Update Aware Device Scheduling for Federated Learning at the Wireless Edge

Mohammad Mohammadi Amiri*, Deniz Gündüz†, Sanjeev R. Kulkarni*, H. Vincent Poor*
*Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA
†Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K.

***Abstract*—We study federated learning (FL) at the wireless edge, where power-limited devices with local datasets train a joint model with the help of a remote parameter server (PS). We assume that the devices are connected to the PS through a bandwidth-limited shared wireless channel. At each iteration of FL, a subset of the devices are scheduled to transmit their local model updates to the PS over orthogonal channel resources. We design novel scheduling policies, that decide on the subset of devices to transmit at each round not only based on their channel conditions, but also on the significance of their local model updates. Numerical results show that the proposed scheduling policy provides a better long-term performance than scheduling policies based only on either of the two metrics individually. We also observe that when the data is independent and identically distributed (i.i.d.) across devices, selecting a single device at each round provides the best performance, while when the data distribution is non-i.i.d., more devices should be scheduled.**

I. Introduction

Today devices at the wireless network edge generate a huge amount of data that can be exploited to make sense of the state of a system. Internet-of-things (IoT) devices, drones, or autonomous driving are prime examples where data from the sensors must be continuously collected and processed. Machine learning (ML) algorithms are being developed to exploit these massive datasets. Current ML approaches are limited to centralized algorithms, where a cloud server collects all the data to train a powerful model. However, such centralized algorithms are becoming increasingly costly since offloading data from the devices to the could server is often not feasible due to latency and privacy constraints. An alternative approach is *federated learning* (FL), which enables ML at the wireless edge while the data never leaves the devices.

FL utilizes the computational capabilities of the edge devices to process their local datasets and collaboratively train a learning model with the help of a parameter server (PS) [1]. Due to unreliable links from the edge devices to the PS with limited energy and bandwidth, it is essential to develop approaches with limited communication requirements [1]–[7]. All these works however ignore the physical layer characteristics of the wireless network, and consider perfect rate-limited links between the devices and the PS.

There have been recent studies on FL considering the physical layer aspects of the wireless medium from the devices to the PS. In [8], optimization over batch size and wireless resources is proposed to speed up FL. FL over a Gaussian multiple access channel (MAC) with limited bandwidth is studied in [9], and novel digital and analog approaches are proposed for the transmissions from the devices. In [10], FL over broadband wireless fading MAC is studied, where devices perform channel inversion with the full knowledge of the channel state information (CSI) to align their signals at the PS. This approach is improved in [11] for bandwidth-limited fading MAC by significantly reducing the communication load. Beamforming techniques at a multi-antenna PS for increasing the number of participating devices and overcoming the lack of CSI at the devices are introduced in [12] and [13], respectively. In [14], resource allocation across devices for FL over wireless channels is studied. Frequency of participation of the devices is introduced as a device scheduling metric in [15]. Convergence analysis of FL over wireless networks under various resource allocation schemes is provided in [16]–[18].

In this paper, we consider FL with digital transmission from the edge devices to the PS over a block fading wireless network with limited resources, where we design novel device scheduling policies. We design resource allocation across the participating devices to perform orthogonal (interference-free) transmission. Due to resource limitations, we develop device scheduling policies taking into account the channel conditions and the significance of the local model updates at the devices to make sure that the resources are allocated across the devices with important messages and proper link capacity to convey their messages. Numerical results illustrate the advantages of considering both the channel conditions and the local model updates at the devices for device scheduling over scheduling based on either of the two metrics individually.

Notation: We denote the set of real, natural and complex numbers by $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{C}$, respectively. We let $[i] \triangleq \{1, \dots, i\}$. An all zero entries vector of length i is denoted by $\mathbf{0}_i$. We denote a circularly symmetric complex Gaussian distribution with real or imaginary component with variance $\sigma^2/2$ by $\mathcal{CN}\left(0, \sigma^2\right)$. Notation $|\cdot|$ represents the cardinality of a set or the magnitude of a complex value, and the l_2 norm of a vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|_2$. For a set $\mathcal{S}$ with only real values, $\max_{[K]}\mathcal{S}$ returns a K-element subset of $\mathcal{S}$ with highest values.

II. System Model

We consider FL across M wireless devices, training a model parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ collaboratively with the

This work was supported in part by the U.S. National Science Foundation under Grants CCF-0939370 and CCF-1908308, and by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 677854).

help of a remote parameter server (PS), to which they are connected through a shared wireless medium, to minimize an empirical loss function $F(\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=1}^{M} F_m(\boldsymbol{\theta})$, where $F_m(\boldsymbol{\theta})$ denotes the loss function at device m, $m \in [M]$.

A. FL System

In FL, each device typically performs a *stochastic gradient descent* (SGD) algorithm to minimize an empirical loss function with respect to its local dataset based on a globally consistent model parameter vector received from the PS.

Let $\mathcal{B}_m$ denote the local dataset at device m, $m \in [M]$, with $|\mathcal{B}_m| = B$. The loss function at device m is given by $F_m(\boldsymbol{\theta}) = \frac{1}{B}\sum_{\boldsymbol{u}\in\mathcal{B}_m} f(\boldsymbol{\theta}, \boldsymbol{u})$, $m \in [M]$, where $f(\cdot,\cdot)$ is an empirical loss function defined by the learning task. During the t-th global iteration, having received global model parameter vector $\boldsymbol{\theta}(t)$ from the PS, device m performs a τ-step local SGD, for some $\tau \in \mathbb{N}$. The i-th step of the local SGD at device m, $m \in [M]$, corresponds to the following update:

$$\boldsymbol{\theta}_m^{i+1}(t) = \boldsymbol{\theta}_m^{i}(t) - \eta_m^i(t)\nabla F_m\left(\boldsymbol{\theta}_m^i(t)\right), i \in [\tau], \quad (1)$$

where $\eta_m^i(t)$ denotes the learning rate, and we set $\boldsymbol{\theta}_m^1(t) = \boldsymbol{\theta}(t)$. We further denote the model parameter vector after the τ-th local update at device m by $\boldsymbol{\theta}_m(t+1)$, i.e., $\boldsymbol{\theta}_m(t+1) = \boldsymbol{\theta}_m^{\tau+1}(t)$, $m \in [M]$.

After receiving the updates from the devices, the PS updates the global model parameter vector $\boldsymbol{\theta}(t+1)$ by averaging the results of the τ-step local updates computed at the device, i.e.,

$$\boldsymbol{\theta}(t+1) = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\theta}_m(t+1). \quad (2)$$

This updated vector is then shared among the devices for further computations until it converges. Having defined the local model update at device m as follows:

$$\Delta\boldsymbol{\theta}_m(t) \triangleq \boldsymbol{\theta}_m(t+1) - \boldsymbol{\theta}(t), \quad m \in [M], \quad (3)$$

the update in (2) corresponds to

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \frac{1}{M}\sum_{m=1}^{M} \Delta\boldsymbol{\theta}_m(t). \quad (4)$$

When FL is implemented over a shared wireless medium, it is not reasonable to expect all the devices to be able to convey their model updates to the PS reliably, due to power and bandwidth constraints. When the available channel resources are shared between the devices, each device would be allocated only a very limited bandwidth. Information that can be conveyed to the PS can be further limited due to fading.

In this paper, we consider scheduling a K-element subset of the devices, denoted by $\mathcal{M}(t) \subset [M]$, where $K = |\mathcal{M}(t)|$, at each global iteration step t, for the most efficient utilization of the limited resources. The PS determines the set of scheduled devices, and informs them for transmission at each round. Accordingly, the PS updates the global model parameter vector based on the received local model updates from only the scheduled devices as follows:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \frac{1}{K}\sum_{m\in\mathcal{M}(t)} \Delta\boldsymbol{\theta}_m(t). \quad (5)$$

B. Wireless Medium

We consider a wireless medium with limited bandwidth from the devices to the PS to transmit the local model updates $\Delta\boldsymbol{\theta}_m(t) \in \mathbb{R}^d$, $\forall m \in [M]$. We assume a single-carrier block fading wireless channel with n symbols (time slots) using TDMA for transmission from the devices to the PS[1]. We denote the length-n input to the channel at device m by $\boldsymbol{x}_m(t) \in \mathbb{C}^n$, where $\boldsymbol{x}_m(t) = \mathbf{0}_n$, if $m \notin \mathcal{M}(t)$. The channel gain from device m to the PS is represented by $h_m(t) \in \mathbb{C}$, which is independent and identically distributed (i.i.d.) according to $\mathcal{CN}(0,1)$. The received signal at the PS is added to an independent noise vector with each entry i.i.d. according to $\mathcal{CN}(0,\sigma^2)$. We assume that, at each iteration step, the CSI is known by the devices and the PS. The channel input of device m at the t-th iteration is a function of the scheduling policy, channel gain $h_m(t)$, local dataset $\mathcal{B}_m$, and $\Delta\boldsymbol{\theta}_m(t)$, $m \in [M]$. For a total of T global iterations, we impose the following average transmit power constraint on device m:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[||\boldsymbol{x}_m^n(t)||_2^2\right] \le \bar{P}, \quad \forall m \in [M], \quad (6)$$

where the expectation is over the randomness of the channel.

The goal at the PS is to recover $\frac{1}{K}\sum_{m\in\mathcal{M}(t)} \Delta\boldsymbol{\theta}_m(t)$, which is used to update the global model parameters as in (5). The PS instead uses an estimate of $\frac{1}{K}\sum_{m\in\mathcal{M}(t)} \Delta\boldsymbol{\theta}_m(t)$ upon receiving the noisy observation $\boldsymbol{y}(t)$ from the wireless medium to update the global model parameter vector, which is then shared among the devices through an error-free shared link for further computations.

We focus on digital transmission from the devices to the PS, where each scheduled device employs data compression followed by channel coding to transmit its local model updates. We design various scheduling policies, and perform resource allocation across the scheduled devices to have interference-free communication from the participating devices to the PS. In this digital approach, a capacity achieving channel code followed by discrete quantization at a resolution afforded by the channel capacity is employed by each scheduled device to communicate over the wireless medium.

With the resource allocation approach, device m is allocated n_m distinct time slots, such that $\sum_{m=1}^{M} n_m = n$, where $n_m = 0$, if $m \notin \mathcal{M}(t)$. For large enough n_m, we assume that the Shannon rate at device m can be achieved; that is, the total amount of information that can be conveyed from device m to the PS is $R_m(t) = n_m C_m(t)$, $m \in [M]$, where $C_m(t) \triangleq \log_2\left(1 + |h_m(t)|^2 P_m(t)/\sigma^2\right)$ and $P_m(t) \triangleq \mathbb{E}\left[||\boldsymbol{x}_m^n(t)||_2^2\right]$.

III. Digital SGD (D-SGD) Quantization Scheme

Here we present the data compression scheme employed by the devices for digital transmission over the wireless channel. We utilize the technique introduced in [19], and extended in [9] for FL over a bandwidth-limited wireless medium.

It is worth noting that, at global iteration t, device m intends to transmit $\Delta\boldsymbol{\theta}_m(t)$, computed after the τ-step SGD algorithm,

[1]The single-carrier assumption is for ease of presentation, and the results in this paper can be extended to multi-carrier systems.

$m \in [M]$. For this purpose, it first quantizes $\Delta\boldsymbol{\theta}_m(t)$ by setting all but the largest $q_m(t)$ and the smallest $q_m(t)$ entries to zero (in practice, we typically have $q_m(t) \ll d$). It then computes the average of the positive and negative entries denoted by $q_m^+(t)$ and $q_m^-(t)$, respectively. If $q_m^+(t) \geq |q_m^-(t)|$, it sets all the negative entries to zero and all the positive entries to $q_m^+(t)$, and vice versa, if $q_m^+(t) < |q_m^-(t)|$. We denote the resultant quantized vector with $q_m(t)$ nonzero entries at device m by $\Delta\widehat{\boldsymbol{\theta}}_m\left(q_m(t)\right)$, $m \in [M]$. To transmit $\Delta\widehat{\boldsymbol{\theta}}_m\left(q_m(t)\right)$, device m requires 32 bits representing the real value $q_m^+(t)$ or $q_m^-(t)$ plus 1 bit for its sign, and no more than $\log_2\binom{d}{q_m(t)}$ bits representing the locations of the nonzero entries, $m \in [M]$. Thus, device m needs to transmit a total of

$$r_m\left(q_m(t)\right) = \log_2\binom{d}{q_m(t)} + 33 \text{ bits}, \quad m \in [M]. \tag{7}$$

The D-SGD quantization scheme at device m is characterized by $q_m(t)$, and represented by D-SGD$\left(q_m(t)\right)$ resulting $\Delta\widehat{\boldsymbol{\theta}}_m\left(q_m(t)\right)$, $m \in [M]$. The value of $q_m(t)$ is a design parameter that is determined for different scheduling policies, described in the next section, to satisfy the capacity limitation of transmission from device m to the PS, $m \in [M]$.

IV. Device Scheduling Policies

Here we present various scheduling policies identifying the devices sharing the wireless resources at each iteration. Having more devices scheduled, each device is allocated less resources and contributes to the learning task with less accurate information about the global model parameters. However, if more devices are scheduled, the global model parameters are trained by exploiting a larger fraction of the data samples. The goal is to identify the scheduled devices resulting in the best performance.

After receiving $\boldsymbol{\theta}(t)$ from the PS, all the devices perform the τ-step SGD algorithm as in (1). However, only $K \leq M$ devices in $\mathcal{M}(t)$ are scheduled for transmission at iteration t, and the PS updates the global model parameter vector as follows:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \frac{1}{K}\sum\nolimits_{m\in\mathcal{M}(t)} \Delta\widehat{\boldsymbol{\theta}}_m(q_m(t)). \tag{8}$$

We take into account the channel conditions and the significance of the local model updates at the devices as the scheduling metrics. We study four different scheduling policies, namely thr *best channel* (BC), *best* l_2 *norm* (BN2), *best channel-best* l_2 *norm* (BC-BN2), and *best* l_2 *norm-channel* (BN2-C) schemes, which we explain below.

Due to the natural symmetry of the considered model across the devices, both in terms of the channel statistics and the local model updates, it is reasonable to assume that the probability of scheduling each device will be the same, K/M.[2] Hence, the average transmit power constraint can be rewritten as follows:

$$\frac{K}{MT}\sum\nolimits_{t=1}^{T} P_m(t) \leq \bar{P}, \quad \forall m \in [M]. \tag{9}$$

[2]We will indeed see below that this assumption holds for all four scheduling policies considered in this paper.

For simplicity, we assume a fixed power over time for the scheduled devices, $P_m(t) = M\bar{P}/K$, $\forall m \in \mathcal{M}(t)$, $\forall t \in [T]$.

A. BC Scheduling Policy

BC schedules devices based only on their channel gains. This generalizes the approach studied in [11], where only a single device is scheduled based on the channel conditions. With BC, the PS does not require any information about the model updates at the devices, and it schedules K devices with the highest channel gain magnitudes; that is,

$$\mathcal{M}(t) = \max\nolimits_{[K]}\left\{|h_1(t)|, ..., |h_M(t)|\right\}. \tag{10}$$

Having no knowledge about the model updates at the devices, the PS allocates the time slots so that the scheduled devices have the same capacity. Given $\mathcal{M}(t) = \{m_1, ..., m_K\}$, we set

$$n_{m_1}R_{m_1}(t) = n_{m_2}R_{m_2}(t) = \cdots = n_{m_K}R_{m_K}(t), \tag{11}$$

which, having $\sum_{k=1}^{K} n_{m_k} = n$, results in

$$n_{m_k} = \frac{\prod_{i=1,i\neq k}^{K} R_{m_i}(t)}{\sum_{j=1}^{K}\prod_{i=1,i\neq j}^{K} R_{m_j}(t)}n, \quad k \in [K]. \tag{12}$$

After the above resource allocation scheme, device m_k performs the quantization scheme D-SGD$(q_{m_k}(t))$, with $q_{m_k}(t)$ set as the largest integer satisfying $r_m\left(q_{m_k}(t)\right) \leq n_{m_k}R_{m_k}(t)$, $k \in [K]$, and transmits the quantized bits to the PS over the time slots allocated to it.

B. BN2 Scheduling Policy

With BN2, the scheduling desicion depends only on the significance of the model updates at the devices captured by the l_2 norm of the model update, $\|\Delta\boldsymbol{\theta}_m(t)\|_2$. The transmission takes place in two phases, where in the first phase, having computed $\Delta\boldsymbol{\theta}_m(t)$, device m sends $\|\Delta\boldsymbol{\theta}_m(t)\|_2$ reliably to the PS, $\forall m \in [M]$. The PS then schedules K devices with the largest $\|\Delta\boldsymbol{\theta}_m(t)\|_2$ values, i.e.,

$$\mathcal{M}(t) = \max\nolimits_{[K]}\left\{\|\Delta\boldsymbol{\theta}_1(t)\|_2, ..., \|\Delta\boldsymbol{\theta}_M(t)\|_2\right\}. \tag{13}$$

The time slots are allocated to the scheduled devices by the PS such that their link capacities are proportional to the significance of their local model updates; that is, for $m_i, m_j \in \mathcal{M}(t)$, $\forall i, j \in [K]$, $i \neq j$, we set

$$\frac{n_{m_i}R_{m_i}(t)}{n_{m_j}R_{m_j}(t)} = \frac{\|\Delta\boldsymbol{\theta}_{m_i}(t)\|_2}{\|\Delta\boldsymbol{\theta}_{m_j}(t)\|_2}. \tag{14}$$

Having $\sum_{k=1}^{K} n_{m_k} = n$, it follows that, for $k \in [K]$,

$$n_{m_k} = \frac{\prod_{i=1,i\neq k}^{K} R_{m_i}(t)\,\|\Delta\boldsymbol{\theta}_{m_i}(t)\|_2}{\sum_{j=1}^{K}\prod_{i=1,i\neq j}^{K} R_{m_j}(t)\,\|\Delta\boldsymbol{\theta}_{m_j}(t)\|_2}n. \tag{15}$$

In the second phase of transmission, device m_k transmits the result of D-SGD$(q_{m_k}(t))$, where $q_{m_k}(t)$ is set as the largest integer satisfying $r_{m_k}\left(q_{m_k}(t)\right) \leq n_{m_k}R_{m_k}(t)$, $k \in [K]$.

C. BC-BN2 Scheduling Policy

BC-BN2 generalizes BC and BN2 by taking into account both the channel conditions and the significance of the model

updates at the devices. The PS first identifies K_c devices with the best channel conditions, for some $K \leq K_c \leq M$. Then, K devices from these K_c devices with the most significant model updates are scheduled. Formally, the PS first selects the best K_c devices according to their channel states as follows:

$$\mathcal{M}_c(t) \triangleq \max_{[K_c]} \{|h_1(t)|, ..., |h_M(t)|\}. \quad (16)$$

Only these selected K_c devices share $\|\Delta\boldsymbol{\theta}_m(t)\|_2$ with the PS, which schedules K devices among them as follows:

$$\mathcal{M}(t) = \max_{[K]} \{\|\Delta\boldsymbol{\theta}_m(t)\|_2, \forall m \in \mathcal{M}_c(t)\}. \quad (17)$$

Having known $\|\Delta\boldsymbol{\theta}_{m_k}(t)\|_2, \forall m_k \in \mathcal{M}(t)$, we follow the same resource allocation scheme as BN2. Thus, device m_k, sends $\Delta\widehat{\boldsymbol{\theta}}_{m_k}(q_{m_k}(t)) = \text{D-SGD}(q_{m_k}(t))$ to the PS, where $q_{m_k}(t)$ is set as the largest integer that satisfies $r_m(q_{m_k}(t)) \leq n_{m_k} R_{m_k}(t)$, $k \in [K]$, where n_{m_k} is given in (15).

We highlight that BC-BN2 for $K_c = K$ and $K_c = M$ corresponds to BC and BN2, respectively.

D. BN2-C Scheduling Policy

With BN2-C, each device performs the D-SGD quantization scheme assuming accessibility to all the time slots, and finds the resultant quantized vector, whose l_2 norm is sent to the PS, based on which it schedules the devices. To be more precise, device m calculates the D-SGD quantization parameter, denoted by $q_m^*(t)$, satisfying $r_m(q_m^*(t)) \leq R_m(t)$, $\forall m \in [M]$, based on which it computes $\Delta\widehat{\boldsymbol{\theta}}_m(q_m^*(t)) = \text{D-SGD}(q_m^*(t))$; it then shares $||\Delta\widehat{\boldsymbol{\theta}}_m(q_m^*(t))||_2$ with the PS. The PS schedules devices according to the following policy:

$$\mathcal{M}(t) = \max_{[K]} \left\{||\Delta\widehat{\boldsymbol{\theta}}_1(q_1^*(t))||_2, ..., ||\Delta\widehat{\boldsymbol{\theta}}_M(q_M^*(t))||_2\right\}, \quad (18)$$

and, given $m_i, m_j \in \mathcal{M}(t)$, allocates the time slots to the scheduled devices such that

$$\frac{n_{m_i} R_{m_i}(t)}{n_{m_j} R_{m_j}(t)} = \frac{||\Delta\widehat{\boldsymbol{\theta}}_{m_i}(q_{m_i}^*(t))||_2}{||\Delta\widehat{\boldsymbol{\theta}}_{m_j}(q_{m_j}^*(t))||_2}, \ \forall i, j \in [K], i \neq j. \quad (19)$$

With $\sum_{k=1}^K n_{m_k} = n$, it follows that, for $k \in [K]$,

$$n_{m_k} = \frac{\prod_{i=1, i\neq k}^K R_{m_i}(t)||\Delta\widehat{\boldsymbol{\theta}}_{m_i}(q_{m_i}^*(t))||_2}{\sum_{j=1}^K \prod_{i=1, i\neq j}^K R_{m_j}(t)||\Delta\widehat{\boldsymbol{\theta}}_{m_j}(q_{m_j}^*(t))||_2} n. \quad (20)$$

Scheduled device m performs the D-SGD$(q_m(t))$ quantization scheme, where $q_m(t)$ is set as the largest integer satisfying $r_m(q_m(t)) \leq n_m R_m(t)$, with n_m given in (20), $\forall m \in \mathcal{M}(t)$.

Remark 1. *We highlight that BN2-C intertwines the channel conditions and the significance of local model updates to schedule the devices. Unlike BN2 and BC-BN2, where $\|\Delta\boldsymbol{\theta}_m(t)\|_2$ is directly used for scheduling, BN2-C utilizes the output of the D-SGD quantization scheme, $||\Delta\widehat{\boldsymbol{\theta}}_m(q_m^*(t))||_2$, for scheduling, where $q_m^*(t)$ is a function of the channel gain. This novel technique comes with a computational cost at the devices due to the extra computation of $\Delta\widehat{\boldsymbol{\theta}}_m(q_m^*(t))$. On the other hand, BC requires the smallest computational overhead at the devices. In this work, we do not study the computational complexity at the devices, and the main goal is to utilize the limited communication resources efficiently.*

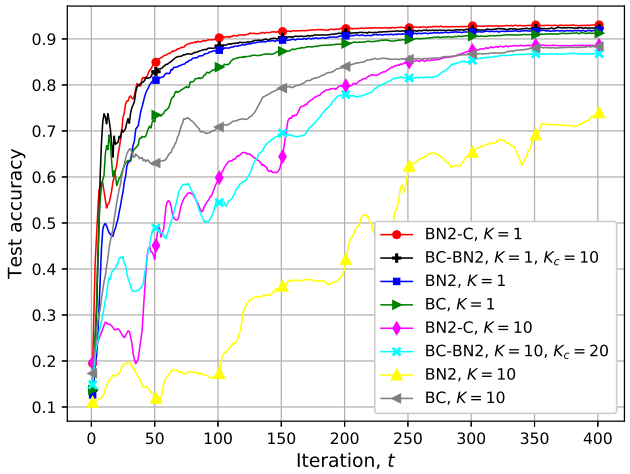


Fig. 1: Performance of different scheduling policies for IID data distribution with $M = 40$, $B = 1000$ and $n = 5 \times 10^3$.

In the longer version of this paper [20], we have established a convergence result for FL with device scheduling, where devices have limited capacity to convey their messages, for a slightly different quantization technique than D-SGD.

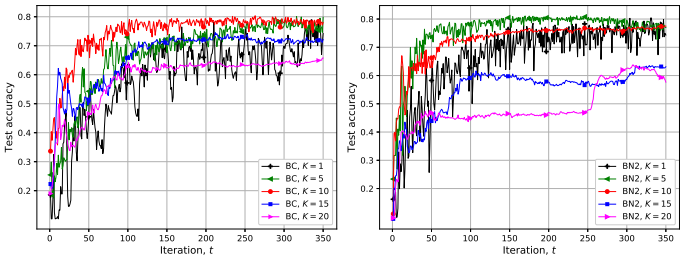
V. Numerical Experiments

Here we compare the performance of different scheduling policies for image classification of the MNIST dataset [21] with 60000 training and 10000 test samples. We train a multi-layer perceptron neural network with a single hidden layer with 256 parameters, in which $d = 203530$, where *softmax* is utilized as the activation function of the output layer.

We consider two data distribution scenarios: *IID data distribution*, where the data samples at each device are selected at random from the training data samples; and *nonIID data distribution*, where at each device two labels/classes are chosen randomly, and half of the local data samples are selected at random from each chosen label/class. We utilize ADAM [22] and AdaGrad [23] optimizers to train the neural network for the IID and nonIID data distribution scenarios, respectively.

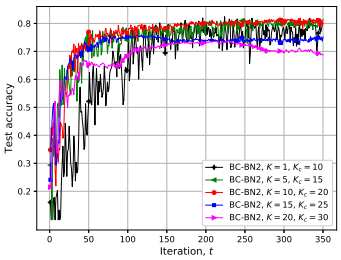
We consider $M = 40$ devices, each with $B = 1000$ training data samples, $n = 5 \times 10^3$ symbols, noise variance $\sigma^2 = 1$ and average power constraint $\bar{P} = 1$. We set the number of local iterations at the devices to $\tau = 3$. We measure the performance as the accuracy with respect to the test data samples, called *test accuracy*, versus the iteration count at the PS, t.

In Fig. 1, we compare the performance of different scheduling policies for IID data distribution. We aim to find the value of K resulting in the best performance for each scheduling policy. To this end, we consider two different values, $K = 1$ and $K = 10$, where for BC-BN2 we set $K_c = 10$ and $K_c = 20$, respectively. We observe that, for each scheduling policy, increasing K deteriorates the accuracy in terms of both the convergence speed and the accuracy level. We did not include the results for other K values, as we have observed that the performance of each scheduling policy deteriorates with K. Thus, we focus on $K = 1$, which,

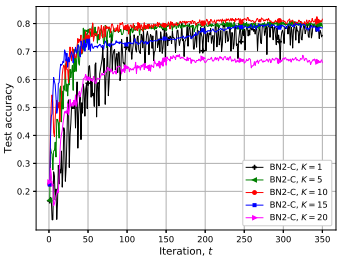


(a) BC scheduling policy

(b) BN2 scheduling policy



(c) BC-BN2 scheduling policy



(d) BN2-C scheduling policy

Fig. 2: Performance of different scheduling policies for nonIID data distribution with $M = 40$, $B = 1000$ and $n = 5 \times 10^3$.

based on our observations, provides the best performance. This illustrates that, with IID local data samples, sending a more accurate update from a single device (which is scheduled at random thanks to the symmetry across the devices in our model) provides a faster convergence rate in the long-term than sending less accurate updates from multiple devices. For comparison, we provide the final accuracy level of each scheduling policy for $K = 1$. These are given by 91.2%, 91.7%, 92.3% and 93.1% for BC, BN2, BC-BN2 and BN2-C, respectively. As can be seen, BN2-C provides the best performance in terms of the convergence speed as well as the final accuracy level. The improvement of BC-BN2 over BN2 is marginal, but both outperform BC. These results illustrate that, given IID data distribution, scheduling devices according to both the significance of their model updates and their channel conditions provides gains in terms of accuracy. Also, from the superiority of BN2 over BC, we conclude that, to obtain the best accuracy performance for the IID scenario, the significance of the model updates plays a more important role than the channel conditions. On the other hand, for large K, such as $K = 10$, it is important to consider the channel conditions for scheduling to make sure that the scheduled devices can send enough bits rather than scheduling devices based only on l_2 norm of their model updates as with BN2.

In Fig. 2, we investigate the performance of these different scheduling policies for a nonIID data distribution scenario. As can be seen, for all the scheduling policies, unlike the IID case, scheduling a single device results in instability of the learning performance appearing as fluctuations in their accuracy levels over iterations. In a nonIID scenario, the local model update at each device is biased due to the biased local datasets, and scheduling a single device provides inaccurate information and causes instability in the performance in the long-term. On the other hand, increasing K (sharing resources among more devices) reduces the accuracy at which the scheduled devices can transmit their model updates. As a result, it is expected that a moderate K value would provide the best performance, which is confirmed with our simulation results. For the setting under consideration, $K = 10$ provides the best accuracy performance for BC, BC-BN2 and BN2-C, while $K = 5$ performs better for BN2, although $K = 10$ shows a more stable accuracy performance with a higher final accuracy level. Similarly to the IID scenario, we observe that it is essential to consider the channel conditions for higher K values in order to make sure that the devices can transmit enough information. Also, as can be seen from the performance of BC for $K = 10$, when scheduling based only on the channel conditions, the performance is more unstable, unless a relatively high number of devices are scheduled, in which case the accuracy level deteriorates. We highlight that, compared to the channel conditions, scheduling based on the significance of the model updates has a greater impact on the performance at the initial iterations when the gradients are more aggressive. On the other hand, it is important to consider the channel conditions at later iterations when approaching the optimum solution, since the transmission is more noisy, and a more accurate estimate of the model update at each participating device is required for robust communication against the noise. For comparison, the best final accuracy levels for BC, BN2, BC-BN2 and BN2-C are 78%, 77.5%, 81.5% and 81.7%, respectively. It can be seen that BN2-C and BC-BN2 outperform BC and BN2 in terms of the accuracy level, highlighting the importance of scheduling devices based on both the channel conditions and the model updates at the devices for the nonIID scenario.

VI. Conclusions

We have studied FL under limited communication resources considering block fading channels from the devices to the PS. We have considered orthogonal digital transmissions from the devices to the PS, and studied various scheduling algorithms to decide which devices participate in the learning process at each round. There is a natural tradeoff between the number of devices participating and the fraction of resources allocated to each device. With more devices scheduled for transmission, the global model parameters are updated at the PS by utilizing a larger fraction of the training data samples; while, each device provides a less accurate estimate of its local model update due to the limited resources available per device. We have proposed novel device scheduling algorithms that consider not only the channel conditions of the devices, but also the significance of their local model updates. Experiments have shown that it is beneficial to schedule devices based on both their channel conditions and the significance of their model updates rather than considering only one of the two metrics. Also, the optimal number of scheduled devices for each considered policy depends on the type of data distribution across devices; for an IID scenario, it is better to schedule a single device, whereas for a nonIID scenario, scheduling a moderate number of devices provides the best performance.

REFERENCES

[1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. NIPS Workshop on Private Multi-Party Machine Learning*, Barcelona, Spain, 2016.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.

[3] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *[Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html*, Apr. 2017.

[4] J. Konecny and P. Richtarik, "Randomized distributed mean estimation: Accuracy vs communication," *Frontiers in Applied Mathematics and Statistics*, vol. 4, Dec. 2018.

[5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

[6] J. Konecny, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv:1511.03575 [cs.LG]*, Nov. 2015.

[7] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," *arXiv:1804.08333 [cs.NI]*, Oct. 2018.

[8] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning system," *arXiv:1905.09712 [cs.LG]*, May 2019.

[9] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, Apr. 2020.

[10] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[11] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[12] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[13] M. M. Amiri, T. M. Duman, and D. Gündüz, "Collaborative machine learning at the wireless edge with blind transmitters," in *Proc. IEEE GlobalSIP*, Ottawa, Canada, Nov. 2019.

[14] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv:1909.07972 [cs.NI]*, Sep. 2019.

[15] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," *arXiv:1910.14648 [cs.IT]*, Oct. 2019.

[16] C. Dinh, et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *arXiv:1910.13067 [cs.LG]*, Nov. 2019.

[17] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[18] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," *arXiv:1911.00856 [cs.NI]*, Nov. 2019.

[19] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *Proc. International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8.

[20] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *arXiv:2001.10402 [cs.IT]*, May 2020.

[21] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980v9 [cs.LG]*, Jan. 2017.

[23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.