Submodel Partitioning in Hierarchical Federated Learning: Algorithm Design and Convergence Analysis

Wenzhi Fang, Dong-Jun Han, and Christopher G. Brinton
Elmore Family School of Electrical and Computer Engineering, Purdue University
Email: {fang375, han762, cgb}@purdue.edu

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10622512

***Abstract*—Hierarchical federated learning (HFL) has demonstrated promising scalability advantages over the traditional "star-topology" architecture-based federated learning (FL). However, HFL still imposes significant computation, communication, and storage burdens on the edge, especially when training a large-scale model over resource-constrained Internet of Things (IoT) devices. In this paper, we propose *hierarchical independent submodel training* (HIST), a new FL methodology that aims to address these issues in hierarchical settings. The key idea behind HIST is a hierarchical version of model partitioning, where we partition the global model into disjoint submodels in each round, and distribute them across different cells, so that each cell is responsible for training only one partition of the full model. This enables each client to save computation/storage costs while alleviating the communication loads throughout the hierarchy. We characterize the convergence behavior of HIST for non-convex loss functions under mild assumptions, showing the impact of several attributes (e.g., number of cells, local and global aggregation frequency) on the performance-efficiency tradeoff. Finally, through numerical experiments, we verify that HIST is able to save communication costs by a wide margin while achieving the same target testing accuracy.**

I. Introduction

The past decade has witnessed a huge breakthrough in various machine learning (ML) applications, from computer vision to natural language processing. As training data for these tasks are often collected by geographically separated clients, developing efficient distributed training strategies has become increasingly important [1]–[3]. In this context, federated learning (FL) is receiving significant attention nowadays as it enables clients to collaboratively train a global model without any raw data exchange [4].

In the traditional cloud-based FL [5], all clients in the system directly communicate with a central cloud server for model aggregations, resulting in communication scalability issues as the size of the network grows. Hierarchical federated learning (HFL) has been proposed as a solution [6]–[8], taking advantage of the fact that clusters of clients (e.g., cells) may be served by intermediate edge servers. The introduction of edge servers in HFL reduces communication and scheduling complexity, as the cloud server now only needs to communicate with the edge servers.

This project was supported in part by NSF under grants CNS-2146171 CPS-2313109, and by ONR under grant N000142212305

However, as the size of the model increases, the HFL training process still suffers from scalability issues. These manifest in several dimensions: (i) computation/storage costs at individual clients, (ii) communication burden between clients and the edge server, and (iii) communication load between edge servers and the cloud server. These are fundamental bottlenecks for the practical deployment of HFL, especially when resource-constrained mobile and IoT devices aim to collaboratively train a large-scale neural network model.

In this paper, we propose HIST, a new FL methodology that integrates *independent submodel training (IST)* in hierarchical networks to address the aforementioned challenges. The core idea of HIST is to partition the global model into disjoint submodels in each training round and distribute them across different cells, so that devices in distinct cells are responsible for training different partitions of the full model. Such a submodel partitioning effectively reduces computation and storage loads at local clients, and also alleviates communication burden on both the links between clients and the edge server and between edge servers and the cloud server. The main contributions of this paper are summarized as follows:

- We propose HIST, a hierarchical independent submodel training methodology that successfully reduces computation, communication, and storage costs during the training process of HFL.
- We analytically characterize the convergence bound of HIST for non-convex loss functions, under milder assumptions than those found in the literature. Based on the result, we analyze the performance-efficiency tradeoff induced by HIST, and provide guidelines on setting the key system parameters of HFL.
- In simulations, we evaluate the effectiveness of the proposed algorithm by training a neural network in two different data distribution setups for hierarchical networks. We show that our proposed HIST achieves significant resource savings for a target trained model accuracy compared with the standard hierarchical FedAvg [8].

Related Works: The exploration of submodel training commenced with the pioneering work [9], where the authors introduced the concept of IST for fully connected neural networks and provided theoretical analysis under centralized

settings. Subsequently, submodel training was extended to graph neural networks [10] and ResNets [11]. Due to its effectiveness in addressing communication, computation, and storage challenges, the concept of IST was subsequently considered in distributed scenarios [12], where the authors empirically show the effectiveness of submodel training in FL. Additionally, several studies also characterized the convergence behavior of distributed submodel training [13]–[15]. However, the aforementioned works either rely on restrictive assumptions [13], [14] or narrow the focus to quadratic models [15]. More importantly, existing works focus on cloud-based FL with a single server, and thus do not provide insights into the hierarchical case. To the best of our knowledge, HIST is the earliest work to integrate IST with HFL and provide theoretical analysis as well as experimental results.

II. System Model and Formulation

We consider a HFL system that consists of a single cloud server, N edge servers indexed by $\{1, 2, \ldots, N\}$, and $\sum_{j=1}^{N} n_j$ clients, where n_j is the number of clients located in the j-th cell. Edge server j is responsible for coordinating the training of n_j clients in cell j. The global server is in charge of model aggregation over N geographically distributed edge servers. Given the loss function $l(\boldsymbol{x}, \xi_i)$ which measures the loss on sample ξ_i with model $\boldsymbol{x} \in \mathbb{R}^d$, the training objective of this HFL system can be formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} f(\boldsymbol{x}) &:= \frac{1}{N}\sum_{j=1}^{N} f_j(\boldsymbol{x}) && \text{Global loss} \\ f_j(\boldsymbol{x}) &:= \frac{1}{n_j}\sum_{i\in\mathcal{C}_j} F_i(\boldsymbol{x}) && \text{Cell loss} \\ F_i(\boldsymbol{x}) &:= \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left[l(\boldsymbol{x}, \xi_i)\right] && \text{Client loss} \end{aligned} \tag{1}$$

where $f : \mathbb{R}^d \leftarrow \mathbb{R}$, $f_j : \mathbb{R}^d \leftarrow \mathbb{R}$, and $F_i : \mathbb{R}^d \leftarrow \mathbb{R}$ represent the global, cell, and client losses, respectively. $\mathcal{C}_j$ denotes the client set of cell j, and $\mathcal{D}_i$ denotes the local data distribution of client i. In this work, we mainly consider the non-i.i.d. scenario, where data distribution is heterogeneous among different clients, i.e., $\mathcal{D}_j \neq \mathcal{D}_{j'}, \forall j' \neq j$.

In conventional HFL, all clients in the system are required to train the full model. To support such model training, each client needs to be equipped with enough computation, storage, and communication resources. However, it is unaffordable for resource-constrained clients to handle the training of large-scale models. This motivates us to develop a more efficient training framework for HFL, which will be discussed in the next section.

III. Hierarchical Independent Submodel Training Algorithm

In this section, we introduce our HIST algorithm tailored to HFL and analyze the communication complexity to demonstrate its efficiency.

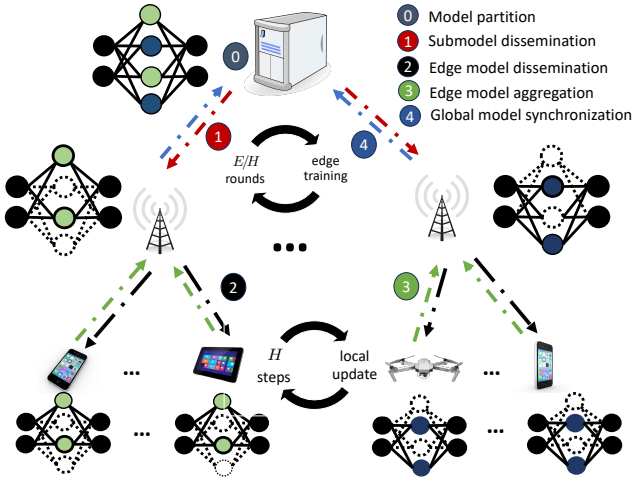


Fig. 1: Overview of proposed hierarchical independent submodel training (HIST). Each cell is responsible for training only a specific partition of the full model, where the submodel partitioning changes over global rounds.

A. Algorithm Description

Inspired by IST [9], we develop a hierarchical federated submodel training algorithm termed HIST, by incorporating hierarchical FedAvg and submodel partitioning. The overview of HIST is presented in Fig. 1 and Algorithm 1. The global cloud server periodically aggregates the models from the edge servers, while each edge server periodically aggregates the models from the clients within the corresponding cell. The key difference with the conventional HFL is that, clients do not need to store, update, and exchange the full model in HIST.

Specifically, in the beginning of t/E-th global round where t represents the iteration number of clients and E denotes the period of the global aggregation, the cloud server initiates the training process by partitioning the current global model $\bar{\boldsymbol{x}}^t$ into N disjoint submodels:

$$\bar{\boldsymbol{x}}_j^t = \boldsymbol{p}_j^t \odot \bar{\boldsymbol{x}}^t, \forall j \in \{1, 2, \ldots, N\}, \tag{2}$$

where $\odot$ denotes a Hadamard Product operation, $\bar{\boldsymbol{x}}_j^t$ represents the j-th submodel for cell j, and $\boldsymbol{p}_j^t$ is a mask that has either 0 or 1 in its element and satisfying

$$\boldsymbol{p}_j^t \odot \boldsymbol{p}_{j'}^t = \boldsymbol{0}, \forall j' \neq j, \text{ and } \sum_{i=1}^{N} \boldsymbol{p}_j^t = \boldsymbol{1}. \tag{3}$$

These submodels are then distributed to the edge servers, and each edge server subsequently disseminates submodel $\bar{\boldsymbol{x}}_j^t$ to the clients within its coverage, such that $\boldsymbol{x}_i^t = \bar{\boldsymbol{x}}_j^t, \forall i \in \mathcal{C}_j$. Once the clients receive the most recent model from the server, they start training with their local datasets. The essential steps performed by clients, edge servers, and the global server in our proposed algorithm are outlined as follows:

Clients: Clients first compute stochastic gradients with respect to their corresponding submodels, and then update the local models for H steps via the following iteration:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \gamma \boldsymbol{p}_j^t \odot \nabla l(\boldsymbol{x}_i^t, \xi_i), \forall i \in \mathcal{C}_j, \forall j. \tag{4}$$

Note that $\boldsymbol{p}_j^t$ keeps invariant during one global round, i.e., $\boldsymbol{p}_j^{mE+e} = \boldsymbol{p}_j^{mE}$, $\forall e = \{1, 2, \ldots, E-1\}$, where m denotes the number of the global rounds. Subsequently, clients upload the updated submodels to the edge server for edge model aggregation.

Edge Servers: After every H steps of local submodel updates, each edge server aggregates the local models within its coverage as

$$\bar{\boldsymbol{x}}_j^{t+1} \leftarrow \frac{1}{n_j} \sum_{i \in \mathcal{C}_j} \boldsymbol{x}_i^{t+1}, \forall j. \tag{5}$$

Subsequently, edge servers determine whether to upload the aggregated model to the cloud server or disseminate it to the clients. The criterion is whether the current iteration number $t+1$ of clients is divisible by E. If not, edge servers just disseminate the edge models in (5) to the corresponding clients; otherwise, edge servers upload their edge models to the cloud server.

Cloud Server: If the client's current iteration number $t+1$ is a multiple of E, the cloud server aggregates the edge models from edge servers according to

$$\bar{\boldsymbol{x}}^{t+1} = \sum_{j}^{N} \boldsymbol{p}_j^t \odot \bar{\boldsymbol{x}}_j^{t+1}. \tag{6}$$

Subsequently, the cloud server partitions the global model $\bar{\boldsymbol{x}}^{t+1}$ based on newly generated masks $\boldsymbol{p}_j^{t+1}$

$$\bar{\boldsymbol{x}}_j^{t+1} = \boldsymbol{p}_j^{t+1} \odot \bar{\boldsymbol{x}}^{t+1}, \forall j \in \{1, 2, \ldots, N\}. \tag{7}$$

Finally, $\bar{\boldsymbol{x}}_j^{t+1}$ will be sent to edge server j for initiating the next round of training. Here, it is worth emphasizing that $\bar{\boldsymbol{x}}^t$ is defined on $t \in \{mE \mid m \in \mathbb{N}\}$ while $\bar{\boldsymbol{x}}_j^t$ is defined on $t \in \{mH \mid m \in \mathbb{N}\}$.

With the proposed algorithm, clients and edge servers are not required to store or manipulate the full model parameters. This enables HIST to reduce the communication, computation, and storage burdens of clients and edge servers compared to the conventional HFL.

B. Communication Complexity Analysis

Let L_0 denote the transmission load of a full model. Each client sends its local model parameter to the corresponding edge server every H iterations, where H denotes the number of local updates. Assume that the mask size, defined as the number of non-zero entries of $\boldsymbol{p}_j^t$, is uniform among N cells. In every H iterations, the total communication load of all the clients within cell j becomes $\frac{n_j L_0}{N}$, which corresponds to the communication complexity of edge server j. The average per-iteration communication load of each client and edge server is $\frac{L_0}{NH}$ and $\frac{n_j L_0}{NH}$, respectively. Additionally, for the cloud server, the communication complexity at every E iterations is L_0. Under the methodology of HIST, the communication complexity of the cloud server is invariant to the number of edge servers. In summary, HIST reduces the communication consumption of the global server, edge server, and client to $\frac{1}{N}$ of what would be required by the standard hierarchical FedAvg algorithm.

Algorithm 1: Hierarchical Independent Submodel Training Algorithm

Input: Initial masks $\{\boldsymbol{p}_1^0, \boldsymbol{p}_2^0, \ldots, \boldsymbol{p}_N^0\}$, initial models $\bar{\boldsymbol{x}}^0$, and $\boldsymbol{x}_i^0 = \bar{\boldsymbol{x}}_j^0 = \boldsymbol{p}_j^0 \odot \bar{\boldsymbol{x}}^0, \forall i \in \mathcal{C}_j, \forall j$, learning rate γ

```
for t ∈ {0, 1, ..., T−1} do
    for each cell and edge server in parallel do
        for each client i ∈ C_j in parallel do
            Update local submodel x_i^{t+1} by (4)
        end
        if H | t+1 then
            Update edge model x̄_j^{t+1} via (5)
            if E | t+1 then
                Upload x̄_j^{t+1} to the cloud server
            else
                Disseminate x̄_j^{t+1} to clients
            end
        end
    end
    if E | t+1 then
        Update the global model x̄^{t+1} via (6)
        Generate masks {p_j^{t+1}} under rule (3)
        Partition the global model by (7) and send the
         obtained submodels x̄_j^{t+1} to clients within cell
         j, x_i^{t+1} = x̄_j^{t+1}, ∀i ∈ C_j, ∀j
    end
end
```

IV. Convergence Analysis

In this section, we provide convergence analysis for the proposed HIST algorithm. Although the proposed HIST algorithm shares a similar training process with the hierarchical FedAvg, we stress that the convergence proof for the latter one cannot be directly extended to our case that adopts *submodel partitioning*, due to the effect of the masks. Specifically, when comparing $\boldsymbol{p}_j^t \odot \nabla F_i(\boldsymbol{p}_j^t \odot \boldsymbol{x})$ and $\nabla F_i(\boldsymbol{x})$, the mask $\boldsymbol{p}_j^t$ compresses not only the gradient but also the model, while many existing works only investigate compressing the gradient. Theoretical analysis on the methods of compressing the model [16] is quite limited. Even in the single-cell scenario, existing proofs of IST [13], [14] rely on some stronger assumptions. Finally, the hierarchical architecture with both multiple steps of client update and multiple steps of edge training we consider makes the analysis further complicated.

A. Assumptions

We focus on a general non-convex loss function and consider a non-i.i.d data setting. Our theoretical analysis relies on the following assumptions.

Assumption 1. *The global loss function $f(\boldsymbol{x})$ has a lower bound f_*, i.e., $f(\boldsymbol{x}) \geq f_*, \forall \boldsymbol{x}$.*

Assumption 2. *F_i is differentiable and L-smooth, i.e., there exists a positive constant L such that for any $\boldsymbol{x}$ and $\boldsymbol{y}$*

$$\begin{aligned}&\|\nabla F_i(\boldsymbol{x})-\nabla F_i(\boldsymbol{y})\|^2 \leq L\|\boldsymbol{y}-\boldsymbol{x}\|, \forall i,\\ &F_i(\boldsymbol{y}) \leq F_i(\boldsymbol{x})+\langle \nabla F_i(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle+\frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2, \forall i.\end{aligned} \tag{8}$$

Assumption 3. *The stochastic gradient $\nabla l(\boldsymbol{x}, \xi_i)$ is an unbiased estimate of the true gradient, i.e., $\mathbb{E}_{\xi_i \sim \mathcal{D}_i}[\nabla l(\boldsymbol{x}, \xi_i)] = \nabla F_i(\boldsymbol{x}), \forall \boldsymbol{x}$.*

Assumption 4. *The variance of the stochastic gradient $\nabla l(\boldsymbol{x}, \xi_i)$ is bounded as*

$$\mathbb{E}_{\boldsymbol{\xi}_i \sim \mathcal{D}_i}\|\nabla l(\boldsymbol{x}, \xi_i)-\nabla F_i(\boldsymbol{x})\|^2 \leq \sigma^2, \forall \boldsymbol{x}. \tag{9}$$

Assumption 5. *The gradient dissimilarity between the global loss and each edge loss f_j can be bounded by a constant δ_1^2, i.e.,*

$$\frac{1}{N}\sum_{j=1}^{N}\|\nabla f_j(\boldsymbol{x})-\nabla f(\boldsymbol{x})\|^2 \leq \delta_1^2, \forall \boldsymbol{x}. \tag{10}$$

Assumption 6. *The gradient dissimilarity between the edge loss f_j and each client loss $F_i(\boldsymbol{x})$ can be bounded by a constant δ_2^2, i.e.,*

$$\frac{1}{n_j}\sum_{i \in \mathcal{C}_j}\|\nabla F_i(\boldsymbol{x})-\nabla f_j(\boldsymbol{x})\|^2 \leq \delta_2^2, \forall \boldsymbol{x}, \forall j. \tag{11}$$

Assumptions 1-4, have been widely adopted in the context of stochastic non-convex and smooth settings [2]. Assumptions 5 and 6 serve to characterize the degree of data heterogeneity between different cells and clients, which is a common characteristic within the HFL literature [8].

B. Theoretical Results

When implementing HIST in practice, $\bar{\boldsymbol{x}}^t$ will not be computed unless t is a multiple of E as the global synchronization occurs every E iterations. We establish the convergence properties of the proposed algorithm by characterizing the evolution of $\|\nabla f(\hat{\boldsymbol{x}}^t)\|^2$, $\hat{\boldsymbol{x}}^t \coloneqq \sum_{j=1}^{N} \boldsymbol{p}_j^t \odot \frac{1}{n_j}\sum_{i \in \mathcal{C}_j} \boldsymbol{x}_i^t, t = \{0, 1, 2, \ldots, T-1\}$, to see how fast the model converges to the stationary point of the general non-convex loss function. The sequence $\{\hat{\boldsymbol{x}}^t \mid t = 0, 1, 2, \ldots, T-1\}$ we use for analysis serves as a virtual global model, which is commonly employed to monitor the convergence of distributed algorithms with delayed global synchronization [17]. Now we state the following main theorem. Due to space limitation, all proofs are provided in our technical report [18].

Theorem 1. *Suppose that Assumptions 1-6 hold, the masks $\{\boldsymbol{p}_1^t, \boldsymbol{p}_2^t, \ldots, \boldsymbol{p}_N^t\}$ are uniformly and randomly generated based on* (3)*, $N \geq 2$, and the step size satisfies*

$$\gamma \leq \min\left\{\frac{1}{32E\sqrt{N-1}L}, \frac{\tilde{N}}{NHL}, \frac{1}{NH^2L}, \frac{1}{(N+1)EL}\right\}. \tag{12}$$

Then, HIST *achieves the following convergence behavior for non-convex loss functions:*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\hat{\boldsymbol{x}}^t)\right\|^2 \leq 4\frac{f(\bar{\boldsymbol{x}}_0)-f_*}{\gamma}+50\gamma\tilde{N}L\sigma^2\\ +24\gamma L\delta_2^2+12\delta_1^2+24(N-1)L^2\frac{E}{T}\sum_{m=0}^{T/E-1}\mathbb{E}\left\|\bar{\boldsymbol{x}}^{mE}\right\|^2,\end{aligned} \tag{13}$$

where $\tilde{N} = \sum_{j=1}^{N}\frac{1}{n_j}$, and $\bar{\boldsymbol{x}}_{mE}$ is the synchronized global model generated by our HIST *algorithm.*

Theorem 1 presents the optimality gap for the time-averaged squared gradient norm. The first term in this upper bound exhibits the influence of the initial optimality gap on convergence performance. The second term reveals the impact of the variance of stochastic gradients on convergence, which can be mitigated by increasing the batch size when computing stochastic gradients. The third and fourth terms indicate that the non-i.i.d. characteristics within the cell and across cells affect convergence performance. The last term demonstrates that the norms of synchronized global models also influence the optimality gap. Note that the last two terms are induced by submodel partition. In addition, the step size γ is a configurable parameter that impacts the first three terms of the derived upper bound. Plugging an appropriate step size into Theorem 1 gives rise to the following corollary.

Corollary 1. *Suppose that Assumptions 1-6 hold, the masks $\{\boldsymbol{p}_1^t, \boldsymbol{p}_2^t, \ldots, \boldsymbol{p}_N^t\}$ are uniformly and randomly generated based on* (3)*, $N \geq 2$, and let the step size $\gamma = (T\tilde{N})^{-\frac{1}{2}}$ in which T is large enough to satisfy* (12)*. Then, the* HIST *algorithm satisfies*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\hat{\boldsymbol{x}}^t)\right\|^2 \leq \mathcal{O}\left(\tilde{N}^{\frac{1}{2}}T^{-\frac{1}{2}}\right)+\mathcal{O}\left(T^{-\frac{1}{2}}\right)\\ +12\delta_1^2+24(N-1)L^2\frac{E}{T}\sum_{m=0}^{T/E-1}\mathbb{E}\left\|\bar{\boldsymbol{x}}^{mE}\right\|^2,\end{aligned} \tag{14}$$

where $\tilde{N}$ and $\bar{\boldsymbol{x}}^{mE}$ are described in Theorem 1.

Remark 1. *In Corollary 1, the retention of $\tilde{N}$ within the convergence rate expression is motivated by the possibility of an arbitrary relationship between the number of clients in each cell, denoted as n_j, and the total number of cells, denoted as N. When the number of clients in each cell is of a comparable magnitude or greater than the total number of cells, the convergence rate of the diminishing terms in the derived upper bound is primarily determined by $\mathcal{O}\left(T^{-\frac{1}{2}}\right)$. However, if the number of clients in each cell is significantly smaller in relation to the total number of cells, $\tilde{N}$ becomes influential, and the convergence rate is dominated by $\mathcal{O}\left(\tilde{N}^{\frac{1}{2}}T^{-\frac{1}{2}}\right)$.*

C. Discussions

Non-diminishing bound: With the step size chosen in Corollary 1, the first three terms in (13) will diminish to zero as long as the number of total iterations, i.e., T, is large enough.

The rest two terms are non-diminishing parts that arise due to submodel training. One can claim that HIST can converge to the neighborhood of a stationary point of the non-convex loss function under the aforementioned conditions. A similar phenomenon has also been reported in the single-cell case [9], [13], [15]. The bound enables us to explore the performance-resource trade-off, where more detailed discussions will be provided in the next paragraph.

The choice of N: As N increases, i.e., as the overall clients in the system are divided into more cells during training, the size of the submodels gets smaller, providing a more lightweight model to the edge servers and clients. As a result, the training costs including computation, communication, and storage will be reduced at each iteration. However, as observed in Corollary 1, a large N causes the sequence to deviate further from the stationary point. Overall, there is a trade-off between the convergence performance and computation, communication, and storage costs.

The optimal values of H and E: The choices of H and E impact the communication frequency. As H increases, the aggregation frequency at the edge servers will become smaller, reducing the communication load between clients and the edge server. On the other hand, a large E induces fewer global synchronizations, which releases the communication burden between edge servers and the cloud server. However, these values cannot be infinitely large. The maximum value of H and E can be derived from the condition of the step size γ. Specifically, to make the step size $\gamma = (T\tilde{N})^{-\frac{1}{2}}$ in Corollary 1 satisfy (12), H and E can be set as on the order of $\min\left\{\mathcal{O}\left((\tilde{N}T)^{\frac{1}{4}}N^{-\frac{1}{2}}\right), \mathcal{O}\left(\tilde{N}^{\frac{3}{2}}T^{\frac{1}{2}}N^{-1}\right)\right\}$ and $\mathcal{O}\left((\tilde{N}T)^{\frac{1}{2}}N^{-1}\right)$ at most, respectively.

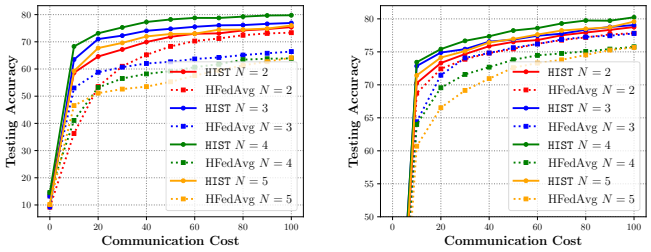
V. Simulations

In this section, we conduct experiments to evaluate the performance of the proposed HIST algorithm.

A. Simulation Settings

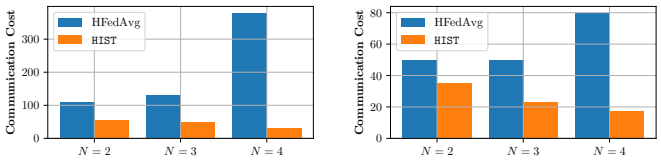
We consider an image classification task on Fashion-MNIST using a two-layer fully connected neural network. In this model, we configure the input layer to have 784 neurons, corresponding to the size of the input image, and the output layer to have 10 neurons, which matches the number of classes. Additionally, we employ a hidden layer with 300 neurons. The cloud server partitions these hidden neurons to construct different submodels. Let the submodels share the same size with each other, which can be achieved by uniformly and randomly partitioning the hidden neurons.

We consider a setup with 60 clients evenly distributed across N cells, where $N \in \{2, 3, 4, 5\}$. We consider two practical data distribution settings: (i) the fully non-i.i.d. case and (ii) the case with i.i.d data across cells but non-i.i.d data across the clients within the same cell. For the former case, the client's dataset construction follows the approach outlined in [2]. The process begins by sorting the training samples based on their corresponding labels. Following this, the training dataset is partitioned into 120 shards, with each shard containing 500 samples. Subsequently, each client is assigned 2 shards, ensuring that each client's dataset comprises 1000 samples. For the latter, we first uniformly and randomly divide the entire training set into N parts, corresponding to N cells, and then distribute each part to the clients within the respective cell in a non-i.i.d. manner following the former case.



(a) Fully non-i.i.d. (b) Cell i.i.d., client non-i.i.d.

Fig. 2: The impact of the number of cells N on the convergence performance.

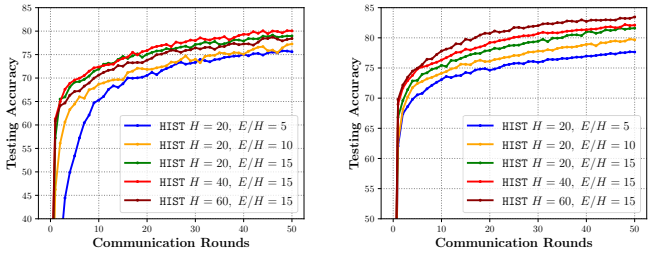


(a) Fully non-i.i.d. (b) Cell i.i.d., client non-i.i.d.

Fig. 3: Communication cost for achieving an accuracy of 75%.

B. Experiment Results and Discussions

Comparison with Baselines: In Fig. 2, we compare our proposed HIST algorithm with the traditional hierarchical FedAvg (denoted as HFedAvg in our figures) where the full model is communicated over the network. We compare their performance in terms of testing accuracy under different numbers of cells, $N \in \{2, 3, 4, 5\}$. The x-axis here represents the communication load which quantifies the volume of parameters transmitted by each client, where the unit is set to the load of a full-model transmission. For each client, the communication cost per global round is equal to $\frac{1}{N}\frac{E}{H}$ times the load of a full-model transmission. Here, E/H represents the number of edge aggregations per global round. We set H and E to 40 and 200, respectively. As shown in Figs. 2a and 2b, the proposed HIST algorithm outperforms hierarchical FedAvg in terms of testing accuracy at the same levels of communication consumption for both data distribution settings. Additionally, as N increases, the non-i.i.d. extent of data among clients becomes more pronounced, leading to performance degradation for hierarchical FedAvg. In contrast, the proposed HIST achieves a higher testing accuracy when N increases from $N = 2$ to $N = 4$. This is because, for HIST, the per-round communication cost per client decreases as the number of cells increases. However, when we increase the number of cells to $N = 5$, HIST also suffers performance degradation. This can



(a) Fully non-i.i.d. (b) Cell i.i.d., client non-i.i.d.

Fig. 4: The impacts of aggregation periods H and E on the convergence performance.

be attributed to the submodel getting too small to effectively handle the task, highlighting the trade-off between training costs and testing accuracy.

Fig. 3 compares the communication cost of HIST and hierarchical FedAvg for achieving the desired accuracy under both data distribution scenarios. This experiment was carried out with $H = 40$ and $E = 200$. The desired testing accuracy is set to be 75%. The Y-axis measures the size of parameters transmitted by each client during the training process as in the x-axis of Fig. 2. It is observed that HIST needs less communication to achieve the preset accuracy, which demonstrates the efficiency of the proposed algorithm over hierarchical FedAvg. In addition, as the number of cells increases from $N = 2$ to $N = 4$, the communication cost shows a decreasing trend, which forms a sharp comparison with hierarchical FedAvg. This further demonstrates the advantage of HIST.

Effects of System Parameters: The impacts of the periods of the edge aggregation H and the global synchronization E/H on the convergence behavior are demonstrated in Fig. 4. The x-axis represents the number of global model synchronizations at the cloud server. We consider the 3-cell case (i.e., $N = 3$) where 60 users are uniformly distributed across these cells without overlapping. As E/H increases from 5 to 10 to 15, HIST attain a better convergence performance, which is witnessed by both Figs. 4a and 4b. This is because a large E/H gives rise to a lower communication load for each round. When H increases from 20 to 40, and from 40 to 60, Fig. 4b shows that the convergence speed of HIST first enjoys an acceleration and then a degradation, where the latter is induced by data heterogeneity. This phenomenon also fits well with our theory, where we show that there is an upper bound for the number of local updates. Fig. 4b shows that HIST has a better performance as H increases from 40 to 60. This is because this data distribution exhibits lower data heterogeneity, which allows for a larger number of local updates.

VI. Conclusion

In this paper, we developed a hierarchical federated submodel training algorithm termed HIST, that is efficient in terms of communication, computation, and storage by integrating independent model training with local training. We investigated its convergence behavior with uniform submodel partitioning under non-convex loss functions and non-i.i.d. data settings, and characterized the impacts of non-i.i.d. extent, the number of periods of edge and global aggregations, and the number of cells on the convergence performance. We show that HIST converges with rate $\max\left\{\mathcal{O}\left(\tilde{N}^{\frac{1}{2}}T^{-\frac{1}{2}}\right), \mathcal{O}\left(T^{-\frac{1}{2}}\right)\right\}$ to a neighborhood of a stationary point of the global loss function. Simulation results show that HIST is able to achieve the target accuracy much faster with less training costs, compared to the standard hierarchical FedAvg.

References

[1] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g: Vision, enabling technologies, and applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, 2021.

[2] W. Fang, Z. Yu, Y. Jiang, Y. Shi, C. N. Jones, and Y. Zhou, "Communication-efficient stochastic zeroth-order optimization for federated learning," *IEEE Transactions on Wireless Communications*, vol. 70, pp. 5058–5073, 2022.

[3] L. Yuan, L. Sun, P. S. Yu, and Z. Wang, "Decentralized federated learning: A survey and perspective," *arXiv preprint arXiv:2306.01603*, 2023.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[5] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE journal on selected areas in communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[6] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical federated learning with quantization: Convergence analysis and system design," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 2–18, 2022.

[7] W. Wen, Z. Chen, H. H. Yang, W. Xia, and T. Q. Quek, "Joint scheduling and resource allocation for hierarchical federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5857–5872, 2022.

[8] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Demystifying why local aggregation helps: Convergence analysis of hierarchical sgd," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8548–8556, 2022.

[9] B. Yuan, C. R. Wolfe, C. Dun, Y. Tang, A. Kyrillidis, and C. Jermaine, "Distributed learning of fully connected neural networks using independent subnet training," *Proceedings of the VLDB Endowment*, vol. 15, no. 8, pp. 1581–1590, 2022.

[10] C. R. Wolfe, J. Yang, F. Liao, A. Chowdhury, C. Dun, A. Bayer, S. Segarra, and A. Kyrillidis, "Gist: Distributed training for large-scale graph convolutional networks," *Journal of Applied and Computational Topology*, pp. 1–53, 2023.

[11] C. Dun, C. R. Wolfe, C. M. Jermaine, and A. Kyrillidis, "Resist: Layer-wise decomposition of resnets for distributed training," in *Uncertainty in Artificial Intelligence*, pp. 610–620, PMLR, 2022.

[12] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *International Conference on Learning Representations*, 2021.

[13] A. Mohtashami, M. Jaggi, and S. Stich, "Masked training of neural networks with partial gradients," in *International Conference on Artificial Intelligence and Statistics*, pp. 5876–5890, PMLR, 2022.

[14] H. Zhou, T. Lan, G. Venkataramani, and W. Ding, "On the convergence of heterogeneous federated learning with arbitrary adaptive online model pruning," *arXiv preprint arXiv:2201.11803*, 2022.

[15] E. Shulgin and P. Richtárik, "Towards a better theoretical understanding of independent subnetwork training," *arXiv preprint arXiv:2306.16484*, 2023.

[16] A. Khaled and P. Richtárik, "Gradient descent with compressed iterates," *arXiv preprint arXiv:1909.04716*, 2019.

[17] S. U. Stich, "Local sgd converges fast and communicates little," in *International Conference on Learning Representations*, 2019.

[18] W. Fang, D.-J. Han, and C. G. Brinton, "Technical report." https://wenzhifang.github.io/ICC/HIST_report.pdf, 2023.