

FedLP: Layer-wise Pruning Mechanism for Communication-Computation Efficient Federated Learning

Zheqi Zhu[†], Yuchen Shi[†], Jiajun Luo[†], Fei Wang[‡], Chenghui Peng[‡], Pingyi Fan^{†*}, and Khaled B. Letaief[§]

[†]Department of Electronic Engineering, Tsinghua University.

Emails: {zhuzq18, shiyc21, luo-jj18}@mails.tsinghua.edu.cn, fpy@tsinghua.edu.cn

[‡]Huawei Wireless Technology Lab. Emails: {wangfei76, pengchenghui}@huawei.com

[§]Department of ECE, Hong Kong University of Science and Technology. Email: eekhaled@ece.ust.hk

Abstract—Federated learning (FL) has prevailed as an efficient and privacy-preserved scheme for distributed learning. In this work, we mainly focus on the optimization of computation and communication in FL from a view of pruning. By adopting layer-wise pruning in local training and federated updating, we formulate an explicit FL pruning framework, FedLP (Federated Layer-wise Pruning), which is model-agnostic and universal for different types of deep learning models. Two specific schemes of FedLP are designed for scenarios with homogeneous local models and heterogeneous ones. Both theoretical and experimental evaluations are developed to verify that FedLP relieves the system bottlenecks of communication and computation with marginal performance decay. To the best of our knowledge, FedLP is the first framework that formally introduces the layer-wise pruning into FL. Within the scope of federated learning, more variants and combinations can be further designed based on FedLP.

Index Terms—federated learning, model pruning, layer-wise aggregation, communication-computation efficiency.

I. INTRODUCTION

A. Backgrounds

By locally training the distributed models and periodically updating the global model, federated learning (FL), first conceptualized in [1], provides an explicit paradigm for cooperative learning without sharing the privacy data. Instead of transmitting the data or intermediate outputs in the networks, only the model parameters are interacted, which significantly improves the communication efficiency.

With the continuous growth of the communication systems and the intelligent devices, it is possible to adopt FL schemes in numerous promising applications such as mobile edge computing (MEC), artificial intelligence of things (AIoT), and autonomous driving [2]. Since leveraging AI in networks is envisioned as a core characteristic of 6G systems, FL has shown its powerful potentials on combinations with deep learning models [3], [4]. On the one hand, FL naturally fits

the structure of multi-user networks with distributed data and can be easily deployed for machine learning tasks [5]. On the other hand, FL schemes are able to achieve the intelligent collaboration for multiagent systems [6].

Though FL has received rapid developments in terms of methods, models and applications, such distributed learning scheme still suffers from challenges in several aspects. This work addresses two key issues, the heterogeneity and the communication-computation efficiency. Firstly, as summarized in [7], the heterogeneity of FL systems mainly comes from local data and the client devices. The heterogeneous local data, also referred to as non-iid data, occurs commonly in real-world distributed scenarios and usually cause the degradation in terms of model performance, convergence and stability. Meanwhile, due to the diversity in computation platforms, communication capabilities and the battery level of the devices, clients may meet different constraints in model scales and local processing. For example, some weak clients are not able to support sufficient local training, which leads to bad model performance and time delay for synchronized aggregation. In contrast, some strong clients may not fully utilize its devices' capability to get better service quality, leading to unfairness of the whole system. Thus, how to design heterogeneous models suitable for various clients is still an open problem.

Besides, communication is also a critical bottleneck for FL networks, especially those with massive number of clients. Since communication and computation are tightly coupled in FL systems, the interplay of these progresses impacts the model quality as well as the system efficiency. Therefore, in recent studies of FL, more attention is paid to optimizing FL schemes through communication compression and computing reduction [8].

B. Motivations & Related Works

As model pruning has been verified to be an efficient approach to reduce the model scales with the cost of marginal loss in accuracy, related technique has also been employed in the context of FL. In this work, we mainly focus on the

* Pingyi Fan is the corresponding author. This work was supported by the National Key Research and Development Program of China (Grant NO.2021YFA1000500(4)).

[§] K. B. Letaief is supported in part by the Hong Kong Research Grant Council under Grant No. 16208921.

Accepted as a conference paper by IEEE International Conference on Communications (ICC) 2023.

pruning mechanism in FL to relieve the communication and computation restrictions.

In the literature, the combination of model compression and FL has already been investigated. Aiming to reduce the client resource requirements in FL systems, Caldas *et al.* proposed Federated Dropout in [9], where the global model is compressed into sub-models for communication. Such schemes were extended in [10], where the authors proposed the dataset-aware dynamic pruning approach to accelerate the inference on edge devices. Jiang *et al.* in [11] and Kumar *et al.* in [12] formulated an adaptive pruning strategies based on gradient information and neuron importance, respectively. An efficient private update scheme of federated sub-model learning was also discussed in [13]. Further, authors of [14], [15] theoretically concerned about the pruning configurations and the communication resource allocation in wireless FL.

However, negative findings in a critical paper [16] argued that dropout-based FL schemes may perform worse than simple ensemble methods. Actually, such pruning schemes in FL are borrowed from dropout in single machine learning. Locally training sub-models and aggregating them to the corresponding parts of the global model lacks explainability. The authors in [17] also pointed out that the order of the parameters cannot be neglected in pruned aggregation. Besides, most existing dropout-based schemes in FL employed the intra-layer pruning, which results in increment of system complexity, e.g., different operations for different functional layers. Chen *et al.* in [18] introduced the block dropout for large-scale neural network training. Inspired by the layer-wise aggregation in [19], we first propose to explore layer-wise pruning mechanism to relieve above quagmires in this work.

C. Contributions & Paper Organization

The main contributions of this work can be summarized as follows:

- We first put forward a universal FL pruning framework, FedLP¹, employing the layer-wise pruning mechanism. FedLP can relieve restrictions of communication as well as computation in FL systems, and also potentially prevents model attacks in some degrees.
- We sketch two basic pruning schemes and the theoretical principle of FedLP for both homogeneous and heterogeneous cases. In particular, the heterogeneous scheme fits the scenarios where the clients vary from device types and computation capabilities, and thus, the local models shall be set adaptively.
- We develop experiments to evaluate the communication-computation efficiency of FedLP¹. The outcomes suggest that such layer-wise pruning mechanism significantly reduces the communication loads and computational complexity with controllable performance loss.

The remaining of this article is organized as follows. In Section II we introduce the preliminaries of this work and illustrate how the basic idea of layer-wise pruning is

formulated through a simple experiment. In Section III, we present two typical schemes of FedLP for homogeneous and heterogeneous scenarios. The corresponding algorithms and a theoretical principle will also be developed. The detailed experimental results and more discussions are presented in Section IV. Finally, in Section V, we conclude this work and point out several potential research directions.

II. PRELIMINARIES AND LAYER-WISE PRUNING

In this section, we first briefly introduce the preliminaries of this work. Then, we illustrate the key idea of layer-wise pruning, which inspires us to sketch FedLP framework.

A. Federated Learning

Recapping a classical horizontal federated learning (HFL) system, there exists N distributed clients with their own local datasets, $\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$, and local models, $\{\theta_1, \dots, \theta_N\}$. For privacy preserving and communication efficiency, FL carries out procedures of local training and periodic model aggregation. A federated period processes as follows: 1) Clients train local models with local data; 2) Parameter server collects local models uploaded by K clients and aggregates them as the global model $\bar{\theta}$; 3) Clients download the updated global model for further training.

As for federated updating, at each global epoch t , HFL selects a set of participators with K clients as P_t , and proceeds the parameter aggregation:

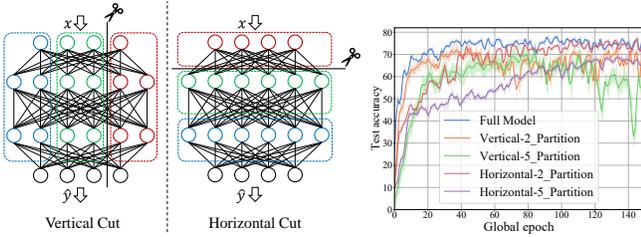
$$\bar{\theta}_t \leftarrow \sum_{k \in P_t} \frac{\omega_k}{\sum_{m \in P_t} \omega_m} \theta_{k,t}, \quad (1)$$

where $\{\omega_k\}$ is the aggregation weights and $\theta_{k,t}$ is the local model of client k after the local training in t -th global epoch. In particular, it reduces to the most popular scheme, FedAvg, when the weights are set as $\omega_k = \frac{|\mathcal{D}_k|}{\sum |\mathcal{D}_m|}$.

B. A Simple Test and Layer-wise Pruning

To further optimize the communication and computation progresses, pruning is a simple but efficient method. However, as mentioned above, existing pruning methods in FL are migrated from the traditional machine learning fields. Namely, the connections between neurons in MLP (multi-layer perceptron) or the filters in CNN (convolutional neural network) are dropped in order to reduce the model scales for training locally or the parameter quantities for transmitting. As shown in the left of Fig. 1(a), we rethink such pruning mechanism as a vertical cut since some inner-layer neurons are detached and the intermediate features of middle layers might be down-scaled. Then, it is natural to consider horizontal scheme, which conducts a layer-wise cut. In single machine learning, detaching a whole layer leads to a completely different model. Thus, such schemes are not termed as a pruning technique. Nevertheless, within the context of FL, clients shall also horizontally cut their model and contribute to the global model together, as shown in the right of Fig. 1(a). Briefly, vertical cut keeps all layers and drops some neurons in each layer, while horizontal cut drops some layers and keeps all neurons for preserved layers.

¹The codes in this work are available at <https://github.com/Zhuzq/FedLP>



(a) Two types of pruning mechanisms. (b) Global model performance.

Fig. 1. A simple comparison for vertical/horizontal pruning in FL.

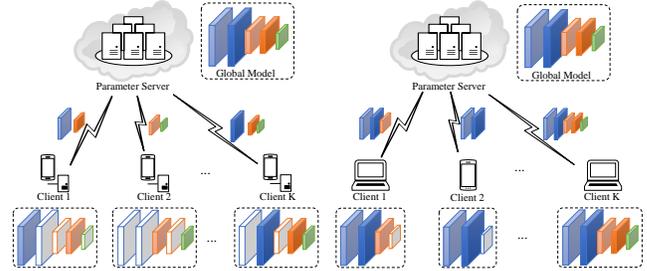
Inspired by such intuition, we develop a simple experiment to check the performance of horizontal cut. The experiment is implemented on a FL system with 100 clients, 0.5 participation rate and non-i.i.d. data divided from Fashion-MNIST dataset. For better model performance, we only consider the pruned aggregation, which means that clients train the full model locally and only upload the pruned model. For vertical cut, the parameters within every layer is uniformly divided into 2/5 partitions and each client only upload one partition for model aggregation. Same for horizontal cut, the full whole layers are splitted into 2/5 partitions and each client upload one partitions. The comparison results are presented in Fig. 1(b). One can find that for same partition count, horizontal cut reaps higher accuracy as well as better stability. Especially, the vertical cut with 5 partition seems not to converge after quite more global epochs.

Based on the above illustration, we deem that the horizontal cut strategies in FL have advantages over the vertical cut. Therefore, we further extend such schemes to layer-wise pruning mechanism and formulate the compressed FL framework, named as FedLP. Apart from the possible outperform beyond traditional pruning mechanism, layer-wise pruning is easier to be applied to different models. For example, clients do not have to figure out whether the model consists of fully-connected (FC) layers or convolutional (Conv) layers before pruning. This is because the whole layer is treated as the smallest pruning unit and the inner-layer structure can be neglected. Thus, we also regard the layer-wise pruning as a model-agnostic mechanism.

III. FEDLP: FRAMEWORKS AND ALGORITHMS

In this section, we will formally propose the basic framework of FedLP. Then, two specific schemes and their corresponding algorithms for homogeneous and heterogeneous models will be designed respectively.

The key idea of FedLP is that the layer-wise pruning mechanism is adopted in the phases of local training and model aggregation. After finishing the local training of every global epoch: 1) Each client only uploads some layers to parameter server; 2) The parameter server aggregates the pruned sub-models and obtains the updated global model; 3) Clients download either the corresponding sub-models or full models to update local models. More specifically, we rewrite a full model θ with L layers as: $\theta := [\theta^1, \theta^2, \dots, \theta^L]$, where θ^l represents the parameters of l -th layer. Assume that each



(a) Homogeneity scheme.

(b) Heterogeneity scheme.

Fig. 2. Two typical FedLP schemes based on different settings of local model. The inactive layers (in gray) are removed from the sub-models for uploading. client k upload layers of index in \mathcal{L}_k after pruning. The pruned model of each client can be represented by:

$$\tilde{\theta}_k = [\theta_k^l], \quad l \in \mathcal{L}_k. \quad (2)$$

Under FedLP aggregation, each layer is operated independently. Let the indicator function $\mathbb{1}_k^l \in \{0, 1\}$ denotes whether layer l is included in the pruned local model of client k . With the pruning information, the layer-wise aggregation rule shall be modified as:

$$\bar{\theta}_t^l \leftarrow \sum_{k \in P_t} \frac{\mathbb{1}_k^l \cdot \omega_k}{\sum_{m \in P_t} \mathbb{1}_m^l \cdot \omega_m} \theta_{k,t}^l. \quad (3)$$

Furthermore, we naturally consider the details of the implementation with the homogeneous clients and the heterogeneous clients. The corresponding two FedLP schemes are displayed in Fig. 2.

A. Homogeneity scenario

All clients possess the full global model, which is also a basic assumption in traditional FL. As shown in Fig. 2(a), each client trains its local model according to local settings. Before uploading the parameters, clients carry out a layer-wise pruning to form the models for aggregation. As presented in Algorithm 1, we formulate a probabilistic rule for homogeneity FedLP, where layer l of client k contributes to aggregation with probability p_k^l , termed as layer-preserving-rate (LPR). Before uploading, each client forms the layer-wise pruned model through the layer preserving indicators $\{\zeta_k^l\}$, which is a binary variable following *Bernoulli*(p_k^l), i.e.,

$$\zeta_k^l = \begin{cases} 1 & \text{with probability } p_k^l, \\ 0 & \text{with probability } 1 - p_k^l. \end{cases} \quad (4)$$

With all pruned local models, parameter server carries out the layer-wise pruned aggregation for each layer according to (3) and obtains the updated global model $\bar{\theta}_t$. While parameter download, clients receive the full global model and update their local models. Besides, guaranteed by Law of large numbers, all layers contribute to the aggregation with the proportion $\{p_k^l\}$.

B. Heterogeneity scenario

Clients train local models of different scales, which is more practical in real-world applications and extends the original FL. For example, in some scenarios, clients may include the

Algorithm 1 FedLP For homogeneous model setting.

Initialization: local models $\{\theta_{k,0}\}$, LPRs $\{p_k^l\}$, system configurations, etc.

```
1: for  $t \leftarrow 1$  to max_epoch do
2:   Select  $K$  clients as participator set,  $P_t$ ;
3:   for client  $k$  in  $P_t$ , parallelly do  $\triangleright$  participator side
4:     Update  $\theta_{k,t} \leftarrow \text{Local\_Train}(\theta_{k,t-1}; \mathcal{D}_k)$ ;
5:     Generate the indicator variables:  $\{\zeta_k^l\} \sim \{p_k^l\}$ ;
6:     Form the pruned local model  $\tilde{\theta}_{k,t}$  through  $\{\zeta_k^l\}$ ;
7:     Upload  $\tilde{\theta}_{k,t}$  to parameter server;
8:   end for
9:   Aggregate each layer  $\bar{\theta}_t^l$  by (3);  $\triangleright$  server side
10:  Update each local model:  $\theta_{k,t} \leftarrow \bar{\theta}_t$ ;  $\triangleright$  client side
11: end for
```

Output: global model: $\bar{\theta}_t$.

mobile devices which has more restrictions on the computation capability, communication bandwidth, energy consumption, etc. Thus, both the local training and the uploading traffic shall be adaptively optimized. Model heterogeneity is always acknowledged as a major challenge in FL because it is hard to design aggregation strategies for various local models. Nevertheless, by adopting the layer-wise pruning mechanism, we can easily build the FedLP schemes for heterogeneous cases. As shown in Fig. 2(b), clients train sub-models with part of layers and upload them for aggregation. We measure the model complexity of client k using the layer count (LC), L_k , which can be determined by the device capability. The local model assigned to client k consists of first L_k layers. To match the data dimensions, the clients with pruned models personalize θ_k^O as their last output layers (in gray). The pruning assignment rule can be formulated as:

$$\tilde{\theta}_k = \begin{cases} [\theta_k^1, \dots, \theta_k^{L_k}, \theta_k^O] & \text{if } L_k < L, \\ [\theta_k^1, \dots, \theta_k^L] & \text{if } L_k = L. \end{cases} \quad (5)$$

Then we develop the FedLP algorithm for heterogeneous scenarios as Algorithm 2. Note that the personalized layers, $\{\theta_k^O\}$, if exists, are only trained locally and will neither be uploaded for aggregation nor updated by downloaded model.

C. Theoretical Result

Due to the page limitation, the detailed theoretical analysis on convergence are skipped and will be presented in our future works. Here, we only give a proposition to show the impact on global gradient caused by FedLP.

Proposition 1. *Assume that local training is independent with pruning operations. In fairness case where $\omega_k \equiv 1/K$ and $p_k^l \equiv p$ for all $k = 1, \dots, K$, FedLP gets $(1-p)^K$ convergence rate decay compared to non-pruned FL.*

Proof. Let $\{g_k^l\}$ denote the accumulated local gradients of $\{\theta_k^l\}$ after local training. We use ζ_k^l in (4) to represent the participation indicator of θ_k^l . By (3), the global aggregated gradient of l -th layer is $\hat{g}^l = \sum_k \frac{\zeta_k^l}{\sum_m \zeta_m^l} g_k^l$. The aggregated

Algorithm 2 FedLP For heterogeneous model setting.

Initialization: model LCs $\{L_k\}$, system configures, etc.

```
1: for each client  $k \leftarrow 1$  to  $N$  do  $\triangleright$  pruning initialization
2:   Assign local model  $\tilde{\theta}_{k,0} \leftarrow [\theta_{k,0}^1, \dots, \theta_{k,0}^{L_k}, (\theta_{k,0}^O)]$ ;
3: end for
4: for  $t \leftarrow 1$  to max_epoch do
5:   Select  $K$  clients as participator set,  $P_t$ ;
6:   for client  $k$  in  $P_t$ , parallelly do  $\triangleright$  participator side
7:     Update  $\tilde{\theta}_{k,t} \leftarrow \text{Local\_Train}(\tilde{\theta}_{k,t-1}; \mathcal{D}_k)$ ;
8:     Upload  $\tilde{\theta}_{k,t}[1 : L_k]$  to parameter server;
9:   end for
10:  Aggregate each layer  $\bar{\theta}_t^l$  by (3);  $\triangleright$  server side
11:  Update each local model:  $\theta_{k,t} \leftarrow \bar{\theta}_t$ ;  $\triangleright$  client side
12: end for
```

Output: global model: $\bar{\theta}_t$.

gradient of non-pruned FL is $\bar{g}^l = \frac{1}{K} \sum_k g_k^l$. Then the expectation of the pruned gradients can be obtained by:

$$\mathbb{E} \hat{g}^l = \mathbb{E} \left\{ \sum_{k=1}^K \frac{\zeta_k^l}{\sum_m \zeta_m^l} g_k^l \right\} = \sum_{k=1}^K \mathbb{E} \left\{ \frac{\zeta_k^l}{\sum_m \zeta_m^l} g_k^l \right\} \quad (6)$$

$$= \sum_{k=1}^K \mathbb{E} \left\{ \mathbb{E} \left\{ \frac{\zeta_k^l}{\sum_m \zeta_m^l} g_k^l \mid \zeta_k^l \right\} \right\} \quad (7)$$

$$= \sum_{k=1}^K p \cdot \mathbb{E} \left\{ \frac{1}{1 + \sum_{m \neq k} \zeta_m^l} \cdot g_k^l \right\} \quad (8)$$

$$= \sum_{k=1}^K p \left[\sum_{m=1}^K \frac{1}{m} \binom{K-1}{m-1} p^{m-1} (1-p)^{K-m} \right] \mathbb{E} g_k^l \quad (9)$$

$$\stackrel{\textcircled{1}}{=} \sum_{k=1}^K \left[\sum_{m=1}^K \frac{1}{K} \binom{K}{m} p^m (1-p)^{K-m} \right] \mathbb{E} g_k^l \quad (10)$$

$$= [1 - (1-p)^K] \sum_{k=1}^K \frac{\mathbb{E} g_k^l}{K} = [1 - (1-p)^K] \mathbb{E} \bar{g}^l \quad (11)$$

where $\textcircled{1}$ holds because $\frac{1}{m} \binom{K-1}{m-1} = \frac{(K-1)!}{(K-m)!m!} = \frac{1}{K} \binom{K}{m}$. (11) means that the aggregated gradient scale of FedLP decreases compared to the non-pruned one, which implies $(1-p)^K$ convergence rate decay. ■

This theoretical result is significant and shows that the impacts of FedLP on convergence can be mitigated by increasing the participation clients or the LPRs $\{p_k^l\}$.

Overall, we give a brief summary for these two FedLP schemes: Homogeneous scheme mainly focuses on the communication progress and concerns less about the local training; On the contrary, for the heterogeneous scheme, both local computation and parameter communication are reduced. It is notable that such a random pruning processing will protect the FL model from attacks in some degrees and strengthen the system robustness as well as security since the attackers are unaware of the exact layer indexes.

IV. EVALUATIONS

In this section, we carry out several experiments to evaluate the performance of FedLP under two mentioned scenarios.

A. Experiment Setups

We develop the experiments in an image classification FL task under CIFAR-10 dataset. For basic configuration, we build up a FL system with 100 clients in total. The participation rate is set as 0.1, which means that 10 clients are randomly selected for aggregation in every global round. Before aggregation, clients proceed 5 epochs to train the local models.

1) *Global Model*: A CNN based model with 6 Conv layers is adopted as the global model. Batch normalization (BN) and maxpooling operations are also conducted following each Conv layer. At the end, a FC layer is placed to assemble the features, followed by another FC as the output layer.

2) *Data Split*: We conduct the experiments under three popular data settings of FL, iid, mixed non-iid and Dirichlet non-iid. For iid split, the training samples are randomly assigned to 100 clients, which means that each client possesses 500 images of uniform categories. Under mixed non-iid (M-niid) split, the training samples are sorted into shards and partitioned to clients. We set the size of each shard as 250 with 5% uniformly sampled from all categories and each client takes 2 shards. Dirichlet non-iid (D-niid) split is also a widely mentioned data partitioning rule [20]. The samples of each category are divided into N parts according to a Dirichlet distribution with parameter $\alpha = 1$, so that the clients own training data of different volumes.

3) *Pruning Strategies*: We adopt FedAvg as the baseline and two proposed layer-wise pruning schemes are implemented. For homogeneous cases, we employ the consistent LPRs, $p_k^l \equiv p$, abbreviated as FedLP-Homo(p). For heterogeneous cases, we prune the global model into 5 ordered layer-sequences with different LCs. These 5 sub-models are assigned to clients according to LC distributions. Specifically, the parameter l of FedLP-Hetero(l) represents the case where sub-models with l LC are mostly assigned with probability 0.6 and other sub-models takes 0.1 respectively. In particular, the parameter 'u' means that all sub-models are chosen uniformly.

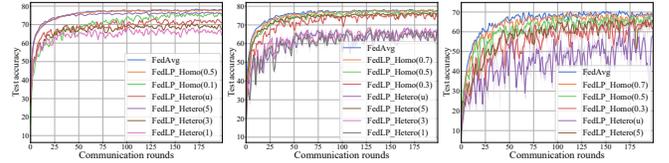
TABLE I

A numerical comparison on accuracy, communication and computation costs.

Schemes	Test accuracy (%) iid / D-niid / M-niid	Comm. #param (k)	Comp. MFLOPs
FedAvg	77.94 / 77.67 / 67.57	1102.93	36.36
FedLP-Homo(0.1)	75.32 / 71.30 / 44.21	606.61	36.36
FedLP-Homo(0.3)	78.20 / 74.92 / 63.24	716.91	36.36
FedLP-Homo(0.5)	77.60 / 77.13 / 66.01	827.20	36.36
FedLP-Homo(0.7)	78.47 / 77.71 / 70.29	937.49	36.36
FedLP-Hetero(1)	66.00 / 67.66 / 37.30	169.60	17.73
FedLP-Hetero(3)	68.82 / 68.29 / 39.54	225.28	24.62
FedLP-Hetero(u)	72.42 / 64.65 / 57.51	318.66	23.83
FedLP-Hetero(5)	76.28 / 76.34 / 65.69	710.80	30.10

B. Accuracy Performance

We evaluate the performance of the proposed layer-wise pruning schemes under three FL data settings. The test accuracy curves of global models are plotted in Fig. 3 and the average numerical results are listed in Table I. One can observe that for higher LPRs (0.7), FedLP-Homo performs even better than original FedAvg under iid and non-iid settings, but saves 30%



(a) Under iid data.

(b) Dirichlet-niid.

(c) Mixed-niid.

Fig. 3. Comparisons under different FL data splits based on CIFAR-10.

communication loads for model upload. Such improvement on generalization capability is also a result of the random LP processing. For intermediate LPRs, FedLP-Homo(0.5) achieves similar accuracy, convergence and stability as communication round increases. The convergence performance of large LPR also fits the Proposition 1. With 0.3 LPR, the performance under iid data keeps the same. The accuracy drops by 3% and 4% under two non-iid settings, compared to non-pruned schemes. In particular, when LPR is set extremely low (0.1), the global models under iid and Dirichlet-niid data only lose 3% and 6% classification accuracy, which is still acceptable. Large performance gap occurs under mixed-niid data training.

For heterogeneous cases, both the local training and the transmission models are pruned. As shown in Fig. 3(a) and 3(b), FedLP schemes still reach high accuracy under iid and Dirichlet-niid data. However, the model accuracy as well as the convergence evidently degrade under mixed-niid data. We explain these phenomena as the result of the non-iid degree and the heterogeneous local models. For severely non-iid data in Fig. 3(c), vanilla FedAvg based approaches cannot handle the imbalanced training data and the divergent local models. Besides, FedLP-Hetero assigns models of different complexity and the personalized output layers are absent for aggregation, which also causes the instability of the global model. Therefore, one can treat such scheme as an alternative solution for the scenarios where the clients are strictly constrained on communication-computation resources and the devices are of variant capability. To further improve the accuracy, the techniques against severely non-iid data shall be added.

C. Communication-Computation Efficiency

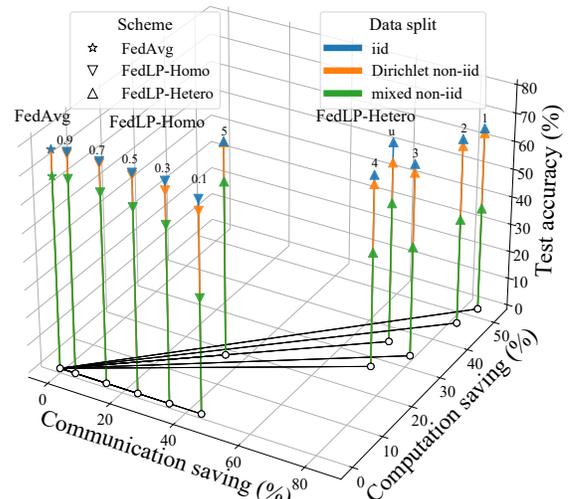


Fig. 4. Trade-offs: accuracy vs. communication-computation efficiency. Homo: $p = \{0.1, 0.3, 0.5, 0.7, 0.9\}$; Hetero: $\{uniform, 1, 2, 3, 4, 5\}$.

We also evaluate the communication-computation efficiency of FedLP. The communication loads are measured by the average parameter count in a global epoch, containing the local model upload and the global model download. The computation complexity is represented by the million floating point of operations (MFLOPs) per local model. These two measures are also listed in Table I. Specifically, FedLP-Homos schemes reduce the data clients transmit to parameter server in different degree and execute the same local training as FedAvg. Meanwhile, under FedLP-Hetero, both local training and the model aggregation are pruned. Thus, both communication and computation costs decrease. Besides, since the traffics of uplinks and downlinks are optimized together, the excessive communication can be further eliminated. For example, the uniform FedLP-Hetero schemes requires only half the communication rates of FedLP-Homo with 0.1 LPR.

Moreover, we sketch a 3D plot of several schemes to visualize the model accuracy and the system efficiency. The x-y axis represents the percentage of communication and computation savings respectively. And the height is the test accuracy of global model. The farther a projection on x-y plane locates towards 0-point, the lower communication/computation capabilities are required, which increases the system efficiency. Fig. 4 intuitively reflects such trade-offs between model performance and system costs which provide guidance for the system designs and layer-wise pruning settings.

Above results suggest that it is not necessary either for clients to possess the full global model or for parameter server to collect all layers of local models. While the data is not highly non-iid, clients are allowed to prune some layers with acceptable accuracy decay, which significantly reduces the communication loads as well as the local computation complexity. In other words, FedLP relaxes the restrictions on communication and computation for practical FL systems.

V. CONCLUSION

In this work, we rethought the pruning strategies in the context of FL and proposed a layer-wise pruning mechanism, FedLP. Instead of dropping the intra-layer parameters vertically, FedLP operates pruning horizontally on each layer to improve the communication-computation efficiency of FL systems. We drew a basic sketch of layer-wise pruning by developing two probabilistic FedLP schemes for homogeneous and heterogeneous scenarios. Theoretical guarantees were also derived to interpret the convergence of FedLP. The experimental outcomes verified that FedLP reduces both communication and computation costs with the controllable loss of model performance. Moreover, FedLP is model-agnostic and can be easily deployed in different FL schemes regardless of the neural network structures and the layer types. Such an explicit pruning mechanism provides alternative ways to implement FL tasks on edge devices with variant capabilities.

Based on FedLP, more works of several aspects can be investigated in the future. Firstly, layer-wise pruning schemes such as the dynamic pruning with adaptive layer weights shall be explored to fit the changeable environments and the

non-iid data. Secondly, the theoretical analysis on learning convergence and system configuration will be presented in our future works. In additional, FedLP's potentials on system robustness and model security can be further discussed.

REFERENCES

- [1] 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*. PMLR, 2017, pp. 1273–1282.
- [2] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.
- [3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.
- [4] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, 2021.
- [5] S. Wan, J. Lu, P. Fan, Y. Shao, C. Peng, J. Chuai *et al.*, "How global observation works in federated learning: Integrating vertical training into horizontal federated learning," *IEEE Internet of Things Journal*, 2023.
- [6] Z. Zhu, S. Wan, P. Fan, and K. B. Letaief, "Federated multiagent actor-critic learning for age sensitive mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1053–1067, 2021.
- [7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [8] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
- [9] S. Caldas, J. Konečný, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.
- [10] S. Yu, P. Nguyen, A. Anwar, and A. Jannesari, "Adaptive dynamic pruning for non-iid federated learning," *arXiv preprint arXiv:2106.06921*, 2021.
- [11] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [12] G. Kumar and D. Toshniwal, "Neuron specific pruning for communication efficient federated learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4148–4152.
- [13] S. Vithana and S. Ulukus, "Efficient private federated submodel learning," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 3394–3399.
- [14] S. Liu, G. Yu, R. Yin, and J. Yuan, "Adaptive network pruning for wireless federated learning," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1572–1576, 2021.
- [15] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 923–927, 2022.
- [16] G. Cheng, Z. Charles, Z. Garrett, and K. Rush, "Does federated dropout actually work?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3387–3395.
- [17] S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane, "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 876–12 889, 2021.
- [18] Y. Chen, Z. Chen, P. Wu, and H. Yu, "Fedobd: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning," *arXiv preprint arXiv:2208.05174*, 2022.
- [19] S. Lee, T. Zhang, C. He, and S. Avestimehr, "Layer-wise adaptive model aggregation for scalable federated learning," *arXiv preprint arXiv:2110.10302*, 2021.
- [20] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.