Federated Hyperparameter Tuning: Challenges, Baselines, and Connections to Weight-Sharing

Authors: Khodak et al.

Presenters: Aurélien Bondis, Parsa Toopchinezhad

Presentation Outline

1- Introduction

2- Background

3- Main

4- Results

5- Conclusion

6- Questions

Introduction

Motivation

- Hyperparameter Optimization (HPO) is an expensive yet crucial part of the ML pipeline
- Standard FL algorithms (e.g., FedAVG) do not handle HPO
- HPO is more difficult in FL

Challenges of Federated Hyperparameter Tuning

- 1: Federated validation data
- 2: Extreme resource limitations
- 3: Evaluating personalization

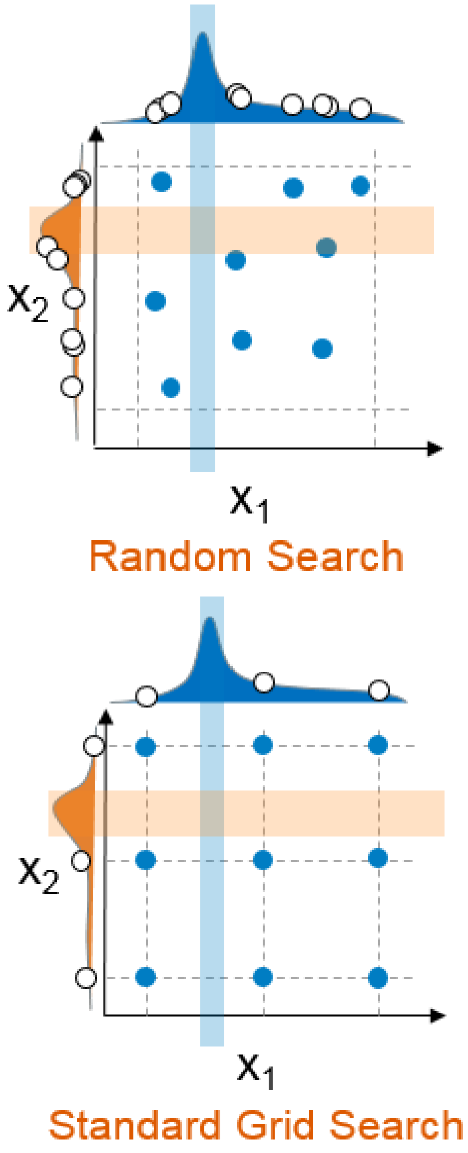
Paper Goals

- Understand the difficulties of HPO in FL
- Formulate the HPO problem for FL
- Create a standard baseline
- Propose FedEx, solving aforementioned challenges
- Empirically test FedEx

Background

Hyperparameter Optimization

- ML-models have two sets of parameters:
  - **θ**: learnable weights found during training
  - **α**: hyper-parameters controlling learning process
- Hyperparameter tuning is more of a dark-art, requiring trial and error
- Popular HPO methods:
  - Manual tuning
  - Random search
  - Grid search
  - SHA
  - Bayesian Optimization



Federated Hyperparameter Tuning

Global setting:

$$\min_{a \in \mathcal{A}} \sum_{i=1}^{n} |V_i| L_{V_i}(\texttt{Alg}_a(\{T_j\}_{j=1}^{n}))$$

Personalized setting:

$$\min_{a=(b,c) \in \mathcal{A}} \sum_{i=1}^{n} |V_i| L_{V_i}(\texttt{Loc}_c(T_i, \texttt{Alg}_a(\{T_j\}_{j=1}^{n}))$$

Federated Hyperparameter Tuning

Global setting:

Validation set size

#Clients

$$\min_{a \in \mathcal{A}} \sum_{i=1}^{n} |V_i| L_{V_i}\left(\texttt{Alg}_a(\{T_j\}_{j=1}^{n})\right)$$

Hyper-param

Validation loss

Training data

Personalized setting:

$$\min_{a=(b,c) \in \mathcal{A}} \sum_{i=1}^{n} |V_i| L_{V_i}\left(\texttt{Loc}_c(T_i, \texttt{Alg}_a(\{T_j\}_{j=1}^{n})\right)$$

Client hyper-param

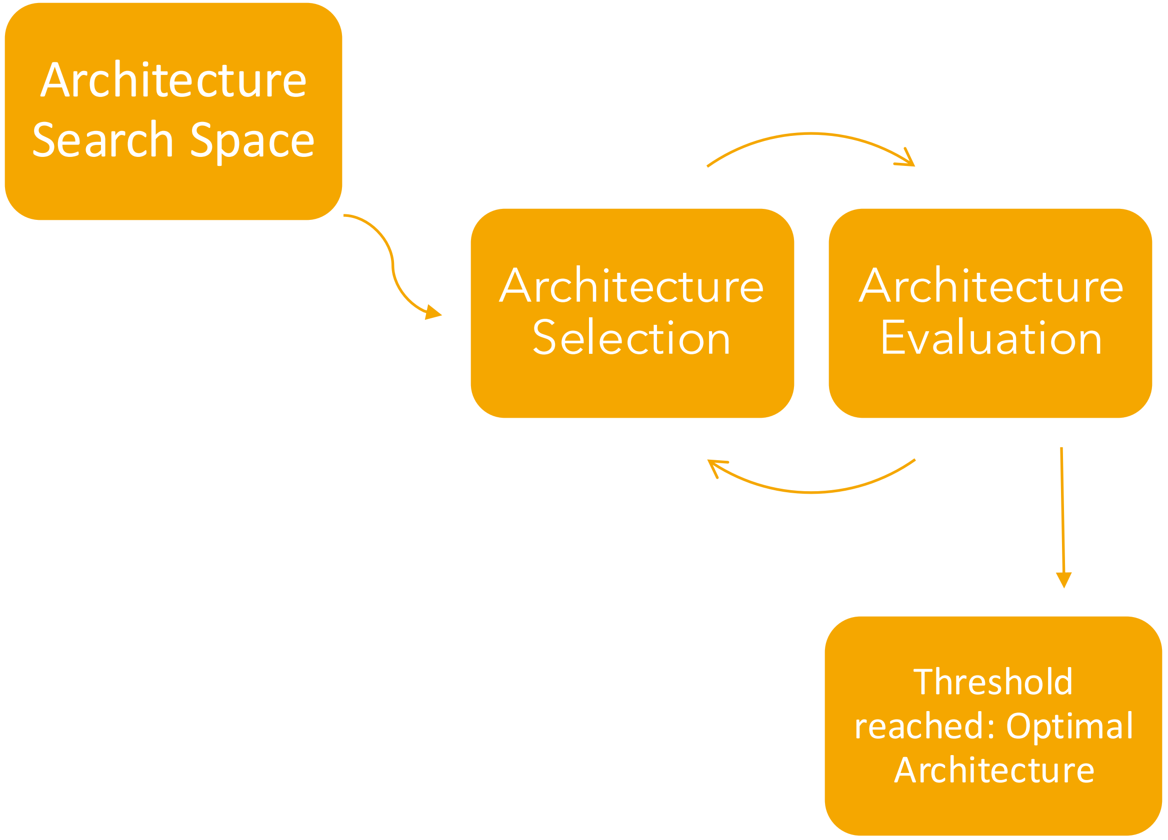
Local model

Server hyper-param

Background > NAS

Neural Architecture Search

- Goal:
  - Find the best network architecture for a problem
  - Automatically
- How: different strategies
  - Reinforcement learning
  - Evolutionary
  - Gradient
  - Weight Sharing

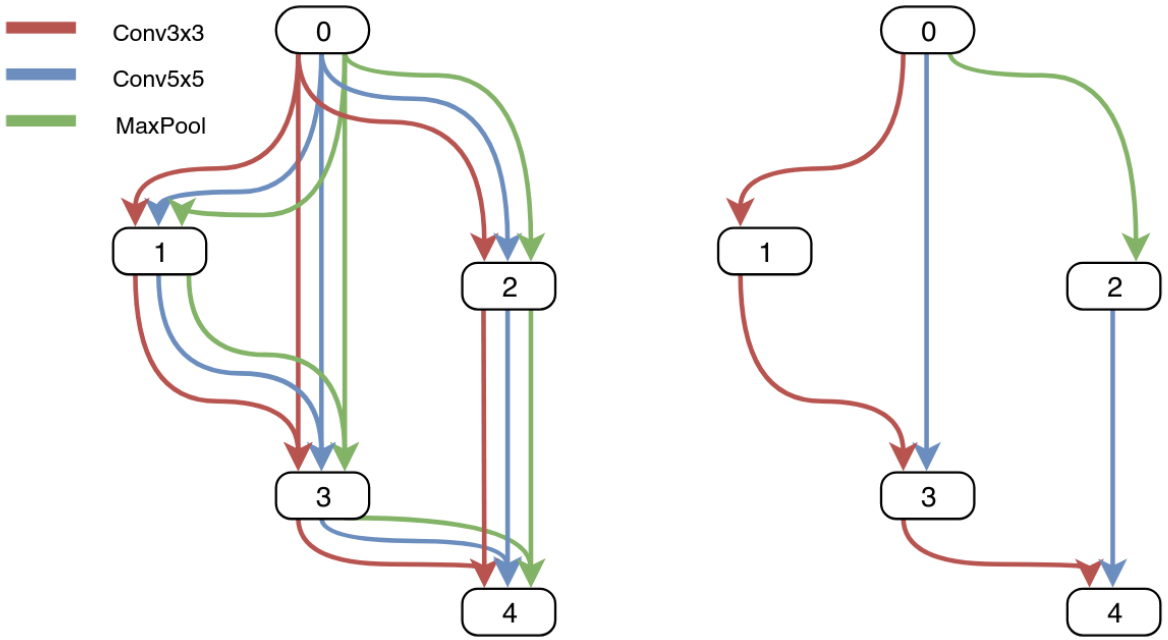


Background > Weight Sharing

Weight Sharing

- Used in NAS
  - Super architecture
  - Shares the weights with smaller sub-architectures
- **Goal**: reduce computation requirements – Train once, validate many
- In FedEx: Train once with selected HP

https://arxiv.org/pdf/1808.05377



One-Shot - Weight Sharing
Source: https://arxiv.org/pdf/1808.05377

Weight Sharing - Simplification

$$\min_{c \in \mathcal{C}} L_{\text{valid}}(\mathbf{w}, c) \quad \text{s.t.} \quad \mathbf{w} \in \arg\min_{\mathbf{u} \in \mathbb{R}^d} L_{\text{train}}(\mathbf{u}, c)$$

Bi-level: heavy

c can be trained

$$\min_{c \in \mathcal{C}, \mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}, c) \quad = \quad \min_{c \in \mathcal{C}, \mathbf{w} \in \mathbb{R}^d} L_{\text{train}}(\mathbf{w}, c) + L_{\text{valid}}(\mathbf{w}, c)$$

Single level: still heavy

Sample configuration from distribution

$$\min_{\theta \in \Theta, \mathbf{w} \in \mathbb{R}^d} \mathbb{E}_{c \sim \mathcal{D}_\theta} L(\mathbf{w}, c)$$

Stochastic relaxation: minimize the weighted average (Expectation) of the losses, for a sampled configuration

Background > SHA

Successive Halving Algorithm

- **Goal**: Evaluate HPs quickly
- **How**: Stop early, Eliminate
- **Disadvantage**: Might skip desirable HPs for the rest of the population

Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \ldots, \tau_R$

```
sample set of η^R hyperparameters H ~ D^[η^R]
initialize a model w_a ∈ R^d for each a ∈ H
for elimination round r ∈ [R] do
    for setting a = (b, c) ∈ H do
        for comm. round t = τ_{r-1} + 1, ..., τ_r do
            for client i = 1, ..., B do
                send w_a, c to client
                w_i ← Loc_c(T_ti, w_a)
                send w_i, L_{V_ti}(w_i) to server
            w_a ← Agg_b(w_a, {w_i}_{i=1}^B)
            s_a ← Σ_{i=1}^B |V_ti| L_{V_ti}(w_i) / Σ_{i=1}^B |V_ti|
    H ← {a ∈ H : s_a ≤ (1/η)-quantile({s_a : a ∈ H})}
```

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$

Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \ldots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$

initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$

for *elimination round* $r \in [R]$ **do**

  for *setting* $a = (b, c) \in H$ **do**

    for *comm. round* $t = \tau_{r-1} + 1, \ldots, \tau_r$ **do**

      for *client* $i = 1, \ldots, B$ **do**

        send $\mathbf{w}_a, c$ to client

        $\mathbf{w}_i \leftarrow \mathtt{Loc}_c(T_{ti}, \mathbf{w}_a)$

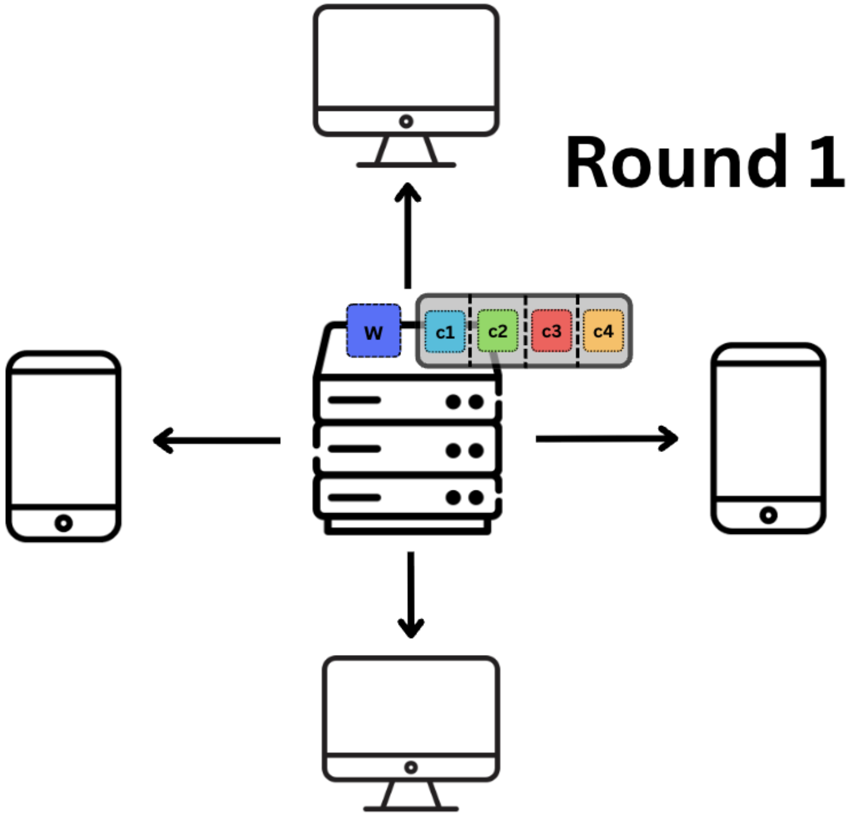
        send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server

      $\mathbf{w}_a \leftarrow \mathtt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$

      $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$

  $H \leftarrow \{a \in H : s_a \leq \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$



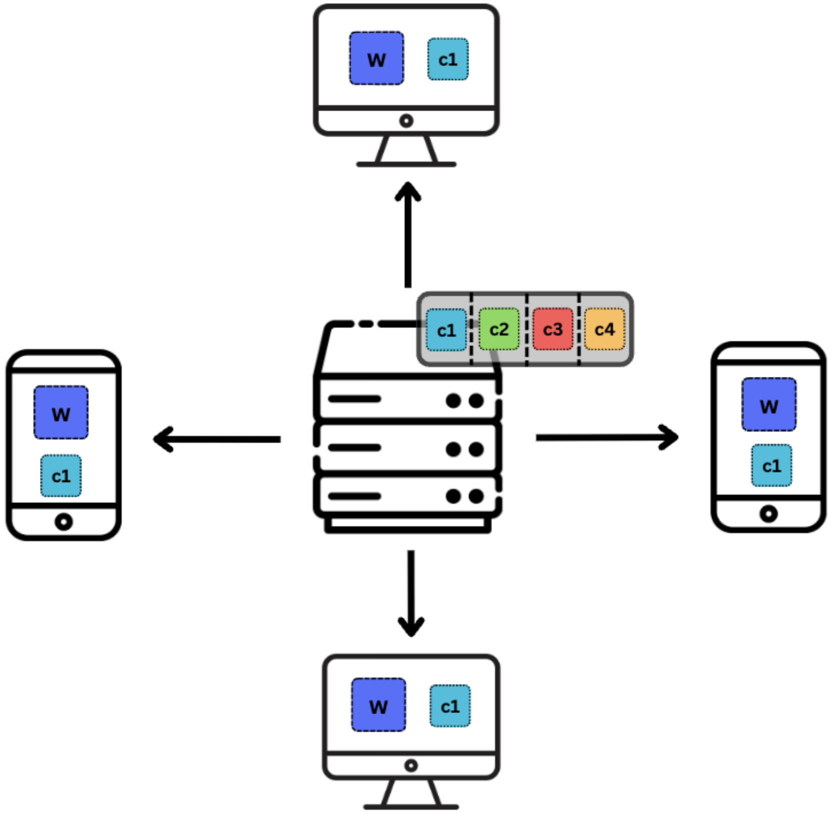
Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \dots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$
initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$
for *elimination round* $r \in [R]$ **do**
  for *setting* $a = (b, c) \in H$ **do**
    for *comm. round* $t = \tau_{r-1} + 1, \dots, \tau_r$ **do**
      for *client* $i = 1, \dots, B$ **do**
        send $\mathbf{w}_a, c$ to client
        $\mathbf{w}_i \leftarrow \mathtt{Loc}_c(T_{ti}, \mathbf{w}_a)$
        send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server
      $\mathbf{w}_a \leftarrow \mathtt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$
      $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$
  $H \leftarrow \{a \in H : s_a \le \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$



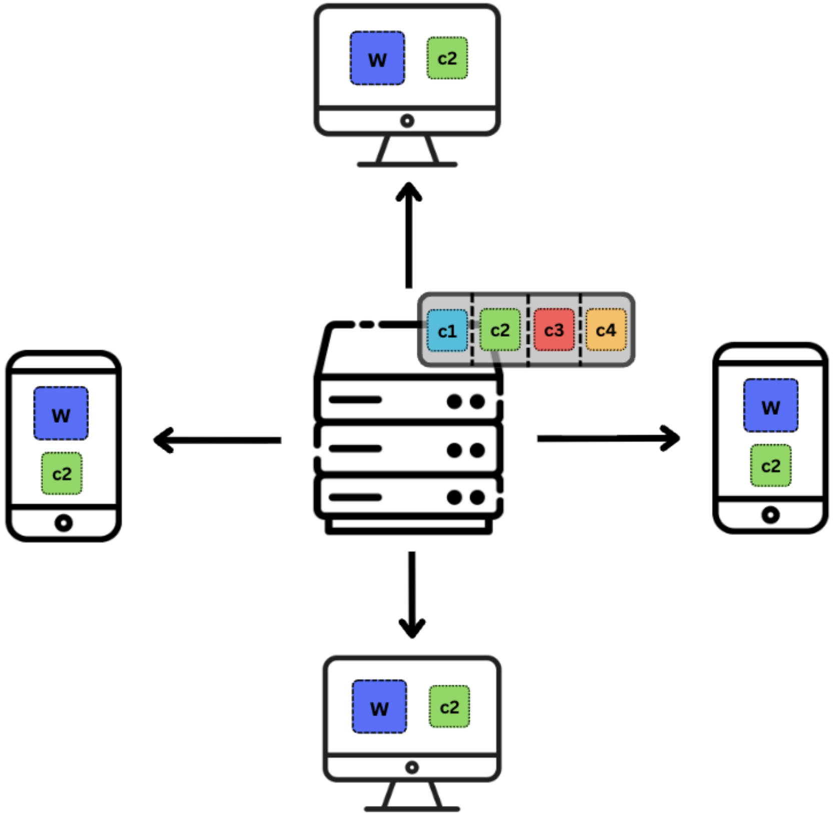
Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \dots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$
initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$
for *elimination round* $r \in [R]$ **do**
    for *setting* $a = (b, c) \in H$ **do**
        for *comm. round* $t = \tau_{r-1} + 1, \dots, \tau_r$ **do**
            for *client* $i = 1, \dots, B$ **do**
                send $\mathbf{w}_a, c$ to client
                $\mathbf{w}_i \leftarrow \mathtt{Loc}_c(T_{ti}, \mathbf{w}_a)$
                send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server
            $\mathbf{w}_a \leftarrow \mathtt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$
            $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$
    $H \leftarrow \{a \in H : s_a \le \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$



Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \ldots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$

initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$

for *elimination round* $r \in [R]$ **do**

  for *setting* $a = (b, c) \in H$ **do**

    for *comm. round* $t = \tau_{r-1} + 1, \ldots, \tau_r$ **do**

      for *client* $i = 1, \ldots, B$ **do**

        send $\mathbf{w}_a, c$ to client

        $\mathbf{w}_i \leftarrow \texttt{Loc}_c(T_{ti}, \mathbf{w}_a)$

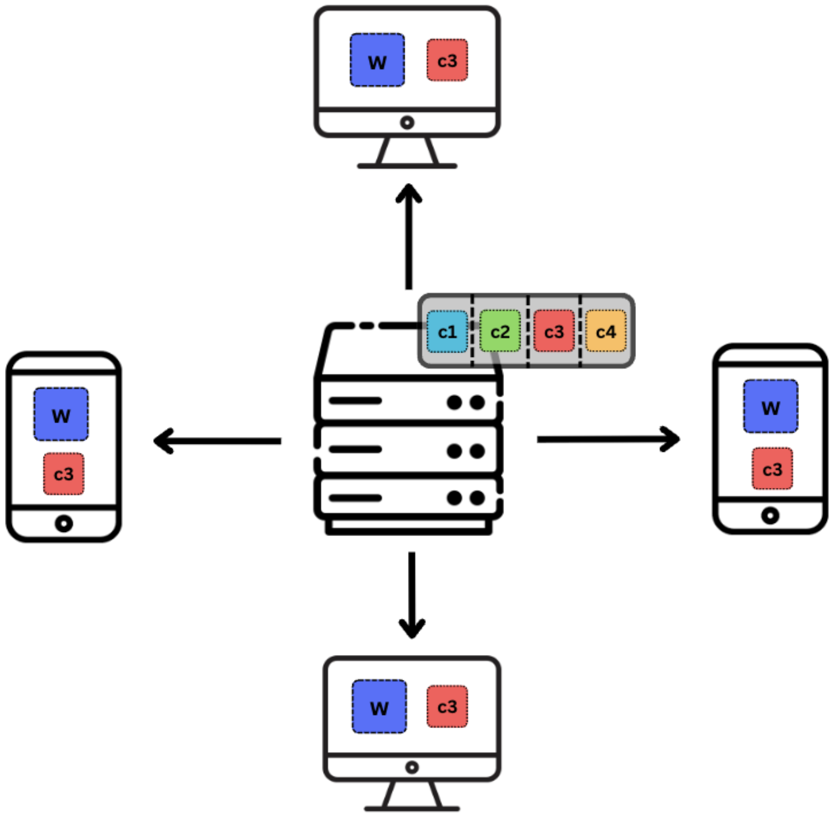
        send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server

      $\mathbf{w}_a \leftarrow \texttt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$

      $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$

  $H \leftarrow \{a \in H : s_a \leq \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$



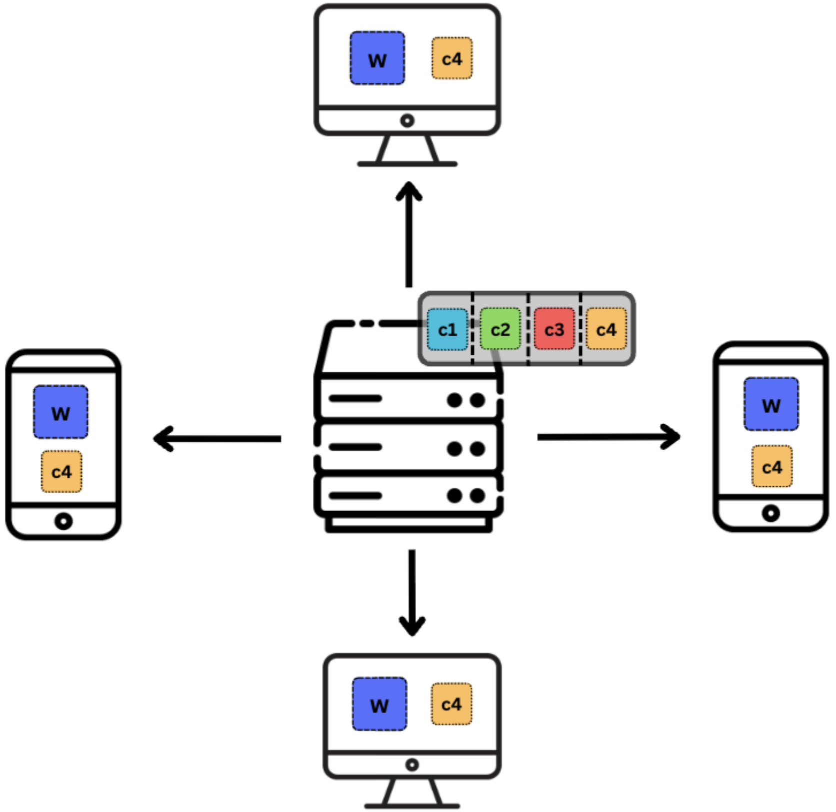
Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \dots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$
initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$
for *elimination round* $r \in [R]$ **do**
  for *setting* $a = (b, c) \in H$ **do**
    for *comm. round* $t = \tau_{r-1} + 1, \dots, \tau_r$ **do**
      for *client* $i = 1, \dots, B$ **do**
        send $\mathbf{w}_a, c$ to client
        $\mathbf{w}_i \leftarrow \mathtt{Loc}_c(T_{ti}, \mathbf{w}_a)$
        send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server
      $\mathbf{w}_a \leftarrow \mathtt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$
      $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$
  $H \leftarrow \{a \in H : s_a \leq \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$



Successive Halving Algorithm

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \dots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$

initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$

for *elimination round* $r \in [R]$ **do**

  for *setting* $a = (b, c) \in H$ **do**

    for *comm. round* $t = \tau_{r-1} + 1, \dots, \tau_r$ **do**

      for *client* $i = 1, \dots, B$ **do**

        send $\mathbf{w}_a, c$ to client

        $\mathbf{w}_i \leftarrow \texttt{Loc}_c(T_{ti}, \mathbf{w}_a)$

        send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server

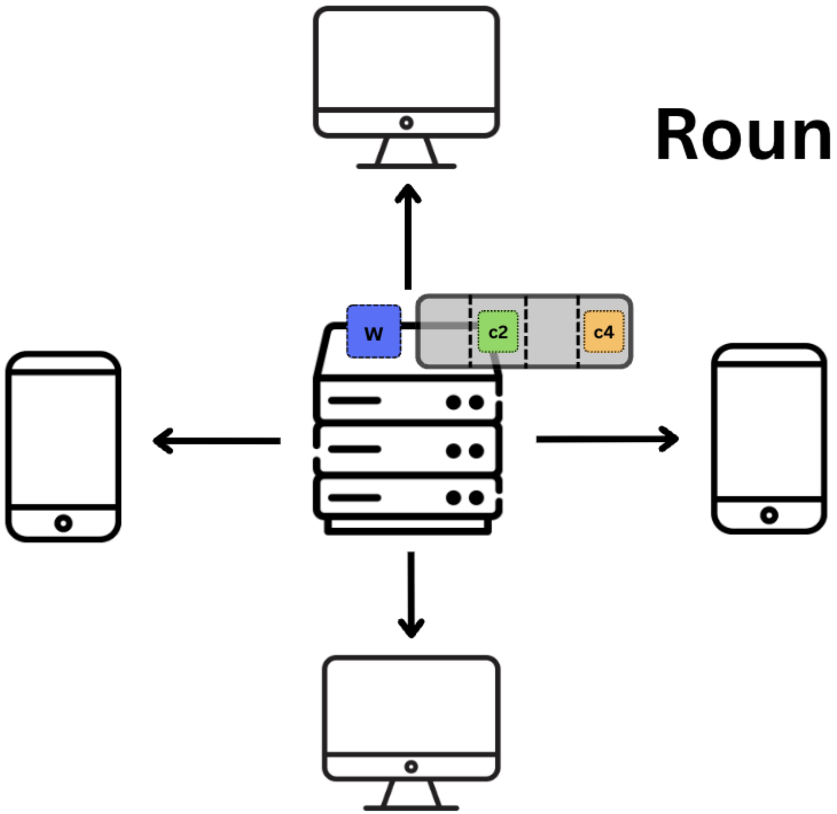
      $\mathbf{w}_a \leftarrow \texttt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$

      $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$

  $H \leftarrow \{a \in H : s_a \leq \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$

Round 2



FedEx and NAS

From Weight Sharing to Fedex

$$\min_{\theta\in\Theta,\mathbf{w}\in\mathbb{R}^d} \mathbb{E}_{c\sim\mathcal{D}_\theta} L(\mathbf{w}, c)$$

$$\min_{a=(b,c)\in\mathcal{A}} \sum_{i=1}^{n} |V_i| L_{V_i}(\texttt{Loc}_c(T_i, \texttt{Alg}_a(\{T_j\}_{j=1}^n)))$$

$$\min_{c\in\mathcal{C},\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} |V_i| L_{V_i}(\texttt{Loc}_c(T_i, \mathbf{w}))$$

$$\min_{\theta\in\Theta,\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} |V_i| \mathbb{E}_{c\in\mathcal{D}_\theta} L_{V_i}(\texttt{Loc}_c(T_i, \mathbf{w}))$$

Goal: Min. sum of the expected validation losses across all clients, weighted by |*Vi*|

Fedex: Pseudocode

Algorithm 2: `FedEx`

Input: configurations $c_1, \dots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \dots, \tau$ **do**

    for *client* $i = 1, \dots, B$ **do**

        send $\mathbf{w}_t, \theta_t$ to client

        sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

        $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

        send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

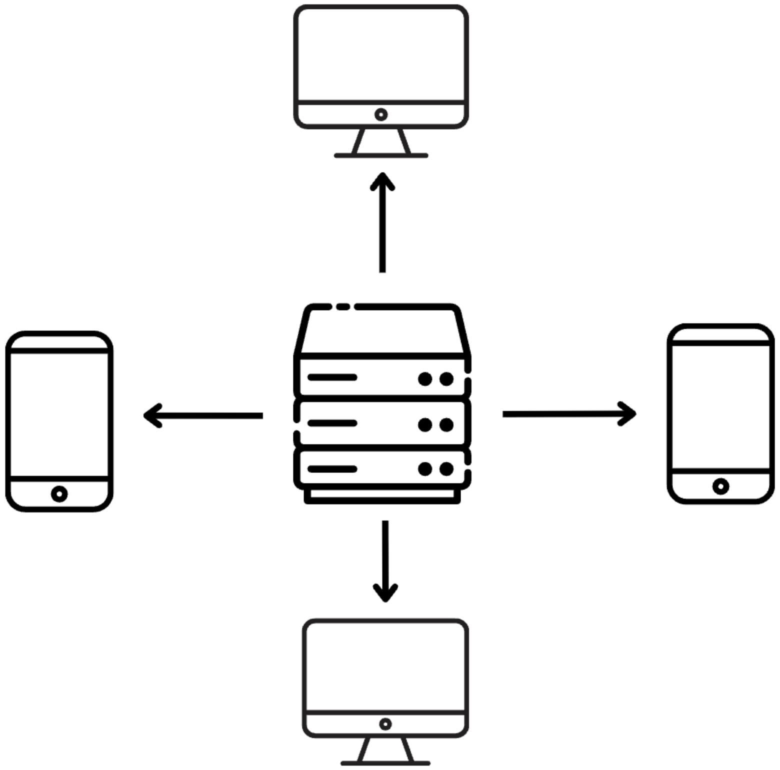
    $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

    $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \ \forall j$

    $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

    $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: `FedEx`

Input: configurations $c_1, \ldots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \ldots, \tau$ **do**

  for *client* $i = 1, \ldots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

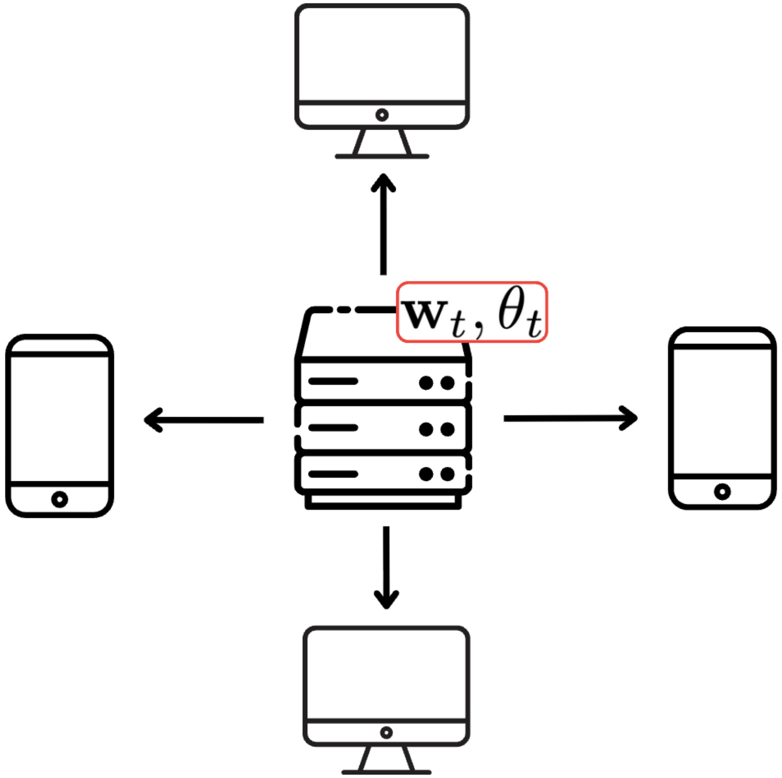
  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \ \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: FedEx

Input: configurations $c_1, \dots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \dots, \tau$ **do**

  for *client* $i = 1, \dots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

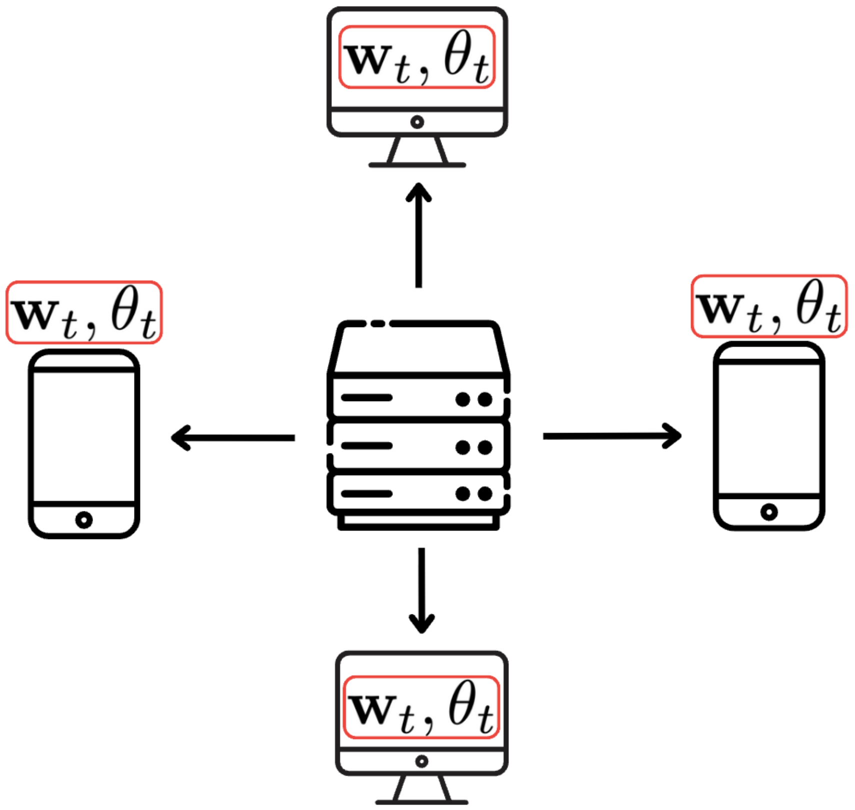
  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: `FedEx`

Input: configurations $c_1, \dots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \dots, \tau$ **do**

  for *client* $i = 1, \dots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

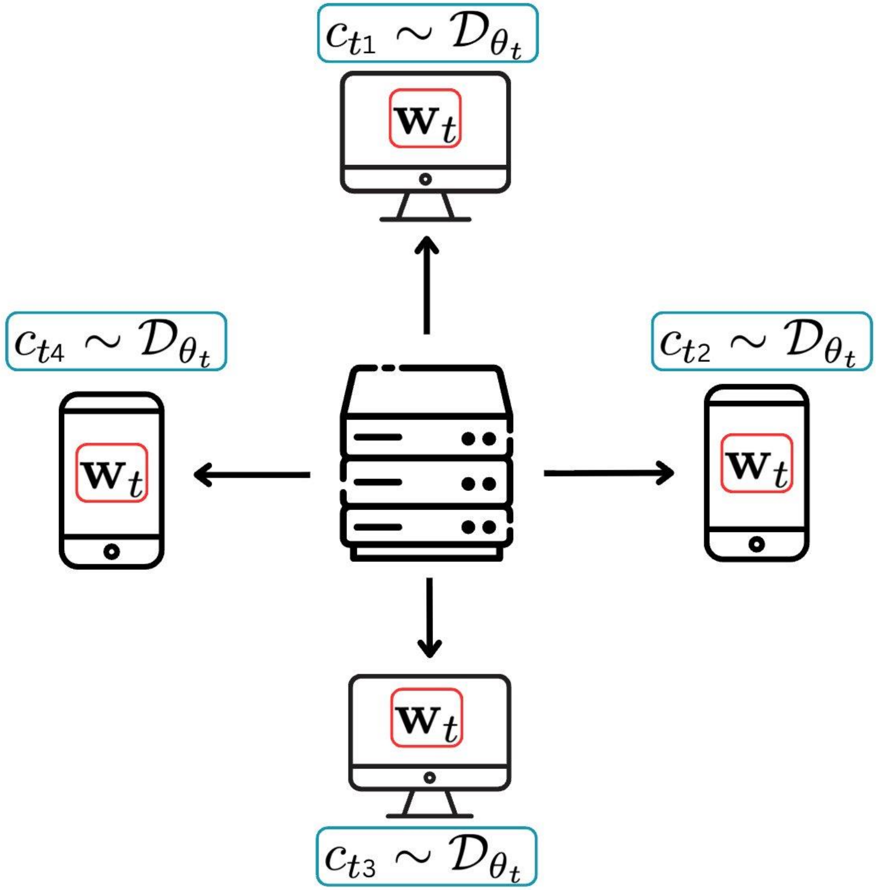
  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: FedEx

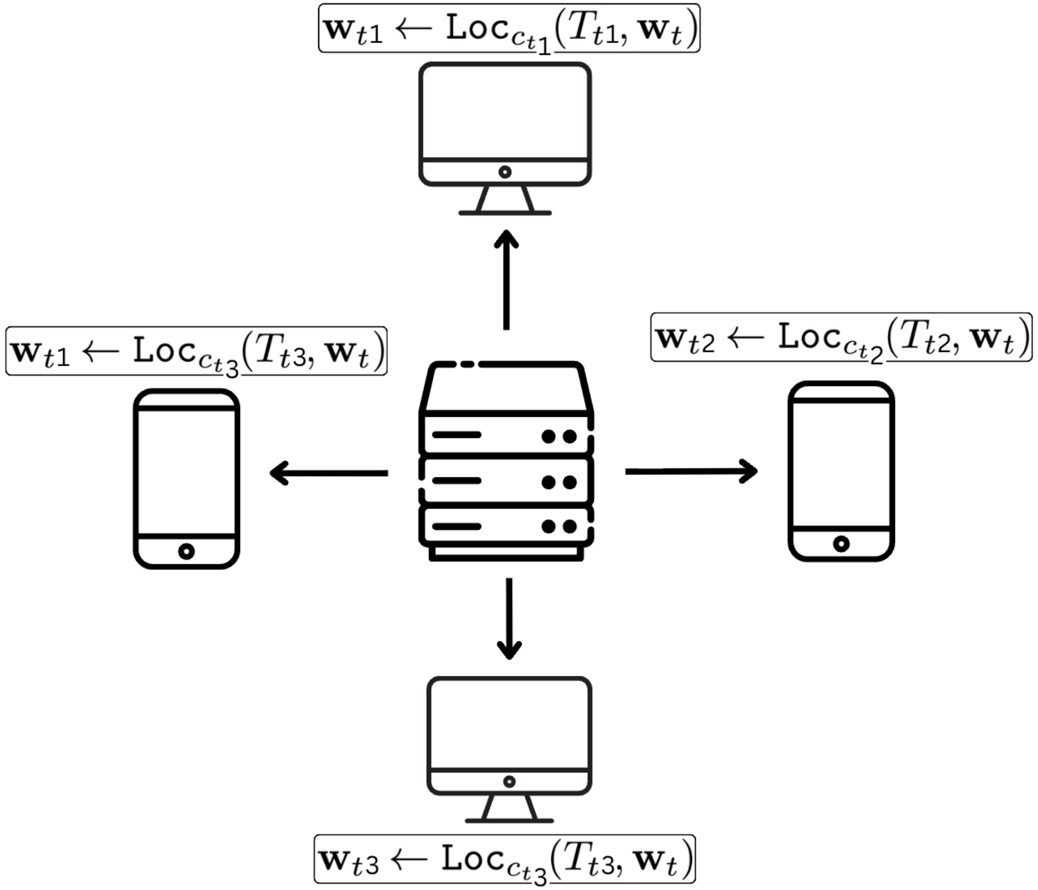
Input: configurations $c_1, \ldots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \ldots, \tau$ **do**

- **for** *client* $i = 1, \ldots, B$ **do**
  - send $\mathbf{w}_t, \theta_t$ to client
  - sample $c_{ti} \sim \mathcal{D}_{\theta_t}$
  - $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$
  - send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server
- $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$
- $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$
- $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$
- $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: FedEx

Input: configurations $c_1, \ldots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \ldots, \tau$ **do**

  for *client* $i = 1, \ldots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

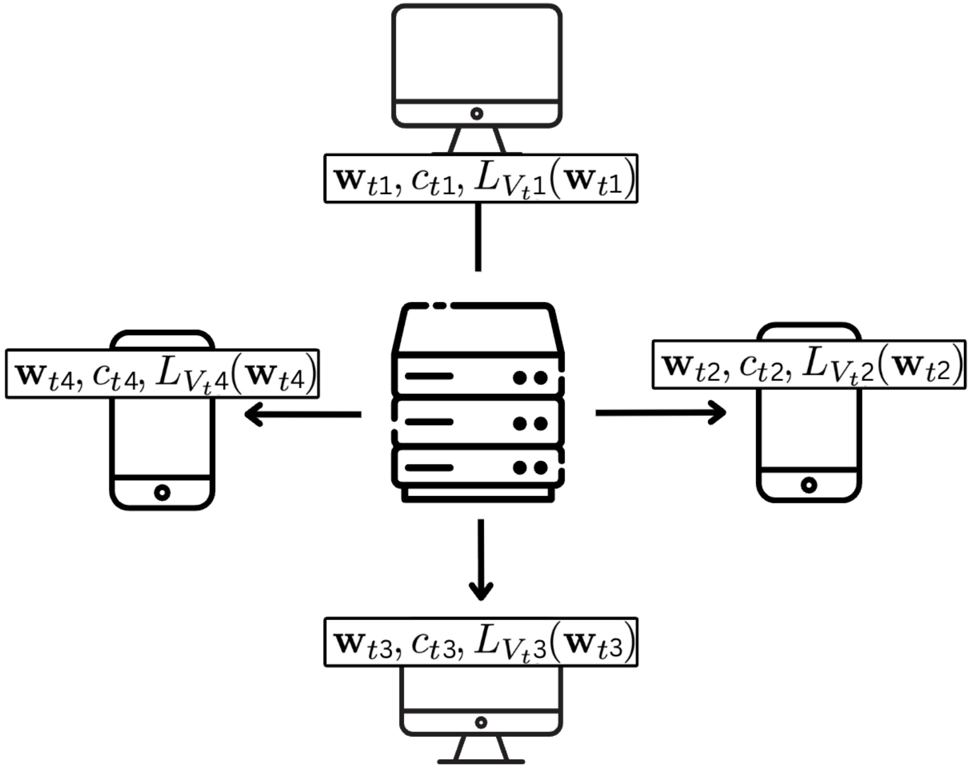
  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ



Fedex: Pseudocode

Algorithm 2: FedEx

Input: configurations $c_1, \dots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \dots, \tau$ **do**

  for *client* $i = 1, \dots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

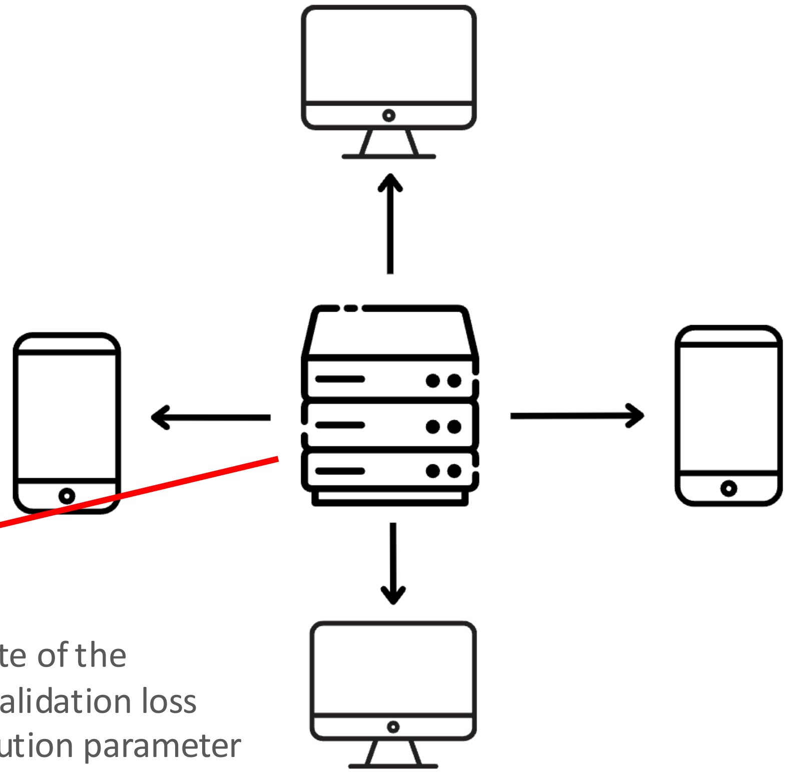
  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ

Calculate unbiased estimate of the gradient of the expected validation loss with respect to the distribution parameter



Questions from Discord

Abdechakour M. Yesterday at 2:53 PM

Since FedEx tunes hyperparameters during training instead of running multiple separate experiments (for example, in traditional hyperparameter tuning you might run 10 full experiments for 10 combinations, which means 10× the computation, time, and communication), how does FedEx compare in terms of extra computation or communication overhead with FedAvg (without hyperparameter tuning)? (edited)

Algorithm 2: `FedEx`

Input: configurations $c_1, \dots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$

for *comm. round* $t = 1, \dots, \tau$ **do**

  for *client* $i = 1, \dots, B$ **do**

    send $\mathbf{w}_t, \theta_t$ to client

    sample $c_{ti} \sim \mathcal{D}_{\theta_t}$

    $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$

    send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server

  $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$

  $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \; \forall j$

  $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$

  $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ

Algorithm 1 FedOpt

Input: x_0, ClientOpt, ServerOpt

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

  Sample a subset $\mathcal{S}$ of clients

  $x_{i,0}^t = x_t$

  for each client $i \in \mathcal{S}$ **in parallel do**

    for $k = 0, \cdots, K-1$ **do**

      Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(x_{i,k}^t)$

      $x_{i,k+1}^t = \text{ClientOpt}(x_{i,k}^t, g_{i,k}^t, \eta_l, t)$

    $\Delta_i^t = x_{i,K}^t - x_t$

  $\Delta_t = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Delta_i^t$

  $x_{t+1} = \text{ServerOpt}(x_t, -\Delta_t, \eta, t)$

How it solves the problem

Algorithm 1: Successive halving algorithm (SHA) applied to personalized FL. For the non-personalized objective (1), replace $L_{V_{ti}}(\mathbf{w}_i)$ by $L_{V_{ti}}(\mathbf{w}_a)$. For random search (RS) with N samples, set $\eta = N$ and $R = 1$.

Input: distribution $\mathcal{D}$ over hyperparameters $\mathcal{A}$, elimination rate $\eta \in \mathbb{N}$, elimination rounds $\tau_0 = 0, \tau_1, \ldots, \tau_R$

sample set of η^R hyperparameters $H \sim \mathcal{D}^{[\eta^R]}$
initialize a model $\mathbf{w}_a \in \mathbb{R}^d$ for each $a \in H$
for *elimination round* $r \in [R]$ **do**
    for *setting* $a = (b, c) \in H$ **do**
        for *comm. round* $t = \tau_{r-1} + 1, \ldots, \tau_r$ **do**
            for *client* $i = 1, \ldots, B$ **do**
                send $\mathbf{w}_a, c$ to client
                $\mathbf{w}_i \leftarrow \texttt{Loc}_c(T_{ti}, \mathbf{w}_a)$
                send $\mathbf{w}_i, L_{V_{ti}}(\mathbf{w}_i)$ to server
            $\mathbf{w}_a \leftarrow \texttt{Agg}_b(\mathbf{w}_a, \{\mathbf{w}_i\}_{i=1}^B)$
            $s_a \leftarrow \sum_{i=1}^B |V_{ti}| L_{V_{ti}}(\mathbf{w}_i) / \sum_{i=1}^B |V_{ti}|$
    $H \leftarrow \{a \in H : s_a \leq \frac{1}{\eta}\text{-quantile}(\{s_a : a \in H\})\}$

Output: remaining $a \in H$ and associated model $\mathbf{w}_a$

Algorithm 2: `FedEx`

Input: configurations $c_1, \ldots, c_k \in \mathcal{C}$, setting b for $\texttt{Agg}_b$, schemes for setting step-size η_t and baseline λ_t, total number of steps $\tau \geq 1$

initialize $\theta_1 = \mathbf{1}_k / k$ and shared weights $\mathbf{w}_1 \in \mathbb{R}^d$
for *comm. round* $t = 1, \ldots, \tau$ **do**
    for *client* $i = 1, \ldots, B$ **do**
        send $\mathbf{w}_t, \theta_t$ to client
        sample $c_{ti} \sim \mathcal{D}_{\theta_t}$
        $\mathbf{w}_{ti} \leftarrow \texttt{Loc}_{c_{ti}}(T_{ti}, \mathbf{w}_t)$
        send $\mathbf{w}_{ti}, c_{ti}, L_{V_{ti}}(\mathbf{w}_{ti})$ to server
    $\mathbf{w}_{t+1} \leftarrow \texttt{Agg}_b(\mathbf{w}, \{\mathbf{w}_{ti}\}_{i=1}^B)$
    $\tilde{\nabla}_j \leftarrow \frac{\sum_{i=1}^B |V_{ti}|(L_{V_{ti}}(\mathbf{w}_{ti}) - \lambda_t) 1_{c_{ti}=c_j}}{\theta_{t[j]} \sum_{i=1}^B |V_{ti}|} \ \forall j$
    $\theta_{t+1} \leftarrow \theta_t \odot \exp(-\eta_t \tilde{\nabla})$
    $\theta_{t+1} \leftarrow \theta_{t+1} / \|\theta_{t+1}\|_1$

Output: model $\mathbf{w}$, hyperparameter distribution θ

Solution to Problem 1

- Recall: 1 - Federated validation data

- Use validation information on every round
  - ⇨ Over time, takes all clients into account
  - ⇨ Prevents early exclusion of HP due to noise

Solution to Problem 2

- Recall: 2 - Extreme resource limitations

- Combine model training and HP search
  - ⇨ Less computation
  - ⇨ Less communication

Solution to Problem 3

- Recall: 3 - Evaluating personalization

- Integrate personalization as part of training

➡ Estimate personalized perf while training

➡ Save computation for evaluation

Wrapping

Wrapping

- FedEx only for Client
- FedEx has HPs
- Server has HPs

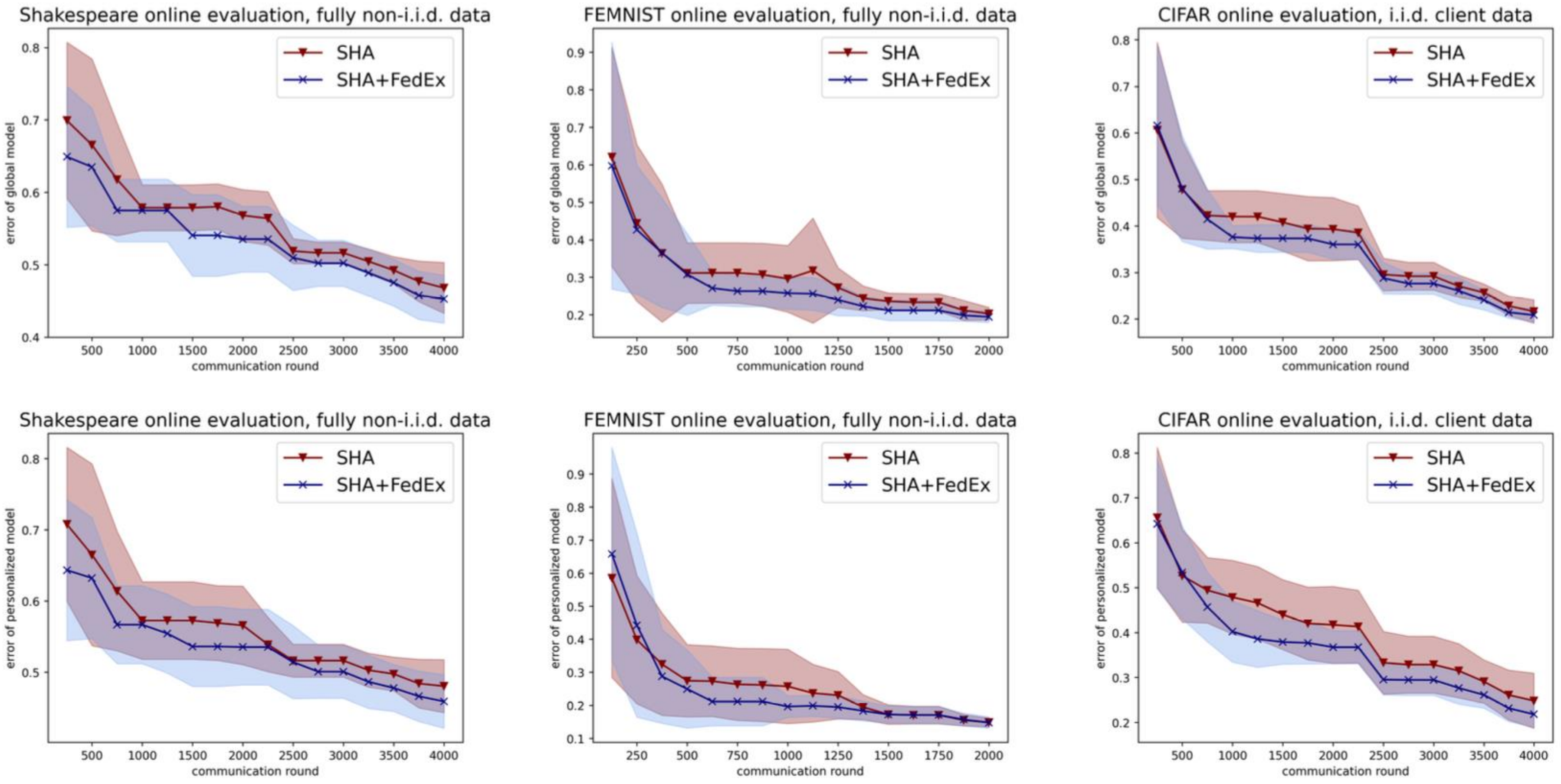
Wrapping in RS or SHA

Empirical Results

Experiment Settings

- Datasets:
  - CIFAR-10 - i.i.d only
  - FEMNIST - i.i.d and non-i.i.d
  - Shakespear - i.i.d and non-i.i.d
- Hyperparameters:
  - Server-side: LR schedule and momentum
  - Client-side: LR, momentum, weight-decay, #local epochs, batch-size, and dropout
- Targets:
  - Global
  - Personalized

Comparison: SHA vs. FedEx

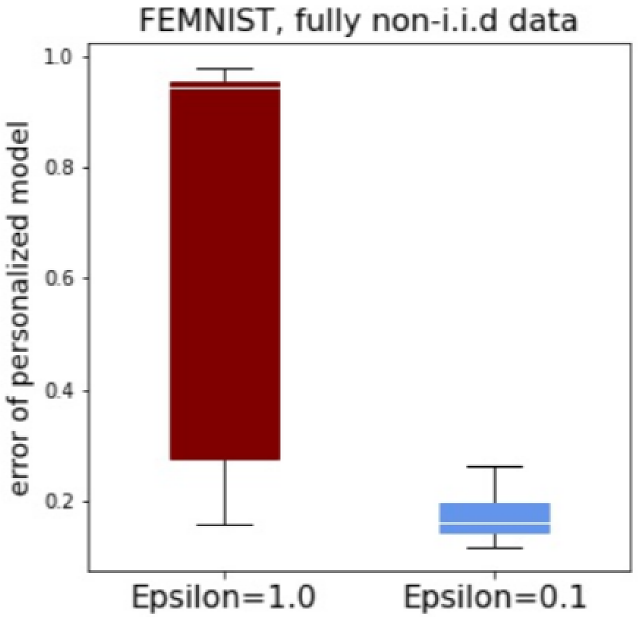


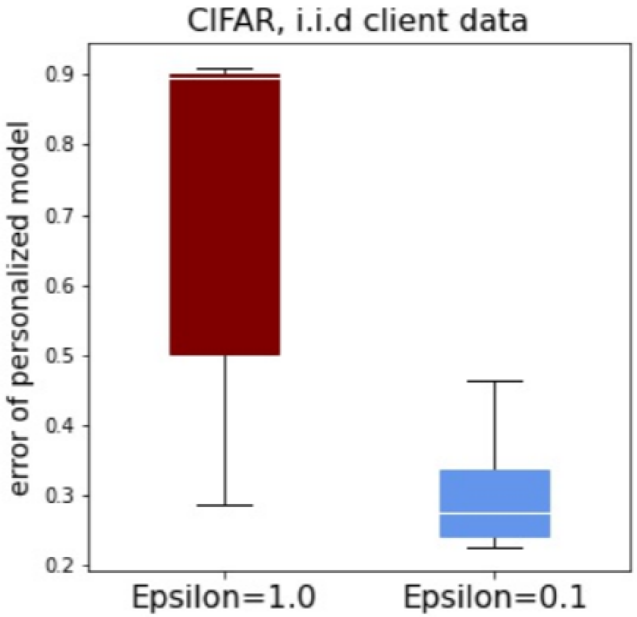
Comparison: Wrapper Method & Target Model

Wrapper method	Target model	Tuning method	Shakespeare i.i.d.	Shakespeare non-i.i.d.	FEMNIST i.i.d.	FEMNIST non-i.i.d.	CIFAR-10 i.i.d.
Random Search (RS)	global	RS (server & client)	60.32 ± 10.03	64.36 ± 14.19	22.81 ± 4.56	22.98 ± 3.41	30.46 ± 9.44
		+ FedEx (client)	53.94 ± 9.13	57.70 ± 17.57	20.96 ± 4.77	22.30 ± 3.66	34.83 ± 14.74
	personalized	RS (server & client)	61.10 ± 9.32	61.71 ± 9.08	17.45 ± 2.82	17.77 ± 2.63	34.89 ± 10.56
		+ FedEx (client)	54.90 ± 9.97	56.48 ± 13.60	16.31 ± 3.77	15.93 ± 3.06	39.13 ± 15.13
Successive Halving (SHA)	global	SHA (server & client)	47.38 ± 3.40	46.79 ± 3.51	18.64 ± 1.68	20.30 ± 1.66	21.62 ± 2.51
		+ FedEx (client)	$\mathbf{44.52} \pm 1.68$	$\mathbf{45.24} \pm 3.31$	19.22 ± 2.05	19.43 ± 1.45	$\mathbf{20.82} \pm 1.37$
	personalized	SHA (server & client)	46.77 ± 3.61	48.04 ± 3.72	$\mathbf{14.79} \pm 1.55$	14.78 ± 1.31	24.81 ± 6.13
		+ FedEx (client)	46.08 ± 2.57	45.89 ± 3.76	14.97 ± 1.31	$\mathbf{14.76} \pm 1.70$	21.77 ± 2.83

Questions from Discord

yazid Yesterday at 9:01 PM
The authors briefly mentioned "local perturbation" in the paper, but it seems to have an important impact on the model. Do we know how sensitive is FedEx to this choice? In other word, is it an important hyper-parameter that needs finetuning?





Conclusion

Conclusion

Advantages

- Reuse training results to find HPs
- Save communication & computation for HPs search
- No skipping
- Can use many Fed Algorithms

Limitations

- Needs an algorithm that can be decomposed in Local training and Aggregation phases
- Unknown privacy risks
- Depends on wrapper

Questions from Discord

> **Jiahao (Kevin) Cheng** Yesterday at 6:10 PM
> Since FedEx was mainly designed and evaluated in the cross-device federated learning setting, how would it perform for hyperparameter optimization in a cross-silo environment (a smaller number but more reliable clients with high capacity)?

Cross-Silo: Privacy needed - FedEx: no guarantee

- More compute/network: training will be faster
- Less clients: less HPs to test at once

Quiz Questions

Question 1

What are the three issues in Federated Learning that FedEx is trying to address?

Question 1

What are the three issues in Federated Learning that FedEx is trying to address?

1. Federated validation data: noisy and partial
2. Extreme resource limitations: due to client availability or privacy measures
3. Evaluating personalization requires multiple exchanges and rounds of training

Question 2

How are the three problems addressed by FedEx?

Question 2

How are the three problems addressed by FedEx?

1. Do not stop early: each round helps to adjust the HPs
2. Train Model and Optimize HPs at once (weight-sharing)
3. Use local training as personalization evaluation

Question 3

Why is FedEx wrapped into another HP search algorithm?

Question 3

Why is FedEx wrapped into another HP search algorithm?

- FedEx focusses on the Local training and validation, and personalization. Since it doesn't account for its own parameters, or the server's, it needs to be wrapped.

Question 4

How is the distribution of training hyperparameters updated?

Question 4

How is the distribution of training hyperparameters updated?

- The server uses the client's validation losses and the hyperparameters they used to update the distribution.