On the Tradeoff between Energy, Precision, and Accuracy in Federated Quantized Neural Networks

Minsu Kim[1], Walid Saad[1], Mohammad Mozaffari[2], and Merouane Debbah[3,4]
[1] Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.
[2] Ericsson Research, Santa Clara, CA, USA.
[3] Technology Innovation Institute, Abu Dhabi, United Arab Emirates.
4 Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates.
Emails: {msukim, walids}@vt.edu, mohammad.mozaffari@ericsson.com, merouane.debbah@tii.ae.

***Abstract*—Deploying federated learning (FL) over wireless networks with resource-constrained devices requires balancing between accuracy, energy efficiency, and precision. Prior art on FL often requires devices to train deep neural networks (DNNs) using a 32-bit precision level for data representation to improve accuracy. However, such algorithms are impractical for resource-constrained devices since DNNs could require execution of millions of operations. Thus, training DNNs with a high precision level incurs a high energy cost for FL. In this paper, a quantized FL framework, that represents data with a finite level of precision in both local training and uplink transmission, is proposed. Here, the finite level of precision is captured through the use of quantized neural networks (QNNs) that quantize weights and activations in fixed-precision format. In the considered FL model, each device trains its QNN and transmits a quantized training result to the base station. Energy models for the local training and the transmission with the quantization are rigorously derived. An energy minimization problem is formulated with respect to the level of precision while ensuring convergence. To solve the problem, we first analytically derive the FL convergence rate and use a line search method. Simulation results show that our FL framework can reduce energy consumption by up to 53% compared to a standard FL model. The results also shed light on the tradeoff between precision, energy, and accuracy in FL over wireless networks.**

I. Introduction

The emergence of federated learning (FL) ushered in a new era of distributed inference that can alleviate data privacy concerns [1]. In FL, massively distributed mobile devices and a central server (e.g., a base station (BS)) collaboratively train a shared model without requiring devices to share raw data. Many FL algorithms employ complex deep neural networks (DNNs) to achieve a high accuracy by allocating many bits for the precision level in data representation [2]. DNN structures, such as convolutional neural networks (CNNs), can have tens of millions of parameters and billions of multiply-accumulate (MAC) operations [3]. In practice, the energy consumed for computation and memory access is proportional to the level of precision [4]. Hence, computationally intensive neural networks with a conventional 32 bits full precision level may not be suitable for deployment on energy-constrained mobile and Internet of Things (IoT) devices. In addition, a DNN may increase the energy consumption of transmitting a training result due to the large model size. To design an energy-efficient FL scheme, one could reduce the level of precision to decrease the energy consumption for the computation and transmission. However, the reduced precision level could introduce quantization error that degrades the accuracy and the convergence rate of FL. Therefore, deploying real-world FL frameworks over wireless systems requires one to *balance precision, accuracy, and energy efficiency* – a major challenge facing future distributed learning frameworks.

Remarkably, despite the surge in research on the use of FL, only a handful of works in [5]–[10] have studied the energy efficiency of FL from a system-level perspective. A novel analytical framework that derived energy efficiency of FL algorithms in terms of the carbon footprint was proposed in [5]. Meanwhile, in [6], the authors formulated an energy minimization problem under heterogeneous power constraints of mobile devices. The work in [7] investigated a resource allocation problem to minimize the total energy consumption considering the convergence rate. In [8], the energy consumption of FL was minimized by controlling workloads of each device, which has heterogeneous computing resources. The work in [9] proposed a quantization scheme for both uplink and downlink transmission in FL and analyzed the impact of the quantization on the convergence rate. The authors in [10] considered a novel FL setting, in which each device trains a binary neural network so as to improve the energy efficiency of transmission by uploading the binary parameters to the server.

However, the works in [5]–[9] did not consider the energy efficiency of their DNN structure during training. Since devices have limited computing and memory resources, deploying an energy-efficient DNN will be a more appropriate way to reduce the energy consumption of FL. Although the work in [11] considered binarized neural networks during training, this work did not optimize the quantization levels of the neural network to balance the tradeoff between precision and energy. To the best of our knowledge, there is no work that jointly considers the tradeoff between precision, energy, and accuracy.

The main contribution of this paper is a novel energy-efficient quantized FL framework that can represent data with a finite level of precision in both local training and uplink transmission. In our FL model, each device trains a quantized neural network (QNN), whose weights and activations are quantized with a finite level of precision, so as to decrease en-

This work was supported by the U.S. National Science Foundation under Grants CNS-1814477 and CNS-2114267.

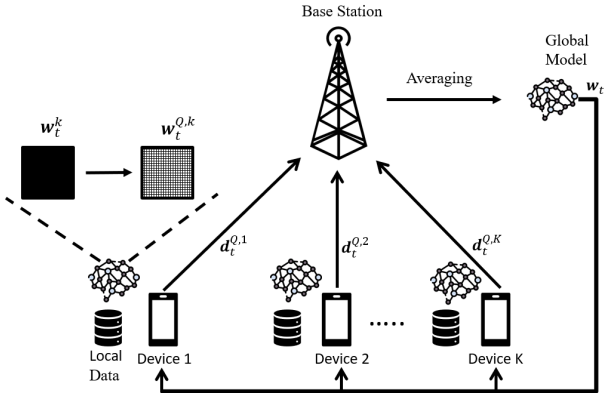

Fig. 1: An illustration of the quantized FL model over wireless network.

ergy consumption for computation and memory access. After training, each device quantizes the result with the same level of precision used in the local training and transmits it to the BS. The BS aggregates the received information to generate a new global model and broadcasts it back to the devices. To quantify the energy consumption, we propose rigorous energy model for the local training based on the physical structure of a processing chip. We also derive the energy model for the uplink transmission considering the quantization. To achieve a high accuracy, FL requires a high level of precision at the cost of increased total energy consumption. Meanwhile, although, a low level of precision can decrease the energy consumption per iteration, it will decrease the convergence rate to achieve a target accuracy. Thus, there is a need for a new approach to analyze and optimize the tradeoff between precision, energy, and accuracy. To this end, we formulate an optimization problem by controlling the level of precision to minimize the total energy consumption while ensuring convergence with a target accuracy. To solve the problem, we first analytically derive the convergence rate of our FL framework and use a line search method to numerically find the local optimal solution. Simulation results show that our FL model can reduce the energy consumption up to 53% compared to a standard FL model, which uses 32-bit full-precision for data representation. The results also shed light on the tradeoff between precision, energy efficiency, and accuracy in FL over wireless networks.

The rest of this paper is organized as follows. Section II presents the system model. In Section III, we describe the studied problem. Section IV provides simulation results. Finally, conclusions are drawn in Section V.

II. System Model

We consider an FL system, in which N devices (e.g. edge or mobile devices) are connected to one BS. As shown in Fig. 1, the BS and devices collaboratively perform an FL algorithm for executing a certain data analysis task. Each device k has local dataset $\mathcal{D}_k = \{\boldsymbol{x}_{kl}, y_{kl}\}$, where $l = 1, \ldots, D_k$. In particular, $\{\boldsymbol{x}_{kl}, y_{kl}\}$ is an input-output pair for image classification, where $\boldsymbol{x}_{kl}$ is an input vector and y_{kl} is the corresponding output. We define a loss function $f(\boldsymbol{w}, \boldsymbol{x}_{kl}, y_{kl})$ to quantify the performance of a machine learning (ML) model with parameters $\boldsymbol{w} \in \mathbb{R}^d$ over $\{\boldsymbol{x}_{kl}, y_{kl}\}$. Since device k has D_k data samples, its local loss function is given by

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

We define the global loss function over N devices as follows:

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

where $D = \sum_{k=1}^{N} D_k$ is the total size of the entire dataset. The FL process aims to find the optimal model parameters $\boldsymbol{w}$ that can minimize the global loss function as follows

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}). \tag{3}$$

Solving problem (3) typically requires an iterative process between the BS and devices. However, in practical systems, such as an IoT, the devices are energy-constrained. They are unable to run a power consuming FL process. Hence, we propose to manage the level of precision of our FL to reduce the energy consumption for computation, memory access, and transmission. As such, we adopt a QNN structure whose weights and activations are quantized in fixed-point format rather than conventional 32-bit floating-point format [11].

A. Quantized Neural Networks

In our model, each device trains a QNN of identical structure using n bits of precision for quantization. We can express data more precisely if we increase n at the cost of more energy usage. We can represent any given number in fixed-point format such as $[\Omega.\omega]$, where Ω is the integer part and ω is the fractional part of the given number [12]. Here, we use one bit to represent the integer part and $(n-1)$ bits for the fractional part. Then, the smallest positive number we can present would be $\kappa = 2^{-n+1}$, and the possible range of numbers with n bits will be $[-1, 1 - 2^{-n+1}]$. Note that a QNN restricts the value of weights to [-1, 1]. We consider a stochastic quantization scheme [12], where any given $w \in \boldsymbol{w}$ is quantized as follows:

$$Q(w) = \begin{cases} \lfloor w \rfloor, & \text{with probability } \frac{\lfloor x \rfloor + \kappa - w}{\kappa}, \\ \lfloor w \rfloor + \kappa, & \text{with probability } \frac{w - \lfloor w \rfloor}{\kappa}, \end{cases} \tag{4}$$

where $\lfloor w \rfloor$ is the largest integer multiple of κ less than or equal to w.

We denote the quantized weights of layer l as $\boldsymbol{w}_{(l)}^{Q,k} = Q(\boldsymbol{w}_{(l)}^{k})$ for device k. Then, the outputs of layer l will be:

$$o_{(l)} = g_{(l)}(\boldsymbol{w}_{(l)}^{Q,k}, o_{(l-1)}^{Q}), \tag{5}$$

where $g(\cdot)$ is the operation of layer l on the input, such as activation and batch normalization, and $o_{(l-1)}^{Q}$ is the quantized output from the previous layer $l-1$. Note that the output $o_{(l)}$ will be quantized and fed into the next layer as an input. For training, we use stochastic gradient descent (SGD) algorithm as follows

$$\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \eta \nabla F_k(\boldsymbol{w}^{Q,k}, \xi^k), \tag{6}$$

Algorithm 1: Quantized FL Algorithm

Input: K, I, initial model $\boldsymbol{w}_0$, $t=0$, target accuracy ϵ

repeat
- The BS randomly select a subset of devices $\mathcal{N}_t$ and broadcasts the $\boldsymbol{w}_t$ to the selected devices;
- Each device $k \in \mathcal{N}_t$ trains $\boldsymbol{w}_t^k$ by running I steps of SGD as (6);
- Each device $k \in \mathcal{N}_t$ tansmits $\boldsymbol{d}_{t+1}^{Q,k}$ to the BS;
- The BS generates a new global model $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{K}\sum_{k\in\mathcal{N}_t} \boldsymbol{d}_{t+1}^{Q,k}$;
- $t \leftarrow t+1$;

until *target accuracy* ϵ *is satisfied*;

where η is the learning rate and ξ is a mini-batch for the current update. Then, we restrict the values of $\boldsymbol{w}^k$ to $[-1,1]$ as $\boldsymbol{w}^k \leftarrow \text{clip}(\boldsymbol{w}^k,-1,1)$, where $\text{clip}(\cdot,-1,1)$ projects each input to 1 (-1) for any input larger (smaller) than 1 (-1), or returns the same value as the input. Otherwise, $\boldsymbol{w}^k$ can become very large without a meaningful impact on quantization [11]. After each training, $\boldsymbol{w}^k$ are quantized as $\boldsymbol{w}^{Q,k}$.

B. FL model

For learning, without loss of generality, we adopt FedAvg [2] to solve problem (3). At each global iteration t, the BS randomly selects a set of devices $\mathcal{N}_t$ with $|\mathcal{N}_t| = K$ and broadcasts the current global model $\boldsymbol{w}_t$ to the scheduled devices. Each device in $\mathcal{N}_t$ trains its local model based on the received global model by running I steps of SGD on its local loss function as below

$$\boldsymbol{w}_{t,\tau}^k = \boldsymbol{w}_{t,\tau-1}^k - \eta_t \nabla F_k(\boldsymbol{w}_{t,\tau-1}^{Q,k}, \xi_\tau^k), \forall \tau = 1,\ldots,I, \quad (7)$$

where η_t is the learning rate at global iteration t. Note that unscheduled devices do not perform local training. Then, K devices calculates the model update $\boldsymbol{d}_{t+1}^k = \boldsymbol{w}_{t+1}^k - \boldsymbol{w}_t^k$, where $\boldsymbol{w}_{t+1}^k = \boldsymbol{w}_{t,I}^k$ and $\boldsymbol{w}_t^k = \boldsymbol{w}_{t,0}^k$ [9]. Typically, $\boldsymbol{d}_{t+1}^k$ has a millions of elements. It is not practical to send $\boldsymbol{d}_{t+1}^k$ with full precision for energy-constrained devices. Hence, we apply the same quantization scheme used in QNNs to $\boldsymbol{d}_{t+1}^k$ and denote its quantization result as $\boldsymbol{d}_{t+1}^{Q,k}$. Then, K devices transmit their model update to the BS. The received model updates are averaged by the BS, and the next global model will be generated as below

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{K}\sum_{k\in\mathcal{N}_t} \boldsymbol{d}_{t+1}^{Q,k}. \quad (8)$$

The FL system repeats this process until the global loss function converges to a target accuracy constraint ϵ. We summarize the aforementioned algorithm in Algorithm 1.

Next, we propose the energy model for the computation and the transmission for our FL system.

C. Computing and Transmission model

1) Computing model: We consider a typical two dimensional processing chip for CNNs as shown in Fig. 2 [4]. This chip has a parallel neuron array, p MAC units, and two levels of memory: a main and a local buffer. A main buffer stores the current layers' weights and activations, while a local buffer caches currently used weights and activations. From

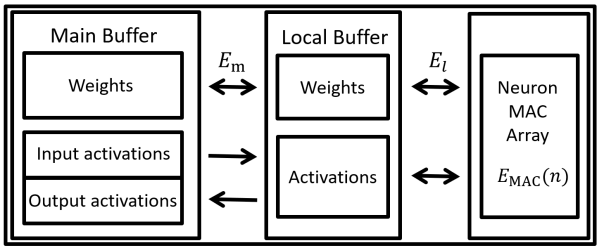

Fig. 2: An illustration of the two dimensional processing chip.

[13], we use the energy model of a MAC operation for n levels of precision $E_{\text{MAC}}(n) = A\,(n/n_{\max})^\alpha$, where $A>0$, $1<\alpha<2$, and $n_{\max}$ is the maximum precision level. Here, a MAC operation includes the operation of a layer such as output calculation, batch normalization, activation, and weight update. Then, the energy consumption for accessing a local buffer E_{l} can be modeled as $E_{\text{MAC}}(n)$, and the energy for accessing a main buffer is $E_{\text{m}} = 2E_{\text{MAC}}(n)$ [4] .

The energy consumption of device k for one iteration of local training is given by $E^{C,k}(n)$ when n bits are used for the precision level in the quantization. Then, $E^{C,k}(n)$ is the sum of the computing energy $E_{\text{C}}(n)$, the access energy for fetching weights from the buffers $E_{\text{W}}(n)$, and the access energy for fetching activations from the buffers $E_{\text{A}}(n)$, as follows [13]:

$$\begin{aligned}
&E^{C,k}(n) = E_{\text{C}}(n) + E_{\text{W}}(n) + E_{\text{A}}(n),\\
&E_{\text{C}}(n) = E_{\text{MAC}}(n)N_c + 4O_s\ E_{\text{MAC}}(n_{\max}),\\
&E_{\text{W}}(n) = E_{\text{m}}N_s + E_{\text{l}}N_c\sqrt{n/pn_{\max}},\\
&E_{\text{A}}(n) = 2E_{\text{m}}O_s + E_{\text{l}}N_c\sqrt{n/pn_{\max}},
\end{aligned} \quad (9)$$

where N_c is the number of MAC operations, N_s is the number of weights, and O_s is the number of intermediate outputs throughout the network. For E_{C}, in a QNN, batch normalization, activation function, gradient calculation, and weight update are done in full-precision $n_{\max}$ to each output O_s [11]. Once we fetch weights from a main to a local buffer, they can be reused in the local buffer afterward as shown in $E_{\text{W}}(n)$. In Fig. 2, a MAC unit fetches weights from a local buffer for computation. Since we are using a two dimensional MAC array of p MAC units, they can share fetched weights with the same row and column, which has $\sqrt{p}$ MAC units respectively. In addition, a MAC unit can fetch more weights due to the quantization with n bits compared with when weights are represented in $n_{\max}$ bits. Thus, we can reduce the access to a local buffer by the amount of $\sqrt{n/pn_{\max}}$. A similar process applies to E_{A} since activations are fetched (stored) from (to) the main buffer.

2) Transmission Model: We use orthogonal frequency domain multiple access (OFDMA) to transmit a model update to the BS. The achievable rate of device k is given by

$$r_k = B\log_2\left(1+\frac{Ph_k}{N_0 B}\right), \quad (10)$$

where B is the allocated bandwidth, h_k is the channel gain between device k and the BS, P is the transmit power of each device, and N_0 is the power spectral density of white noise.

After local training, device k will transmit $\boldsymbol{d}_t^{Q,k}$ to the BS at given global iteration t. Then, the transmission time T_k for uploading $\boldsymbol{d}_t^{Q,k}$ is given by

$$T_k(n) = \frac{||\boldsymbol{d}_t^{Q,k}||}{r_k} = \frac{||\boldsymbol{d}_t^{k}||n}{r_k n_{\max}}. \tag{11}$$

Note that $\boldsymbol{d}_t^{Q,k}$ is quantized with n bits of precision while $\boldsymbol{d}_t^{k}$ is represented with $n_{\max}$ bits. Then, the energy consumption for the uplink transmission is given by

$$E^{UL,k}(n) = T_k(n) \times P = \frac{P||\boldsymbol{d}_t^{k}||n}{B\log_2\left(1+\frac{Ph_k}{N_0 B}\right) n_{\max}}. \tag{12}$$

In the following section, we formulate an energy minimizing problem based on the derived energy models.

III. Proposed Approach for Energy-Efficient Federated QNN

We formulate an energy minimization problem while ensuring convergence under a target accuracy. A tradeoff exists between the energy consumption and the convergence rate with respect to n. Hence, finding the optimal n is important to balance the tradeoff and achieve the target accuracy. We propose a numerical method to solve this problem.

We aim to minimize the expected total energy consumption until convergence under the target accuracy as follows:

$$\min_{n} \quad \mathbb{E}\left[\sum_{t=1}^{T}\sum_{k\in\mathcal{N}_t} E^{UL,k}(n) + IE^{C,k}(n)\right] \tag{13a}$$

$$\text{s.t.} \quad n \in [1,\dots,n_{\max}], \tag{13b}$$

$$\mathbb{E}[F(\boldsymbol{w}_T)] - F(\boldsymbol{w}^*) \le \epsilon, \tag{13c}$$

where I is the number of local iterations, $\mathbb{E}[F(\boldsymbol{w}_T)]$ is the expectation of global loss function after T global iteration, $F(\boldsymbol{w}^*)$ is the minimum value of F, and ϵ is the target accuracy.

Since K devices are randomly selected at each global iteration, we can derive the expectation of the objective function of (13a) as follows

$$\begin{aligned} f_E(n) &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{k\in\mathcal{N}_t} E^{UL,k}(n) + IE^{C,k}(n)\right] \\ &= \frac{KT}{N}\sum_{k=1}^{N}\left\{E^{UL,k}(n) + IE^{C,k}(n)\right\}. \end{aligned} \tag{14}$$

To represent T with respect to ϵ, we assume that the loss function is L-smooth, μ-strongly convex and that the variance and the squared norm of the stochastic gradient are bounded by σ_k^2 and G for device k, $\forall k \in N$, respectively. Before we present the expression of T, in the following lemma, we will first analyze the quantization error of the stochastic quantization in Sec. II.

Lemma 1. *For the stochastic quantization $Q(\cdot)$, a scalar value w, and a vector $\boldsymbol{w} \in \mathbb{R}^d$, we have*

$$\mathbb{E}[Q(w)] = w, \quad \mathbb{E}[(Q(w)-w)^2] \le \frac{1}{2^{2n}}, \tag{15}$$

$$\mathbb{E}[Q(\boldsymbol{w})] = \boldsymbol{w}, \quad \mathbb{E}[||Q(\boldsymbol{w})-\boldsymbol{w}||^2] \le \frac{d}{2^{2n}}. \tag{16}$$

Proof. We first derive $\mathbb{E}[Q(w)]$ as

$$\mathbb{E}[Q(w)] = \lfloor w \rfloor \frac{\lfloor w \rfloor + \kappa - w}{\kappa} + (\lfloor w \rfloor + \kappa)\frac{w - \lfloor w \rfloor}{\kappa} = w. \tag{17}$$

Similarly, $\mathbb{E}[(Q(w)-w)^2]$ can be obtained as

$$\begin{aligned} \mathbb{E}[(Q(w)-w)^2] &= (\lfloor w \rfloor - w)^2 \frac{\lfloor w \rfloor + \kappa - w}{\kappa} + (\lfloor w \rfloor + \kappa - w)^2 \frac{w - \lfloor w \rfloor}{\kappa} \\ &= (w - \lfloor w \rfloor)(\lfloor w \rfloor + \kappa - w) \\ &\le \frac{\kappa^2}{4} = \frac{1}{2^{2n}}, \end{aligned} \tag{18}$$

where (18) follows from the arithmetic mean and geometric mean inequality. Since expectation is a linear operator, we have $\mathbb{E}[Q(\boldsymbol{w})] = \boldsymbol{w}$ from (17). From the definition of the square norm, $\mathbb{E}[||Q(\boldsymbol{w}) - \boldsymbol{w}||^2]$ can obtained as

$$\mathbb{E}[||Q(\boldsymbol{w}) - \boldsymbol{w}||^2] = \sum_{j=1}^{d} \mathbb{E}[(Q(w_j) - w_j)^2] \le \frac{d}{2^{2n}}. \tag{19}$$

□

From Lemma 1, we can see that our quantization scheme is unbiased as its expectation is zero. However, the quantization error can still increase for a large model. We next leverage the results of Lemma 1, [9], and [14] so as to derive T with respect to ϵ in the following proposition.

Proposition 1. *For learning rate $\eta_t = \frac{\beta}{t+\gamma}, L < \frac{2\eta_t}{2\eta_t^2+1}, \beta > \frac{1}{\mu}$, and $\gamma > 0$, we have*

$$\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \le \frac{L}{2}\frac{v}{T+\gamma}, \tag{20}$$

where v is

$$v = \sum_{k=1}^{N}\frac{\sigma_k^2}{N^2} + \frac{d}{2^{2n}}\left(1 + \frac{2IG^2}{K}\right) + 4(I-1)^2G^2 + \frac{4(N-K)}{K(N-1)}I^2G^2. \tag{21}$$

Proof. The complete proof is omitted due to space limitations. Essentially, proposition 1 can be proven by using Lemma 1, the convergence result with a quantized model update [9], and replacing the SGD weight update in [14] with (7). □

From Proposition 1, We let (20) be upper bounded by ϵ in (13c) as follows

$$\mathbb{E}[F(\boldsymbol{w}_T) - F(\boldsymbol{w}^*)] \le \frac{L}{2}\frac{v}{T+\gamma} \le \epsilon. \tag{22}$$

We then take equality in (22) to obtain $T = Lv/(2\epsilon) - \gamma$ and approximate the problem as

$$\min_{n} \quad \frac{K}{N}\left(\frac{Lv}{2\epsilon} - \gamma\right)\sum_{k=1}^{N}\left\{E^{UL,k}(n) + IE^{C,k}(n)\right\} = f_E(n) \tag{23a}$$

$$\text{s.t.} \quad n \in [1,\dots,n_{\max}]. \tag{23b}$$

Note that any optimal solution n^* from problem (23a) can satisfy problem (13a) [15]. For any feasible T from (22), we can always choose $T_0 > T$ such that T_0 satisfies (13c).

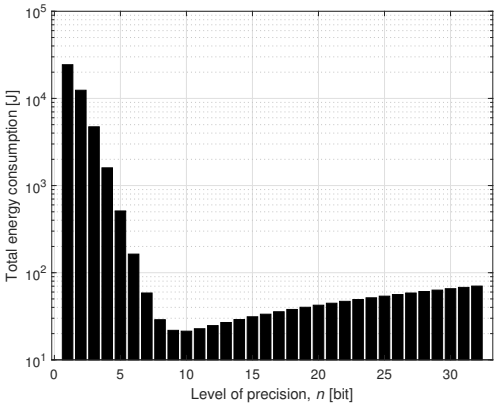

Fig. 3: Total energy consumption for varying level of precision.

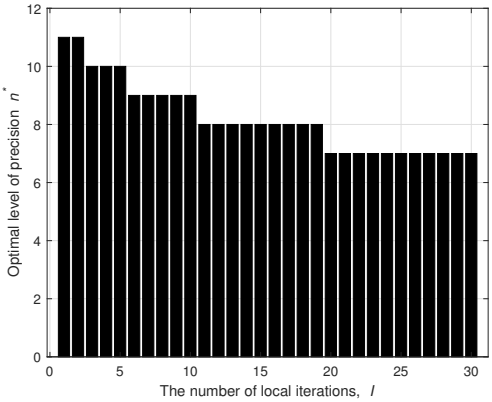

Fig. 4: Optimal level of precision for varying the number of local iterations.

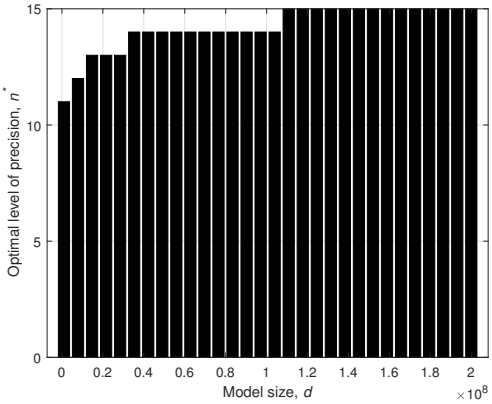

Fig. 5: Optimal level of precision for varying model size.

Now, we relax n as a continuous variable, which will be rounded back to an integer value. From (9), (12), and (21), we can observe that, as n increases, $E^{UL,k}(n)$ and $E^{C,k}(n)$ becomes larger while T decreases. Hence, we can know that a local optimal n^* may exist for minimizing $f_E(n)$. Since $f_E(n)$ is differentiable with respect to n in the given range, we can find n^* by solving $\partial f_E(n)/\partial n = 0$ from Fermat's Theorem [16]. Although it is difficult to derive n^* analytically, we can obtain it numerically using a line search method. Hence, we can find a local optimal solution, which minimizes the total energy consumption under the given target accuracy.

IV. Simulation Results

For our simulations, we uniformly deploy $N = 50$ devices over a square area of size 100 m $\times$ 100 m serviced by one BS at the center, and we assume a Rayleigh fading channel with a path loss exponent of 2. Unless stated otherwise, we use $P = 100$ mW, $B = 10$ MHz, $N_0 = -100$ dBm, $K = 30$, $I = 5$, $n_{\max} = 32$ bits, $\epsilon = 0.01$, $\beta = 5$, $L = 1$, $\mu = 1$, $\gamma = 1$, $\sigma_k = 1$, and $G = 0.02$, $\forall k = 1, \ldots, N$ [17]. For the computing model, we set $A = 3.7$ pJ and $\alpha = 1.25$ as done in [13], and we assume that each device has the same architecture of the processing chip. All statistical results are averaged over 10000 independent runs

Figure 3 shows the total energy consumption of the FL system until convergence for varying levels of precision n. In Fig. 3, we assume a QNN structure with two convolutional layers: 32 kernels of size 3×3 with one padding and three of strides and 32 kernels of size 3×3 with one padding and two of strides, each followed by 2×2 max pooling. Then, we have one dense layer of 220 neurons and one fully-connected layer. In this setting, we have $N_c = 20.64 \times 10^6, N_s = 0.18 \times 10^6$, and $O_s = 1354$. From this figure, we can see that the total energy consumption decreases and then increases with n. This is because when n is small, quantization error becomes large as shown in Lemma 1, which slows down the convergence rate in (20). However, as n increases, the energy consumption for the local training and transmission also increases. Hence, a very small or very high n may induce undesired large quantization error or unnecessary energy consumption due to a high level of precision. From this figure, we can see that $n = 10$ can be optimal for minimizing energy consumption for our system.

Figure 4 shows the optimal level of precision n^* when varying the number of local iterations I. We use the same CNN architecture in Fig. 3. We can observe that n^* increases with I. This is because, as I increases, the local models converge to the local optimal faster as SGD averages out the effect of quantization error [11]. Hence, a lower n can be chosen by leveraging the increased I to minimize the total energy consumption. We can observe that only $n = 7$ is required at $I = 20$ while we need $n = 10$ at $I = 3$.

Figure 5 presents the optimal level of precision n^* for varying model size d. Note that d equals to the number of model parameters N_s. To scale the number of MAC operations for increasing d accordingly, we set $N_c = 0.5 \times 10^3 N_s$. From Fig. 5, we can see that n^* increases with d. From Lemma 1, the quantization error accumulates as d increases. This directly affects the convergence rate in (20) resulting in both increased global iterations and the total energy consumption. Therefore, to mitigate the increasing quantization error from increasing d, a larger level of precision may be chosen.

Figure 6 shows the total energy consumption and n^* when varying a target accuracy ϵ. For these results, we use the same CNN architecture as Fig. 3. We can see that a higher accuracy level requires larger total energy consumption and

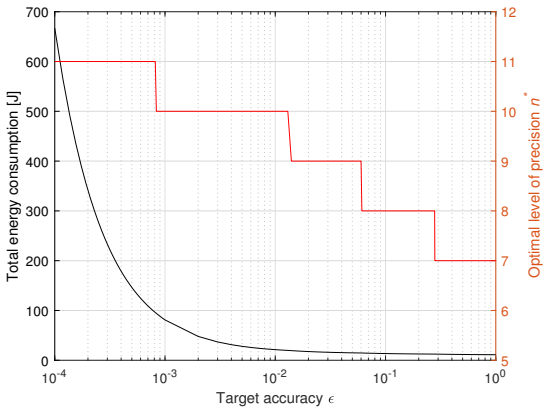

Fig. 6: Total energy consumption and optimal level of precision for varying target accuracy.

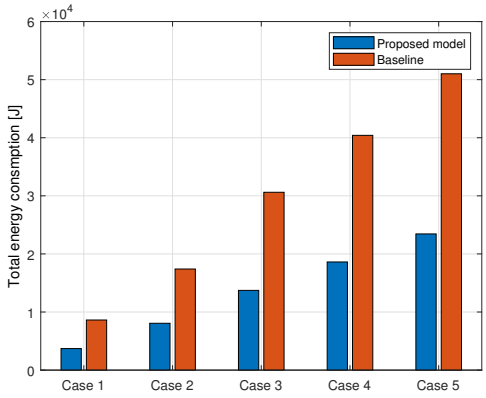

Fig. 7: Total energy consumption for various CNN models.

more bits for data representation to mitigate the quantization error. In addition, the FL system needs a more number of global iterations to achieve ϵ from (22). As ϵ becomes looser, a lower n can be chosen. From Fig. 6, we can see that an additional 127 J energy is required to increase ϵ from 0.01 to 0.001 while one additional bit of precision is needed.

Figure 7 compares the total energy consumption until convergence for varying the size of CNN models with the baseline that uses standard 32 bits for data representation. Case 1 is assumed to be CNN of two layers and have $d = 25.6 \times 10^6$ and $N_c = 1.8 \times 10^9$. Case 2 and 3 are assumed to be CNN of five layers. They have 61.6×10^6 and 83.5×10^6 number of parameters and 0.35×10^9 and 83.5×10^6 number of MAC operations, respectively. Case 4 is 7 layers of CNN with $d = 115.1 \times 10^6$ and $N_c = 10.4 \times 10^9$. Lastly, we assume Case 5 is 9 layers of CNN with $d = 138.6 \times 10^6$ and 15.5×10^9. The corresponding n^* are 13, 14, 14 15, 15, respectively. We can see that our FL scheme is more effective for CNN models with a large model size. Note that Case 5 has 138.8 M weights while Case 1 has 25.6 M weights. In particular, for Case 5, we can reduce the total energy consumption up to 53% compared to the baseline.

V. Conclusion

In this paper, we have studied the problem of energy-efficient quantized FL over wireless networks. We have presented the energy model for the quantized FL based on the physical structure of a processing chip and the convergence rate. Then, we have formulated an energy minimization problem that considers a level of precision in the quantized FL. To solve this problem, we have used a line search method. Simulation results have shown that our model requires much less energy than a standard FL model for convergence. The results particularly show significant improvements when the local models rely on large neural networks. In essence, this work provides the first holistic study of the tradeoff between energy, precision, and accuracy for FL over wireless networks.

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2017.

[3] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *arXiv preprint arXiv:1907.10597*, 2019.

[4] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, "Minimum energy quantized neural networks," in *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Apr. 2017.

[5] S. Savazzi, S. Kianoush, V. Rampa, and M. Bennis, "A framework for energy and carbon footprint analysis of distributed and federated edge learning," *arXiv preprint arXiv:2103.1034*, 2021.

[6] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. of IEEE Conf. on Computer Commun.*, Paris, France, May 2019.

[7] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[8] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with cpu-gpu heterogeneous computing," *IEEE Trans. Wireless Commun.*, 2021, to appear.

[9] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, Jul. 2021.

[10] Y. Yang, Z. Zhang, and Q. Yang, "Communication-efficient federated learning with binary neural networks," *IEEE J. Sel. Areas Commun.*, 2021, to appear.

[11] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations." *arXiv preprint arXiv:1609.07061*, 2016.

[12] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proc. of International Conference on Machine Learning (ICML)*, Lille, France, Jul. 2015.

[13] B. Moons, D. Bankman, and M. Verhelst, *Embedded Deep Learning, Algorithms, Architectures and Circuits for Always-on Neural Network Processing*. Springer, 2018.

[14] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. of International Conference on Learning Representations (ICLR)*, May 2020.

[15] B. Luo, X. Li, S. Wang, J. Huangy, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. of IEEE Conf. on Computer Commun.*, Vancouver, BC, Canada, May 2021.

[16] H. H. Bauschke, P. L. Combettes *et al.*, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011, vol. 408.

[17] L. M. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtarik, K. Scheinberg, and M. Takac, "Sgd and hogwild! convergence without the bounded gradients assumption," in *Proc. of International Conference on Machine Learning (ICML)*, Stockholm, Sweden, Jul. 2018.