Delay Minimization for Federated Learning Over Wireless Communication Networks

Zhaohui Yang [*1] **Mingzhe Chen** [*2 3] **Walid Saad** [4] **Choong Seon Hong** [5] **Mohammad Shikh-Bahaei** [1] **H. Vincent Poor** [2] **Shuguang Cui** [3]

Abstract

In this paper, the problem of delay minimization for federated learning (FL) over wireless communication networks is investigated. In the considered model, each user exploits limited local computational resources to train a local FL model with its collected data and, then, sends the trained FL model parameters to a base station (BS) which aggregates the local FL models and broadcasts the aggregated FL model back to all the users. Since FL involves learning model exchanges between the users and the BS, both computation and communication latencies are determined by the required learning accuracy level, which affects the convergence rate of the FL algorithm. This joint learning and communication problem is formulated as a delay minimization problem, where it is proved that the objective function is a convex function of the learning accuracy. Then, a bisection search algorithm is proposed to obtain the optimal solution. Simulation results show that the proposed algorithm can reduce delay by up to 27.3% compared to conventional FL methods.

[*]Equal contribution [1]Centre for Telecommunications Research, Department of Engineering, King's College London, UK [2]Electrical Engineering Department, Princeton University, USA [3]Shenzhen Research Institute of Big Data and School of Science and Engineering, the Chinese University of Hong Kong, China [4]Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA [5]Department of Computer Science and Engineering, Kyung Hee University, Rep. of Korea. This work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the Natural Science Foundation of China under Grants NSFC-61629101, NSFC-61871109, NSFC-61941115, by the EPSRC through the Scalable Full Duplex Dense Wireless Networks (SENSE) grant EP/P003486/1, and by the U.S. National Science Foundation under Grants CCF-1908308 and CNS-1909372. Correspondence to: Zhaohui Yang <yang.zhaohui@kcl.ac.uk>, Mingzhe Chen <mingzhec@princeton.edu.>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

1. Introduction

In future wireless systems, due to privacy constraints and limited communication resources for data transmission, it is impractical for all wireless devices to transmit all of their collected data to a data center that can implement centralized machine learning algorithms for data analysis (Wang et al., 2018; Chen et al., 2019a; Huang et al., 2020; Dong et al., 2019; Gao et al., 2020). To this end, distributed edge learning approaches, such as federated learning (FL), were proposed (Saad et al., to appear, 2020; Park et al., 2019; Chen et al., 2020; Samarakoon et al., 2018; Gündüz et al., 2019; Chen et al., 2019b). In FL, the wireless devices individually establish local learning models and cooperatively build a global learning model by uploading the local learning model parameters to a base station (BS) instead of sharing training data(McMahan et al., 2016; Yang et al., 2020; Wang et al., 2019). To implement FL over wireless networks, the wireless devices must transmit their local training results over wireless links (Zhu et al., 2018a), which can affect the FL performance, because both local training and wireless transmission introduce delay. Hence, it is necessary to optimize the delay for wireless FL implementation.

Some of the challenges of FL over wireless networks have been studied in (Zhu et al., 2018b; Ahn et al., 2019; Yang et al., 2018; Zeng et al., 2019; Chen et al., 2019; Tran et al., 2019). To minimize latency, a broadband analog aggregation multi-access scheme for FL was designed in (Zhu et al., 2018b). The authors in (Ahn et al., 2019) proposed an FL implementation scheme between devices and access point over Gaussian multiple-access channels. To improve the statistical learning performance for on-device distributed training, the authors in (Yang et al., 2018) developed a sparse and low-rank modeling approach. The work in in (Zeng et al., 2019) proposed an energy-efficient strategy for bandwidth allocation with the goal of reducing devices' sum energy consumption while meeting the required learning performance. However, the prior works (Konečný et al., 2016; Zhu et al., 2018b; Ahn et al., 2019; Yang et al., 2018; Zeng et al., 2019) focused on the delay/energy consumption for wireless consumption without considering the delay/energy tradeoff between learning and transmission. Recently, in (Chen et al., 2019) and (Tran et al., 2019), the authors considered both local

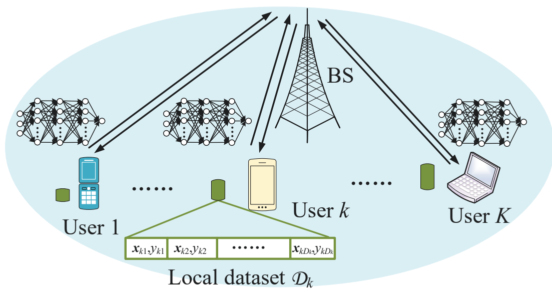

Figure 1. FL over wireless communication networks.

learning and wireless transmission energy. In (Chen et al., 2019), the authors investigated the FL loss function minimization problem with taking into account packet errors over wireless links. However, this prior work ignored the computation delay of local FL model. The authors in (Tran et al., 2019) considered the sum learning and transmission energy minimization problem for FL, where all users transmit learning results to the BS. However, the solution in (Tran et al., 2019) requires all users to upload their learning model synchronously.

The main contribution of this paper is a framework for optimizing FL over wireless networks. In particular, we consider a wireless-powered FL algorithm in which each user locally computes its FL model parameters under a given learning accuracy and the BS broadcasts the aggregated FL model parameters to all users. Considering the tradeoff between local computation delay and wireless transmission delay, we formulate a joint transmission and computation optimization problem aiming to minimize the delay for FL.

2. System Model and Problem Formulation

Consider a cellular network that consists of one BS serving a set $\mathcal{K}$ of K users, as shown in Fig. 1. Each user k has a local dataset $\mathcal{K}_k$ with D_k data samples. For each dataset $\mathcal{K}_k = \{\boldsymbol{x}_{kl}, y_{kl}\}_{l=1}^{D_k}$, $\boldsymbol{x}_{kl} \in \mathbb{R}^d$ is an input vector of user k and y_{kl} is its corresponding output[1].

2.1. FL Model

For FL, we define a vector $\boldsymbol{w}$ to capture the parameters related to the global FL model that is trained by all datasets. Hereinafter, the FL model that is trained by all users' data set is called *global FL model*, while the FL model that is trained by each user's dataset is called *local FL model*. We introduce the loss function $f(\boldsymbol{w}, \boldsymbol{x}_{kl}, y_{kl})$, that captures the FL performance over input vector $\boldsymbol{x}_{kl}$ and output y_{kl}. For different learning tasks, the loss function will be different. Since the dataset of user k is $\mathcal{K}_k$, the total loss function of user k will be:

$$F_k(\boldsymbol{w}) = \frac{1}{D_k} \sum_{l=1}^{D_k} f(\boldsymbol{w}, \boldsymbol{x}_{kl}, y_{kl}). \tag{1}$$

[1]For simplicity, this paper only considers an FL algorithm with a single output. Our approach can be extended to the case with multiple outputs (Konečnỳ et al., 2016).

Algorithm 1 FL Algorithm

1: Initialize global regression vector $\boldsymbol{w}^0$ and iteration number $n = 0$.
2: **repeat**
3: Each user k computes $\nabla F_k(\boldsymbol{w}^{(n)})$ and sends it to the BS.
4: The BS computes $\nabla F(\boldsymbol{w}^{(n)}) = \frac{1}{K}\sum_{k=1}^{K} \nabla F_k(\boldsymbol{w}^{(n)})$, which is broadcast to all users.
5: **parallel for** user $k \in \mathcal{K}$
6: Solve local FL problem (3) with a given learning accuracy η and the solution is $\boldsymbol{h}_k^{(n)}$.
7: Each user sends $\boldsymbol{h}_k^{(n)}$ to the BS.
8: **end for**
9: The BS computes $\boldsymbol{w}^{(n+1)} = \boldsymbol{w}^{(n)} + \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{h}_k^{(n)}$ and broadcasts the value to all users.
10: Set $n = n + 1$.
11: **until** the accuracy ϵ_0 of problem (2) is obtained.

In order to deploy FL, it is necessary to train the underlying model. Training is done in order to compute the global FL model for all users without sharing their local datasets due to privacy and communication issue. The FL training problem can be formulated as follows (Wang et al., 2018):

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) = \sum_{k=1}^{K} \frac{D_k}{D} F_k(\boldsymbol{w}) = \frac{1}{D} \sum_{k=1}^{K} \sum_{l=1}^{D_k} f(\boldsymbol{w}, \boldsymbol{x}_{kl}, y_{kl}), \tag{2}$$

where $D = \sum_{k=1}^{K} D_k$ is the total data samples of all users.

To solve problem (2), we adopt the FL algorithm in (Konečnỳ et al., 2016), which is summarized in Algorithm 1. In Algorithm 1, at each iteration of the FL algorithm, each user downloads the global FL model parameters from the BS for local computing, while the BS periodically gathers the local FL model parameters from all users and sends the updated global FL model parameters back to all users. We define $\boldsymbol{w}^{(n)}$ as the global FL parameter at a given iteration n. Each user computes the local FL problem:

$$\min_{\boldsymbol{h}_k \in \mathbb{R}^d} \quad G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k) \triangleq F_k(\boldsymbol{w}^{(n)} + \boldsymbol{h}_k) - (\nabla F_k(\boldsymbol{w}^{(n)}) - \xi \nabla F(\boldsymbol{w}^{(n)}))^T \boldsymbol{h}_k, \tag{3}$$

by using the gradient method with a given accuracy. In problem (3), ξ is a constant value. The solution $\boldsymbol{h}_k$ in problem (3) means the updated value of local FL parameter for user k in each iteration, i.e., $\boldsymbol{w}^{(n)} + \boldsymbol{h}_k$ denotes user k' local FL parameter at the n-th iteration. Since it is hard to obtain the actual optimal solution of problem (3), we obtain a solution of (3) with some accuracy. The solution $\boldsymbol{h}_k^{(n)}$ of problem (3) at the n-th iteration with accuracy η means that

$$G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)}) - G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)*}) \geq \eta(G_k(\boldsymbol{w}^{(n)}, \boldsymbol{0}) - G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n)*})), \tag{4}$$

where $\boldsymbol{h}_k^{(n)*}$ is the actual optimal solution of problem (3).

In Algorithm 1, the iterative method involves a number of global iterations (i.e., the value of n in Algorithm 1) to achieve a global accuracy ϵ_0 of global FL model. The solution $\boldsymbol{w}^{(n)}$ of problem (2) with accuracy ϵ_0 means that

$$F(\boldsymbol{w}^{(n)}) \quad F(\boldsymbol{w}^*) \geq \epsilon_0(F(\boldsymbol{w}^{(0)}) \quad F(\boldsymbol{w}^*)), \tag{5}$$

where $\boldsymbol{w}^*$ is the actual optimal solution of problem (2).

To analyze the convergence of Algorithm 1, we assume that $F_k(\boldsymbol{w})$ is L-Lipschitz continuous and γ-strongly convex, i.e.,

$$\gamma \boldsymbol{I} \propto \;\; {}^2 F_k(\boldsymbol{w}) \propto L\boldsymbol{I}, \quad Ɗk \forall \{ \;. \tag{6}$$

Under assumption (6), we provide the following lemma about convergence rate of Algorithm 1.

Lemma 1 *If we run Algorithm 1 with $0 < \xi \geq \frac{\gamma}{L}$ for*

$$n \succeq \frac{a}{1 \quad \eta} \triangleq I_0, \tag{7}$$

iterations with $a = \frac{2L^2}{\gamma^2 \xi} \ln \frac{1}{\epsilon_0}$, we have $F(\boldsymbol{w}^{(n)}) \quad F(\boldsymbol{w}^*) \geq \epsilon_0(F(\boldsymbol{w}^{(0)}) \quad F(\boldsymbol{w}^*))$.

The proof of Lemma 1 can be found in (Yang et al., 2019). From Lemma 1, we can find that the number of global iterations n increases with the local accuracy. This is because more iterations are needed if the local computation has a low accuracy.

2.2. Computation and Transmission Model

The FL procedure between the users and their serving BS consists of three steps in each iteration: Local computation at each user (using several local iterations), local FL parameter transmission for each user, and result aggregation and broadcast at the BS. During the local computation step, each user calculates its local FL parameters by using its local dataset and the received global FL parameters.

2.2.1. Local Computation

We solve the local learning problem (3) by using the gradient method. In particular, the gradient procedure in the $(i+1)$-th iteration is given by:

$$\boldsymbol{h}_k^{(n),(i+1)} = \boldsymbol{h}_k^{(n),(i)} \quad \delta \quad G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n),(i)}), \tag{8}$$

where δ is the step size, $\boldsymbol{h}_k^{(n),(i)}$ is the value of $\boldsymbol{h}_k$ at the i-th local iteration with given vector $\boldsymbol{w}^{(n)}$, and $\quad G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k^{(n),(i)})$ is the gradient of function $G_k(\boldsymbol{w}^{(n)}, \boldsymbol{h}_k)$ at point $\boldsymbol{h}_k = \boldsymbol{h}_k^{(n),(i)}$. We set the initial solution $\boldsymbol{h}_k^{(n),(0)} = \boldsymbol{0}$.

Next, we provide the number of local iterations needed to achieve a local accuracy η in (4). We set $v = \frac{2}{(2-L\delta)\delta\gamma}$.

Lemma 2 *If we set step $\delta < \frac{2}{L}$ and run the gradient method for $i \succeq v \log_2(1/\eta)$ iterations at each user, we can solve local FL problem (3) with an accuracy η.*

The proof of Lemma 2 can be found in paper (Yang et al., 2019). Let f_k be the computation capacity of user k, which is measured by the number of CPU cycles per second. The computation time at user k needed for data processing is:

$$\tau_k = \frac{vC_k D_k \log_2(1/\eta)}{f_k} = \frac{A_k \log_2(1/\eta)}{f_k}, \quad Ɗk \forall \{ \;, \tag{9}$$

where C_k (cycles/bit) is the number of CPU cycles required for computing one sample data at user k, $v \log_2(1/\eta)$ is the number of local iterations for each user as given by Lemma 2, and $A_k = vC_k D_k$.

2.2.2. Wireless Transmission

After local computation, all users upload their local FL parameters to the BS via frequency domain multiple access (FDMA). The achievable rate of user k can be given by:

$$r_k = b_k \log_2 \Big) 1 + \frac{g_k p_k}{N_0 b_k} \sum, \quad Ɗk \forall \{ \;, \tag{10}$$

where b_k is the bandwidth allocated to user k, p_k is the transmit power of user k, g_k is the channel gain between user k and the BS, and N_0 is the power spectral density of the Gaussian noise. Due to the limited bandwidth, we have $\sum_{k=1}^{K} b_k \geq B$, where B is the total bandwidth.

In this step, user k needs to upload the local FL parameters to the BS. Since the dimensions of the vector $\boldsymbol{h}_k^{(n)}$ are fixed for all users, the data size that each user needs to upload is constant, and can be denoted by s. To upload data of size s within transmit time t_k, we must have: $t_k r_k \succeq s$.

2.2.3. Information Broadcast

In this step, the BS aggregates the global prediction model parameters. The BS broadcasts the global prediction model parameters to all users in the downlink. Due to the high power of the BS and large downlink bandwidth, we ignore the downlink time. Note that the local data $\mathcal{K}_k$ is not accessed by the BS, so as to protect the privacy of users, as is required by FL. The delay of each user includes the local computation time and transmit time. Based on (7) and (9), the delay T_k of user k will be:

$$T_k = I_0(\tau_k + t_k) = \frac{a}{1 \quad \eta} \Big) \frac{A_k \log_2(1/\eta)}{f_k} + t_k \sum. \tag{11}$$

We define $T = \max_{k \in \mathcal{K}} T_k$ as the delay for training the whole FL algorithm.

2.3. Problem Formulation

We now pose the delay minimization problem:

$$\min_{T, \boldsymbol{t}, \boldsymbol{b}, \boldsymbol{f}, \boldsymbol{p}, \eta} \; T \tag{12}$$

$$\text{s.t.} \quad \frac{a}{1 \quad \eta} \Big) \frac{A_k \log_2(1/\eta)}{f_k} + t_k \sum \geq T, \quad Ɗk \forall \{ \;, \tag{12a}$$

$$t_k b_k \log_2 \Big) 1 + \frac{g_k p_k}{N_0 b_k} \sum \succeq s, \quad Ɗk \forall \{ \;, \tag{12b}$$

$$\sum_{k=1}^{K} b_k \geq B, \tag{12c}$$

$$0 \geq f_k \geq f_k^{\max}, 0 \geq p_k \geq p_k^{\max}, \; Ɗk \forall \{ \;, \tag{12d}$$

$$0 \geq \eta \geq 1, \tag{12e}$$

$$t_k \succeq 0, b_k \succeq 0, \quad Ɗk \forall \{ \;, \tag{12f}$$

where $\boldsymbol{t} = [t_1, \cdots, t_K]^T$, $\boldsymbol{b} = [b_1, \cdots, b_K]^T$, $\boldsymbol{f} = [f_1, \cdots, f_K]^T$, and $\boldsymbol{p} = [p_1, \cdots, p_K]^T$. $f_k^{\max}$ and $p_k^{\max}$ are, respectively, the maximum local computation capacity and maximum transmit power of user k. (12a) indicates that the execution time of the local tasks and the transmit time for all users should not exceed the delay of the whole FL algorithm. The data transmission constraint is given by (12b), while the bandwidth constraint is given by (12c). (12d) represents the maximum local computation capacity and transmit power limits of all users. The accuracy constraint is given by (12e).

3. Optimal Resource Allocation

Although the delay minimization problem (12) is nonconvex due to constraints (12a)-(12b), the globally optimal solution is shown to be obtained by using the bisection method.

3.1. Optimal Resource Allocation

Let $(T^*, \boldsymbol{t}^*, \boldsymbol{b}^*, \boldsymbol{f}^*, \boldsymbol{p}^*, \eta^*)$ be the optimal solution of problem (12). We provide the following lemma about the feasibility conditions of problem (12).

Lemma 3 *Problem (12) with fixed $T < T^*$ is always feasible, while problem (12) with fixed $T > T^*$ is infeasible.*

Proof: Assume that $(\bar{T}, \bar{\boldsymbol{t}}, \bar{\boldsymbol{b}}, \bar{\boldsymbol{f}}, \bar{\boldsymbol{p}}, \bar{\eta})$ is a feasible solution of problem (12) with $T = \bar{T} < T^*$. Then, solution $(\bar{T}, \bar{\boldsymbol{t}}, \bar{\boldsymbol{b}}, \bar{\boldsymbol{f}}, \bar{\boldsymbol{p}}, \bar{\eta})$ is feasible with lower value of the objective function than solution $(T^*, \boldsymbol{t}^*, \boldsymbol{b}^*, \boldsymbol{f}^*, \boldsymbol{p}^*, \eta^*)$, which contradicts the fact that $(T^*, \boldsymbol{t}^*, \boldsymbol{b}^*, \boldsymbol{f}^*, \boldsymbol{p}^*, \eta^*)$ is the optimal solution. For problem (12) with $T = \bar{T} > T^*$, we can always construct a feasible solution $(\bar{T}, \boldsymbol{t}^*, \boldsymbol{b}^*, \boldsymbol{f}^*, \boldsymbol{p}^*, \eta^*)$ to problem (12) by checking all constraints. □

According to Lemma 3, we can use the bisection method to obtain the optimal solution of problem (12). Denote

$$T_{\min} = 0, T_{\max} = \max_{k \in \mathcal{K}} \frac{2aA_k}{f_k^{\max}} + \frac{2aKs}{B \log_2\left(1 + \frac{g_k p_k^{\max} K}{N_0 B}\right)}. \tag{13}$$

If $T > T_{\max}$, problem (12) is always feasible by setting $f_k = f_k^{\max}$, $p_k = p_k^{\max}$, $b_k = \frac{B}{K}$, $\eta = \frac{1}{2}$, and

$$t_k = \frac{Ks}{B \log_2\left(1 + \frac{g_k p_k^{\max} K}{N_0 B}\right)}. \tag{14}$$

Hence, the optimal T^* of problem (12) must lie in the interval $(T_{\min}, T_{\max})$. At each step, the bisection method divides the interval in two by computing the midpoint $T_{\text{mid}} = (T_{\min} + T_{\max})/2$. There are now only two possibilities: 1) if problem (12) with $T = T_{\text{mid}}$ is feasible, we have $T^* \in (T_{\min}, T_{\text{mid}}]$ and 2) if problem (12) with $T = T_{\text{mid}}$ is infeasible, we have $T^* \in (T_{\text{mid}}, T_{\max})$. The bisection method selects the subinterval that is guaranteed to be a bracket as the new interval to be used in the next step. As such an interval that contains the optimal T^* is reduced in width by 50% at each step. The process continues until the interval is sufficiently small.

With a fixed T, we still need to check whether there exists a feasible solution satisfying constraints (12a)-(12g). From constraints (12a) and (12c), we can see that it is always efficient to utilize the maximum computation capacity, i.e., $f_k^* = f_k^{\max}, \forall k \in \mathcal{K}$. In addition, from (12b) and (12d), we can see that minimizing the delay can be done by having: $p_k^* = p_k^{\max}, \forall k \in \mathcal{K}$. Substituting the maximum computation capacity and maximum transmission power into (12), delay minimization problem becomes:

$$\min_{T, \boldsymbol{t}, \boldsymbol{b}, \eta} \quad T \tag{15}$$

$$\text{s.t.} \quad t_k \leq \frac{(1-\eta)T}{a} + \frac{A_k \log_2 \eta}{f_k^{\max}}, \quad \forall k \in \mathcal{K}, \tag{15a}$$

$$\frac{s}{t_k} \leq b_k \log_2\left(1 + \frac{g_k p_k^{\max}}{N_0 b_k}\right), \quad \forall k \in \mathcal{K}, \tag{15b}$$

$$\sum_{k=1}^{K} b_k \leq B, \tag{15c}$$

$$0 \leq \eta \leq 1, \tag{15d}$$

$$t_k \geq 0, b_k \geq 0, \quad \forall k \in \mathcal{K}. \tag{15e}$$

We provide the sufficient and necessary condition for the feasibility of set (15a)-(15e) using the following lemma.

Lemma 4 *With a fixed T, set (15a)-(15e) is nonempty if an only if*

$$B \geq \min_{0 \leq \eta \leq 1} \sum_{k=1}^{K} u_k(v_k(\eta)), \tag{16}$$

where

$$u_k(\eta) = \frac{(\ln 2)\eta}{W\left(-\frac{(\ln 2)N_0\eta}{g_k p_k^{\max}} e^{-\frac{(\ln 2)N_0\eta}{g_k p_k^{\max}}}\right) + \frac{(\ln 2)N_0\eta}{g_k p_k^{\max}}}, \tag{17}$$

and

$$v_k(\eta) = \frac{s}{\frac{(1-\eta)T}{a} + \frac{A_k \log_2 \eta}{f_k^{\max}}}. \tag{18}$$

Proof: To prove this, we first define a function $y = x \ln\left(1 + \frac{1}{x}\right)$ with $x > 0$. Then, we have

$$y' = \ln\left(1 + \frac{1}{x}\right) - \frac{1}{x+1}, y'' = -\frac{1}{x(x+1)^2} < 0. \tag{19}$$

According to (19), y' is a decreasing function. Since $\lim_{t_i \to +\infty} y' = 0$, we have $y' > 0$ for all $0 < x < +\infty$. Hence, y is an increasing function, i.e., the right hand side of (15b) is an increasing function of bandwidth b_k. To ensure that the maximum bandwidth constraint (15c) can be satisfied, the left hand side of (15b) should be as small as possible, i.e., t_k should be as long as possible. Based on (15a), the optimal time allocation should be:

$$t_k^* = \frac{(1 \quad \eta)T}{a} + \frac{A_k \log_2 \eta}{f_k^{\max}}, \quad \mathcal{D}k \,\forall\, \{ . \tag{20}$$

Substituting (20) into (15b), we can construct the following problem:

$$\min_{\boldsymbol{b},\eta} \quad \sum_{k=1}^{K} b_k \tag{21}$$

$$\text{s.t. } v_k(\eta) \geq b_k \log_2 \Big) 1 + \frac{g_k p_k^{\max}}{N_0 b_k} \Sigma, \ \mathcal{D}k \,\forall\, \{ , \tag{21a}$$

$$0 \geq \eta \geq 1, \tag{21b}$$

$$b_k \preceq 0, \quad \mathcal{D}k \,\forall\, \{ , \tag{21c}$$

where $v_k(\eta)$ is defined in (18). We can observe that set (15a)-(15e) is nonempty if an only if the optimal objective value of (21) is less than B. Since the right hand side of (15b) is an increasing function, (15b) should hold with equality for the optimal solution of problem (21). Setting (15b) with equality, problem (21) reduces to (16). □

To effectively solve (16) in Lemma 4, we provide the following lemma.

Lemma 5 *In (17), $u_k(v_k(\eta))$ is a convex function.*

Proof: We first prove that $v_k(\eta)$ is a convex function. To show this, we define:

$$\phi(\eta) = \frac{s}{\eta}, \quad 0 \geq \eta \geq 1, \tag{22}$$

and

$$\varphi_k(\eta) = \frac{(1 \quad \eta)T}{a} + \frac{A_k \log_2 \eta}{f_k^{\max}}, \quad 0 \geq \eta \geq 1. \tag{23}$$

According to (18), we have: $v_k(\eta) = \phi(\varphi_k(\eta))$. Then, the second-order derivative of $v_k(\eta)$ can be given by:

$$v_k''(\eta) = \phi''(\varphi_k(\eta))(\varphi_k'(\eta))^2 + \phi'(\varphi_k(\eta))\varphi_k''(\eta). \tag{24}$$

According to (22) and (23), we have:

$$\phi'(\eta) = \quad \frac{s}{\eta^2} \geq 0, \quad \phi''(\eta) = \frac{2s}{\eta^3} \preceq 0, \tag{25}$$

and

$$\varphi_k''(\eta) = \quad \frac{A_k}{(\ln 2) f_k^{\max} \eta^2} \geq 0. \tag{26}$$

Combining (24)-(26), we can find that $v_k''(\eta) \preceq 0$, i.e., $v_k(\eta)$ is a convex function.

Then, we can show that $u_k(\eta)$ is an increasing and convex function. According to the proof of Lemma 4, $u_k(\eta)$ is the inverse function of the right hand side of (15b). If we further define function:

$$z_k(\eta) = \eta \log_2 \Big) 1 + \frac{g_k p_k^{\max}}{N_0 \eta} \Sigma, \quad \eta \preceq 0, \tag{27}$$

$u_k(\eta)$ is the inverse function of $z_k(\eta)$, which gives $u_k(z_k(\eta)) = \eta$.

According to (19), function $z_k(\eta)$ is an increasing and concave function, i.e., $z_k'(\eta) \preceq 0$ and $z_k''(\eta) \geq 0$. Since $z_k(\eta)$ is an increasing function, its inverse function $u_k(\eta)$ is also an increasing function.

Based on the definition of concave function, for any $\eta_1 \preceq 0$, $\eta_2 \preceq 0$ and $0 \geq \theta \geq 1$, we have:

$$z_k(\theta\eta_1 + (1 \quad \theta)\eta_2) \preceq \theta z_k(\eta_1) + (1 \quad \theta) z_k(\eta_2). \tag{28}$$

Applying the increasing function $u_k(\eta)$ on both sides of (28) yields:

$$\theta\eta_1 + (1 \quad \theta)\eta_2 \preceq u_k(\theta z_k(\eta_1) + (1 \quad \theta) z_k(\eta_2)). \tag{29}$$

Denote $\bar{\eta}_1 = z_k(\eta_1)$ and $\bar{\eta}_2 = z_k(\eta_2)$, i.e., we have $\eta_1 = u_k(\bar{\eta}_1)$ and $\eta_2 = u_k(\bar{\eta}_2)$. Thus, (29) can be rewritten as:

$$\theta u_k(\bar{\eta}_1) + (1 \quad \theta) u_k(\bar{\eta}_1) \preceq u_k(\theta\bar{\eta}_1 + (1 \quad \theta)\bar{\eta}_2), \tag{30}$$

which indicates that $u_k(\eta)$ is a convex function. As a result, we have proven that $u_k(\eta)$ is an increasing and convex function, which shows:

$$u_k'(\eta) \preceq 0, \quad u_k''(\eta) \preceq 0. \tag{31}$$

To show the convexity of $u_k(v_k(\eta))$, we have:

$$u_k''(v_k(\eta)) = u_k''(v_k(\eta))(v_k'(\eta))^2 + u_k'(v_k(\eta))v_k''(\eta) \preceq 0,$$

according to $v_k''(\eta) \preceq 0$ and (31). As a result, $u_k(v_k(\eta))$ is a convex function. □

Algorithm 2 Delay Minimization

1: Initialize $T_{\min}$, $T_{\max}$, and the tolerance ϵ_0.
2: **repeat**
3: Set $T = \frac{T_{\min}+T_{\max}}{2}$.
4: Check the feasibility condition (32).
5: If set (15a)-(15e) has a feasible solution, set $T_{\max} = T$. Otherwise, set $T_{\min} = T$.
6: **until** $(T_{\max} - T_{\min})/T_{\max} \leq \epsilon_0$.

Lemma 5 implies that the optimization problem in (16) is a convex problem, which can be effectively solved. By finding the optimal solution of (16), the sufficient and necessary condition for the feasibility of set (15a)-(15e) can be simplified using the following theorem.

Theorem 1 *With a fixed T, set (15a)-(15e) is nonempty if and only if*

$$B \preceq \sum_{k=1}^{K} u_k(v_k(\eta^*)), \tag{32}$$

where η^* is the solution to $\sum_{k=1}^{K} u_k'(v_k(\eta^*))v_k'(\eta^*) = 0$.

Theorem 1 directly follows from Lemmas 4 and 5. Due to the convexity of function $u_k(v_k(\eta))$, $\sum_{k=1}^{K} u_k'(v_k(\eta^*))v_k'(\eta^*)$ is an increasing function of η^*. As a result, the unique solution of η^* to $\sum_{k=1}^{K} u_k'(v_k(\eta^*))v_k'(\eta^*) = 0$ can be effectively solved via the bisection method. Based on Theorem 1, the algorithm for obtaining the minimal delay is summarized in Algorithm 2.

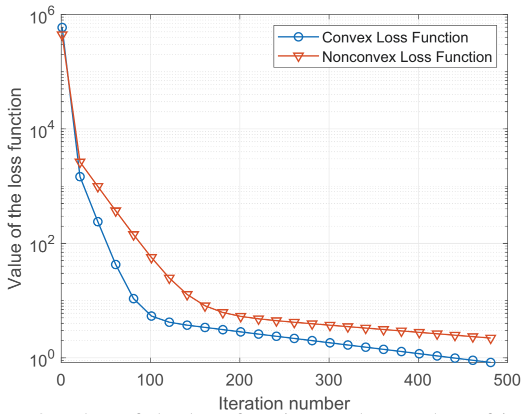

Figure 2. Value of the loss function as the number of iterations varies for convex and nonconvex loss functions.

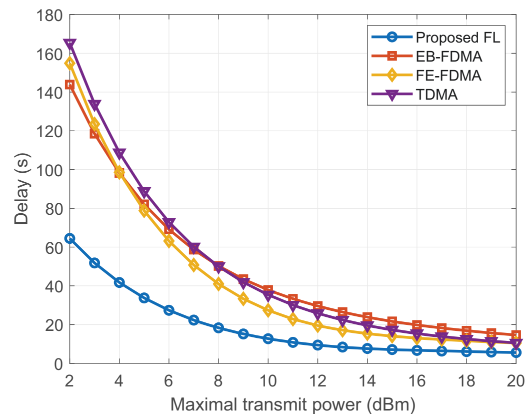

Figure 3. Delay versus maximum average transmit power of each user.

4. Simulation Results

For our simulations, we deploy $K = 50$ users uniformly in a square area of size 500 m $\leq$ 500 m with the BS located at its center. The path loss model is $128.1 + 37.6 \log_{10} d$ (d is in km) and the standard deviation of shadow fading is 8 dB (Yang et al., 2020). In addition, the noise power spectral density is $N_0 = \quad 174$ dBm/Hz. We use the real open blog feedback dataset in (Buza, 2014). This dataset with a total number of 60,021 data samples originates from blog posts and the dimensional of each data sample is 281. The prediction task associated with the data is the prediction of the number of comments in the upcoming 24 hours. Parameter C_k is uniformly distributed in $[1, 3] \leq 10^4$ cycles/sample. The effective switched capacitance in local computation is $\kappa = 10^{-28}$. In Algorithm 1, we set $\xi = 1/10$, $\delta = 1/10$, and $\epsilon_0 = 10^{-3}$. Unless specified otherwise, we choose an equal maximum average transmit power $p_1^{\max} = \times\times\times= p_K^{\max} = p^{\max} = 10$ dBm, an equal maximum computation capacity $f_1^{\max} = \times\times\times= f_K^{\max} = f^{\max} = 2$ GHz, a transmit data size $s = 28.1$ kbits, and a bandwidth $B = 20$ MHz. Each user has $D_k = 500$ data samples, which are randomly selected from the dataset with equal probability. All statistical results are averaged over 1000 independent runs.

In Fig. 2, we show the value of the loss function as the number of iterations varies for convex and nonconvex loss functions. For this feedback prediction problem, we consider two different loss functions: convex loss function $f_1(\boldsymbol{w}, \boldsymbol{x}, y) = \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{w} \quad y)^2$, and nonconvex loss function $f_2(\boldsymbol{w}, \boldsymbol{x}, y) = \frac{1}{2}(\max\}\boldsymbol{x}^T\boldsymbol{w}, 0\nabla \quad y)^2$. From this figure, we can see that, as the number of iterations increases, the value of the loss function first decreases rapidly and then decreases slowly for both convex and nonconvex loss functions. According to Fig. 2, the initial value of the loss function is $F(\boldsymbol{w}^{(0)}) = 10^6$ and the value of the loss function decreases to $F(\boldsymbol{w}^{(500)}) = 1$ for convex loss function after 500 iterations. For our prediction problem, the optimal model $\boldsymbol{w}^*$ is the one that predicts the output without any error, i.e., the value of the loss function value should be $F(\boldsymbol{w}^*) = 0$. Thus, the actual accuracy of the proposed algorithm is $\frac{F(\boldsymbol{w}^{(500)}) - F(\boldsymbol{w}^*)}{F(\boldsymbol{w}^{(0)}) - F(\boldsymbol{w}^*)} = 10^{-6}$ after 500 iterations. Meanwhile, Fig. 2 clearly shows that the FL algorithm with a convex loss function can converge faster than that the one having a nonconvex loss function. According to Fig. 2, the loss function monotonically decreases as the number of iterations varies for even nonconvex loss function, which indicates that the proposed FL scheme can also be applied to the nonconvex loss function.

We compare the proposed FL scheme with the FL FDMA scheme with equal bandwidth $b_1 = \times\times\times= b_K$ (labelled as 'EB-FDMA'), the FL FDMA scheme with fixed local accuracy $\eta = 1/2$ (labelled as 'FE-FDMA'), and the FL time division multiple access (TDMA) scheme in (Tran et al., 2019) (labelled as 'TDMA'). Fig. 3 shows how the delay changes as the maximum average transmit power of each user varies. We can see that the delay of all schemes decreases with the maximum average transmit power of each user. This is because a large maximum average transmit power can decrease the transmission time between users and the BS. We can clearly see that the proposed FL scheme achieves the best performance among all schemes. This is because the proposed approach jointly optimizes bandwidth and local accuracy η, while the bandwidth is fixed in EB-FDMA and η is not optimized in FE-FDMA. Compared to TDMA, the proposed approach can reduce the delay by up to 27.3%.

5. Conclusions

In this paper, we have investigated the delay minimization problem of FL over wireless communication networks. The tradeoff between computation delay and transmission delay is determined by the learning accuracy. To solve this problem, we first proved that the total delay is a convex function of the learning accuracy. Then, we have obtained the optimal solution by using the bisection method. Simulation results show the various properties of the proposed solution.

References

Ahn, J.-H., Simeone, O., and Kang, J. Wireless federated distillation for distributed edge learning with heterogeneous data. *arXiv preprint arXiv:1907.02745*, 2019.

Buza, K. Feedback prediction for blogs. In *Data analysis, machine learning and knowledge discovery*, pp. 145–152. Springer, 2014.

Chen, M., Challita, U., Saad, W., Yin, C., and Debbah, M. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Commun. Surveys Tut.*, pp. 1–1, 2019a. ISSN 1553-877X. doi: 10.1109/COMST.2019.2926625.

Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., and Cui, S. Performance optimization of federated learning over wireless networks. In *Proc. IEEE Global Commun. Conf.*, pp. 1–6, Waikoloa, HI, USA, Dec. 2019b.

Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., and Cui, S. A joint learning and communications framework for federated learning over wireless networks. *arXiv preprint arXiv:1909.07972*, 2019.

Chen, M., Semiari, O., Saad, W., Liu, X., and Yin, C. Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks. *IEEE Trans. Wireless Commun.*, to appear, 2020.

Dong, P., Zhang, H., Li, G. Y., Gaspar, I. S., and Naderi-Alizadeh, N. Deep CNN-based channel estimation for mmWave massive MIMO systems. *IEEE J. Sel. Topics Signal Process.*, 13(5):989–1000, Sept. 2019.

Gao, S., Dong, P., Pan, Z., and Li, G. Y. Reinforcement learning based cooperative coded caching under dynamic popularities in ultra-dense networks. *IEEE Trans. Veh. Technol.*, 69(5):5442–5456, 2020.

Gündüz, D., de Kerret, P., Sidiropoulos, N. D., Gesbert, D., Murthy, C. R., and van der Schaar, M. Machine learning in the air. *IEEE J. Sel. Areas Commun.*, 37(10):2184–2199, Oct. 2019.

Huang, C., Mo, R., and Yuen, C. Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning. *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2020.

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Park, J., Samarakoon, S., Bennis, M., and Debbah, M. Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11):2204–2239, Nov. 2019.

Saad, W., Bennis, M., and Chen, M. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, to appear, 2020.

Samarakoon, S., Bennis, M., Saad, W., and Debbah, M. Distributed federated learning for ultra-reliable low-latency vehicular communications. *arXiv preprint arXiv:1807.08127*, 2018.

Tran, N. H., Bao, W., Zomaya, A., and Hong, C. S. Federated learning over wireless networks: Optimization model design and analysis. In *Proc. IEEE Conf. Computer Commun.*, pp. 1387–1395, Paris, France, June 2019.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE Conf. Computer Commun.*, pp. 63–71, Honolulu, HI, USA, Apr. 2018.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.*, 37(6):1205–1221, June 2019.

Yang, H. H., Liu, Z., Quek, T. Q. S., and Poor, H. V. Scheduling policies for federated learning in wireless networks. *IEEE Trans. Commun.*, to appear, 2020.

Yang, K., Jiang, T., Shi, Y., and Ding, Z. Federated learning via over-the-air computation. *arXiv preprint arXiv:1812.11750*, 2018.

Yang, Z., Chen, M., Saad, W., Hong, C. S., and Shikh-Bahaei, M. Energy efficient federated learning over wireless communication networks. *arXiv preprint arXiv:1911.02417*, 2019.

Yang, Z., Chen, M., Saad, W., Xu, W., Shikh-Bahaei, M., Poor, H. V., and Cui, S. Energy-efficient wireless communications with distributed reconfigurable intelligent surfaces, 2020.

Zeng, Q., Du, Y., Leung, K. K., and Huang, K. Energy-efficient radio resource allocation for federated edge learning. *arXiv preprint arXiv:1907.06040*, 2019.

Zhu, G., Liu, D., Du, Y., You, C., Zhang, J., and Huang, K. Towards an intelligent edge: Wireless communication meets machine learning. *arXiv preprint arXiv:1809.00343*, 2018a.

Zhu, G., Wang, Y., and Huang, K. Low-latency broadband analog aggregation for federated edge learning. *arXiv preprint arXiv:1812.11494*, 2018b.