

Federated Learning with Communication Delay in Edge Networks

Frank Po-Chen Lin, Christopher G. Brinton, Nicolò Michelusi
School of Electrical and Computer Engineering, Purdue University, USA
Email: {lin1183,cgb,michelusi}@purdue.edu

Abstract—Federated learning has received significant attention as a potential solution for distributing machine learning (ML) model training through edge networks. This work addresses an important consideration of federated learning at the network edge: communication delays between the edge nodes and the aggregator. A technique called *FedDelAvg* (federated delayed averaging) is developed, which generalizes the standard federated averaging algorithm to incorporate a weighting between the current local model and the delayed global model received at each device during the synchronization step. Through theoretical analysis, an upper bound is derived on the global model loss achieved by *FedDelAvg*, which reveals a strong dependency of learning performance on the values of the weighting and learning rate. Experimental results on a popular ML task indicate significant improvements in terms of convergence speed when optimizing the weighting scheme to account for delays.

Index Terms—Federated learning, edge intelligence, distributed machine learning, convergence analysis, edge-cloud computing

I. INTRODUCTION

Rapid developments of communications technology in conjunction with machine learning (ML) algorithms has resulted in an exponential rise in data generated by user devices [1]. A paradigm shift is occurring from the conventional cloud computing architecture to a hybrid cloud-edge model where ML data processing is carried out on edge devices for many applications, particularly latency-sensitive tasks [2].

Federated learning (FL) has emerged as a promising solution for distributing the training of ML models across edge devices. FL techniques, such as the popular *FedAvg* algorithm [3], generally consist of three steps repeated in sequence: (i) several iterations of parallel, local model training at each device using their own local datasets, (ii) aggregation of the local models at an edge server into a single, global model, and (iii) synchronization of the local models at each device with this global model.

Several research efforts have been conducted on federated learning in recent years to address challenges such as reducing communication overhead and analyzing convergence rates. In this paper, we consider another important aspect of FL that arises in edge networks: communication delays between the edge devices and server performing the aggregations. Our proposed algorithm, *FedDelAvg*, serves the dual purpose of quantifying the impact of such delays and optimizing model performance in their presence.

1) *Related work*: We divide related work on federated learning into two categories: studies on (i) reducing communication bandwidth requirements and (ii) obtaining model

convergence bounds. For a recent, comprehensive survey of works on FL, see [4].

Reducing communication requirements. Edge networks can have unreliable communication environments (e.g., variable wireless connections), which can impact distributed ML techniques. For this reason, [3], [5] proposed methods to reduce the number of upstream and downstream communication rounds required in FL. Other efforts have focused on reducing communication demand per round; in particular, [6]–[8] proposed gradient compression methods to reduce the bandwidth required in each transmission. Recently, [9] proposed a network-aware FL architecture which trades off communication demand with model convergence.

Model convergence bounds. Other work on FL has studied model convergence under different data distributions and local update models. [10] showed that the error bound of FL in the case of non-Independent and Identically Distributed (non-i.i.d.) data samples is worse than i.i.d., while [11] analyzed the convergence of the *FedAvg* algorithm, suggesting conditions on the learning rate to achieve the optimum. [12] discovered that training may converge faster with batch gradient descent if all devices participate throughout the training process. [13] analyzed the convergence bound of FL in the presence of edge network resource constraints, and [14] considered how wireless channel characteristics impact convergence.

The above-mentioned works do not consider the effect of network delays. In practice, delays between the edge devices and the cloud are non negligible – usually from hundreds of milliseconds to several seconds depending on the network bandwidth [6] – which might severely degrade the performance of FL schemes. This aspect is the focus of our work.

2) *Outline of contributions*: We propose *FedDelAvg*, a novel technique that adapts FL in the presence of network delays. Specifically, we develop a new algorithm for the synchronization step that combines local and global models to account for the effects of delay (Section II). Then, we characterize convergence of *FedDelAvg* and provide suggestions for the optimal weight used in the synchronization phase (Section III). Finally, the delay-robustness of *FedDelAvg* on convergence speed is demonstrated numerically when the synchronization weighting is adjusted for delay (Section IV).

II. FedDelAvg: FEDERATED DELAYED AVERAGING

In this section, we introduce the federated learning system model, the machine learning task model, and develop

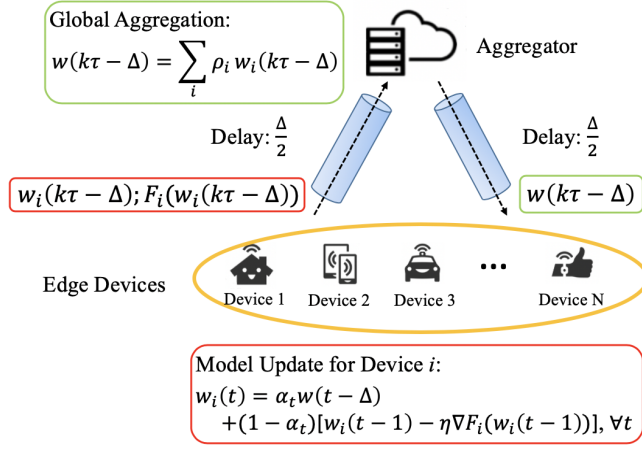


Fig. 1: System architecture of FedDelAvg. Model synchronization combines global and local models to account for communication delays.

FedDelAvg, our federated delayed averaging algorithm.

A. Edge Network Model

The federated learning (FL) system architecture consists of a single edge server and N edge devices indexed by $i = 1, 2, \dots, N$, as shown in Fig.1. The edge devices collect data and perform local updates to optimize a loss function $F(\cdot)$ corresponding to a machine learning task (described next). The edge server (the cloud) plays the role of an aggregator, collecting the locally trained parameters \mathbf{w}_i and the evaluations of the local loss functions $F_i(\mathbf{w}_i)$ from the edge devices to perform a global update. Local updates are taken to be gradient descent steps on the local loss functions $F_i(\mathbf{w})$, while global updates are constituted of aggregation followed by synchronization. Aggregation denotes the computation of a global model obtained using the weighted average of local models, while synchronization represents the update of local models at the edge after aggregation [9].

In an edge network, the aggregation of the local model parameters at the cloud followed by the synchronization at the edge incurs communication delay, which we aim to model in our formulation.

B. Machine Learning Model

1) *Data structure*: Each device i carries a dataset \mathcal{D}_i with $D_i = |\mathcal{D}_i|$ data points. Each data point $(\mathbf{x}, y) \in \mathcal{D}_i$ consists of an m -dimensional feature vector $\mathbf{x} \in \mathbb{R}^m$ and a label $y \in \mathbb{R}$.

2) *Loss function*: We let $f_i(\mathbf{x}, y; \mathbf{w})$ be the loss associated with the data point $(\mathbf{x}, y) \in \mathcal{D}_i$ based on a model parameter vector $\mathbf{w} \in \mathbb{R}^m$. For instance, in linear regression, the loss function is the squared error $f_i(\mathbf{x}, y; \mathbf{w}) = \frac{1}{2}(y - \mathbf{w}^T \mathbf{x})^2$. We define the loss function across the local dataset \mathcal{D}_i as

$$F_i(\mathbf{w}) = \frac{1}{D_i} \sum_{(\mathbf{x}, y) \in \mathcal{D}_i} f_i(\mathbf{x}, y; \mathbf{w}), \quad (1)$$

and the global loss function across all nodes can then be expressed as

$$F(\mathbf{w}) = \sum_{i=1}^N \rho_i F_i(\mathbf{w}), \quad (2)$$

where $\rho_i = D_i / \sum_j D_j$ is the weight associated with the i th node, proportional to the size of the local dataset.

3) *Learning objective*: The goal of the machine learning task is to find a \mathbf{w}^* that minimizes $F(\mathbf{w})$, i.e.,

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} F(\mathbf{w}). \quad (3)$$

To aid our analysis in Section III, we make a few standard assumptions [3] on the local loss functions F_i .

Assumption 1. F_i is continuously differentiable, convex, L -Lipschitz and β -smooth, implying that

$$|F_i(\mathbf{w}_1) - F_i(\mathbf{w}_2)| \leq L \|\mathbf{w}_1 - \mathbf{w}_2\|, \quad (4)$$

$$\|\nabla F_i(\mathbf{w}_1) - \nabla F_i(\mathbf{w}_2)\| \leq \beta \|\mathbf{w}_1 - \mathbf{w}_2\|, \quad (5)$$

where $L \in [0, \infty)$ and $\beta \in [0, \infty)$ are the Lipschitz and smoothness constants, respectively.

By (2), Assumption 1 holds for the global loss function F as well. One example of F_i that satisfies Assumption 1 is the well-known logistic regression loss.

We now state an important assumption on the dissimilarity of data [4], [13] at the edge, in addition to Assumption 1:

Assumption 2. The gradients of the local and global loss functions exhibit a similarity of

$$\|\nabla F_i(\mathbf{w}) - \nabla F(\mathbf{w})\| \leq \delta_i, \quad \forall \mathbf{w}, \quad (6)$$

where $\delta_i \geq 0$ is the dissimilarity parameter for node i .

We let $\delta = \sum_i \rho_i \delta_i$ be the average data dissimilarity across the edge network.

4) *Centralized gradient descent*: Loss functions are typically minimized by gradient descent (GD) iterations. In a centralized case, where the global loss F can be optimized directly, this is defined as

$$\mathbf{w}_{GD}(t) = \mathbf{w}_{GD}(t-1) - \eta \nabla F(\mathbf{w}_{GD}(t-1)), \quad (7)$$

where $\mathbf{w}_{GD}(0)$ is an initialization, $t \geq 1$ is the iteration index, and $\eta > 0$ is the learning rate. Under Assumption 1 with $\eta \leq \frac{1}{\beta}$, $F(\mathbf{w}_{GD}(t))$ is guaranteed to converge to $F(\mathbf{w}^*)$ with rate $\mathcal{O}(1/T)$ under gradient descent, where T is the number of iterations [15].

However, centralized gradient descent cannot be directly applied to the FL framework in Fig.1 since no device has direct access to all the data. In addition, communication to the cloud is costly in terms of network resources, so the aggregation and synchronization processes can only be done periodically. Finally, communication delay between edge and cloud is usually non-negligible, which we address next in developing the FedDelAvg algorithm.

C. FedDelAvg Algorithm

In FedDelAvg, i.e., Federated Delayed Averaging, the effect of communication delay between edge and cloud on learning performance is incorporated into the design of the FL system. We divide the learning process into discrete time intervals $t \in \{1, 2, \dots, T\}$, where the duration between two consecutive aggregations is denoted as τ . The communication delay between the time when edge devices send their updates to the cloud and the resulting synchronization is denoted Δ , where $\tau \geq \Delta \geq 0$. In Fig.1, we assume a symmetric delay of $\tau/2$ upstream and downstream.

1) *Distributed gradient descent incorporating delay*: We let $\mathbf{w}_i(t)$ be the local model parameter vector of edge device i at time t , initialized as $\mathbf{w}_i(-\Delta) = \mathbf{w}(-\Delta)$ at time $-\Delta$ across all devices. Let

$$\mathbf{w}(t) = \sum_i \rho_i \mathbf{w}_i(t) \quad (8)$$

be the weighted average of the parameter vectors across the edge devices at t . To compute it, each device i sends its local $\mathbf{w}_i(t)$ at times $t \in \{k\tau - \Delta, \forall k \geq 0\}$; these local models are received at the aggregator with delay $\Delta/2$ to compute $\mathbf{w}(t)$; $\mathbf{w}(t)$ is then sent back to the nodes with delay $\Delta/2$, so that each edge device receives it at time $t + \Delta$. We partition the time interval $(-\Delta, T - \Delta]$ of duration T into $K = T/\tau$ periods, each of duration τ (without loss of generality, we assume T is an integer multiple of τ).

Consider the k th period, $k \in \{0, \dots, K-1\}$, spanning the time interval $\mathcal{T}_k = \{k\tau - \Delta + 1, \dots, (k+1)\tau - \Delta\}$. At each time $t \in \mathcal{T}_k \setminus \{k\tau\}$, each edge device performs local GD updates as

$$\mathbf{w}_i(t) = \mathbf{w}_i(t-1) - \eta \nabla F_i(\mathbf{w}_i(t-1)). \quad (9)$$

At time $t = k\tau$ (synchronization), each edge device receives the delayed global parameter vector $\mathbf{w}(t - \Delta)$ from the cloud. To update the local model, each device first performs a local GD update, followed by a weighted average between the local and global variables; mathematically, at time $t = k\tau$,

$$\begin{aligned} \mathbf{w}_i(t) = & \alpha \mathbf{w}(t - \Delta) \\ & + (1 - \alpha) [\mathbf{w}_i(t-1) - \eta \nabla F_i(\mathbf{w}_i(t-1))], \end{aligned} \quad (10)$$

where $\alpha \in [0, 1]$ is a weight parameter weighting the local vs. global updates. Note that, when $\alpha = 1$ and $\Delta = 0$, we obtain [3] as a special case. Letting

$$\alpha_t = \begin{cases} \alpha, & t = k\tau, \exists k \in \{0, 1, \dots, K-1\} \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

we can then define the updates at all times $t \in \{-\Delta + 1, \dots, T - \Delta\}$ as

$$\begin{aligned} \mathbf{w}_i(t) = & \alpha_t \sum_j \rho_j \mathbf{w}_j(t - \Delta) \\ & + (1 - \alpha_t) [\mathbf{w}_i(t-1) - \eta \nabla F_i(\mathbf{w}_i(t-1))]. \end{aligned} \quad (12)$$

Since edge devices send their local parameters $\mathbf{w}_i(t)$ and the corresponding local loss functions $F_i(\mathbf{w}_i(t))$ to the cloud at

Algorithm 1: Federated Delayed Averaging

Input: α_t, τ, N, T

Output: \mathbf{w}^K

Initialize $\mathbf{w}_i(-\Delta), \forall i$;

for $k = 0 : K - 1$ **do**

for $t = k\tau - \Delta + 1 : (k+1)\tau - \Delta$ **do**

 For each edge device $i \in 1, 2, \dots, N$ in parallel,
 update local model with (12);

if $t = (k+1)\tau - \Delta$ **then**

 Each edge device i send local parameters
 \mathbf{w}_i and $F_i(\mathbf{w}_i)$ to the cloud;

else if $t = (k+1)\tau - \Delta/2$ **then**

 Compute $\mathbf{w}((k+1)\tau - \Delta)$ with (8) and
 send it to the edge for synchronization;

 Update \mathbf{w}^K with (13);

end

end

$t = k\tau - \Delta$, the cloud has access to the global model $\mathbf{w}(t)$ and $F(\mathbf{w}(t))$ at times $t = k\tau - \Delta$. Then, the final model parameter chosen from FedDelAvg, after K global aggregations, is

$$\mathbf{w}^K = \arg \min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w}), \quad (13)$$

where $\mathcal{W} \equiv \{\mathbf{w}(k\tau - \Delta), k = 0, 1, \dots, K-1\}$.

The full FedDelAvg algorithm is summarized in Alg.1.

III. CONVERGENCE ANALYSIS OF FedDelAvg

In this section, we study the convergence of FedDelAvg in terms of the optimality gap $F(\mathbf{w}^K) - F(\mathbf{w}^*)$ between the global objective function at the algorithm output \mathbf{w}^K and at the globally optimal parameter vector \mathbf{w}^* .

Definition 1. For each period $k \in \{0, \dots, K-1\}$, we define $\mathbf{c}_k(t)$ as the centralized gradient descent algorithm during the time interval $t \in \mathcal{T}_k$, i.e.,

$$\mathbf{c}_k(t) = \mathbf{c}_k(t-1) - \eta \nabla F(\mathbf{c}_k(t-1)), \quad (14)$$

initialized as $\mathbf{c}_k(k\tau - \Delta) = \mathbf{w}(k\tau - \Delta)$.

Note that this is the delayed counterpart of $\mathbf{v}_{[k]}(t)$ defined in [13] for the case $\Delta = 0$.

The main result is stated in Theorem 1, which upper bounds the optimality gap under delay.

Theorem 1. Under Assumption 1 and with $\eta \leq \frac{1}{\beta}$,

$$\begin{aligned} & F(\mathbf{w}^K) - F(\mathbf{w}^*) \\ & \leq \frac{1}{2\eta\phi T} + \sqrt{\frac{1}{4\eta^2\phi^2 T^2} + \frac{L\Psi(\alpha)}{\eta\phi T}} + L\psi(\alpha, K), \end{aligned} \quad (15)$$

where

$$\Psi(\alpha) \triangleq \sum_{k=1}^K \psi(\alpha, k) \quad (16)$$

$$= K\psi(\alpha, \infty) - [(1 + \eta\beta)^\tau - 1] \frac{(1 - \alpha)^2}{\alpha} \epsilon^{(K)},$$

$$\psi(\alpha, k) \triangleq (1 - \alpha)\epsilon^{(k)}[(1 + \eta\beta)^\tau - 1] + (1 - \alpha)h(\tau) + \alpha h(\tau - \Delta) + \alpha\eta\Delta L(1 + \eta\beta)^{\tau-\Delta}, \quad (17)$$

$$\phi \triangleq \omega \left(1 - \frac{\beta\eta}{2}\right), \quad (18)$$

$$\omega \triangleq \frac{1}{\max_{k \in \{0, \dots, K-1\}} \|c_k(k\tau - \Delta) - \mathbf{w}^*\|^2}, \quad (19)$$

$$h(x) \triangleq \frac{\delta}{\beta}[(1 + \eta\beta)^x - 1] - \eta\delta x, \quad (20)$$

$$\epsilon^{(k)} \triangleq [1 - (1 - \alpha)^k]2\eta L(\tau/\alpha - \Delta). \quad (21)$$

Proof. To derive (15), consider $\eta \leq \frac{1}{\beta}$ and let $\xi^* > 0$ be defined such that $T\eta\phi - \frac{L\Psi(\alpha)}{\xi^{*2}} > 0$ and

$$\xi^* = \frac{1}{T\eta\phi - \frac{L\Psi(\alpha)}{\xi^{*2}}}. \quad (22)$$

Solving, we obtain

$$\xi^* = \frac{1}{2\eta\phi T} + \sqrt{\frac{1}{4\eta^2\phi^2 T^2} + \frac{L\Psi(\alpha)}{\eta\phi T}} \quad (23)$$

(which indeed satisfies $T\eta\phi - \frac{L\Psi(\alpha)}{\xi^{*2}} > 0$). Now, let $\xi > \xi^*$, and assume that, under such ξ , the conditions of Proposition 1 in Appendix A are all satisfied. Then, it follows that

$$F(\mathbf{w}((K+1)\tau - \Delta)) - F(\mathbf{w}^*) < \xi^* < \xi. \quad (24)$$

In other words, this shows a contradiction with condition (4) of Proposition 1. Therefore, at least one of the conditions of Proposition cannot be satisfied, for any $\xi > \xi^*$. Conditions (1) and (2) are clearly satisfied since $\eta \leq 1/\beta$ and $\xi > \xi^* > 0$. Therefore, either conditions (3) or (4) are violated, implying that

$$\begin{aligned} & \min\{F(\mathbf{w}((K+1)\tau - \Delta)), \min_k F(c_k((k+1)\tau - \Delta))\} \\ & \leq F(\mathbf{w}^*) + \xi^*. \end{aligned} \quad (25)$$

Using Lemma 1 in Appendix A, we have that

$$F(\mathbf{w}((k+1)\tau - \Delta)) \leq F(c_k((k+1)\tau - \Delta)) + L\psi(\alpha, K),$$

($\psi(\alpha, k)$ is increasing in k) so that

$$\begin{aligned} & \min_k F(c_k((k+1)\tau - \Delta)) \\ & \geq \min_k \{F(\mathbf{w}((k+1)\tau - \Delta)) - L\psi(\alpha, K)\}, \end{aligned}$$

and (25) implies

$$\min_{k \leq K} \{F(\mathbf{w}((k+1)\tau - \Delta))\} - L\psi(\alpha, K) - F(\mathbf{w}^*) \leq \xi^*.$$

The result of the theorem then directly follows. \square

Due to space constraints, we are abbreviating some details in the steps of this proof. For a more detailed version of the proof, please refer to our online technical report [16].

Note that the upper bound of (15) is a function of ω , which depends on the outcome of Algorithm 1. The usefulness of this bound requires that ω does not converge to 0 when $K \rightarrow \infty$, i.e., $\|c_k(k\tau - \Delta) - \mathbf{w}^*\|^2$ does not diverge with k . We verified through numerical analysis that $\|c_k(k\tau - \Delta) - \mathbf{w}^*\|^2$ decreases as k increases which implies that ω does not increase over k . However, the fact that ω depends on the outcome of the algorithm makes it hard to be analyzed analytically. Since this issue also arises in [13], it motivates future work on investigating upper bounding (15) with terms independent of the algorithm operation.

Theorem 1 demonstrates that the performance of FedDelAvg under communication delay is strongly dependent on the learning rate η and on the value of the weighting α used in the synchronization phase, indicating that these algorithm parameters should be carefully selected with respect to the communication delay in a given learning environment. One way to design α and η is to minimize the asymptotic optimality gap, achieved in the limit $T \rightarrow \infty$. In this case, we obtain

$$\lim_{K \rightarrow \infty} F(\mathbf{w}^K) - F(\mathbf{w}^*) \leq \sqrt{\frac{L}{\eta\phi\tau}} \sqrt{\psi(\alpha, \infty)} + L\psi(\alpha, \infty), \quad (26)$$

where

$$\begin{aligned} \psi(\alpha, \infty) &= (1 - \alpha)2\eta L(\tau/\alpha - \Delta)[(1 + \eta\beta)^\tau - 1] \\ &+ (1 - \alpha)h(\tau) + \alpha h(\tau - \Delta) + \alpha\eta\Delta L(1 + \eta\beta)^{\tau-\Delta}. \end{aligned} \quad (27)$$

Note that (26) is increasing in $\psi(\alpha, \infty)$. Therefore, the optimal value of α is the minimizer of $\psi(\alpha, \infty)$ in (27). When the delay is negligible ($\Delta = 0$), we obtain

$$\begin{aligned} \psi(\alpha, \infty) &= h(\tau) + (\alpha^{-1} - 1)2\eta L\tau[(1 + \eta\beta)^\tau - 1] \\ &\geq \psi(1, \infty), \end{aligned} \quad (28)$$

and therefore the asymptotic optimality gap in (26) is minimized by choosing $\alpha = 1$. In this case, we obtain the FedAvg algorithm derived in [13, Theorem 2] as a special case of our analysis. Intuitively, $\alpha = 1$ is optimal for this special case because the global model obtained by the cloud at $t = k\tau$ is built based on the weighted average of up-to-date local models.

Our analysis generalizes that in [13, Theorem 2] by incorporating communication delay, and by allowing $\alpha \in (0, 1]$ in the synchronization phase. When $\Delta > 0$, the term $\psi(\alpha, \infty)$ is an increasing function of $\alpha \in (0, 1]$ iff

$$\alpha^2 \zeta \geq 2\eta L\tau[(1 + \eta\beta)^\tau - 1],$$

where

$$\begin{aligned} \zeta &= \eta\Delta L[2(1 + \eta\beta)^\tau + (1 + \eta\beta)^{\tau-\Delta} - 2] \\ &- \frac{\delta}{\beta}(1 + \eta\beta)^{\tau-\Delta}[(1 + \eta\beta)^\Delta - 1] + \eta\delta\Delta. \end{aligned}$$

Note that, if

$$\delta \geq \eta\beta L \frac{\Delta(1 + \eta\beta)^{\tau-\Delta} - 2[(1 + \eta\beta)^{\tau} - 1](\tau - \Delta)}{(1 + \eta\beta)^{\tau} - (1 + \eta\beta)^{\tau-\Delta} - \eta\beta\Delta},$$

then $\psi(\alpha, \infty)$ is a decreasing function of α , minimized at $\alpha = 1$. This result demonstrates that, if data dissimilarity δ among various edge devices is larger than a certain threshold, the global parameters across the overall federated system dominates the learning performance such that $\alpha = 1$ becomes the optimum. The large dissimilarity reduces the importance of any particular local model since it becomes harder for any of them to truly reflect the characteristics of the overall data. Only by gathering different local models across the network can the learning system form a representative whole for all data participated in the training. Otherwise, the optimal $\alpha \in (0, 1]$ is

$$\alpha = \sqrt{\frac{2\eta L\tau[(1 + \eta\beta)^{\tau} - 1]}{\zeta}}.$$

In Theorem 1, notice that $F(\mathbf{w}^K)$ does not converge to the optimum as T increases to infinity. This is due to the fact that the model parameters obtained by FedDelAvg with any fixed learning rate η will converge to a sub-optimal point. Reference [11] proved that the decay of learning rate in each training iteration is necessary for FedAvg to converge even when assuming the loss function to be strongly convex and smooth. We leave the algorithm design and convergence analysis for this more general case for future work.

IV. EXPERIMENTAL EVALUATION

To verify our theoretical results, we conduct numerical experiments to examine the effect of communication delay on the convergence of federated learning. The simulation is carried out using the TensorFlow Federated (TFF) framework [17]. Considering the fact that cloud would only have access to the global model at $t = k\tau - \Delta$ for all k when the delay between edge and cloud are carefully considered, we evaluate the averaged model Δ iterations before each global synchronization on the corresponding global loss function.

We consider a federated learning system with $N = 10$ edge devices. The number of local update steps between two global aggregations is set to $\tau = 10$, communication delay is set to $\Delta = 9$, and the total number of global aggregation steps is set to $K = 100$.

Dataset. The MNIST dataset [18] containing 70K images (60K for training and 10K for testing) of hand-written digits is considered in the simulation. We distribute the dataset among the edge devices in a manner such that each obtains a subset corresponding to a specific writer. Since each writer has a unique writing style, the data exhibits dissimilarity ($\delta > 0$) among devices (data dissimilarity is referred to as “non-iid” in the literature, see [4]).

ML model. We consider a generic multinomial logistic regression machine learning model to predict the label of each image out of $s = 10$ possible classes. The cross-entropy

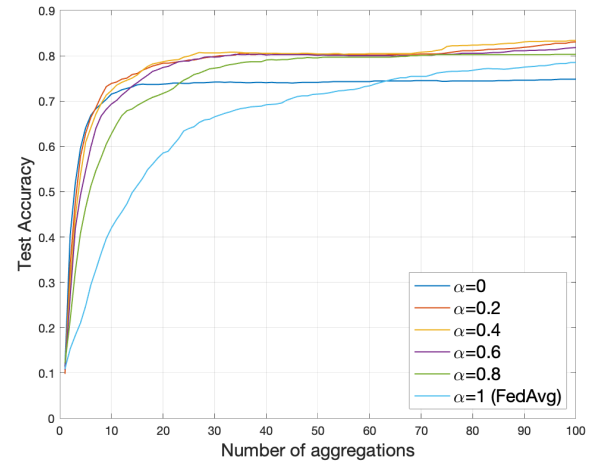


Fig. 2: Accuracy with respect to different α , for a fixed $\Delta = 9$. Selection of α strongly affects rate at which accuracy converges.

loss $f_i(\mathbf{x}, y; \mathbf{w}) = -\sum_{j=1}^s \chi(y = j) \log \left[e^{\mathbf{w}_j^T \mathbf{x}} \left(\sum_{l=1}^s e^{\mathbf{w}_l^T \mathbf{x}} \right)^{-1} \right]$, which satisfies Assumptions 1 and 2, is applied as the loss function at each edge device. During the local updating process, each edge device performs gradient descent with full batch size and a fixed learning rate $\eta = 0.02$.

We study the effects of the two key parameters – the weighting α which we control, and the communication delay Δ which is an artifact of the system – on Federated Delayed Average Learning. The convergence of the testing accuracy, defined as the fraction of classes predicted correctly relative to the total number of predictions, is demonstrated with respect to different values of α and Δ in Figs. 2&3.

Fig. 2 depicts the improvement in testing accuracy by aggregation for different values of α . The convergence speed varies with respect to different values of α . Compared with FedAvg ($\alpha = 1$), FedDelAvg gains the best improvement with $\alpha = 0.2$: the model reaches an accuracy of roughly 80% in 78% fewer training iterations.

Observing the best selection of α in Fig. 2, we set $\alpha = 0.2$ and compare FedDelAvg with FedAvg when delay is either negligible or non-negligible ($\Delta = 0$ and $\Delta = 9$) in terms of accuracy. As shown in Fig. 3, when communication delay between edge and cloud is negligible ($\Delta = 0$), FedAvg obtains the best performance in terms of convergence rate, verifying our conclusion in Theorem 1 that $\alpha = 1$ is the optimum when $\Delta = 0$. To further demonstrate the delay robustness of FedDelAvg, we set the performance of FedAvg with no delay ($\Delta = 0$) as benchmark and compare it with FedDelAvg under delay ($\Delta = 9$) when α is optimized. We observe that, even when the delay is non-negligible, FedDelAvg achieves an accuracy of 80% while only requiring 10% extra training iterations compared with the benchmark. After 100 aggregations, FedDelAvg achieves an accuracy within 3% of the benchmark, whereas FedAvg has a severely degraded performance, thus demonstrating the delay-robustness of the proposed algorithm.

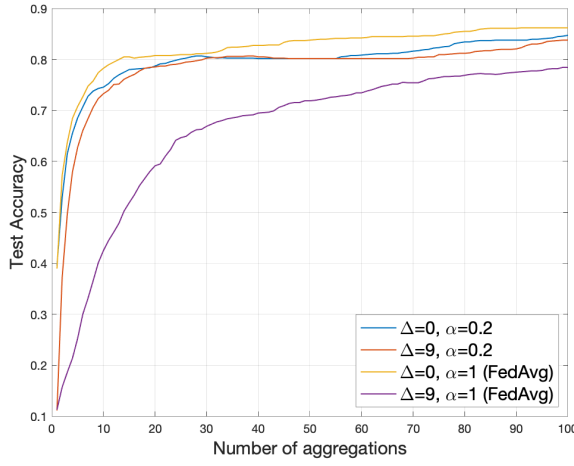


Fig. 3: Accuracy with respect to different communication delays. FedDelAvg under delay is able to obtain a similar convergence rate to the case of no delay when α is tuned.

V. CONCLUSION

This paper proposed FedDelAvg, a federated averaging algorithm that incorporates communication delay in edge networks. Analysis of the convergence bound of FedDelAvg was conducted with respect to dissimilarities of local data at each device. The experimental results demonstrated the impact of delays on federated learning and the delay-robustness of FedDelAvg. Overall, we found that the global model converges significantly faster when the synchronization weighting is optimized for the delay compared with existing FL algorithms where local models are not considered during synchronization.

APPENDIX A

AUXILIARY RESULTS IN THE PROOF OF THEOREM 1

In order to prove Theorem 1, we introduce several properties of FedDelAvg through a supporting lemma and proposition. The statements and proofs of these are provided in this section, with some steps abbreviated due to space constraints. For a more detailed version of the proofs, please refer to our online technical report [16].

Proposition 1. *Let*

$$\omega = \frac{1}{\max_{k \in \{0, \dots, K-1\}} \|\mathbf{c}_k(k\tau - \Delta) - \mathbf{w}^*\|^2}.$$

Under Assumption 1, and if the following conditions are satisfied,

- 1) $\eta \leq \frac{1}{\beta}$
- 2) $T\eta\phi - \frac{L\Psi(\alpha)}{\xi^2} > 0$
- 3) $F(\mathbf{c}_k((k+1)\tau - \Delta)) - F(\mathbf{w}^*) \geq \xi$ for all k
- 4) $F(\mathbf{w}((K+1)\tau - \Delta)) - F(\mathbf{w}^*) \geq \xi$

for some $\xi > 0$, the convergence upper bound of FedDelAvg is

$$F(\mathbf{w}((K+1)\tau - \Delta)) - F(\mathbf{w}^*) \leq \frac{1}{T\eta\phi - \frac{L\Psi(\alpha)}{\xi^2}}, \quad (29)$$

where $\Psi(\alpha)$ and ϕ are defined in Theorem 1.

Proof. First, note that, if $\omega = \infty$, i.e., $\mathbf{c}_k(k\tau - \Delta) = \mathbf{w}^*, \forall k$, then $\mathbf{w}((K+1)\tau - \Delta) = \mathbf{c}_{[K+1]}((K+1)\tau - \Delta) = \mathbf{w}^*$, hence $F(\mathbf{w}((K+1)\tau - \Delta)) = F(\mathbf{w}^*)$. Now, let us consider the case $\omega < \infty$. For every interval k and $t \in [k\tau - \Delta, (k+1)\tau - \Delta]$, we define the sub-optimality gap of the centralized GD scheme,

$$\Gamma_{[k]}(t) = F(\mathbf{c}_k(t)) - F(\mathbf{w}^*). \quad (30)$$

Note that $\Gamma_{[k]}(t) \geq 0, \forall k$. Since $\mathbf{w}((K+1)\tau - \Delta) = \mathbf{c}_{[K+1]}((K+1)\tau - \Delta)$, we want to prove that

$$\Gamma_{[K+1]}((K+1)\tau - \Delta)^{-1} \geq T\eta\phi - \frac{L\Psi(\alpha)}{\xi^2}. \quad (31)$$

To determine this bound, note that [13, Lemma 6]

$$\Gamma_{[k]}(t+1)^{-1} - \Gamma_{[k]}(t)^{-1} \geq \eta\phi, \quad (32)$$

and therefore

$$\Gamma_{[k]}((k+1)\tau - \Delta)^{-1} - \Gamma_{[k]}(k\tau - \Delta)^{-1} \geq \tau\eta\phi. \quad (33)$$

It follows that

$$\sum_{k=1}^K [\Gamma_{[k]}((k+1)\tau - \Delta)^{-1} - \Gamma_{[k]}(k\tau - \Delta)^{-1}] \geq T\eta\phi,$$

where $K\tau = T$. Therefore, to prove (31), it is sufficient to show that

$$\sum_{k=1}^K [\Gamma_{[k]}((k+1)\tau - \Delta)^{-1} - \Gamma_{[k+1]}((k+1)\tau - \Delta)^{-1}] \leq \frac{L\Psi(\alpha)}{\xi^2}, \quad (34)$$

which we now prove. Note that, since $\Psi(\alpha) = \sum_{k=1}^K \psi(\alpha, k)$, a sufficient condition which implies (34) is

$$\begin{aligned} & \Gamma_{[k+1]}((k+1)\tau - \Delta) - \Gamma_{[k]}((k+1)\tau - \Delta) \\ & \leq \frac{L\psi(\alpha, k)}{\xi^2} \Gamma_{[k]}((k+1)\tau - \Delta) \Gamma_{[k+1]}((k+1)\tau - \Delta). \end{aligned} \quad (35)$$

Note that, from conditions (3) and (4) of the proposition statement,

$$\Gamma_{[k]}((k+1)\tau - \Delta) \geq \xi, \quad \forall k, \quad (36)$$

$$\Gamma_{[K+1]}((K+1)\tau - \Delta) \geq \xi. \quad (37)$$

Moreover, from (33) with $k < K-1$,

$$\Gamma_{[k+1]}((k+1)\tau - \Delta) \geq \Gamma_{[k+1]}((k+2)\tau - \Delta) \geq \xi.$$

Therefore, to prove (35), it is sufficient to show

$$\Gamma_{[k+1]}((k+1)\tau - \Delta) - \Gamma_{[k]}((k+1)\tau - \Delta) \leq L\psi(\alpha, k).$$

Indeed,

$$\begin{aligned} & \Gamma_{[k+1]}((k+1)\tau - \Delta) - \Gamma_{[k]}((k+1)\tau - \Delta) \\ & \leq L\|\mathbf{w}((k+1)\tau - \Delta) - \mathbf{c}_k((k+1)\tau - \Delta)\|, \end{aligned} \quad (38)$$

so that the result directly follows from Proposition 1. The Proposition is thus proved. \square

Lemma 1. Under Assumption 1 and $\eta \leq 1/\beta$, we have

$$\|\mathbf{w}((k+1)\tau - \Delta) - \mathbf{c}_k((k+1)\tau - \Delta)\| \leq \psi(\alpha, k), \quad (39)$$

such that

$$\|F(\mathbf{w}((k+1)\tau - \Delta)) - F(\mathbf{c}_k((k+1)\tau - \Delta))\| \leq L\psi(\alpha, k),$$

where $\psi(\alpha, k)$ is defined in (17).

Proof. Let $t \in (k\tau - \Delta, (k+1)\tau - \Delta]$. Then from (10) we have

$$\begin{aligned} \mathbf{w}(t) - \mathbf{c}_k(t) &= (1 - \alpha_t)[\mathbf{w}(t-1) - \mathbf{c}_k(t-1)] \\ &\quad - (1 - \alpha_t)\eta \sum_i \rho_i [\nabla F_i(\mathbf{w}_i(t-1)) - \nabla F_i(\mathbf{c}_k(t-1))] \\ &\quad + \eta\alpha_t \sum_{r=0}^{\Delta-1} \nabla F(\mathbf{c}_k(k\tau - \Delta + r)). \end{aligned}$$

Taking the norm, using the triangular inequality and Assumption 1, we obtain the inequality

$$\begin{aligned} \|\mathbf{w}(t) - \mathbf{c}_k(t)\| &\leq (1 - \alpha_t)\|\mathbf{w}(t-1) - \mathbf{c}_k(t-1)\| \\ &\quad + (1 - \alpha_t)\eta\beta \sum_i \rho_i \|\mathbf{w}_i(t-1) - \mathbf{c}_k(t-1)\| + \alpha_t\eta L\Delta. \end{aligned}$$

Using induction, we then obtain

$$\begin{aligned} \|\mathbf{w}((k+1)\tau - \Delta) - \mathbf{c}_k((k+1)\tau - \Delta)\| &\leq (1 - \alpha)\eta\beta \sum_{\ell=k\tau-\Delta}^{k\tau-1} \sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\| \\ &\quad + \eta\beta \sum_{\ell=k\tau}^{(k+1)\tau-\Delta-1} \sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\| + \alpha\eta L\Delta. \quad (40) \end{aligned}$$

We now bound the term $\sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\|$. For $\ell \in (k\tau - \Delta, (k+1)\tau - \Delta]$, we have

$$\begin{aligned} \mathbf{w}_i(\ell) - \mathbf{c}_k(\ell) &= (1 - \alpha_\ell)[\mathbf{w}_i(\ell-1) - \mathbf{c}_k(\ell-1)] \\ &\quad - (1 - \alpha_\ell)\eta [\nabla F_i(\mathbf{w}_i(\ell-1)) - \nabla F_i(\mathbf{c}_k(\ell-1))] \\ &\quad - (1 - \alpha_\ell)\eta [\nabla F_i(\mathbf{c}_k(\ell-1)) - \nabla F(\mathbf{c}_k(\ell-1))] \\ &\quad + \alpha_\ell\eta \sum_{r=0}^{\Delta-1} \nabla F(\mathbf{c}_k(k\tau - \Delta + r)). \end{aligned}$$

Taking the norm, using the triangular inequality, Assumption 1, Definition 2 and computing the sum $\sum_i \rho_i$, we obtain the inequality

$$\begin{aligned} \sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\| &\leq (1 - \alpha_\ell)\eta\delta + \alpha_\ell\eta\Delta L \\ (1 - \alpha_\ell)[1 + \eta\beta] \sum_i \rho_i \|\mathbf{w}_i(\ell-1) - \mathbf{c}_k(\ell-1)\|. \end{aligned}$$

Using induction, it then follows that

$$\sum_{\ell=k\tau-\Delta}^{k\tau-1} \sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\| \leq \epsilon^{(k)} \frac{[1 + \eta\beta]^\Delta - 1}{\eta\beta} + \frac{h(\Delta)}{\eta\beta},$$

and

$$\begin{aligned} &\sum_{\ell=k\tau}^{(k+1)\tau-\Delta-1} \sum_i \rho_i \|\mathbf{w}_i(\ell) - \mathbf{c}_k(\ell)\| \\ &\leq (1 - \alpha) \left[[1 + \eta\beta]^\Delta \epsilon^{(k)} \frac{[1 + \eta\beta]^{\tau-\Delta} - 1}{\eta\beta} + \frac{h(\tau) - h(\Delta)}{\eta\beta} \right] \\ &\quad + \alpha \frac{h(\tau - \Delta)}{\eta\beta} + \alpha\Delta L \frac{[1 + \eta\beta]^{\tau-\Delta} - 1}{\beta}. \end{aligned}$$

Combining these bounds with (40), we finally obtain

$$\begin{aligned} \|\mathbf{w}((k+1)\tau - \Delta) - \mathbf{c}_k((k+1)\tau - \Delta)\| &\leq (1 - \alpha)\epsilon^{(k)} \{ [1 + \eta\beta]^\tau - 1 \} \\ &\quad + (1 - \alpha)h(\tau) + \alpha h(\tau - \Delta) + \alpha\eta\Delta L [1 + \eta\beta]^{\tau-\Delta}, \quad (41) \end{aligned}$$

thus proving the lemma. \square

REFERENCES

- [1] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, 2018.
- [2] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet Thing J.*, vol. 3, no. 6, pp. 854–864, 2016.
- [3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. 20th Int. Conf. Artif. Intell. Stat.*, vol. 54, pp. 1273–1282, 2017.
- [4] P. Kairouz *et al.*, "Advances and open problems in federated learning," *arXiv:1912.04977*, 2019.
- [5] J. Konečný *et al.*, "Federated learning: Strategies for improving communication efficiency," in *Proc. NIPS Workshop on Private Multi-Party Mach. Learn.*, 2016.
- [6] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [7] J. Konečný and P. Richtárik, "Randomized distributed mean estimation: Accuracy vs. communication," *Front. Appl. Math. Stat.*, vol. 4, p. 62, 2018.
- [8] S. Horvath, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtárik, "Natural compression for distributed deep learning," *arXiv:1905.10988*, 2019.
- [9] Y. Tu, Y. Ruan, S. Wagle, C. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," *Proc. IEEE INFOCOM*, pp. 1–9, 2020.
- [10] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local gd on heterogeneous data," *arXiv:1909.04715*, 2019.
- [11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv:1907.02189*, 2019.
- [12] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization," *arXiv:1905.03817*, 2019.
- [13] 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 J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.
- [14] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM*, 2019, pp. 1387–1395.
- [15] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.
- [16] F. P.-C. Lin, C. Brinton, and N. Michelusi, "Federated learning with communication delay in edge networks," *arXiv:2008.09323*, 2020.
- [17] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," *Proc. USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, vol. 16, pp. 265–283, 2016.
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.