Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation*#

Peixi LIU[1,3], Jiamo JIANG[†‡2], Guangxu ZHU[†‡3], Lei CHENG[4,5],
Wei JIANG[1], Wu LUO[1], Ying DU[2], Zhiqin WANG[2]

[1]State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronics, Peking University, Beijing 100871, China

[2]China Academy of Information and Communications Technology, Beijing 100191, China

[3]Shenzhen Research Institute of Big Data, Shenzhen 518172, China

[4]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

[5]Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking, Hangzhou 310027, China

[†]E-mail: jiangjiamo@caict.ac.cn; gxzhu@sribd.cn

Received Nov. 19, 2021; Revision accepted Feb. 10, 2022; Crosschecked Apr. 15, 2022

Abstract: Training a machine learning model with federated edge learning (FEEL) is typically time consuming due to the constrained computation power of edge devices and the limited wireless resources in edge networks. In this study, the training time minimization problem is investigated in a quantized FEEL system, where heterogeneous edge devices send quantized gradients to the edge server via orthogonal channels. In particular, a stochastic quantization scheme is adopted for compression of uploaded gradients, which can reduce the burden of per-round communication but may come at the cost of increasing the number of communication rounds. The training time is modeled by taking into account the communication time, computation time, and the number of communication rounds. Based on the proposed training time model, the intrinsic trade-off between the number of communication rounds and per-round latency is characterized. Specifically, we analyze the convergence behavior of the quantized FEEL in terms of the optimality gap. Furthermore, a joint data-and-model-driven fitting method is proposed to obtain the exact optimality gap, based on which the closed-form expressions for the number of communication rounds and the total training time are obtained. Constrained by the total bandwidth, the training time minimization problem is formulated as a joint quantization level and bandwidth allocation optimization problem. To this end, an algorithm based on alternating optimization is proposed, which alternatively solves the subproblem of quantization optimization through successive convex approximation and the subproblem of bandwidth allocation by bisection search. With different learning tasks and models, the validation of our analysis and the near-optimal performance of the proposed optimization algorithm are demonstrated by the simulation results.

Key words: Federated edge learning; Quantization optimization; Bandwith allocation; Training time minimization
https://doi.org/10.1631/FITEE.2100538 **CLC number:** TN929.5

[‡] Corresponding authors

[*] Project supported by the National Key R&D Program of China (No. 2020YFB1807100), the National Natural Science Foundation of China (No. 62001310), and the Guangdong Basic and Applied Basic Research Foundation, China (No. 2022A1515010109)

[#] Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2100538) contains supplementary materials, which are available to authorized users

ORCID: Peixi LIU, https://orcid.org/0000-0002-9047-8889; Jiamo JIANG, https://orcid.org/0000-0002-4986-7081; Guangxu ZHU, https://orcid.org/0000-0001-9532-9201

1 Introduction

1.1 Background

The evolution of wireless networks from the first generation (1G) to the fifth generation (5G) advanced (5G-advanced) has witnessed a paradigm shift, from connecting people targeting human-type communications towards connecting intelligence targeting machine-type communications, to attain the vision of Artificial Intelligence of Things (AIoT). This has driven the rapid development of an emerging area called edge intelligence, positioned at the intersection of two disciplines, namely artificial intelligence (AI) and wireless communications (Letaief et al., 2019; Park et al., 2019). In edge intelligence, AI technologies are pushed toward the network edge so that the edge servers can quickly access real-time data generated by edge devices for fast training and real-time inference (Zhu et al., 2020b). A promising framework for distributed edge learning, called federated edge learning (FEEL), has recently been pushed into the spotlight; it distributes the task of model training to edge devices and keeps the data locally at the edge devices to avoid data uploading and thus to preserve user privacy (Zhu et al., 2020a; Chen et al., 2021b; Liu and Simeone, 2021). Specifically, a typical training process of FEEL involves multiple rounds of wireless communication between the edge server and devices. In a particular round, the edge server first broadcasts the global model under training to the edge devices for local stochastic gradient descent (SGD) execution using local data, and then the edge devices upload their local models/gradients to the edge server for aggregation and global model updating. After the convergence criterion, such as attaining a desired level of accuracy or reaching a pre-defined value of the loss, is met, the entire training process is completed, and then on-device models can be tweaked for the edge devices' personalization.

In edge networks, the computation resources of the edge devices are constrained, and the wireless resources of the network, e.g., frequency bandwidth, are also limited; therefore, training an AI model by FEEL is usually a time-consuming and expensive task that can take anywhere from hours to weeks (Chen et al., 2021a). Hence, training time (This is also called wall-clock time in some literature (Kairouz et al., 2019; Nori et al., 2021). In this work, we use "total training time," "training time," and "wall-clock time" interchangeably.) minimization of AI models is one of the critical concerns in FEEL. The whole training process of FEEL typically consists of multiple communication rounds, and each round lasts a period of time consisting of computation time and communication time, which we call per-round latency. To reduce the total training time, we should not only bring down the number of communication rounds, i.e., increase the convergence rate of the learning algorithm, but also shorten the per-round latency. Communication-efficient transmission achieved through compression is usually included into the FEEL pipeline to alleviate the transmission burden of the edge devices, and thus the per-round latency is reduced (Park et al., 2021). Two main lossy compression techniques, namely quantization and sparsification, as well as the combination of them, have been considered in the literature (Alistarh et al., 2017; Stich et al., 2018; Wangni et al., 2018; Amiri and Gündüz, 2020a, 2020b; Basu et al., 2020; Reisizadeh et al., 2020; Zhu et al., 2021). Specifically, in the case of quantization, the gradient vector entries are transmitted after being quantized to finite bits, instead of the full floating-point values (Alistarh et al., 2017; Reisizadeh et al., 2020; Zhu et al., 2021). Sparsification reduces the communication overhead by sending only significant entries of the gradient vector (Stich et al., 2018; Wangni et al., 2018). Although the compressed transmission can decrease the per-round latency, the lossy compression degrades the convergence speed of FEEL as well, which leads to an increase in the number of communication rounds needed to achieve a certain accuracy on a given task (Basu et al., 2020). As a result, the compression level balances the number of communication rounds and the per-round latency during minimization of the total training time in FEEL. Besides, edge devices in the edge network are usually heterogeneous, such that some of the edge devices with lower computation power become laggards in the synchronous model/gradient aggregation due to their longer computation time, which increases the per-round latency. It is necessary to consider the optimization of wireless resources over different edge devices to reduce the communication time of the lagging edge devices and compensate for the longer computation time (Dinh et al., 2021; Nguyen et al., 2021; Wan et al., 2021). With the goal of minimizing the total training time, we analyze the following question: how to balance the number of communication rounds and the per-round latency via joint quantization level and bandwidth allocation optimization in the presence of device heterogeneity.

1.2 Related works

Recently, extensive efforts have been made to minimize the total training time of FEEL by resource allocation or gradient/model compression, which fall mainly into three categories in terms of their objectives.

1.2.1 Minimization of the number of communication rounds

This is equivalent to accelerating the convergence of FEEL algorithms. Chen et al. (2021b) minimized the global loss function by optimizing the communication resources, e.g., power and bandwidth, and the computation resources under the given per-round latency constraint, and thus the convergence speed was maximized. Salehi and Hossain (2021) and Wang YM et al. (2022) considered quantization and bandwidth optimization to accelerate the algorithm convergence of FEEL with device sampling in the presence of outage probability. In Chang and Tandon (2020), stochastic gradient quantization was adopted to compress the local gradient, and the quantization levels of each device were optimized to minimize the optimality gap under multiple-access channel capacity constraints. These efforts focused on speeding up the convergence of the FEEL algorithms, i.e., reducing the number of communication rounds, without considering minimization of the total training time, which is a more practical and important issue in FEEL.

1.2.2 Minimization of the per-round latency

Dinh et al. (2021) and Nguyen et al. (2021) studied the trade-off between the per-round latency and the energy consumption by introducing a weight factor, and these two objectives tend to form a competitive interaction. Zhu et al. (2020a) analyzed the per-round latency of different multiple-access schemes in FEEL, i.e., the proposed broadband analog aggregation (BAA) and the traditional orthogonal frequency-division multiple access (OFDMA), and proved that the proposed BAA can significantly reduce the per-round latency compared to the traditional OFDMA. The resource allocation over each single communication round was considered in Zhu et al. (2020a), Dinh et al. (2021), and Nguyen et al. (2021). Nevertheless, FEEL is a long-term process consisting of many communication rounds that determine the total training time.

1.2.3 Minimization of the total training time

Wan et al. (2021) minimized the total training time by optimizing the communication and computation resource allocation; however, no compression was considered therein. Chen et al. (2021a) minimized the training time for a fixed number of communication rounds by solving a joint learning, wireless resource allocation, and device selection problem. Some other works did not minimize the training time directly, but the minimization of the loss function in a given training time was considered. For example, Nori et al. (2021) studied the communication trade-off between compression and local update steps in a fixed training time, but communication resource allocation was not taken into consideration. Jin et al. (2020) adopted the idea of signSGD with majority vote (Bernstein et al., 2018) and optimized the power allocations and central processing unit (CPU) frequencies under the trade-off between the number of communication rounds and the outage probability per communication round for a fixed training time.

Despite the above research efforts, these prior works have overlooked the inherent trade-off in minimizing the total training time of communication-efficient FEEL between the number of communication rounds and the per-round latency, which is balanced by the quantization level at the edge devices. Moreover, the communication resource allocation among different edge devices and the compression setup for minimizing the total training time of communication-efficient FEEL should be jointly considered. This thus motivates the current work.

1.3 Our contributions

This paper studies a FEEL system consisting of multiple edge devices with heterogeneous computational capabilities and one edge server for coordinating the learning process. We consider quantized FEEL, whereby a stochastic quantization scheme is adopted for updated gradient compression, which can save per-round communication cost but may come at the cost of increasing the number of communication rounds. Thus, we make a comprehensive analysis of the total training time by taking into account the communication time, computation time, and the number of communication rounds, based on which the intrinsic trade-off between the number of communication rounds and the per-round latency is characterized. Then, based on the analytical results, a joint quantization and bandwidth allocation optimization problem is formulated and solved. The main contributions are elaborated as follows:

1. Training time analysis in quantized FEEL

The challenge of analyzing the total training time lies mainly in estimating the number of required communication rounds for model convergence. To tackle this challenge, we analyze the convergence behavior of quantized FEEL in terms of the optimality gap and establish the expression for the minimum number of communication rounds to obtain a particular optimality gap of the loss function. However, the derived results are generally too loose to be used for further optimization. To yield an accurate estimate of the number of required communication rounds, we propose a joint data-and-model-driven fitting method to further refine and tighten the derived results. Owing to the refinement, an accurate estimate of the total training time can be attained and the trade-off between the number of communication rounds and the per-round latency can be better characterized.

2. Training time minimization via joint optimization of quantization and bandwidth

Next, based on the derived analytical results, we formulate the total training time minimization problem by jointly optimizing the quantization level and bandwidth allocation, subject to a maximum bandwidth constraint in the FEEL network. The problem is non-convex, and thus is challenging to solve. To tackle this challenge, we adopt the alternating optimization technique to decompose the problem into two subproblems, and each optimizes one of the two control variables with the other variable fixed. The subproblem of bandwidth allocation with a fixed quantization level can be solved by bisection search efficiently. For the subproblem of quantization optimization with bandwidth allocation fixed, an algorithm based on successive convex approximation (SCA) is proposed.

3. Performance evaluation

Finally, we conduct extensive simulations to evaluate the performance of task-oriented resource allocation for quantized FEEL by considering logistic regression (convex loss function) on a synthetic dataset and a convolutional neural network (non-convex loss function) on the Canadian Institute For Advanced Research (CIFAR)-10 dataset. It is shown that the results of the proposed joint data-and-model-driven fitting method fit the curve of the actual optimality gap well. In addition, it is shown that the optimal quantization level found by solving the formulated optimization problem matches well the simulation results. The benefits of optimizing the bandwidth allocation for coping with device heterogeneity are also demonstrated.

Notations $\mathbb{R}$ represents the set of real numbers. $[K]$ denotes the set $\{1, 2, \ldots, K\}$. $\varnothing$ denotes the empty set. $\operatorname{sgn}(\cdot)$ denotes the sign of a scalar. $\boldsymbol{w}^{\mathrm{T}}$ is the transpose of vector $\boldsymbol{w}$. $\nabla f(\boldsymbol{w})$ denotes the gradient of function f at point $\boldsymbol{w}$. $\|\boldsymbol{w}\|$ denotes the ℓ_2 norm of vector $\boldsymbol{w}$. $\lceil x \rceil$ is the ceiling operator. $x \sim \mathcal{CN}(0, \sigma^2)$ denotes the zero-mean circularly symmetric complex Gaussian (CSCG) random variable with the variance of σ^2.

2 System model

2.1 Federated learning

We consider a quantized FEEL system consisting of K edge devices and a single edge server, as shown in Fig. 1. With the coordination of the edge server, the edge devices collaboratively train a shared model, which is represented by the parameter vector $\boldsymbol{w} \in \mathbb{R}^d$, with d denoting the model size. The training process is performed to minimize the following empirical loss function:

$$F(\boldsymbol{w}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\boldsymbol{w}),$$

where $F_k(\boldsymbol{w})$ denotes the local loss function at edge device k, $k \in [K]$. Suppose that device k holds the training dataset $\mathcal{D}_k$ with a uniform size of D, i.e., $|\mathcal{D}_k| = D$. The local loss function $F_k(\boldsymbol{w})$ is given by the following expression:

$$F_k(\boldsymbol{w}) = \frac{1}{D} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_k} f(\boldsymbol{w}; \boldsymbol{x}_i, y_i), \tag{1}$$

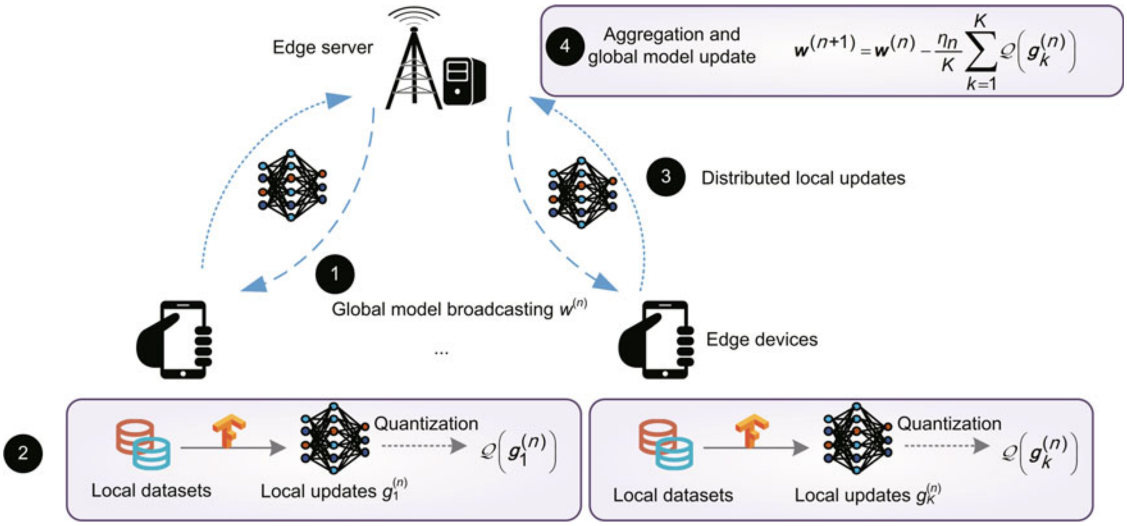
Edge server
4 Aggregation and global model update
$w^{(n+1)}=w^{(n)}-\frac{\eta_n}{K}\sum_{k=1}^{K}\mathcal{Q}\left(g_k^{(n)}\right)$
3 Distributed local updates
1 Global model broadcasting $w^{(n)}$
...
Edge devices
2
Local datasets
Local updates $g_1^{(n)}$
Quantization
$\mathcal{Q}\left(g_1^{(n)}\right)$
Local datasets
Local updates $g_k^{(n)}$
Quantization
$\mathcal{Q}\left(g_k^{(n)}\right)$

Fig. 1 Quantized federated edge learning (FEEL) system with K edge devices

where $f(\boldsymbol{w};\boldsymbol{x}_i,y_i)$ denotes the sample-wise loss function specified by the learning task and quantifies the training loss of model $\boldsymbol{w}$ on the training data $\boldsymbol{x}_i$ and its ground-true label y_i.

In FEEL, the training process is implemented iteratively in a distributed manner using the federated SGD (FedSGD) algorithm as elaborated in the following paragraphs. Besides FedSGD, federated averaging (FedAvg) is another approach for FEEL, which enables multiple local updates at edge devices together while updating the model updates at the edge server. This study considers FedSGD (instead of FedAvg), mainly to facilitate the convergence analysis and to obtain insightful results. Nevertheless, following the analytical work in Reisizadeh et al. (2020), FedAvg can be mathematically converted to biased FedSGD. By exploiting such a relationship between FedSGD and FedAvg, the analysis presented in this work can be readily extended to the case with FedAvg. Consider a particular iteration or communication round n, all the devices first download the current model $\boldsymbol{w}^{(n)}$ from the server. Then, each device computes a local stochastic gradient, $\boldsymbol{g}_k^{(n)}$, using a randomly chosen mini-batch of samples from dataset $\mathcal{D}_k$ in a uniform manner. We denote the set of mini-batch samples used by device k at round n as $\tilde{\mathcal{D}}_k^{(n)}$ and the size of each mini-batch as $m_{\rm b}$. Then we have

$$\boldsymbol{g}_k^{(n)}=\frac{1}{m_{\rm b}}\sum_{(\boldsymbol{x}_i,y_i)\in\tilde{\mathcal{D}}_k^{(n)}}\nabla f\left(\boldsymbol{w}^{(n)};\boldsymbol{x}_i,y_i\right)+\lambda\nabla R\left(\boldsymbol{w}^{(n)}\right). \tag{2}$$

Next, each device transmits a quantized version of its local gradient, i.e., $\mathcal{Q}\left(\boldsymbol{g}_k^{(n)}\right)$, to the edge server. The quantization scheme will be elaborated in Section 2.2. Upon reception of these data, the edge server aggregates the local gradients and updates the global model as follows:

$$\boldsymbol{w}^{(n+1)}=\boldsymbol{w}^{(n)}-\frac{\eta_n}{K}\sum_{k=1}^{K}\mathcal{Q}\left(\boldsymbol{g}_k^{(n)}\right),$$

where η_n denotes the learning rate at round n. Then the updated global model is broadcasted back to all edge devices for initializing the next round of training. The above procedure continues until the convergence criterion is met.

2.2 Stochastic quantization

We consider a widely used stochastic quantizer for local gradient quantization (Alistarh et al., 2017). For any arbitrary vector $\boldsymbol{g}\in\mathbb{R}^d$, the stochastic quantizer $\mathcal{Q}(\boldsymbol{g}):\mathbb{R}^d\to\mathbb{R}^d$ is defined as

$$\mathcal{Q}\left(g_i\right)=\|\boldsymbol{g}\|\cdot\operatorname{sgn}\left(g_i\right)\cdot\xi_i\left(\boldsymbol{g},q\right),\ \forall i\in[d], \tag{3}$$

where the output of quantizer $\mathcal{Q}(\boldsymbol{g})$ consists of three parts, i.e., the vector norm $\|\boldsymbol{g}\|$, the sign of each entry $\operatorname{sgn}(g_i)$ with g_i denoting the $i^{\rm th}$ entry of $\boldsymbol{g}$, and the quantization value of each entry $\xi_i(\boldsymbol{g},q)$. Further, the terms $\xi_i\left(\boldsymbol{g},q\right)$ are independent random variables defined as

$$\xi_i\left(\boldsymbol{g},q\right)=\begin{cases}\frac{l+1}{q}, & \text{with probability } \frac{|g_i|}{||\boldsymbol{g}||}q-l,\\ \frac{l}{q}, & \text{otherwise.}\end{cases}$$

Here, q denotes the quantization level and $0 \le l < q$ is an integer such that $\frac{g_i}{\|\boldsymbol{g}\|} \in \left[\frac{l}{q}, \frac{l+1}{q}\right)$.

As proved in Lemma 3.1 in Alistarh et al. (2017), the random quantizer $\mathcal{Q}(\boldsymbol{g})$ is unbiased, i.e., $\mathbb{E}[\mathcal{Q}(\boldsymbol{g})] = \boldsymbol{g}$ for any given vector $\boldsymbol{g}$. Moreover, assuming that $d \ge q^2$, the quantizer has a bounded variance, i.e., $\mathbb{E}[\|\mathcal{Q}(\boldsymbol{g}) - \boldsymbol{g}\|]^2 \le \frac{\sqrt{d}}{q}\|\boldsymbol{g}\|^2$. Note that we do not claim that this quantization scheme is optimal in terms of the communication efficiency. Rather, we adopt it as a simple and general scheme that facilitates the subsequent analysis of the tradeoff between the number of communication rounds and the per-round latency, controlled by the quantization level.

2.3 Wireless transmission model

In the quantized FEEL system, each edge device connects to the edge server through a shared wireless medium. We assume that the spectrum is divided into distinct and non-overlapping flat fading channels with different bandwidths, so that the edge devices share the spectrum through frequency-division multiple access to avoid interferences with each other. In general, modern neural network models are of high dimension, with d in the order of 10^6–10^9. Hence, it usually takes much time (longer than the coherence period) to transmit a complete model. For example, a single long-term evolution (LTE) frame of 5 MHz bandwidth and 10 ms duration can carry only 6000 complex symbols (Amiri and Gündüz, 2020a). To transmit a moderate neural network model with 10^6 parameters encoded by 32-bit floating-point values, it will take approximately 6 s, which is much longer than the frame length, i.e., 10 ms. Moreover, in Internet of Things (IoT) networks, which are typically limited by bandwidth and power (Dhillon et al., 2017), it takes more time to transmit a machine learning model. Hence, it is reasonable to model the wireless uplink channels as independent and identically distributed (i.i.d.) fast Rayleigh fading channel, in the course of training; i.e., the channel coefficients remain constant over each coherence period and vary in an i.i.d. manner across different coherence periods, and the codeword or frame will span multiple coherence periods (Tse and Viswanath, 2005). Specifically, the channel propagation coefficient between the edge server and device k is generally modeled as $h_k = \sqrt{\phi_k}\overline{h}_k$; here, ϕ_k describes the large-scale propagation effects, including path loss and shadowing, and $\overline{h}_k$ denotes the small-scale fading modeled as i.i.d. CSCG random variables with zero mean and unit variance, i.e., $\overline{h}_k \sim \mathcal{CN}(0,1)$. The large-scale propagation coefficient ϕ_k remains unchanged in the whole time frame, while the small-scale fading $\overline{h}_k$ varies from one coherence block to another in a time frame. Moreover, we assume that the channel coefficients, which can be obtained by channel estimation at the server, are known only at the edge server.

In this situation, the ergodic capacity can be assigned to the fast fading channel and achieved in practice by the interleaving technique (Tse and Viswanath, 2005). The ergodic capacity of device k is given by

$$R_k = \mathbb{E}_{h_k}\left[b_k \log_2\left(1 + \frac{p_k|h_k|^2}{b_k N_0}\right)\right], \tag{4}$$

where b_k denotes the frequency bandwidth allocated for device k with $\sum_{k=1}^{K} b_k = B_0$, p_k denotes the transmit power at device k, N_0 denotes the noise power spectral density, and the expectation is taken over the channel distribution.

3 Training time analysis

The training time of each device for one communication round comprises computation time T_k^{comp} and communication time T_k^{comm}, as illustrated in Fig. 2. Since the server broadcasts the same global model to all the devices using the entire frequency bandwidth, the downlink delay due to global model broadcasting is ignorable compared with the uplink delay due to uploading of updates from many devices to the server. Assume that the delay requirement for running one round of training is T_{d}, i.e., $T_k^{\mathrm{comp}} + T_k^{\mathrm{comm}} \le T_{\mathrm{d}}$. We define N_ϵ as the minimum number of communication rounds when the ϵ-optimality gap is achieved, i.e., $F(\boldsymbol{w}^{(N_\epsilon)}) - F(\boldsymbol{w}_*) \le \epsilon$, where $\boldsymbol{w}_*$ denotes the optimal model expressed as $\boldsymbol{w}_* = \arg\min_{\boldsymbol{w}} F(\boldsymbol{w})$. The training process stops when the ϵ-optimality gap is achieved. Then, the requirement of the total training time over N_ϵ rounds is given by the following expression:

$$T = N_\epsilon T_{\mathrm{d}}. \tag{5}$$

In the following subsections, we give the expression of per-round training time T_{d} and obtain the minimum number of communication rounds N_ϵ by analyzing the convergence of FEEL, which paves the way for minimizing the total training time in Section 4.

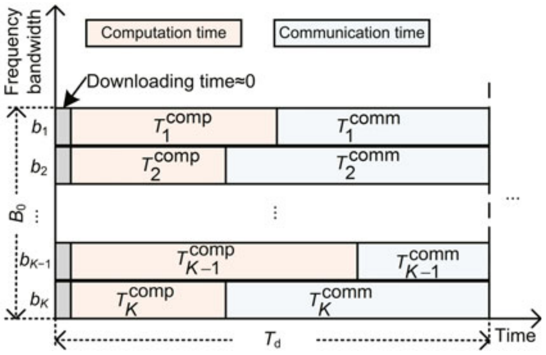

Fig. 2 Computation time and communication time in one communication round of quantized FEEL

3.1 Computation time

Let ν denote the number of processing cycles for one particular edge device to execute the SGD operation on one batch of samples, and f_k the CPU frequency of device k. Accordingly, the computation time for running one-round SGD at device k is given by the following expression (Ren et al., 2020):

$$T_k^{\text{comp}} = \frac{\nu}{f_k}.$$

3.2 Communication time

Let S denote the number of bits for transmission after stochastic quantization. For quantizing any element g_i in gradient vector $\boldsymbol{g} \in \mathbb{R}^d$, as noted in Eq. (2), we need to encode the vector norm $\|\boldsymbol{g}\|$, the element-wise sign $\operatorname{sgn}(g_i)$, and the normalized quantization value $\xi_i(\boldsymbol{g}, q)$ into bits. Particularly, it takes one bit to encode each of the $\operatorname{sgn}(g_i)$. Since $\xi_i(\boldsymbol{g}, q)$ takes value from $\{0, 1/q, 2/q, \ldots, 1\}$, it takes at most $\log_2(1+q)$ bits to encode each $\xi_i(\boldsymbol{g}, q)$ (Cover and Thomas, 2006). Since each vector contains d entries, it takes totally $(1+\log_2 q)d$ bits to encode these two parts. By contrast, the overhead in encoding the single scalar vector norm $\|\boldsymbol{g}\|$ is typically negligible for large models of size d (Shlezinger et al., 2021). To facilitate subsequent analysis, for large d, we approximate S using the following expression:

$$S = (1 + \log_2(q+1))\, d. \tag{6}$$

The communication delay in one round is

$$T_k^{\text{comm}} = \frac{S}{R_k}, \tag{7}$$

where R_k is the ergodic capacity defined in Eq. (4).

3.3 Minimum number of communication rounds

In this subsection, we derive N_ϵ by analyzing the convergence of quantized FEEL. To this end, we make the following assumptions on the local loss functions $\{F_k(\boldsymbol{w})\}$:

Assumption 1 (Smoothness) The local loss functions $\{F_k(\boldsymbol{w})\}$ are all L-smooth: for all $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$, $F_k(\boldsymbol{w}_i) \leq F_k(\boldsymbol{w}_j) + (\boldsymbol{w}_i - \boldsymbol{w}_j)^{\mathrm{T}} \nabla F_k(\boldsymbol{w}_j) + \frac{L}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2$, $\forall k$.

Assumption 2 (Strong convexity) The local loss functions $\{F_k(\boldsymbol{w})\}$ are all μ-strongly convex: for all $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$, $F_k(\boldsymbol{w}_i) \geq F_k(\boldsymbol{w}_j) + (\boldsymbol{w}_i - \boldsymbol{w}_j) + (\boldsymbol{w}_i - \boldsymbol{w}_j)^{\mathrm{T}} \nabla F_k(\boldsymbol{w}_j) + \frac{\mu}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2$, $\forall k$.

Assumption 3 (First and second moments of local gradients) The mean and variance of stochastic gradients $\boldsymbol{g}_k^{(n)}$ of the local loss functions $F_k(\boldsymbol{w})$, $\forall n \in [N]$ and $\forall k$, satisfy the following conditions:

$$\text{(Unbiased gradient) } \mathbb{E}[\boldsymbol{g}_k^{(n)}] = \nabla F_k(\boldsymbol{w}^{(n)}),$$
$$\text{(Bounded variance) } \mathbb{E}\left[\left\|\boldsymbol{g}_k^{(n)} - \nabla F_k\left(\boldsymbol{w}^{(n)}\right)\right\|^2\right] \leq \delta_k^2.$$

Assumptions 1 and 2 on local loss functions are standard, and they can be satisfied by many typical learning models, such as logistic regression, linear regression, and softmax classifier. Assumption 3 is general enough to cope with both i.i.d. and non-i.i.d. data distributions across edge devices, following the work in Li et al. (2020), Luo et al. (2021), and Salehi and Hossain (2021). Under Assumptions 1–3, the convergence rate of quantized FEEL is established in the following theorem:

Theorem 1 Consider a quantized FEEL system with fixed quantization level $q \geq 2$. The optimality gap of the loss function after N communication rounds is upper bounded by

$$\begin{aligned} &\mathbb{E}\left[F\left(\boldsymbol{w}^{(N)}\right)\right] - F(\boldsymbol{w}_*) \\ \leq &\frac{\alpha\kappa}{N + 2\alpha\kappa - 1}\left(L\left\|\boldsymbol{w}^{(0)} - \boldsymbol{w}_*\right\|^2 + \frac{2\mathit{\Gamma}}{\mu}\right), \end{aligned}$$

where $\alpha = \frac{\sqrt{d}}{qK} + 1$, $\kappa = \frac{L}{\mu}$, $F_\delta = F(\boldsymbol{w}_*) - \frac{1}{K}\sum_{k=1}^K F_k^*$ with $F_k^* = \min\limits_{\boldsymbol{w}} F_k(\boldsymbol{w})$, $\mathit{\Gamma} = 2LF_\delta + \frac{1}{K}\sum_{k=1}^K \delta_k^2$, and

$\boldsymbol{w}^{(0)}$ is the initial point of the training process. The learning rate is set to be a diminishing one, i.e., $\eta_n = \frac{2}{\mu(n+2\alpha\kappa-1)}$.

Proof The proof of Theorem 1 can be found in the supplementary materials.

Remark 1 (Convergence rate) Theorem 1 quantizes the impact of gradient quantization on the convergence rate of the FEEL, which is captured by the term $\alpha = \frac{\sqrt{d}}{qK} + 1$. An aggressive quantization scheme, e.g., with a small q, will lead to an enlarged optimality gap and thus needs more rounds to converge. Nevertheless, the quantized FEEL can still achieve the asymptotic convergence rate of $\mathcal{O}(\frac{1}{N})$ as federated learning without quantization (Li et al., 2020).

Although we can readily establish a bound on N_ϵ based on Theorem 1 and apply this bound for subsequent resource optimization as was done in prior works, e.g., Wang SQ et al. (2019) and Luo et al. (2021), such an approach has two key drawbacks: First, the gap between the derived bound and its true value could be large as several relaxations are made in deriving the bound. Thus, ignoring the gap may lead to a highly suboptimal solution in the subsequent resource optimization stage. Second, even if we adopt the upper bound and ignore the effects of the gap, it is still difficult to obtain the exact value of the upper bound, since it involves calculating a bunch of data- and model-related parameters, such as μ, L, F_δ, $\boldsymbol{w}_*$, and Γ.

To address the above issues, we propose a joint data-and-model-driven fitting approach, which uses a small number of pre-training rounds to yield a good estimate of the optimality gap based on the bound derived in Theorem 1. To this end, we first denote the upper bound function derived in Theorem 1 as follows:

$$\hat{U}(N) = \frac{\alpha\kappa\left(L\left\|\boldsymbol{w}_0 - \boldsymbol{w}_*\right\|^2 + \frac{2\Gamma}{\mu}\right)}{N + 2\alpha\kappa - 1}.$$

Before we derive the tight estimate of the optimality gap, it is first observed that the upper bound function $\hat{U}(N)$ satisfies the following properties:

(1) $\hat{U}(N)$ is a decreasing function of N, and it converges to zero at a rate of $\mathcal{O}(\frac{1}{N})$;

(2) $\hat{U}(N)$ has a fractional structure, where the numerator and denominator are both linear increasing functions of α.

We assume that the ground-true optimality gap follows the same properties as its upper bound $\hat{U}(N)$. Based on the above assumption, the exact optimality gap can be well estimated by the following function:

$$\mathbb{E}\left[F(\boldsymbol{w}^{(N)})\right] - F(\boldsymbol{w}_*) = \frac{\alpha A + D}{n + \alpha B + C} \triangleq U(N), \quad (8)$$

where $A > 0$, $B > 0$, $C \geq 0$, and $D \geq 0$ are the tuning parameters to be fitted and implicitly related to parameters such as μ, L, F_δ, and Γ. It can be seen that $U(N)$ generalizes all the functions that satisfy the two properties mentioned above.

Next, we apply a joint data-and-model-driven fitting method to fit the values of the tuning parameters as follows: First, we randomly choose two quantization levels, say q_1 and q_2, and run the quantized FEEL with q_1 and q_2, separately, from an initial model $\boldsymbol{w}_0$. Then, we sample the value of loss at each round until the number of communication rounds reaches a pre-defined value $\tilde{N}$. The corresponding loss values are denoted as $F_{i,n}$ ($i \in \{1,2\}$, $n \in [1, \tilde{N}]$) for round n when the quantization level is q_i. According to Eq. (8), we have

$$F_{i,n} - Z \approx \frac{X_i}{n + Y_i}, \ \forall i \in \{1,2\}, \ n \in [1, \tilde{N}], \quad (9)$$

where $\alpha_i = \frac{\sqrt{d}}{q_i K} + 1$, $Z = F(\boldsymbol{w}_*)$,

$$X_i = \alpha_i A + D, \quad (10)$$

$$Y_i = \alpha_i B + C. \quad (11)$$

Then we aim to find the proper values of X_i, Y_i, and Z to fit expression (9) well. The method we choose is to solve the following nonlinear regression problem:

$$\min_{X_i, Y_i, Z} \sum_{i=1}^{2}\sum_{n=1}^{\tilde{N}} \big((F_{i,n} - Z)(n + Y_i) - X_i\big)^2. \quad (12)$$

For any fixed Z, this problem can be divided into two linear regression problems, i.e.,

$$\min_{X_i, Y_i} \sum_{n=1}^{\tilde{N}} \big((F_{i,n} - Z)(n + Y_i) - X_i\big)^2, \quad (13)$$

and the optimal $\{X_i\}$ and $\{Y_i\}$ can be given by

$$X_i = \frac{\sum_{n=1}^{\tilde{N}} \chi_{i,n} \sum_{n=1}^{\tilde{N}} \psi_{i,n}^2 - \sum_{n=1}^{\tilde{N}} \chi_{i,n}\psi_{i,n} \sum_{n=1}^{\tilde{N}} \psi_{i,n}}{N\sum_{n=1}^{\tilde{N}} \psi_{i,n}^2 - \left(\sum_{n=1}^{\tilde{N}} \psi_{i,n}\right)^2}, \quad (14)$$

$$Y_i = \frac{\sum_{n=1}^{\tilde{N}} \chi_{i,n} \sum_{n=1}^{\tilde{N}} \psi_{i,n} - N \sum_{n=1}^{\tilde{N}} \chi_{i,n}\psi_{i,n}}{N \sum_{n=1}^{\tilde{N}} \psi_{i,n}^2 - \left(\sum_{n=1}^{\tilde{N}} \psi_{i,n}\right)^2}, \tag{15}$$

where $\chi_{i,n} = (F_{i,n} - Z)n$ and $\psi_{i,n} = F_{i,n} - Z$. Hence, problem (12) can be solved by a one-dimensional search of Z. Since the computations in Eqs. (14) and (15) involve only limited algebraic operations, the computation time for solving problem (12) is negligible compared with the time needed in the whole training process. With $\{X_i\}$ and $\{Y_i\}$ at hand, A, B, C, and D can be obtained from Eqs. (10) and (11) as follows:

$$\begin{cases} A = \dfrac{X_1 - X_2}{\alpha_1 - \alpha_2}, \\ B = \dfrac{Y_1 - Y_2}{\alpha_1 - \alpha_2}, \\ C = \dfrac{\alpha_2 Y_1 - \alpha_1 Y_2}{\alpha_2 - \alpha_1}, \\ D = \dfrac{\alpha_2 X_1 - \alpha_1 X_2}{\alpha_2 - \alpha_1}. \end{cases}$$

Based on the estimated optimality gap in Eq. (8) with the well-fitted parameters, we can derive N_ϵ in the following proposition:

Proposition 1 In the quantized FEEL system, the minimum number of communication rounds needed to achieve the ϵ-optimality gap is given by

$$N_\epsilon = \left\lceil \left(\frac{\sqrt{d}}{qK} + 1\right)\left(\frac{A}{\epsilon} - B\right) + \frac{D}{\epsilon} - C \right\rceil. \tag{16}$$

Proof By setting the fitted optimality gap in Eq. (8) to be less than ϵ, i.e., $U(N) \leq \epsilon$, and respecting the fact that the minimum number of communication rounds should be an integer, we obtain Eq. (16).

Remark 2 (Impact of the quantization level and device number) Proposition 1 unveils the impact of the quantization level and the number of participating devices on the minimum number of communication rounds as reflected by the term $\frac{\sqrt{d}}{qK} + 1$. On one hand, N_ϵ decreases with the increase in the quantization level q. This is due to the fact that increasing the quantization levels leads to smaller quantization errors, which speeds up the convergence. On the other hand, we can observe that as the number of devices tends to infinity, i.e., $K \to \infty$, the impact of quantization diminishes since the quantization errors average out due to the update aggregation mechanism. Moreover, we can obtain $N_\epsilon = \left\lceil \frac{A+D}{\epsilon} - B - C \right\rceil$ when $q \to \infty$ or $K \to \infty$. In other words, $\left\lceil \frac{A+D}{\epsilon} - B - C \right\rceil$ can be used to evaluate the minimum number of communication rounds under high-resolution quantization or a sufficiently large number of devices, and this value offers us a lower bound of the minimum number of communication rounds under practical setup of quantization levels and number of devices.

4 Training time minimization

In this section, we aim to jointly optimize the quantization level and bandwidth allocation by minimizing the training time defined in Eq. (5) to achieve an ϵ-optimality gap.

4.1 Problem formulation

The training time minimization problem is mathematically formulated as follows:

$$\text{(P1)} \quad \min_{q\in\mathbb{Z}^+, \{b_k\}, T_\mathrm{d}} \; T_\mathrm{d} N_\epsilon \tag{17}$$

$$\text{s.t. } T_k^{\text{comp}} + T_k^{\text{comm}} \leq T_\mathrm{d}, \ \forall k \in [K], \tag{17a}$$

$$\sum_{k=1}^{K} b_k = B_0, \tag{17b}$$

$$q \geq 2, \tag{17c}$$

where the objective function in problem (17) is the total training time needed to achieve the ϵ-optimality gap. Constraint (17a) indicates that the training time of each device per communication round cannot exceed the delay requirement T_d. Constraint (17b) means that the total bandwidth allocated to all the devices is B_0. The constraint on the quantization level q is described by constraint (17c).

The objective function in problem (17) is complicated due to the coupling of the control variables T_d and q. Moreover, q can take values only from positive integers. Therefore, problem P1 is non-convex, and is challenging to solve optimally. To yield a good solution to problem P1, we divide it into two subproblems: One is to find the optimal bandwidth allocation $\{b_k\}$ and T_d with fixed quantization level q, and the other is to find the optimal quantization level q with fixed bandwidth allocation $\{b_k\}$ and T_d. We find that the first subproblem can be solved optimally and efficiently with a unique solution, and the second one can be transformed into a non-convex problem that can be solved by the method of SCA (Razaviyayn, 2014). Then, by alternatively solving each subproblem, we can obtain good sub-optimal solutions for joint quantization level and bandwidth allocation optimization.

Remark 3 (Trade-off between the minimum number of communication rounds and the per-round latency) As noted in Remark 2, the minimum number of communication rounds N_ϵ can be reduced by increasing the quantization level q, but at a cost of increasing the per-round latency. Therefore, when minimizing the total training time, there exists a fundamental trade-off between reducing the minimum number of communication rounds and suppressing the per-round latency. The trade-off is manipulated by the setting of the quantization level q.

Remark 4 (Resource allocation over heterogeneous devices in FEEL) The computation time of the devices varies due to their heterogeneous computation capacities. To enforce the per-round latency constraint, a wider frequency bandwidth should be allocated to the devices with lower computation power to compensate for the long computation time with short communication time, and vice versa. Hence, the bandwidth allocation among the devices should jointly account for the channel condition and the computation resources, which is in sharp contrast to the classic bandwidth allocation problem accounting for only the channel condition, e.g., the problem in Gong et al. (2011).

4.2 Bandwidth allocation optimization

Since N_ϵ is independent of $\{b_k\}$ and T_d, problem (17) under a fixed quantization level q is reduced to

$$\text{(P2)} \min_{\{b_k\},T_\mathrm{d}} T_\mathrm{d} \tag{18}$$

$$\text{s.t. } T_k^\text{comp} + T_k^\text{comm} \le T_\mathrm{d},\ \forall k \in [K], \tag{18a}$$

$$\sum_{k=1}^{K} b_k = B_0. \tag{18b}$$

Since $T_k^\text{comm} = \frac{S}{R_k}$ as defined in Eq. (7), constraint (18a) can be rewritten as

$$T_k^\text{comp} + \frac{S}{R_k} \le T_\mathrm{d},\ \forall k \in [K].$$

To obtain a closed-form expression of R_k, it can be rewritten as

$$\begin{aligned} R_k &= \int_0^{+\infty} b_k \log_2\left(1 + \frac{p_k x}{b_k N_0}\right) f_{|h_k|^2}(x)\mathrm{d}x \\ &= \frac{b_k}{\ln 2}\frac{p_k}{b_k N_0}\int_0^{+\infty} \frac{1 - F_{|h_k|^2}(x)}{1 + \frac{p_k x}{b_k N_0}}\mathrm{d}x, \end{aligned}$$

where $f_{|h_k|^2}(x)$ and $F_{|h_k|^2}(x)$ are the probability density function (PDF) and the cumulative distribution function (CDF), respectively, of the random variable $|h_k|^2$. It can be verified that $|h_k|^2$ follows an exponential distribution, i.e., $|h_k|^2 \sim \exp(1/\phi_k)$. Hence, we have $F_{|h_k|^2}(x) = 1 - \mathrm{e}^{-x/\phi_k}$. Then, R_k can be calculated as follows:

$$\begin{aligned} R_k &= \frac{b_k}{\ln 2}\frac{p_k}{b_k N_0}\int_0^{+\infty} \frac{\mathrm{e}^{-x/\phi_k}}{1 + \frac{p_k x}{b_k N_0}}\mathrm{d}x \\ &= \frac{b_k}{\ln 2}\int_0^{+\infty} \frac{\mathrm{e}^{-x/\phi_k}}{x + \frac{b_k N_0}{p_k}}\mathrm{d}x. \end{aligned}$$

According to Section 8.212 in Gradshteyn and Ryzhik (2014), we have, for real numbers a and $b > 0$, $\int_0^{+\infty} \frac{\mathrm{e}^{-bx}}{a+x}\mathrm{d}x = -\mathrm{e}^{ab}\mathrm{Ei}(-ab)$, where $\mathrm{Ei}(x) = \int_{-\infty}^{x} \frac{\mathrm{e}^{\rho}}{\rho}\mathrm{d}\rho$ is the exponential integral function. R_k can be rewritten in the closed form as follows:

$$R_k = -\frac{b_k}{\ln 2}\mathrm{e}^{b_k\theta_k}\mathrm{Ei}(-b_k\theta_k), \tag{19}$$

where $\theta_k = \frac{N_0}{p_k\phi_k}$.

It can be verified that the transmission rate R_k in Eq. (19) is an increasing function of b_k, denoted as $R_k(b_k)$. Hence, $T_k^\text{comm} = \frac{S}{R_k(b_k)}$ decreases with increasing b_k. The following lemma will be beneficial for solving problem P2:

Lemma 1 Constraint (18a) in problem P2 can be replaced by

$$T_k^\text{comp} + T_k^\text{comm} = T_\mathrm{d},\ \forall k \in [K].$$

Proof The proof can be found in the supplementary materials.

From Lemma 1, each b_k can be represented as a function of T_d, i.e.,

$$b_k(T_\mathrm{d}) = R_k^{-1}\left(\frac{S}{T_\mathrm{d} - T_k^\text{comp}}\right), \tag{20}$$

where $R_k^{-1}(\cdot)$ denotes the inverse function of $R_k(\cdot)$. Since $\sum_{k=1}^{K} b_k = B_0$ holds, we can find T_d by solving the equation as follows:

$$\sum_{k=1}^{K} b_k(T_\mathrm{d}) = B_0. \tag{21}$$

It can be verified that $b_k(T_\mathrm{d})$ is a decreasing function of T_d. Therefore, Eq. (21) can be efficiently solved by bisection search. Note that although $R_k(\cdot)$ is an increasing function with closed-form expression, it is nontrivial to obtain a tractable expression of $R_k^{-1}(\cdot)$, so we cannot obtain b_k with given T_d directly from Eq. (20). Instead, b_k with given T_d can be obtained by solving

$$R_k(b_k) = \frac{S}{T_\mathrm{d} - T_k^\mathrm{comp}},$$

with the tool of bisection search. This is feasible due to the monotonicity of R_k.

In consequence, problem P2 can be solved by two-layer bisection search, as summarized in Algorithm 1. In the outer layer bisection, we search T_d in the range of $[T_\mathrm{d}^-, T_\mathrm{d}^+]$, where $T_\mathrm{d}^- = \max\limits_k\{T_k^\mathrm{comp}\}$ and $T_\mathrm{d}^+ = \max\limits_k\{T_k^\mathrm{comp} + R_k(B_0/K)\}$. The inner layer bisection at step 5 is implemented in the range of $[0, B_0]$. Since only the search of single variable b_k is involved in each bisection, the process is straightforward, and thus we omit the detailed steps for simplicity.

4.3 Quantization level optimization

Next, we focus on optimizing the quantization level q under fixed bandwidth allocation. First, similar to Wang YM et al. (2022), we relax the value of q from integer to real number in the interval $q \in [2, +\infty)$. For the convenience of optimization, we approximate N_ϵ in Proposition 1 as $N_\epsilon \approx \frac{\sqrt{d}}{qK}H_1 + H_2$, where $H_1 = \frac{A}{\epsilon} - B$ and $H_2 = \frac{A+D}{\epsilon} - B - C$, by getting rid of the ceiling operation. Then, we introduce an intermediate variable $\tilde{T}$, and problem P1 is reduced to the following:

$$\text{(P3)} \quad \min_{q,\tilde{T}} \tilde{T} \tag{22}$$

$$\text{s.t.} \ (T_k^\mathrm{comp} + T_k^\mathrm{comm})\left(\frac{\sqrt{d}}{qK}H_1 + H_2\right) \le \tilde{T}, \tag{22a}$$

$$q \ge 2. \tag{22b}$$

Problem P3 is non-convex due to the nonconvexity of constraint (22a). To tackle this problem, the SCA technique can be applied to obtain a stationary point (Razaviyayn, 2014). The above procedure is summarized in Algorithm 2. The key idea is that at each iteration, the original problem is approximated by a tractable convex one at a given local point as elaborated below. To start with, we substitute $T_k^\mathrm{comm} = \frac{(1+\log_2(1+q))d}{R_k}$ into constraint (22a), and obtain

$$\left(T_k^\mathrm{comp} + \frac{(1+\log_2(1+q))\,d}{R_k}\right)\left(\frac{\sqrt{d}}{qK}H_1 + H_2\right) \le \tilde{T}.$$

After taking the logarithm of both sides and rearrangement, it yields

$$J_k(q) - \ln(qK) - \ln\tilde{T} \le 0, \tag{23}$$

where $J_k(q) = \ln\left(T_k^\mathrm{comp} + \frac{(1+\log_2(1+q))d}{R_k}\right) + \ln\left(qKH_2 + H_1\sqrt{d}\right)$. It can be verified that $J_k(q)$ is a concave function of q. Recall that any concave function is globally upper-bounded by its first-order Taylor expansion at any point. Therefore, with a given local point $q^{(r)}$, we can establish an upper bound of $J_k(q)$ as

$$J_k(q) \le J_k\left(q^{(r)}\right) + J_k'\left(q^{(r)}\right)\left(q - q^{(r)}\right) \triangleq \hat{J}_k(q),$$

where $J_k'\left(q^{(r)}\right)$ is the derivative of $J_k(q)$ at $q^{(r)}$, i.e.,

$$J_k'\left(q^{(r)}\right) = \frac{KH_2}{q^{(r)}KH_2 + H_1\sqrt{d}} + \frac{1}{\ln 2\left[\log_2(1+q^{(r)}) + \frac{1}{d}R_kT_k^\mathrm{comm} + 1\right](1+q^{(r)})}.$$

By replacing $J_k(q)$ in inequality (23) with its upper bound $\hat{J}_k(q)$, and with given local point $q^{(r)}$ at the r^th iteration, the next point at the $(r+1)^\mathrm{th}$ iteration can be obtained by solving the following problem:

$$\text{(P3.1)} \quad q^{(r+1)} = \arg\min_q \tilde{T} \tag{24}$$

$$\text{s.t.} \ \hat{J}_k(q) - \ln(qK) - \ln(\tilde{T}) \le 0, \ \forall k \in [K], \tag{24a}$$

$$q \ge 2. \tag{24b}$$

Since the left side of constraint (24a) is jointly convex with respect to q and $\tilde{T}$, problem P3.1 is convex, and can be solved by standard convex optimization tools such as CVXPY (Diamond and Boyd, 2016). After the iterations converge (e.g., the gap between $\tilde{T}^{(r)}$ and $\tilde{T}^{(r+1)}$ is lower than a given threshold), problem P3 is deemed solved.

4.4 Joint optimization algorithm

Since the unique solution to problem P2 can be obtained by Algorithm 1 and a stationary point of problem P3 can be reached by Algorithm 2, the sub-optimal bandwidth allocation and quantization level can be jointly obtained by alternately solving problems P2 and P3, which is summarized in Algorithm 3. It can be verified that the sub-optimal solution from Algorithm 3 is a stationary point of the original problem P1.

Note that solving problem P1 may lead to a non-integer q, and thus we need a further rounding technique to yield an integer q for practical implementation. One possible rounding technique is discussed as follows: We denote $T(q,\{b_k\})$ as the total training time in problem P1 when q and $\{b_k\}$ are substituted. After finding the optimized quantization level $\hat{q}$ and the optimal bandwidth allocation $\{b_k^*\}$, the final quantization level q^* is obtained as

$$q^* = \arg \min_{q\in\{\lceil\hat{q}\rceil-1,\lceil\hat{q}\rceil\}} T(q,\{b_k^*\}).$$

Algorithm 1 Two-layer bisection search for solving problem P2

```
Input: $B_0$, $\{\theta_k\}$, $\{T_k^{\rm comp}\}$, $T_{\rm d}^+ = \max_k\{T_k^{\rm comp} + R_k(B_0/K)\}$, $T_{\rm d}^- = \max_k\{T_k^{\rm comp}\}$, accuracy threshold $\varepsilon$, and temporary variable $\bar{B}_0 = 0$
while $|B_0 - \bar{B}_0| > \varepsilon$ do
    $T_{\rm d} = \frac{1}{2}(T_{\rm d}^+ + T_{\rm d}^-)$
    for $k = 1 : K$ do
        Solve $R_k(b_k) = \frac{S}{T_{\rm d} - T_k^{\rm comp}}$ by bisection search with respect to $b_k$
    end for
    $\bar{B}_0 = \sum_{k=1}^{K} b_k$
    if $\bar{B}_0 > B_0$ then
        $T_{\rm d}^- = T_{\rm d}$
    else
        $T_{\rm d}^+ = T_{\rm d}$
    end if
end while
return $b_k$
```

Algorithm 2 SCA method for solving problem P3

```
Find a feasible initial quantization level $q^{(0)}$ in problem (22), and set $r = 0$ and threshold $\varepsilon$
repeat
    Set $q^{(r+1)}$ as the solution to problem P3.1
    $r \leftarrow r + 1$
until $|\tilde{T}^{(r)} - \tilde{T}^{(r-1)}| \le \varepsilon$
return $q^{(r)}$
```

Algorithm 3 Joint optimization algorithm for solving problem P1

```
Initialization: quantization level $q^{(0)}$ and bandwidth allocation $\{b_k^{(0)}\}$. Set $r = 0$ and accuracy threshold $\varepsilon$
repeat
    Update the bandwidth allocation $\{b_k^{r+1}\}$ by Algorithm 1
    Update the quantization level $q^{(r+1)}$ and the total training time $\tilde{T}\left(q^{(r+1)}, \{b_k^{(r+1)}\}\right)$ by Algorithm 2
    $r \leftarrow r + 1$
until $\left\| T\left(q^{(r)}, \{b_k^{(r)}\}\right) - T\left(q^{(r-1)}, \{b_k^{(r-1)}\}\right) \right\| \le \varepsilon$
return $\{b_k^* = b_k^{(r)}\}$ and $q^* = \arg\min_{q\in\{\lceil q^{(r)}\rceil-1,\lceil q^{(r)}\rceil\}} T(q,\{b_k^*\})$
```

5 Simulation evaluation

In this section, we provide numerical results of two simulations under different wireless communication scenarios and learning tasks, in which the real system's heterogeneity is captured, to examine our theoretical results. In simulation 1, we consider a learning task with a strongly convex loss function and a training model of small size. In simulation 2, to stretch the theory, we consider a learning task with a non-convex loss function and a training model of large size. Although our analysis is developed based on the assumption of strongly convex loss functions, we show that the proposed algorithm can work well in the case of non-convex loss functions. Both simulations are implemented by PyTorch using Python 3.8 on a Linux server with one NVIDIA® GeForce® RTX 3090 graphics processing unit (GPU) 24 GB and one Intel® Xeon® Gold 5218 CPU.

5.1 Setup

1. FEEL system

We consider a FEEL system with an edge server covering a circular area of $r = 500$ m. Within this area, $K = 6$ edge devices are placed randomly and distributed uniformly over the circular area with the exclusion of a central disk of $r_{\rm d} = 100$ m. The transmit power of each device is 1 dBm. To expose the heterogeneity of the edge devices, the CPU frequency of each device is assumed to be uniformly distributed from 100 MHz to 1 GHz. The number of processing cycles of device k for executing the SGD operation on one batch of samples is $\nu = 10^8$ in simulation 1 and $\nu = 2.5 \times 10^{10}$ in simulation 2.

2. Wireless propagation

The large-scale propagation coefficient in decibels from device k to the edge server is modeled as $[\phi_k]_{\rm dB} = [{\rm PL}_k]_{\rm dB} + [\zeta_k]_{\rm dB}$, where $[{\rm PL}_k]_{\rm dB} = 128.1 + 37.6 \lg {\rm dist}_k$ (${\rm dist}_k$ is the distance in kilometer) is the path loss in decibels, and $[\zeta_k]_{\text{dB}}$ is the shadow fading in decibels (Yang et al., 2021). In this simulation, $[\zeta_k]_{\text{dB}}$ is a Gauss-distributed random variable with mean zero and variance $\sigma_\zeta^2 = 8$ decibels. The noise power spectral density is $N_0 = -174$ dBm/Hz, and the total bandwidth is $B_0 = 10$ KHz (Dhillon et al., 2017). All the simulation parameters are summarized in Table S1 in the supplementary materials.

3. Learning tasks and models

In simulation 1, we consider the ℓ_2 regularized logistic regression task on synthetic data (Wangni et al., 2018). The local loss function in Eq. (1) at device k is given by

$$F_k(\boldsymbol{w}) = \frac{1}{D}\sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}_k} \log_2\left(1+\exp(-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}y_i)\right)+\lambda\|\boldsymbol{w}\|_2^2,$$

where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. The ℓ_2 regularization parameter λ is set to $\lambda = 10^{-6}$. It can be verified that the local loss function $F_k(\boldsymbol{w})$ is smooth and strongly convex. Each data sample $(\boldsymbol{x}_i, y_i)$ is generated in four steps as follows:

(1) Dense data generation: $\bar{x}_{ij} \sim \mathcal{N}(0, 1), \forall\, j \in [d]$;

(2) Magnitude sparsification: $\Theta_j \sim$ Uniform [0, 1], $\Theta_j \leftarrow \Delta_1\Theta_j$ if $\Theta_j \leq \Delta_2, \forall\, j \in [d]$;

(3) Data sparsification: $x_{ij} \leftarrow \bar{x}_{ij}\Theta_j, \forall\, j \in [d]$;

(4) Label generation: $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, $y_i \leftarrow \text{sgn}(\bar{\boldsymbol{x}}_i^{\mathrm{T}}\boldsymbol{w})$.

Note that the parameters Δ_1 and Δ_2 control the sparsity of data points and the gradients (Wangni et al., 2018). From our simulations, the effect of stochastic quantization on SGD convergence depends heavily on the sparsity structure of the gradients. Therefore, we choose this dataset in simulation 1 to better validate our theoretical results. Moreover, to the best of our knowledge, how the sparsity structure of the gradients affects the learning algorithm that uses stochastic quantization as the compression scheme has not been revealed in the literature, and it is an interesting topic but beyond the scope of this work. The parameters Δ_1 and Δ_2 are set to 0.9 and 0.25 in this simulation, respectively. Moreover, the dimension of each data point is set to $d = 1024$. Hence, the model contains 1024 parameters in total. We generate 48 000 data points for training and 12 000 data points for validation.

In simulation 2, we consider the learning task of image classification using the well-known CIFAR-10 dataset, which consists of 50 000 training images and 10 000 validation images in 10 categories of colorful objectives such as airplanes and cars. ResNet-20 (269 722 parameters in total) with batch normalization (The implementation of ResNet-20 follows this GitHub repository https://github.com/hclhkbu/GaussianK-SGD (Shi et al., 2019)) is applied as the classifier model (He et al., 2016).

4. Training and optimization parameters

We consider a decaying learning rate as $\eta_n = \frac{5}{n+10}$ in simulation 1, where n is the number of communication rounds, and the learning rate is set to $\eta_n = \frac{100}{n+1000}$ in simulation 2. To deliver rigorous results, we strictly control all the unrelated variables in both simulations.

5.2 Results of simulation 1

5.2.1 Estimation of data-related parameters

In the optimization in Section 4.3, we need to obtain the values of H_1 and H_2 using the proposed joint data-and-model-driven fitting method in Section 3.3. With the joint data-and-model-driven fitting method, for any given two quantization levels q_1 and q_2, and the threshold of loss optimality gap upper bound ϵ, we can obtain the estimates of H_1 and H_2, which are used in Algorithm 3. Moreover, the optimal loss value can be obtained by the estimate of Z in Eq. (9), i.e., $F(\boldsymbol{w}_*) \approx Z \approx 0.247$. The threshold of the loss optimality gap upper bound is set to $\epsilon = 0.012$. One can choose any combination of q_1 and q_2 to implement the estimation in theory. As a reminder to the readers, however, in our experience, the combination of q_1 and q_2 with a large difference leads to better estimation accuracy. In our results, we choose $(q_1, q_2) = (4, 6)$ in the joint data-and-model-driven fitting method and obtain $H_1 \approx 43.01$ and $H_2 \approx 48.79$. To show the robustness of our estimation method, we plot the fitted loss function and the actual loss values when (q_1, q_2)=(4,6) and (q_1, q_2)=(8,16), as shown in Fig. 3, and we can see that the fitted loss function fits the actual loss well.

5.2.2 Optimization of the quantization level

Fig. 4 plots the total training time vs. quantization level in simulation 1 when the bandwidth allocation is the optimal. We run the same training process at least five times on each quantization level. Fig. 4 shows that there exists an optimal quantization level that minimizes the total training time. Recall the facts from Section 3 that the total training time is given by $T = N_\epsilon T_\mathrm{d}$ and that N_ϵ is a decreasing function of the quantization level q, while T_d is an increasing function of q. In other words, Fig. 4 demonstrates the trade-off between the total number of the communication rounds N_ϵ and per-round latency T_d in the FEEL system. Moreover, it can be observed from the figure that the optimal quantization level obtained in theory from Algorithm 3 in Section 4 matches the results of the simulation, which confirms the validity of our proposed algorithm. From Fig. S1 in the supplementary materials, it can be observed that the training loss of the optimal quantization level under the optimal bandwidth allocation reaches the predefined threshold in a short time.

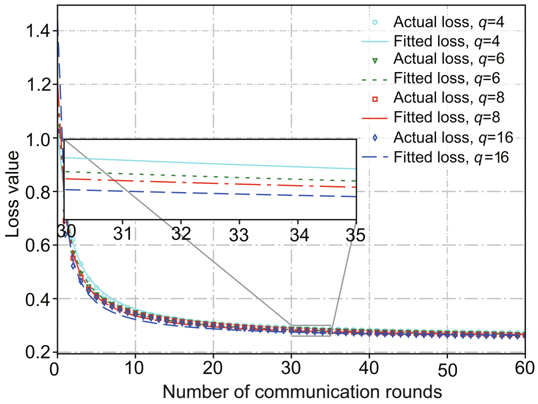

Fig. 3 Robustness of the joint data-and-model-driven fitting method (the fitted loss function and the actual loss vs. the number of communication rounds when the quantization levels are set as q_1 and q_2)

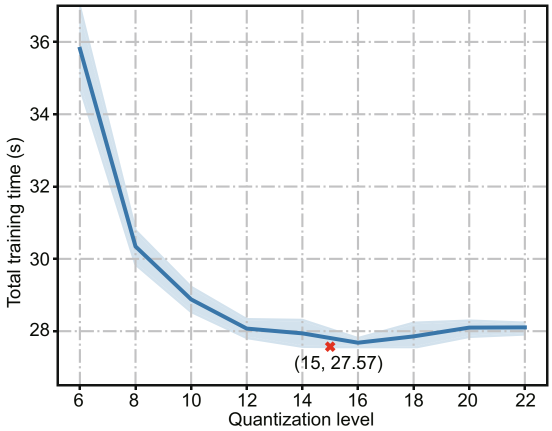

Fig. 4 Total training time vs. the quantization level q in simulation 1 when the bandwidth allocation is optimal (The point with the optimal quantization level and the corresponding training time from Algorithm 3 in theory is annotated by "x")

5.2.3 Optimization of bandwidth

Fig. S1 depicts the comparison between the schemes with the optimal and equal bandwidth allocation in terms of the loss optimality gap and test accuracy. We can observe that the scheme with the optimal bandwidth allocation can reach the predefined threshold and obtain higher test accuracy in a shorter time. This indicates that our bandwidth allocation algorithm is effective and necessary in the FEEL system. To show how the heterogeneous computation power of edge devices affects the communication resource allocation, we present the CPU frequency of each edge device and its corresponding optimal allocated bandwidth in Fig. 5. It can be observed that the edge devices with a lower CPU frequency will be allocated with a larger bandwidth, which in spirit has similarity to the well-known phenomenon of "water-filling" in the problem of power allocation in wireless communication (Gong et al., 2011).

5.3 Results of simulation 2

We conduct simulation 2 to evaluate our method and algorithm on the learning model with a non-convex loss function. In this simulation, we choose $(q_1, q_2) = (15, 20)$ and obtain $H_1 \approx 96.26$ and $H_2 \approx 808.53$ by our joint data-and-model-driven fitting method. The threshold of the upper bound of

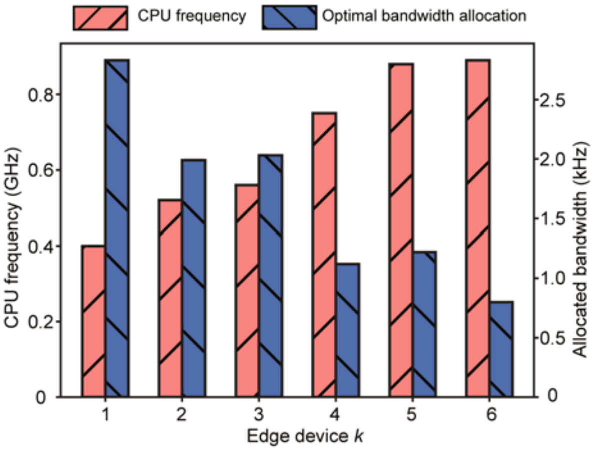

Fig. 5 Optimal bandwidth allocation (bars on the right) and CPU frequency (bars on the left) of each edge device in simulation 1

the loss optimality gap is set to $\epsilon = 0.22$. Fig. 6 shows the total training time vs. the quantization level during the simulation when the bandwidth allocation is optimal. Fig. S2 in the supplementary materials shows the optimality gap and test accuracy vs. training time in simulation 2 with different quantization levels and bandwidth allocations. We can obtain similar observations from Fig. 6 and Fig. S2 to those in simulation 1, which reveals that our method and algorithm work well in the non-convex setting, although they are derived under a strongly convex setting.

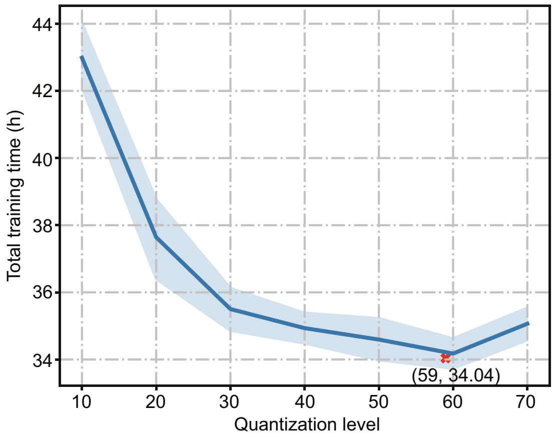

Fig. 6 Total training time vs. quantization level *q* in simulation 2 when the bandwidth allocation is optimal (The point with the optimal quantization level and the corresponding training time from Algorithm 3 in theory is annotated by "x")

6 Conclusions

In this study, we studied the minimization of training time for quantized FEEL with optimized quantization level and bandwidth allocation. On the basis of the convergence analysis of quantized FEEL and our proposed joint data-and-model-driven fitting method, we derived the closed-form expression of the total training time and characterized the trade-off between the number of communication rounds and per-round latency, which is governed by the quantization level. Next, we minimized the total training time by optimizing the quantization level and bandwidth allocation, for which high-quality near-optimal solutions were obtained by alternating optimization. The theoretical results developed can be used to guide system optimization and contribute to the understanding of how a wireless communication system can properly coordinate resources to accomplish learning tasks. This also opens several directions for further research. One future research direction is to implement device sampling in quantized FEEL, in which how the bandwidth is allocated to minimize the training time is completely a different story. Another direction is to consider error compensation in quantized FEEL to mitigate the effects of compression errors.

Contributors

Peixi LIU, Jiamo JIANG, Guangxu ZHU, Wei JIANG, and Wu LUO designed the research. Guangxu ZHU, Wei JIANG, and Wu LUO supervised the research. Peixi LIU and Guangxu ZHU implemented the simulations. Peixi LIU drafted the paper. Jiamo JIANG and Guangxu ZHU helped organize the paper. Lei CHENG, Ying DU, and Zhiqin WANG revised and finalized the paper.

Compliance with ethics guidelines

Peixi LIU, Jiamo JIANG, Guangxu ZHU, Lei CHENG, Wei JIANG, Wu LUO, Ying DU, and Zhiqin WANG declare that they have no conflict of interest.

References

Alistarh D, Grubic D, Li JZ, et al., 2017. QSGD: communication-efficient SGD via gradient quantization and encoding. Proc 31st Int Conf on Neural Information Processing Systems, p.1707-1718.

Amiri MM, Gündüz D, 2020a. Federated learning over wireless fading channels. *IEEE Trans Wirel Commun*, 19(5):3546-3557. https://doi.org/10.1109/TWC.2020.2974748

Amiri MM, Gündüz D, 2020b. Machine learning at the wireless edge: distributed stochastic gradient descent over-the-air. *IEEE Trans Signal Process*, 68:2155-2169. https://doi.org/10.1109/TSP.2020.2981904

Basu D, Data D, Karakus C, et al., 2020. Qsparse-local-SGD: distributed SGD with quantization, sparsification, and local computations. *IEEE J Sel Areas Inform Theory*, 1(1):217-226. https://doi.org/10.1109/JSAIT.2020.2985917

Bernstein J, Wang YX, Azizzadenesheli K, et al., 2018. signSGD: compressed optimisation for non-convex problems. Proc 35th Int Conf on Machine Learning, p.560-569.

Chang WT, Tandon R, 2020. Communication efficient federated learning over multiple access channels. https://arxiv.org/abs/2001.08737

Chen MZ, Poor HV, Saad W, et al., 2021a. Convergence time optimization for federated learning over wireless networks. *IEEE Trans Wirel Commun*, 20(4):2457-2471. https://doi.org/10.1109/TWC.2020.3042530

Chen MZ, Yang ZH, Saad W, et al., 2021b. A joint learning and communications framework for federated learning over wireless networks. *IEEE Trans Wirel Commun*, 20(1):269-283. https://doi.org/10.1109/TWC.2020.3024629

Cover TM, Thomas JA, 2006. Elements of Information Theory (2nd Ed.). John Wiley & Sons, Hoboken, USA.

Dhillon HS, Huang H, Viswanathan H, 2017. Wide-area wireless communication challenges for the Internet of Things. *IEEE Commun Mag*, 55(2):168-174. https://doi.org/10.1109/MCOM.2017.1500269CM

Diamond S, Boyd S, 2016. CVXPY: a python-embedded modeling language for convex optimization. *J Mach Learn Res*, 17(1):2909-2913.

Dinh CT, Tran NH, Nguyen MNH, et al., 2021. Federated learning over wireless networks: convergence analysis and resource allocation. *IEEE/ACM Trans Netw*, 29(1):398-409. https://doi.org/10.1109/TNET.2020.3035770

Gong XW, Vorobyov SA, Tellambura C, 2011. Optimal bandwidth and power allocation for sum ergodic capacity under fading channels in cognitive radio networks. *IEEE Trans Signal Process*, 59(4):1814-1826. https://doi.org/10.1109/TSP.2010.2101069

Gradshteyn IS, Ryzhik IM, 2014. Table of Integrals, Series, and Products. Academic Press, Cambridge, USA.

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/CVPR.2016.90

Jin R, He X, Dai H, 2020. On the design of communication efficient federated learning over wireless networks. https://arxiv.org/abs/2004.07351v1

Kairouz P, McMahan HB, Avent B, et al., 2019. Advances and open problems in federated learning. *Found Trends® Mach Learn*, 14(1-2):1-210. https://doi.org/10.1561/2200000083

Letaief KB, Chen W, Shi YM, et al., 2019. The roadmap to 6G: AI empowered wireless networks. *IEEE Commun Mag*, 57(8):84-90. https://doi.org/10.1109/MCOM.2019.1900271

Li X, Huang KX, Yang WH, et al., 2020. On the convergence of FedAvg on non-IID data. Proc 8th Int Conf on Learning Representations, p.1-26.

Liu DZ, Simeone O, 2021. Privacy for free: wireless federated learning via uncoded transmission with adaptive power control. *IEEE J Sel Areas Commun*, 39(1):170-185. https://doi.org/10.1109/JSAC.2020.3036948

Luo B, Li X, Wang SQ, et al., 2021. Cost-effective federated learning design. IEEE Conf on Computer Communications, p.1-10. https://doi.org/10.1109/INFOCOM42981.2021.9488679

Nguyen VD, Sharma SK, Vu TX, et al., 2021. Efficient federated learning algorithm for resource allocation in wireless IoT networks. *IEEE Int Things J*, 8(5):3394-3409. https://doi.org/10.1109/JIOT.2020.3022534

Nori MK, Yun S, Kim IM, 2021. Fast federated learning by balancing communication trade-offs. *IEEE Trans Commun*, 69(8):5168-5182. https://doi.org/10.1109/TCOMM.2021.3083316

Park J, Samarakoon S, Bennis M, et al., 2019. Wireless network intelligence at the edge. *Proc IEEE*, 107(11):2204-2239. https://doi.org/10.1109/JPROC.2019.2941458

Park J, Samarakoon S, Elgabli A, et al., 2021. Communication-efficient and distributed learning over wireless networks: principles and applications. *Proc IEEE*, 109(5):796-819. https://doi.org/10.1109/JPROC.2021.3055679

Razaviyayn M, 2014. Successive Convex Approximation: Analysis and Applications. PhD Thesis, University of Minnesota, Minnesota, USA.

Reisizadeh A, Mokhtari A, Hassani H, et al., 2020. FedPAQ: a communication-efficient federated learning method with periodic averaging and quantization. Proc 23rd Int Conf on Artificial Intelligence Statistics, p.2021-2031.

Ren JK, He YH, Wen DZ, et al., 2020. Scheduling for cellular federated edge learning with importance and channel awareness. *IEEE Trans Wirel Commun*, 19(11):7690-7703. https://doi.org/10.1109/TWC.2020.3015671

Salehi M, Hossain E, 2021. Federated learning in unreliable and resource-constrained cellular wireless networks. *IEEE Trans Commun*, 69(8):5136-5151. https://doi.org/10.1109/TCOMM.2021.3081746

Shi SH, Chu XW, Cheung KC, et al., 2019. Understanding top-k sparsification in distributed deep learning. https://arxiv.org/abs/1911.08772v1

Shlezinger N, Chen MZ, Eldar YC, et al., 2021. UVeQFed: universal vector quantization for federated learning. *IEEE Trans Signal Process*, 69:500-514. https://doi.org/10.1109/TSP.2020.3046971

Stich SU, Cordonnier JB, Jaggi M, 2018. Sparsified SGD with memory. Proc 32nd Int Conf on Neural Information Processing Systems, p.4452-4463.

Tse D, Viswanath P, 2005. Fundamentals of Wireless Communication. Cambridge University Press, New York, USA. https://doi.org/10.1017/CBO9780511807213

Wan S, Lu JX, Fan PY, et al., 2021. Convergence analysis and system design for federated learning over wireless networks. *IEEE J Sel Areas Commun*, 39(12):3622-3639. https://doi.org/10.1109/JSAC.2021.3118351

Wang SQ, Tuor T, Salonidis T, et al., 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE J Sel Areas Commun*, 37(6):1205-1221. https://doi.org/10.1109/JSAC.2019.2904348

Wang YM, Xu YQ, Shi QJ, et al., 2022. Quantized federated learning under transmission delay and outage constraints. *IEEE J Sel Areas Commun*, 40(1):323-341. https://doi.org/10.1109/JSAC.2021.3126081

Wangni JQ, Wang JL, Liu J, et al., 2018. Gradient sparsification for communication-efficient distributed optimization. https://arxiv.org/abs/1710.09854v1

Yang ZH, Chen MZ, Saad W, et al., 2021. Energy efficient federated learning over wireless communication networks. *IEEE Trans Wirel Commun*, 20(3):1935-1949. https://doi.org/10.1109/TWC.2020.3037554

Zhu GX, Wang Y, Huang KB, 2020a. Broadband analog aggregation for low-latency federated edge learning. *IEEE Trans Wirel Commun*, 19(1):491-506. https://doi.org/10.1109/TWC.2019.2946245

Zhu GX, Liu DZ, Du YQ, et al., 2020b. Toward an intelligent edge: wireless communication meets machine learning.

IEEE Commun Mag, 58(1):19-25. https://doi.org/10.1109/MCOM.001.1900103

Zhu GX, Du YQ, Gündüz D, et al., 2021. One-bit over-the-air aggregation for communication-efficient federated edge learning: design and convergence analysis. *IEEE Trans Wirel Commun*, 20(3):2120-2135. https://doi.org/10.1109/TWC.2020.3039309

List of supplementary materials

Proof S1 Proof of Theorem 1
Proof S2 Proof of Lemma 1
Table S1 Simulation parameters
Fig. S1 Optimality gap and test accuracy in simulation 1
Fig. S2 Optimality gap and test accuracy in simulation 2