Communication-Efficient Federated Learning with Gradual Layer Freezing

Erich Malan, Valentino Peluso, *Member, IEEE,* Andrea Calimera, *Member, IEEE*, Enrico Macii *Fellow, IEEE*

***Abstract*—Federated Learning (FL) is a collaborative, privacy-preserving method for training deep neural networks at the edge of the Internet-of-Things (IoT). Despite the many advantages, existing FL implementations suffer high communication costs that prevent adoption at scale. Specifically, the frequent model updates between the central server and the many end nodes are a source of channel congestion and high energy consumption. This brief tackles this aspect by introducing Federated Learning with Gradual Layer Freezing (FedGLF), a novel FL scheme that gradually reduces the portion of the model sent back and forth, relieving the communication bundle yet preserving the quality of the training service. The results collected on two image classification tasks learned with different data distributions prove FedGLF outperforms conventional FL schemes, with data volume savings ranging from 14% to 59% or up to 2.5% higher accuracy.**

***Index Terms*—Federated Learning, Internet of Things, Convolutional Neural Networks, Communication, Optimization.**

I. Introduction

THE advancement of cheaper sensing and connecting technologies in the Internet of Things (IoT) brought rapid growth of wireless sensor networks capable of gathering and sharing large volumes of unstructured raw physical data. The availability of more data with higher quality and capillarity empowered the development of Machine Learning (ML) applications built upon Convolutional Neural Networks (CNNs). Nonetheless, a profitable interaction between IoT and ML poses challenges beyond functionality, and data privacy is the most critical one. For instance, security cameras, smart speakers, and smartphones collect sensitive information that end-users might not be willing to share with service providers. Unfortunately, it is a prerogative for any supervised learning strategy to have centralized and fully accessible training data at its disposal, which is in conflict with privacy and ownership rights. A natural yet challenging evolution calls for learning methods that operate the data locally, on the embedded devices deployed at the edge of the IoT.

In this context, Federated Learning (FL) is an emerging paradigm where many clients collaborate in training a global model hosted on a central server without distributing the raw data. Although FL has proven feasible in real-life use cases [1], [2], to manage the limited resource budgets of IoT end-nodes remains an open issue. FL involves frequent upload and download of models to achieve competitive accuracy, generating massive communication volumes that are unaffordable for most embedded devices. The communication burden is a primary concern for typical IoT systems, where download and upload bandwidths are limited to $0.75\,\mathrm{MB/s}$ and $0.25\,\mathrm{MB/s}$, respectively [3]. Moreover, the volume of exchanged data impacts the communication energy [4], with a negative effect on the battery life of the end nodes.

This brief introduces a novel FL scheme for CNNs to reduce the volume of data transferred during the learning process. We propose Federated Learning with Gradual Layer Freezing (FedGLF), a layer gating strategy where the number of layers that need to be trained and then synchronized with the server reduces as the learning flow evolves, letting clients skip massive download/upload of layers that already reached (or that are close to reach) convergence. FedGLF was inspired by observing that shallower layers converge sooner than deeper ones, as reported in the theoretical study in [5], and that layer over-training brings limited (or no) improvements in model accuracy. As main contribution, FedGLF introduces a novel scheme to coordinate the freezing mechanism at the server side depending on the global state of the learning process.

The experimental validation conducted on two tasks for image classification (i.e., CIFAR-10 and CIFAR-100) reveals the effectiveness of FedGLF. First, FedGLF reduces the communication bundle achieving the same accuracy as a standard FL scheme, with savings from 14% to 59% (depending on the accuracy constraint and the data distribution). Second, FedGLF improves the quality of the training procedure achieving up to +2.5% higher accuracy while consuming the same data volume of a standard FL scheme.

II. Related Works

Recent works addressed the optimization of the communication cost in FL. Table I shows a brief taxonomy of the main approaches, emphasizing which synchronization phase (download, upload, or both) benefits from the optimization. For a complete review, interested readers can refer to [3]. In this section, we briefly overview the principal works and their key underlying strategy.

FederatedAveraging (FedAvg) [6] is a de-facto standard for training neural networks under a federated setting. FedAvg plays with the synchronization frequency between the clients and the server with the aim of limiting the volume of exchanged data while ensuring no (or marginal) loss of accuracy. Some variants of FedAvg foresee a dynamic tuning of the synchronization frequency during the learning process [7], eventually following a per-layer basis [8], while others play on the end-node side, by means of an optimal client selection mechanism applied before [9] or after aggregation [10], even

Manuscript received XX, 20XX; revised XX, 20XX; accepted XX, 20XX. Date of publication XXXX XX, 20XX; date of current version XXXX XX, 20XX. This brief was recommended by Associate Editor X. XXXX. (Corresponding author: Andrea Calimera.)

Erich Malan and Andrea Calimera are with the Department of Control and Computer Engineering, Politecnico di Torino, Italy, 10129 (e-mail: erich.malan@polito.it, andrea.calimera@polito.it)
Valentino Peluso and Enrico Macii are with the Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Italy, 10129 (e-mail: valentino.peluso@polito.it, enrico.macii@polito.it).

TABLE I
OVERVIEW OF COMMUNICATION-EFFICIENT FL.

Optimization	Download	Upload	Both
Synchronization			[6] [7] [8]
Clients selection		[9]	[10]
Model Compression	[14]	[12] [13]	[11]
Training and Aggregation			[15] [16] [17]
Layer Freezing			This work

Algorithm 1: Communication Round in FedGLF

```
1 for $l \in [1, L]$ do
2     if $globalTimestamp[l] > localTimestamp[l]$ then
3         Download $\mathcal{W}_l^{r-1}$ from the Server
4 $L_{\min} = \min(\max(1, \lceil \frac{r-K}{F} \rceil + 1), L)$
5 for $epoch \in [1, E]$ do
6     Train $W_{l,i}^r, l \in [L_{\min}, L]$
7 for $l \in [L_{min}, L]$ do
8     Upload $\mathcal{W}_{l,i}^r$ to the Server
```

considering multiple criteria, like memory, energy, and latency. Some alternatives to FedAvg focus on the compression of the local model through techniques like pruning [11], quantization [12], or a combination [13]. The model compression works at the server-side, reducing the downloading communication cost [14]. As a downside, the computational workload increases due to extra model compression and decompression stages, both on the client side and the server side. Other recent works investigated alternative training functions [15], different aggregation methods [16], or a combination of them [17], to reach higher accuracy with fewer communication rounds.

The FedGLF strategy proposed in this work is orthogonal to the aforementioned methods, as its underlying freezing mechanism can be combined with other techniques already in place. As a distinctive feature, FedGLF concurrently optimizes the arithmetic workload and the communication costs (both client-to-server and server-to-client), as the number of arithmetic operations for the back-propagation and the data transmitted to update the model is inversely proportional to the number of frozen layers.

III. System Model & Optimization

A. System Model

In a synchronous FL system, the central server coordinates a set of I embedded devices, the clients. Each client i holds a local dataset $\mathcal{D}_i$ with exclusive access. FL implements an iterative process with multiple interactions between the clients and the server, referred to as communication rounds. Each round r consists of three steps. First, the clients download from the server the global model generated in the previous round $r-1$; the global model consists of L layers, each of them with a set of trainable weights $\mathcal{W}_l^{r-1}$, $l \in [1, L]$. Second, each client trains the model with local data achieving a new local version of the weights set $\mathcal{W}_{l,i}^r$. Third, the server collects the updated models that have been sent back by the clients and it enforces a model aggregation via weight averaging. To alleviate the resource usage, only a fraction of available clients participate in a round r. The training flow implements several rounds until the model reaches convergence above a target accuracy A_{th}.

Considering a standard 32-bit floating-point training (i.e., 4 bytes for each weight), the communication cost $CC_{i,r}$ of the i-th client at the round r is given by:

$$CC_{i,r} = 4 \times \sum_{l=1}^{L} (|\mathcal{W}_l^{r-1}| + |\mathcal{W}_{l,i}^r|) \tag{1}$$

with $|\cdot|$ the number of weights. The two terms in Equation (1), i.e., $|\mathcal{W}_l^{r-1}|$ and $|\mathcal{W}_{l,i}^r|$ refers to the download and upload costs, respectively. In a standard FL framework, the communication cost of a round keeps constant during the whole learning flow as the entire model gets synchronized at each round, and the overall communication cost is the sum of the data volume generated by the clients over all the communication rounds R:

$$CC = \sum_{r=1}^{R} \sum_{i=1}^{I} p_{i,r} \times CC_{i,r} \tag{2}$$

with $p_{i,r} = 1$ if the client i is selected to participate for round r, $p_{i,r} = 0$ otherwise.

B. FedGLF

FedGLF optimizes the communication cost CC under a prediction accuracy constraint A_{th}, resulting in the following minimization problem:

$$\text{minimize } CC \text{ s.t. } A \geq A_{\text{th}} \tag{3}$$

In pursuing this objective, FedGLF implements a heuristic gating scheme that freezes the layers of a CNN incrementally, in topological order, from input to output, after a predefined number of rounds K and with a predefined frequency F. The two hyper-parameters K and F are global state variables set at the server side, and they act as optimization knobs enabling trade-off between accuracy and communication cost.

The Algorithm 1 shows the proposed synchronization and training procedure on the client side. The pseudo-code refers to a generic CNN with L trainable layers and describes the three stages of each communication round r: model download (lines 1–3), model training (lines 5–6), and model upload (7–8). The client downloads the timestamp (*globalTimestamp*) of the last update for each layer of the global model (line 1). If the local version of the layer (*localTimestamp*) is older than that available in the global model (line 2), the client downloads the updated layer's parameters (line 3) from the server. Even though the timestamp is a source of additional information (not required in conventional FL schemes), it counts 8 bytes per layer, a marginal overhead compared to the size of a CNN (from kBs to MBs). The number of layers trained at each iteration changes during the evolution of the learning progress. For $r < K$, i.e., at the first iteration $L_{\min} = 1$, all the layers get involved in the training. This is the initiation phase, when FedGLF works like a conventional FL strategy and the whole model is trained for E epochs and then sent to the server. After the first K rounds (r=K), the first layer, i.e., the input layer, is frozen. This is the beginning of the second iteration ($L_{\min} = 2$, line 4) when the L-1 layers participate. Every F communication rounds, a new layer is frozen in topological order, until only the output layer is trained alone and communicated to the server ($L_{\min} = L$). On the server side, the model versions produced by the clients are aggregated and sent back to the clients upon request. Although FedGLF can work with any training loss and aggregation function, we adopted those proposed by FedAvg for a fair quantitative comparison.

In FedGLF, the communication cost varies over time, depending on the number of frozen layers. Specifically, the FedGLF scheme reduces the communication cost during the downloads, as the client must just synchronize the layers that differ from the global model, and during the uploads, as the clients just send the new learned layers. Moreover, the local training workload reduces with the number of frozen layers.

IV. Experimental Results

This section presents a comparative analysis between the proposed FedGLF and FedAvg, the most common baseline used for the assessment of FL optimizations. We describe the experimental setup, the evaluation metrics adopted, and the collected results.

A. Benchmark, Datasets, and Training

The neural network used as benchmark is a five-layer CNN suited for tiny IoT applications and borrowed from [6]. The network consists of two convolutional (conv) layers with 64 5×5 filters, two fully connected (fc) layers with 394 and 192 neurons, respectively, and a final linear layer that returns the prediction logits. The same CNN was trained for image classification using two different datasets, CIFAR-10 and CIFAR-100 [18]. We applied a standard split for the training and testing set and data augmentation based on random crop followed by random horizontal flip.

We considered different data distributions which reflect two possible federating learning scenarios: (i) independent and identically distributed (IID) and balanced, i.e., each client trains the CNN using samples belonging to all the classes (IID) and all the clients have the same number of samples (balanced); (ii) non-IID and unbalanced, i.e., each client trains the CNN using samples belonging to a subset of classes (non-IID) and the clients have a different number of samples (unbalanced). For non-IID distributions, the splits were generated following a Dirichlet distribution with coefficient 0.3.

The training is distributed across 100 clients with a participation rate of 10% at each round. The on-device training is done using Stochastic Gradient Descent (SGD) with a number of epochs of 5 and batch-size 50. The learning rate follows a polynomial decay schedule with initial value 0.01. We selected the same hyper-parameters in both FedAvg and FedGLF experiments. The training is iterated until the communication cost reaches an upper-bound $60.5\,\mathrm{GB}$, which is equivalent to 1000 communication rounds in FedAvg.

The solutions achievable with FedGLF were explored using different values of K and F, enabling a trade-off analysis in the accuracy vs. communication cost space. Specifically, we opted for a grid-search approach with $K \in \{350, 400, 450, 500\}$ and $F \in \{25, 50, 75\}$.

B. Evaluation metrics

We probed the evolution of the classification accuracy of the global model on the test set and, like in [16], we computed the moving average over a fixed window of 30 communication rounds. The filtering suppresses the oscillations due to intrinsic instability of the training loop enabling a more reliable assessment. Moreover, we measured the communication cost needed by the clients for uploading and downloading the model and we collected the number of communication rounds in both FedAvg and the proposed FedGLF to achieve a predefined level of accuracy within communication upper-bound ($60.5\,\mathrm{GB}$).

C. Results

IID setting. Tables II-III report the comparison between FedAvg and FedGLF on the CIFAR-10 and CIFAR-100 datasets in the IID scenario. The tables summarize the achieved performance for different accuracy thresholds (A_{th}) and the communication cost (CC) in GB. Concerning FedGLF, we reported the setting parameters K and F (column K/F) under which the training reaches the target accuracy with the lowest communication cost. We also reported the number of rounds for the two schemes (R_{AVG} and R_{GLF}).

The resulting trends are similar for both datasets and can be summarized as follows. First, FedGLF outperforms FedAvg achieving the same accuracy at a lower communication cost, with savings ranging from 14.1% to 28.1% for CIFAR-10 and from 23.3% to 39.5% for CIFAR-100. These findings validate the efficiency of the adopted layer freezing strategy. The gradual layer freezing has no impact on accuracy if compared to a full model training, but substantially reduces the volume of exchanged data achieving higher efficiency. Second, for higher target accuracies the communication cost increases for both FedGLF and FedAvg, but FedGLF is by far less data-hungry. To increase the accuracy by 0.5% on CIFAR-10 (CIFAR-100), FedAvg calls for an average communication cost increment of 26.6% (20.6%), while FedGLF calls for a smaller increment of 19.4% (11.8%). This reflects the nature of the two methods: in FedAvg, the only viable option to improve accuracy is to increase the number of rounds, with a constant cost of $62.24\,\mathrm{MB}$ per round, whereas FedGLF offers a new way to reach a better trade-off, namely, the speed of the layers freezing that is controlled with the two hyper-parameters K and F. The results collected on CIFAR-100 reinforce these considerations. To notice that, even though FedGLF requires more rounds than FedAVG ($R_{\text{GLF}} > R_{\text{AVG}}$) for most of the accuracy levels under analysis, the communication cost gets lower ($CC_{\text{GLF}} < CC_{\text{AVG}}$) due to the freezing mechanism. As an example, with $K/F = 450/25$ (third row in Table III), we observed that after 550 rounds the last layer of the network was already trained, lowering the communication cost per round to $0.15\,\mathrm{MB}$. As a side note, the increased number of rounds does not necessarily imply longer training time or worse performance. By reducing the arithmetic workload and the size of the transmitted payloads, FedGLF progressively shortens the training and communication time, hence the overall round duration. Therefore, the shorter duration of the rounds mitigates the larger number of rounds.

TABLE II
COMMUNICATION COST IN GB (CC), NUMBER OF ROUNDS (R) OF FEDAVG (AVG) AND FEDGLF (GLF) UNDER DIFFERENT ACCURACY THRESHOLDS (A_{TH}) FOR CIFAR-10 IN IID SETTING.[1]

A_{th}	CC_{AVG}	CC_{GLF}	K/F	R_{AVG}	R_{GLF}
$\geq 80.0\%$	27.24	**23.40** (-14.1%)	350/25	448	386
$\geq 80.5\%$	33.26	**27.30** (-17.9%)	350/50	547	469
$\geq 81.0\%$	41.16	**30.58** (-25.7%)	400/50	677	593
$\geq 81.5\%$	55.14	**39.64** (-28.1%)	500/75	907	753

TABLE III
COMMUNICATION COST IN GB (CC), NUMBER OF ROUNDS (R) OF FEDAVG (AVG) AND FEDGLF (GLF) UNDER DIFFERENT ACCURACY THRESHOLDS (A_{TH}) FOR CIFAR-100 IN IID SETTING.[1]

A_{th}	CC_{AVG}	CC_{GLF}	K/F	R_{AVG}	R_{GLF}
$\geq 40.0\%$	32.91	**25.23** (-23.3%)	350/25	530	507
$\geq 40.5\%$	39.74	**26.17** (-34.1%)	350/25	640	1155
$\geq 41.0\%$	47.19	**32.54** (-31.0%)	450/25	760	1368
$\geq 41.5\%$	57.68	**34.92** (-39.5%)	450/50	929	968

Non-IID setting. FedGLF achieves larger savings in the non-IID scenario: up to 59.0% for CIFAR-10 and 54.8% for CIFAR-100 (Tables IV-V). Moreover, FedGLF reaches higher accuracy levels than FedAvg; in some cases FedAvg cannot reach the target accuracy within the communication budget (60.5 GB). As general trend, the training on non-IID data calls for more rounds than the IID case, limiting FedAvg. FedGLF ensures better performance instead, reaching higher accuracy while consuming fewer data. These gains are achievable controlling the layer freezing rate through K and F. On CIFAR-10, FedGLF reaches 78.0% accuracy with 39.82 GB, while FedAvg reaches a mere 74.5% consuming more data, 45.35 GB. Similar trends hold for CIFAR-100, where FedGLF is up to 2.5% more accurate with comparable communication cost (38.73 GB vs. 34.77 GB). Those results confirm the sub-optimality of standard FL schemes that work over all the layers during the entire learning process, promoting FedGLF as a more efficient approach.

Limitations and future directions. One open issue for FedGLF is how to estimate the values of K and F to minimize the communication cost under a given accuracy constraint. Nonetheless, the grid search adopted in this work reveals some useful insight. First, lower values of K and F can be used to prioritize solutions with fewer communication costs and slightly lower accuracy, while higher values of K and F enable higher accuracy, but a larger communication overhead. Second, K represents a coarse-grain knob, whereas F allows a finer control in the accuracy vs. communication cost space. Future works will explore optimal layer freezing policies.

V. CONCLUSIONS

This letter presented FedGLF, a distributed training strategy based on gradual layer freezing that reduces the communication cost of conventional FL. By tuning two simple algorithmic control knobs, the FedGLF schedule offers multiple solutions in the accuracy vs. cost space, which outperform FedAvg in terms of attainable accuracy and communication costs.

TABLE IV
COMMUNICATION COST IN GB (CC), NUMBER OF ROUNDS (R) OF FEDAVG (AVG) AND FEDGLF (GLF) UNDER DIFFERENT ACCURACY THRESHOLDS (A_{TH}) FOR CIFAR-10 IN NON-IID SETTING.[1]

A_{th}	CC_{AVG}	CC_{GLF}	K/F	R_{AVG}	R_{GLF}
$\geq 74.5\%$	45.35	**24.58** (-45.8%)	350/25	746	528
$\geq 75.0\%$	51.19	**24.59** (-52.0%)	350/25	842	603
$\geq 75.5\%$	60.13	**24.65** (-59.0%)	350/25	989	986
$\geq 76.0\%$	-	**24.7**	350/25	-	1362
$\geq 76.5\%$	-	**27.7**	350/50	-	1525
$\geq 77.0\%$	-	**30.65**	350/75	-	1520
$\geq 77.5\%$	-	**33.76**	450/50	-	1524
$\geq 78.0\%$	-	**39.82**	500/75	-	1984

TABLE V
COMMUNICATION COST IN GB (CC), NUMBER OF ROUNDS (R) OF FEDAVG (AVG) AND FEDGLF (GLF) UNDER DIFFERENT ACCURACY THRESHOLDS (A_{TH}) FOR CIFAR-100 IN NON-IID SETTING.[1]

A_{th}	CC_{AVG}	CC_{GLF}	K/F	R_{AVG}	R_{GLF}
$\geq 40.0\%$	34.77	**25.25** (-27.4%)	350/25	560	522
$\geq 40.5\%$	42.90	**25.68** (-40.1%)	350/25	691	815
$\geq 41.0\%$	47.69	**26.19** (-45.1%)	350/25	768	1169
$\geq 41.5\%$	59.30	**26.81** (-54.8%)	350/25	955	1601
$\geq 42.0\%$	-	**29.81**	350/50	-	1623
$\geq 42.5\%$	-	**38.73**	500/50	-	1495

REFERENCES

[1] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.
[2] S. Ramaswamy *et al.*, "Federated learning for emoji prediction in a mobile keyboard," *arXiv preprint arXiv:1906.04329*, 2019.
[3] J. Wang *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.
[4] J. Huang *et al.*, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 225–238.
[5] M. Raghu *et al.*, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *NIPS*, 2017.
[6] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
[7] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.
[8] Y. Chen *et al.*, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 10, pp. 4229–4238, 2019.
[9] W. Luping *et al.*, "Cmfl: Mitigating communication overhead for federated learning," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*. IEEE, 2019, pp. 954–964.
[10] S. AbdulRahman *et al.*, "Fedmccs: Multicriteria client selection model for optimal iot federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4723–4735, 2020.
[11] F. Sattler *et al.*, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.
[12] J. Xu *et al.*, "Ternary compression for communication-efficient federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[13] J. Mills *et al.*, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.
[14] M. Asad *et al.*, "Fedopt: towards communication efficiency and privacy preservation in federated learning," *Applied Sciences*, vol. 10, no. 8, p. 2864, 2020.
[15] T. Li *et al.*, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, I. Dhillon *et al.*, Eds., vol. 2, 2020, pp. 429–450.
[16] S. J. Reddi *et al.*, "Adaptive federated optimization," in *International Conference on Learning Representations*, 2021.
[17] A. E. Durmus *et al.*, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.
[18] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," 2009.

[1]Best results in bold. A dash indicates the target accuracy is not met.