 2020)
stocBiO (Ji et al., 2020)

Stochastic **MiniMax** Optimization

SGDA (Lin et al., 2020)
SMD (Rafique et al., 2021)

Stochastic **Compositional** Optimization

SCGD (Wang et al., 2017)
NASA (Ghadimi et al., 2020)

Conclusion and Future Work

- **Conclusion:**
  - **FedNest** gives a new framework for federated bilevel, minimax, and compositional optimization.
  - **FedNest** matches the sample complexity of the alternating SGD.

	FedNest			
	Bilevel	Minimiax	Compositional	Single-Level
batch size	$\mathcal{O}(1)$			
samples complexity	$\mathcal{O}(\epsilon^{-2})$			

- **Future Work:**
  - Other applications and properties of FedNest such as federated actor-critic reinforcement learning.
  - Sparsification or quantization for communication-efficient FBO

Paper: ICML 2022, arXiv
Code: github.com/ucr-optml/FedNest

References

Chen, T., Sun, Y., and Yin, W. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34, 2021.

Ghadimi, S. and Wang, M. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Ghadimi, S., Ruszczynski, A., and Wang, M. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

Hou, C., Thekumparampil, K. K., Fanti, G., and Oh, S. Efficient algorithms for federated saddle point optimization. *arXiv preprint arXiv:2102.06333*, 2021.

Ji, K., Yang, J., and Liang, Y. Provably faster algorithms for bilevel optimization and applications to meta-learning. *ArXiv*, abs/2010.07962, 2020.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Konečný, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *International Conference on Learning Representations*, 2018.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Lin, T., Jin, C., and Jordan, M. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pp. 6083–6093. PMLR, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Mitra, A., Jaafar, R., Pappas, G., and Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34, 2021.

Rafique, H., Liu, M., Lin, Q., and Yang, T. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, pp. 1–35, 2021.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 2020.

Wang, M., Fang, E. X., and Liu, H. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017.