Federated Optimal Framework with Low-bitwidth Quantization for Distribution System

Ping Feng[‡], Jiahong Ning[‡], Tingting Yang[‡§(✉)], Jiabao Kang[‡], Jiale Wang[‡], and Yicheng Li[‡]

[‡]Dalian Maritime University, Dalian, China
[§]Peng Cheng Laboratory, Shenzhen, China
yangtingting820523@163.com

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437303

***Abstract*—Federated learning is an attractive solution for efficient data processing and optimization in distributed networks. However, communication bottlenecks often result in sluggish optimization and increased resource consumption, leading to reduced system efficiency. To address these challenges, this paper proposes a novel approach based on Federated Low-bitwidth Quantization (FLQ) framework, which optimizes resource utilization in distributed communications. FLQ quantizes network parameters and gradients, significantly reducing computational and communication costs related to parameter broadcasting and gradient uploading. By utilizing an 8-bit low-bitwidth training and binary vector compression technique, our algorithm greatly enhances network convergence and is well-suited to the energy consumption characteristics of servers and end devices. Moreover, to enhance overall device coordination, we develop a dynamic resource allocation scheme that adapts to changing requirements of individual nodes. Results from extensive experiments demonstrate that our approach leads to faster convergence, lower computational and communication costs and optimized joint network deployment in Internet of Things scenarios.**

***Keywords*—Federated learning, Resource optimization, Low-bitwidth quantization, Gradient compression, Internet of things**

I. Introduction

As intelligent devices in the Internet of Things (IoT) become increasingly interconnected, a large number of edge intelligent services are emerging. Federated Learning, as an efficient distributed privacy algorithm, uses a distributed architecture, where each device performs local model training with its own data and only sends updated model parameter to a central server for aggregation [1]. This approach ensures that privacy-sensitive data remains on the user's device. Thus, in the rapidly developing edge intelligent IoT landscape, Federated Learning is one of the most effective solutions.

However, unlike traditional centralized machine learning, Federated Learning distributes device data across a network, with each device responsible for training a subset of the model. This requires frequent communication between devices and the central server, with communication bottlenecks being the most significant factor affecting neural network performance. Additionally, edge devices such as smartphones and smart sensors have limited computing power and communication resources, which presents significant challenges to the resource scheduling capability of distributed systems.

Ping Feng and Jiahong Ning contribute equally to this work. This work is supported by the National Key R and D Program of China under Grant (2020YFB1806800), Key project of Guangdong Province Basic and Applied Basic Research Fund Joint Fund (2019B1515120084).

To address the challenge of limited communication and computing costs encountered by federated learning in resource-constrained IoT scenarios, we propose an efficient Federated Low-bitwidth Quantization (FLQ) framework. Our contribution lies in:

- We introduce FLQ, an innovative framework that leverages low-bitwidth distributed parameter training to accelerate the convergence speed of the quantization process. By optimizing weights, gradients, and activation parameters with low-bitwidth designs, FLQ enables faster convergence and efficient deployment on resource-limited terminal devices. This significantly reduces computation costs and communication rounds.
- For distributed communication, we utilize the Lloyd-Max algorithm for mobility aggregation quantization of gradients. This algorithm effectively projects binary values onto a limited set of values, resulting in efficient compression of gradients. By minimizing the resources required for communication, such as transmitted bits and communication bandwidth, we achieve highly efficient data transfer in the system.
- To address varying communication costs among devices, we have developed a resource allocation and scheduling scheme. This scheme dynamically selects devices that require quantization compression for communication and optimizes the utilization of energy and computing resources. By considering the specific communication costs of each device, we ensure efficient resource.

II. System Model

Consider a wireless multi-user system comprising a base station (BS) and N participating devices, each storing a local data set denoted as S_n. The total data size is defined as $S = \sum_{n=1}^{N} S_n$. In a typical learning problem, given the input-output pairs (x_i, y_i), the objective is to find the model parameters w using a loss function. Examples of loss functions include $f_i(w) = \frac{1}{2}\left(wx_i^T - y_i\right)^2$ for linear regression, where $y_i \in \mathbb{R}$. The loss function of device n is defined as:

$$F_n(w) = \frac{1}{S_n} \sum_{i \in \mathcal{S}_n} f_i(w) \tag{1}$$

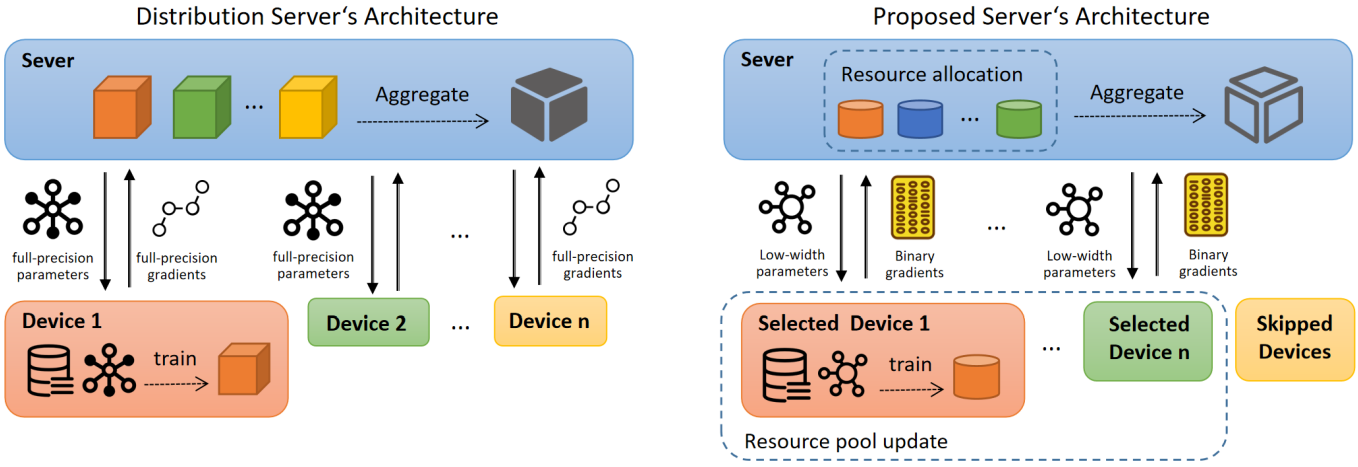

Fig. 1. The proposed framework compared to the traditional distributed server's architecture

Here, $\mathcal{S}_n$ represents the dataset of device n. The learning model aims to minimize the following global loss function:

$$\min_{w\in\mathbb{R}^d} F(w) = \sum_{n=1}^{N} \frac{S_n}{S} F_n(w) \tag{2}$$

This optimization problem seeks to minimize the weighted average of the loss functions across all devices.

According to the information mentioned earlier, federated learning involves three main steps during the training process: parameter broadcasting, gradient uploading, and gradient aggregation. The computation formula for parameter broadcasting is as follows:

$$w_{t+1} = \frac{1}{M} \sum_{i=1}^{M} w_{i,t} \tag{3}$$

Here,w_{t+1} represents the updated model parameter, $w_{i,t}$ represents the model parameter of the i-th participating device at time t, and M denotes the number of participating devices. The formula for gradient uploading in federated learning is given by:

$$\Delta w_{t+1} = \sum_{i=1}^{M} \frac{n_i}{\sum_{k=1}^{M} n_k} \Delta w_{i,t+1} \tag{4}$$

In the given equation,Δw_{t+1} symbolizes the federated average of the parameter update, n_i is the local training data size of the i-th node, M is the aggregate of local training data sizes across all nodes, and $\Delta w_{i,t+1}$ is the parameter update of the i-th node post local training.

Nonetheless, the requisite to wait for all nodes to complete their uploads during gradient aggregation can slow down the upload processes, resulting in delays in gradient broadcasting. In turn, this causes further systemic lags, a phenomenon we refer to as a systemic disaster [2]. Furthermore, considering the constraints of communication and computational resources of edge nodes in edge computing environments, our primary concern is to minimize communication costs and enhance computational speed. Thus, we focus on reducing the interaction communication cost between edge devices and server nodes using quantization compression.

Contrary to traditional distributed service architectures [3] , this research compresses the parameters broadcasted by the server and the training gradients uploaded by terminal devices, consequently conserving communication bandwidth and computational resources. Specifically, during the broadcasting of model parameters, we emphasize network performance by setting the parameter quantization precision to a low-bitwidth of 8 bits. However, for the upload of gradients, we opt for higher compression ratios and convert the gradients into binary vectors that align better with the computational capabilities and energy resources of terminal devices. Additionally, when coupled with our device selection and scheduling scheme, resource allocation is further optimized. Figure 1 provides a lucid comparison between the proposed algorithm and traditional distributed service architecture.

As illustrated in Figure 1, our suggested distributed resource optimization solution comprises three components: low-bitwidth parameter quantization, binary gradient quantization, and a device resource allocation algorithm. In the parameter quantization section, we segregate convolutional kernels, weights, activations, and network gradients into four categories. We leverage bit convolutional kernels for low-bitwidth neural network computation and implement the Straight-Through Estimator (STE) for quantizing weights and activations. In the gradient quantization section, we prioritize nonlinear mappings with a larger numerical range and display the stable convergence of binary gradients. We present a gradient binaryization algorithm rooted in the Lloyd-Max principle and ultimately develop a device resource selection and allocation algorithm to realize global resource optimization.

III. Low-Bitwidth Distributed Parameters Training

We propose the training of low-bitwidth neural networks to enhance the effectiveness of inference algorithms and

optimize memory usage in resource-constrained IoT devices. By limiting the number of bits required for weight parameters and activation functions, low-bitwidth networks significantly reduce communication and computation requirements compared to traditional full-precision networks. Our research focuses on leveraging the benefits of low-bitwidth parameters in optimizing communication resource utilization during server parameter broadcasting, particularly in IoT devices with limited resources.

1) Using Bit Convolution Kernels: In the context of IoT federated scenarios, employing bit convolution kernels for low-bitwidth neural network calculations is an effective solution [4]. Let $\mathbf{x}$ be an m-bit fixed-point integer sequence defined as $\mathrm{x} = \sum_{m=0}^{M-1} c_m(\mathrm{x}) 2^m$, and let $\mathbf{y}$ be a k-bit fixed-point integer sequence defined as $\mathbf{y} = \sum_{k=0}^{K-1} c_k(\mathbf{y}) 2^k$. The dot product of $\mathbf{x}$ and $\mathbf{y}$ can be expressed as $\mathrm{bitcount}(x_{nor})$.

When $\mathbf{x}$ and $\mathbf{y}$ are vectors consisting of -1,1, there exists a variant of Equation 1 that replaces the bitwise and operation with x_{nor}. Thus, the dot product can be represented as:

$$\mathbf{x} \cdot \mathbf{y} = N - 2 \times \mathrm{bitcount}(x_{nor}(\mathbf{x}, \mathbf{y})), x_i, y_i \in \{-1, 1\} \forall i \quad (5)$$

The computational complexity of this formula is $O(MK)$, which is proportional to the bit-widths of x and y. Consequently, utilizing bit convolution kernels for low-bitwidth neural network calculations is highly efficient. This approach improves computation speed and reduces the consumption of computing resources, aligning with the requirements of IoT federated scenarios.

2) Low Biwtidth Quantization of Wight: In this section, we provide a detailed description of our method for obtaining low-bitwidth weights using the STE. The STE has been widely employed in previous research for weight binarization [5]. The definition of the STE is given by:

$$p_o = \frac{1}{2^k - 1} \mathrm{round}\left(\left(2^k - 1\right) p_i\right) \quad (6)$$

Here, p_i represents the input real number, p_o represents the quantized output, and k denotes the number of bits for quantization. The output p_o is a real number that can be represented using k bits.

In our low-bitwidth weight quantization process, we utilize the STE to quantize weights. The input real number within the range [0,1] is quantized to an output within the same range [0,1] with the specified bitwidth.

$$p_o = \mathrm{sign}\left(p_i\right) \quad (7)$$

Here, $\mathrm{sign}\left(p_i\right)$ is a function that returns either -1 or 1 based on the sign of p_i. Then, the weights are scaled after the binarization:

$$p_o = \mathrm{sign}\left(p_i\right) \times \mathrm{E}\left(|p_i|\right) \quad (8)$$

In the equation above, p_i represents the input weight, and p_o represents the quantized weight. The function $\mathrm{E}\left(|p_i|\right)$ denotes the mean of absolute value of each output channel of weights. We utilize a k-bit representation of weights with $k > 1$, we apply the function f_ω^k to the weights, which is defined as:

$$p_o = f_\omega^k\left(p_i\right) = 2 \text{ quantize }_{\mathrm{k}}\left(\frac{\tanh\left(p_i\right)}{2 \max\left(\left|\tanh\left(p_i\right)\right|\right)} + \frac{1}{2}\right) - 1 \quad (9)$$

In the above equation, p_i represents the input weight, and p_o represents the quantized weight. The function $\tanh\left(p_i\right)$ calculates the hyperbolic tangent of the input weight, and quantize_k refers to the quantization operation with k bits.

3) Low Biwtidth Quantization of Activation: In federated IoT scenarios, using low-bitwidth activations as inputs to convolutional layers is crucial for replacing computationally-intensive floating-point convolutions with less demanding bit convolutions. To achieve this, the STE is applied to the input activation r of each weight layer, assuming that the output from the previous layer has already undergone a bounded activation function h to ensure that $r \in [0, 1]$. The activation quantization process can be represented by the equation:

$$f_\alpha^k(r) = \text{ quantize } e_k(p) \quad (10)$$

where $f_\alpha^k(p)$ represents the quantized activation using k bits, and quantize $e_k(p)$ denotes the quantization operation.

By combining our proposed method for low-bitwidth activation quantization with the previously described low-bitwidth weight quantization technique, we can achieve significant compression of memory requirements and computational speedup for neural networks in federated IoT scenarios.

4) Low Biwtidth Quantization of Gradients: When dealing with low-bitwidth gradients, stochastic quantization is necessary to maintain effectiveness [6]. Gradients, which can have a larger value range than activations, are unbounded. Unlike activations, we cannot directly map the gradient range to [0,1] using a differentiable non-linear function. To address this issue, we propose a k-bit quantization function:

$$f_\gamma^k(\ \mathrm{d}r) = 2 \max_0(|\ \mathrm{d}r|) \left[\text{ quantize }_{\ k}\left(\frac{\mathrm{d}r}{2 \max_0(|\ \mathrm{d}r|)} + \frac{1}{2}\right) - \frac{1}{2}\right] \quad (11)$$

Here, $\mathrm{d}r = \frac{\partial c}{\partial r}$ represents the backpropagation gradient output of a layer with respect to its input activation r. The function first applies an affine transformation to the gradient, mapping it to the range [0,1], and then reverses the transformation after quantization.

To compensate for potential bias introduced by gradient quantization, we introduce an additional noise function $N(k) = \sigma^2 \frac{k-1}{2}$. This noise function has the same magnitude as the possible quantization error. The selection of artificial noise is crucial for achieving good performance. Finally, the expression for quantizing gradients to k-bit values is as follows:

$$f_V^k(\ \mathrm{d}r) = 2 \max_0(|\ \mathrm{d}r|) \left[\text{ quantize }_{\ k}\left[\frac{\mathrm{d}r}{2 \max_0(|\ \mathrm{d}r|)} + \frac{1}{2} + N(k)\right] - \frac{1}{2}\right] \quad (12)$$

In summary, our proposed method for stochastic gradient quantization utilizes the STE and a noise function. This ensures that the quantized gradients maintain the required precision for effective training and inference in low-bitwidth neural networks.

IV. Binary Gradient Communication Quantization

Most of the popular and efficient federated learning methods for communication are based on simple gradient updates, with key emphasis on gradient compression to reduce communication costs. One feasible method for achieving efficient communication in resource-limited IoT scenarios is gradient quantization. In order to further reduce communication costs and ensure rapid convergence, we propose a low-bitwidth gradient quantization algorithm. In this section, we first demonstrate how our algorithm can maintain a good linear convergence rate while operating under communication limitations, and we then provide a detailed introduction to the proposed low-bitwidth gradient quantization algorithm.

1) Federated Binary Gradient Convergence Proof: We import the Federated Binary Gradient Convergence Proof, a novel method for optimizing computational and communication costs in federated learning by quantizing low-bit gradients and dynamically allocating resources. The algorithm can be summarized as follows:

Initialization: For each node $i = 1, ..., N$, we initialize $\mathrm{x}^0 \in \mathcal{X}$ and $\mathbf{q}_i^0 = \mathbf{c}_i^0 = C_i\left(\mathbf{x}_i^0\right)$. At each iteration $k \in N$, the following computations are performed:

$$\begin{aligned} \mathbf{x}^{k+1} &= A_i\left(\mathbf{q}^k, \mathbf{x}^k\right) \mathbf{c}_i^{k+1} \\ = C_i\left(\mathbf{x}^{k+1}\right) \mathbf{q}_i^{k+1} &= \text{ quant }_i\left(\mathbf{c}_i^{k+1}, \mathbf{q}_i^k, r^k, b\right) \end{aligned} \tag{13}$$

Here, $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_N)$, and the quantization function $\text{quant}_i\left(\mathbf{c}_i, \mathbf{q}_i, r, b\right)$ quantizes $\mathbf{q}_i^{k+1}$ to b bits per dimension. The quantization ensures that

$$\mid \text{ quant }_i\left(\mathbf{c}_i, \mathbf{q}_i, r, b\right) - \mathbf{c}_i | \infty \leq \frac{r}{2^b - 1} \tag{14}$$

This theorem establishes a fundamental relationship between the number of quantization bits b and the precision of quantization. Our approach consistently maintains the algorithm's convergence rate, even when using a fixed number of b bits for quantization and communication in each iteration. We demonstrate the significant benefits of our method in optimizing computational and communication resources and provide a comprehensive formal convergence proof.

2) Binary Gradient Communication Quantization: We propose a novel binary gradient quantization scheme for distributed optimization, aiming to reduce communication costs while preserving the accuracy of gradient updates. Prior to aggregation, we apply quantization to the gradient vectors uploaded by the nodes, resulting in binary vectors. Each element of the gradient vector undergoes thresholding, transforming it into binary value. Subsequently, the binary vector is scaled to recover the magnitude of the gradient.

The mathematical representation of the proposed binary gradient quantization scheme is as follows:

$$G_m^B(\theta^k) = \frac{|\nabla f_m(\theta^k)|}{l} \text{sign}(\nabla f_m(\theta^k)) \tag{15}$$

Here, $\nabla f_m(\theta^k)$ represents the gradient of the loss function for worker m at iteration k, and l is a scalar normalization factor.

To further reduce communication costs, we impose a restriction on the number of bits used to represent the quantized gradient vector. In contrast to the typical full-precision floats (eg. 64 bits) representation employed by most computers, we utilize b bits to quantize each coordinate of the gradient vector. The quantization operator is denoted as $\mathcal{G}(\cdot)$. During iteration k, the quantized gradient of worker m is given by $G_m(\theta^k) = \mathcal{G}(\nabla f_m(\theta^k), G_m(\hat{\theta}_m^{k-1}))$, where $\hat{\theta}_m^{k-1}$ denotes the previously quantized value.

Quantization is performed by mapping the gradient to the nearest point on a uniformly discretized grid. This grid forms a d-dimensional hypercube centered at $G_m(\hat{\theta}_m^{k-1})$ with a radius of $R_m^k = |\nabla f_m(\theta^k) G_m(\hat{\theta} m^{k-1})|_\infty$. The quantization granularity is defined as 2^{-b+1}, and the gradient innovation $f_m(\theta^k) - G_m(\hat{\theta}_m^{k-1})$ is quantized to b bits at each coordinate.

The quantized gradient is an integer within the range of $\left[0, 2^b - 1\right]$, allowing it to be represented using b bits. To ensure non-negativity in the numerator, we incorporate R_m^k into the numerator of the quantization formula. Additionally, we add 1/2 to the formula to achieve rounding to the nearest point. Consequently, the quantized gradient m can be expressed as:

$$G_m^k = \left\lfloor \frac{f_m\left(\boldsymbol{theta}^k\right) - G_m\left(\hat{\boldsymbol{theta}}_m^{k-1}\right) + R_m^k + 1/2}{2^{-b+1}} \right\rfloor \tag{16}$$

By employing d bits for quantization, we can transmit the quantized gradient using only low-bitwidth $8 + b$ bits for each of the p dimensions, instead of the previous $64d$ bits. The quantized gradient $G_m(\theta^k)$ can be recovered by adding the quantization error δG_m^k to the previous quantized value $G_m\left(\hat{\boldsymbol{theta}}_m^{k-1}\right)$.

In summary, our proposed binary gradient quantization scheme is an effectively method for reducing communication costs in distributed optimization. By quantizing and scaling the gradient vector, we can transmit the quantized gradient using fewer bits without sacrificing accuracy in gradient updates, making it a very practical and efficient solution.

V. Resource Optimization Process

Based on the aforementioned principles of federated resource minimization and distributed modeling [7], we design a weight gradient low-bitwidth quantization-based resource optimization algorithm, suited for use in a federated learning environment. This algorithm leverages low-bitwidth network parameters, binary gradient updates, and optimizations of computational resources to enhance the efficiency and communication cost of distributed machine learning tasks. The steps of the algorithm are listed in Algorithm 1.

Algorithm 1 Federated Low-bitwidth Quantization

Initialization: Low-width network parameter θ, Number of device N, Quantilization parameter B, Systerm parameters f, Transmit power vector P, Resource allocation R.

For $k = 1, 2, \ldots, K$ **do**
  Selects devices $n \in N$ for iteraration.
  Servers broadcasts θ^k to the selected devices.
  For $n = 1, 2, \ldots, N$ **do** on the selected devices
    Calculate distributed parameters B base on low-bitwidth quantization.
    Device n uploads binary gradients δG_m^k via Eqn.(16).
    Calculate optimal resource allocation R according the systerm parameters f and power vector P.
  End for
  Server updates θ on the selected devices.
End for

We propose a dynamic resource optimization algorithm where the resource allocation R and transmission power P are optimized based on the system parameters and the uplink/downlink capacity of each device. Specifically, each device employs the Hamming algorithm to solve the optimization problem, leveraging the previously determined global model and gradient to obtain an approximate solution. Following this, each device transmits its local solution and corresponding gradient to the edge server. The server, in turn, updates the global model and gradient based on the received local solutions and gradients, providing feedback to all devices. This iterative process is carried out over a specified number of rounds, culminating in an optimal solution for resource allocation R and transmission power P. The advantages of this algorithmic approach are manifold, including reduced communication costs, enhanced computational efficiency, and optimal resource allocation achieved through the device selection process. Notably, employing low-bitwidth network parameters and binary gradient updates maintains model accuracy while simultaneously reducing network bandwidth and storage requirements.

VI. EXPERIMENT

A. Comparative experiments of resource optimization

For this paper, a series of experiments have been conducted to verify the performance of the FLQ algorithm, and Figure 2 shows the comparison results of the convergence of FLQ with two well-known federal quantization algorithms, QGD [8] and LAQ [9]. The convergence result graph presented in this study highlights the significant improvements achieved through the FLQ in federated learning for IoT devices. The graph compares the convergence rates of several traditional federated learning approaches with the FLQ approach, with different numbers of nodes and the same workload for each node.

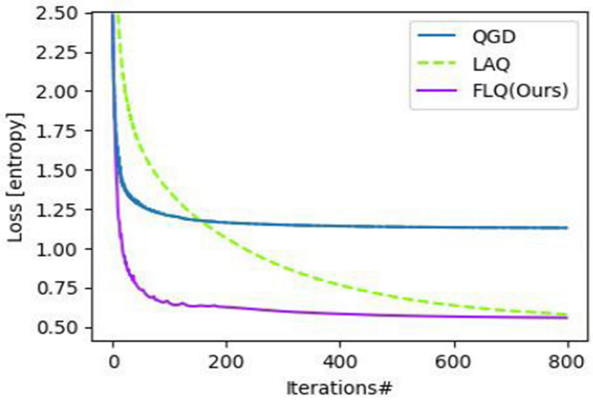

Fig. 2. Comparison of convergence results

TABLE I
COMPARISON TABLE FOR RESOURCE OPTIMIZATION

Method	Iteration	Broadcast (bits)	Upload (bits)	Accuracy (%)
LAQ	1000	4.14×10^9	1.94×10^8	90.81
(Stochastic)	1500	5.62×10^9	1.42×10^{10}	91.84
FLQ	1000	6.03×10^8	7.54×10^7	90.66
(Binary)	1500	8.76×10^8	1.09×10^8	90.69
FLQ	1000	5.96×10^8	5.96×10^8	93.38
(Low-bitwidth)	3000	$1,73 \times 10^9$	$1,73 \times 10^9$	93.73

As observed in the graph, the FLQ approach exhibits faster convergence rates than traditional approaches, demonstrating its potential to significantly improve the efficiency and performance of federated learning. The results show that, despite being applied to resource-limited IoT devices, the FLQ approach is competitive with traditional approaches applied to more powerful computing devices. In addition, it is worth noting that the proposed resource allocation and scheduling scheme further contributes to the overall improvements achieved by the FLQ approach. It allows for dynamic distribution of resources to nodes based on their evolving requirements, ensuring that the system operates in an efficient and effective manner.

Table I presents a comparison of different methods for optimizing resource usage in the context of machine learning models. The aim of this comparison is to evaluate the efficiency and accuracy of each method. We quantize the gradients as 8-bitwide floats and binary values, denoted as FLQ (Low-bitwidth) and FLQ (Binary), respectively. Our results show that FLQ achieves higher accuracy than LAQ and Stochastic methods with a lower number of iterations, while consuming significantly less broadcast and upload resources. Specifically, with only 1000 iterations, FLQ (Binary) and FLQ (Low-bitwidth) consume over 10 times less uplink and downlink resources compared to LAQ (Stochastic) with 1500 iterations. Furthermore, our FLQ (Low-bitwidth) method achieves the highest accuracy among all methods tested, with 93.73 accuracy, while maintaining a reasonable amount of resource consumption. Overall, these results demonstrate the superior performance of our proposed FLQ algorithm compared to other methods in terms of both efficiency and accuracy,

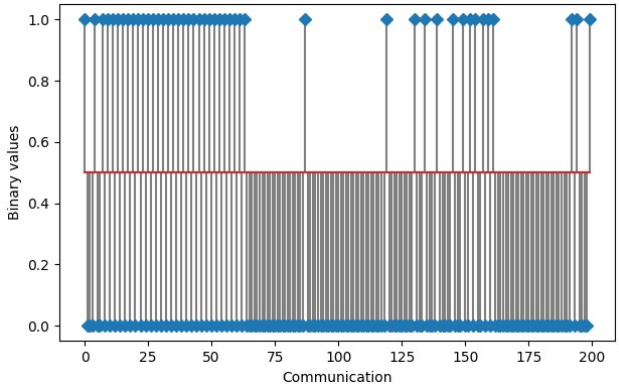

Fig. 3. Results of gradient quantification in communication

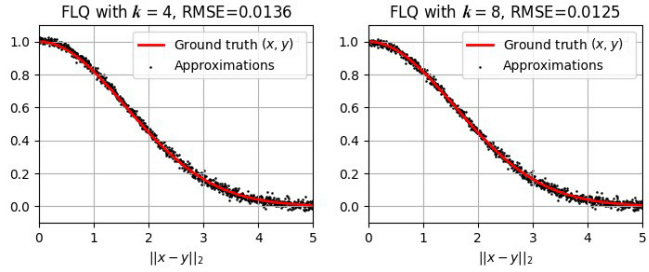

Fig. 4. Quantification of different hyper-parameters

making it a promising choice for optimizing resource usage in distributed machine learning applications.

B. Analytical experiments of quantifies results

In distributed communication, quantization results of gradients can be represented by a graph as shown below, where the horizontal axis represents communication rounds and the vertical axis represents the quantized outcomes. The gradients are quantized as either 0 or 1.

In Figure 3, the quantized results of the gradients effectively illustrate the compressed full-precision gradient values, leading to a significant reduction in communication costs. During the initial communication rounds, there is notable volatility observed in the quantization results of the gradients. However, as the number of communication rounds increases, a discernible upward trend becomes evident. This trend suggests that with a greater number of communication rounds, the gradients are transmitted and utilized more efficiently, thereby enhancing the performance of distributed learning. The quantization results, derived from 200 communications, serve as an objective evaluation and adjustment basis for the gradient quantization strategy. Furthermore, they provide valuable guidance for further advancements and optimizations of the distributed communication algorithm.

The following experiment represents the fitting of quantization results obtained by different hyper-parameters k, where k represents the number of quantization levels used in the quantization process of network gradients during distributed communication.

Our study determined that the selection of $k = 8$ represents a balance between quantization error and communication efficiency. By examining Figure 4, it is evident that varying values of k result in different RMSE (root-mean-square-error) values. The RMSE measures the dissimilarity between the original full-precision gradients and the quantized gradients obtained with different k values. A lower RMSE value indicates a better approximation of the quantized gradients to the full-precision gradients, reflecting improved quantization performance.

The graph clearly demonstrates a decreasing trend in the RMSE value as kk increases. However, we selected $k = 8$ as it provides a balance between quantization accuracy and communication costs in the context of distributed learning. We will conduct further studies to better understand the relationship between k value and RMSE beyond $k = 8$.

VII. CONCLUSION

This paper proposes Federated Low-bitwidth Quantization (FLQ), a comprehensive solution to optimize federated quantization networks for IoT devices with limited resources. Our proposed Federated Low-bitwidth Quantization uses low-bitwidth parameters optimization to accelerate convergence while reducing computation and communication costs, making it an efficient and cost-effective solution. Moreover, we present a novel approach for mobility aggregation quantization of gradient data using Lloyd-Max arithmetic, which saves a significant amount of communication costs. Our resource allocation and scheduling scheme dynamically distributes resources to nodes based on their communication and computational needs, thereby effectively utilizing bandwidth, energy, and computing resources, ensuring optimal performance in IoT scenarios. Our approach aims to faster convergence, reduced computation and communication costs, and streamlined deployment of federated quantization networks in IoT scenarios, thus demonstrating significant application value.

REFERENCES

[1] Tonellotto, Nicola, et al. ”Neural network quantization in federated learning at the edge.” Information Sciences 575 (2021): 417-436.
[2] Shlezinger, Nir, et al. ”UVeQFed: Universal vector quantization for federated learning.” IEEE Transactions on Signal Processing 69 (2020): 500-514.
[3] Tran, Nguyen H., et al. ”Federated learning over wireless networks: Optimization model design and analysis.” IEEE Infocom 2019-IEEE conference on computer communications. IEEE, (2019).
[4] Zhou, Shuchang, et al. ”Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients.” arXiv preprint arXiv:1606.06160 (2016).
[5] Bengio, Yoshua, Nicholas Léonard, and Aaron Courville. ”Estimating or propagating gradients through stochastic neurons for conditional computation.” arXiv preprint arXiv:1308.3432 (2013).
[6] Gupta, Suyog, et al. ”Deep learning with limited numerical precision.” International conference on machine learning. PMLR, (2015).
[7] Yang, Howard H., et al. ”Scheduling policies for federated learning in wireless networks.” IEEE transactions on communications 68.1 (2019): 317-333.
[8] Magnússon, Sindri, et al. ”On maintaining linear convergence of distributed learning and optimization under limited communication.” IEEE Transactions on Signal Processing 68 (2020): 6101-6116.
[9] Sun, Jun, et al. ”Communication-efficient distributed learning via lazily aggregated quantized gradients.” Advances in Neural Information Processing Systems 32 (2019).