Approximate Wireless Communication for Federated Learning

Xiang Ma
Utah State University
Logan, UT, USA
xiang.ma@usu.edu

Haijian Sun
University of Georgia
Athens, GA, USA
hsun@uga.edu

Rose Qingyang Hu
Utah State University
Logan, UT, USA
rose.hu@usu.edu

Yi Qian
University of Nebraska-Lincoln
Lincoln, NE, USA
yi.qian@unl.edu

ABSTRACT

This paper presents an approximate wireless communication scheme for federated learning (FL) model aggregation in the uplink transmission. We consider a realistic channel that reveals bit errors during FL model exchange in wireless networks. Our study demonstrates that random bit errors during model transmission can significantly affect FL performance. To overcome this challenge, we propose an approximate communication scheme based on the mathematical and statistical proof that machine learning (ML) model gradients are bounded under certain constraints. This bound enables us to introduce a novel encoding scheme for float-to-binary representation of gradient values and their QAM constellation mapping. Besides, since FL gradients are error-resilient, the proposed scheme simply delivers gradients with errors when the channel quality is satisfactory, eliminating extensive error-correcting codes and/or retransmission. The direct benefits include less overhead and lower latency. The proposed scheme is well-suited for resource-constrained devices in wireless networks. Through simulations, we show that the proposed scheme is effective in reducing the impact of bit errors on FL performance and saves at least half the time than transmission with error correction and retransmission to achieve the same learning performance. In addition, we investigated the effectiveness of bit protection mechanisms in high-order modulation when gray coding is employed and found that this approach considerably enhances learning performance.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Networks** → **Wireless access networks**.

KEYWORDS

approximate communication, federated learning, wireless networks, bit error rate, modulation

ACM Reference Format:
Xiang Ma, Haijian Sun, Rose Qingyang Hu, and Yi Qian. 2023. Approximate Wireless Communication for Federated Learning. In *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning (WiseML '23), June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3586209.3591399

WiseML '23, June 1, 2023, Guildford, United Kingdom

ACM ISBN 979-8-4007-0133-7/23/06.
https://doi.org/10.1145/3586209.3591399

1 INTRODUCTION

Federated learning (FL) [5] enables local devices to perform machine learning (ML) tasks while still benefiting from the learning generalization ability provided by model parameter sharing. It does not require sharing locally collected data among devices and the server. Instead, only model parameters are shared, thereby effectively protecting data privacy. The FL system is composed of a central parameter server (PS) and a large number of smart local clients (LCs). LCs gather data from onboard sensors and execute a predefined ML task based on the global model broadcast from the PS. After computation, each LC sends local models to the PS for aggregation, and then the updated global model is redistributed back to LCs. This process is repeated until the global model converges.

For edge devices such as UAVs serving as LCs, wireless networks are usually employed to connect them to the PS. However, the nature of wireless channels often results in erroneous information transmission. To address this issue, modern wireless communications utilize forward error correction (FEC) methods, such as convolutional code and low-density parity check code (LDPC), to detect and correct received bit errors. The basic principle of FEC is to encode the message with redundant information in the form of an error correction code (ECC). The receiver can correct the error bits without knowing the actual bits sent by the transmitter. Packet retransmission can be employed when the number of errors exceeds the correction capability of ECC. Although FEC and packet retransmission are powerful, they increase computation and communication overhead, leading to extra power consumption and transmission delays during FL model aggregation. In [12], the authors focused on transmission bit errors in FL but only in a packet erasure channel.

Stochastic gradient descent (SGD) is a widely used optimization method in distributed ML. In FL, each client performs SGD on ML tasks, then a single-step gradient is calculated and sent to the central PS in every communication round. This method, called FedSGD [7], serves as a baseline algorithm for FL. However, for large-scale distributed ML models with millions of parameters, transmitting gradients can cause high delay. Advanced transmission schemes such as non-orthogonal multiple access (NOMA) [13] are good options, but they need to equip with complex decoding methods. Gradient compression is a promising approach to addressing this challenge, where extensive research has shown the effectiveness

of gradient sparsification and quantization with little performance loss. For instance, 1-bit SGD was applied in [10] to reduce gradient transmission size, and in [1], it was shown that 99% of gradients could be dropped. Therefore, we are motivated to apply *approximate wireless communication* to transmit those gradients, i.e., allowing lossy transmission (with errors) in exchange for low latency, low overhead, and less FEC computation in this paper. The tolerance of gradient quantization errors is based on the assumption that the gradient magnitude is small enough. Empirical studies in [6, 14] have shown that the gradients are close to Gaussian distribution, and most gradient values fall in the range of $(-1, 1)$ or even $(-0.01, 0.01)$. Approximate wireless communication for media data transmission has been proposed in [9, 11], and similar ideas can be applied to FL gradient transmission.

In this study, we present a theoretical analysis of bounded ML gradients in commonly used ML settings. Specifically, we prove that gradients in fully connected neural network models and convolutional neural network models are bounded under commonly used conditions. Based on this analysis, we set a limit on the erroneous gradient, together with the approximate transmission in practical wireless networks. Simulation results demonstrate that our proposed method is effective in reducing the impact of bit errors on FL performance, and saves half the time than transmission with Error Correction and ReTransmission (ECRT). The rest of this paper is organized as follows. Section 2 introduces the FL model in wireless networks, Section 3 presents the theoretical analysis of bounded gradients, Section 4 describes the proposed method, Section 5 presents simulation results, and Section 6 concludes the paper.

2 SYSTEM MODEL

The FL model is considered as follows. The FL system consists of M LCs, which are connected to the PS through wireless channels. The overall data amount D is distributed among M devices, with each device m containing D_m data.

2.1 FL System Model

FL is an iterative ML algorithm that performs local computation and global aggregation in each round. Local computation is followed by the LC model uploading, and after global aggregation, the global model is downloaded to each LC. This process is repeated until the model converges. The objective function of FL can be defined as:

$$\min_{\boldsymbol{w}\in R^d} f(\boldsymbol{w}) \quad \text{where} \quad f(\boldsymbol{w}) \overset{\text{def}}{=} \frac{1}{|D|}\sum_{i=1}^{|D|} f_i(\boldsymbol{w}), \tag{1}$$

where $|D|$ is the size of dataset D. $f_i(\boldsymbol{w}) = C(\boldsymbol{x_i}, \boldsymbol{y_i}; \boldsymbol{w})$ is the cost or loss function used to measure the inference error between the data sample $(\boldsymbol{x_i}, \boldsymbol{y_i})$ and the inference made by model parameters $\boldsymbol{w}$. For classification problems in ML, the cross-entropy function is commonly used as the loss function C, particularly in neural network models. In multiclass classification, the label $\boldsymbol{y_i}$ is typically one-hot encoded to ensure each label carries equal weight.

As the data is distributed among M LCs and not on the same device, the objective function (1) needs to be rewritten as follows:

$$f(\boldsymbol{w}) = \sum_{m=1}^{M} \frac{|D_m|}{|D|} F_m(\boldsymbol{w}), \tag{2}$$

where $F_m(\boldsymbol{w}) = \frac{1}{|D_m|}\sum_{i\in D_m} f_i(\boldsymbol{w})$. By distributing the data and computation across multiple devices, a conventional centralized ML problem can be transformed into a distributed FL problem.

Since the cost function for neural networks is typically non-convex, it is challenging to solve directly and find the global minimum. Therefore, the gradient descent method is an iterative optimization algorithm commonly used in ML to find a local minimum point. Stochastic gradient descent (SGD) is a variant of the gradient descent method that can be helpful in escaping local minimums by selecting data samples randomly. As a result, gradients play a central role in the learning process. The gradient is defined as:

$$g = \nabla_{\boldsymbol{w}} C(\boldsymbol{x_i}, \boldsymbol{y_i}; \boldsymbol{w}). \tag{3}$$

The local gradient at each LC in each round can be written as

$$g_t^m = \nabla F_m(w_t). \tag{4}$$

And the global gradient after aggregation is

$$g_t = \sum_{m=1}^{M} \frac{|D_m|}{|D|} g_t^m. \tag{5}$$

The PS stores the model weights from the last round w_t, and then updates the global model as follows

$$w_{t+1} = w_t - \eta g_t. \tag{6}$$

Here, η is the learning rate, which typically falls within the range of $(0, 1)$.

2.2 Wireless Channel Model

Federated Learning is an upper-layer algorithm that does not have knowledge of the lower-layer gradient transmission details. Typically, transmission takes place over wireless channels when LCs are smart sensors or UAVs. For the uplink channel from LCs to PS, we consider the fading channel, which can lead to random bit errors. For the downlink channel, we assume the PS can deliver global gradients to LCs with negligible errors, this can be justified by higher PS transmit power (hence higher SNR) [12].

In the uplink, a time division scheme can be used where each user is assigned to a specific time slot while sharing the same channel. The received signal at the PS can be expressed as follows

$$r_t^m = \sqrt{p_t^m (d^m)^{-\alpha}} h_t^m g_t^m + n_t^m, \tag{7}$$

where r_t^m represents the signal received at the PS from client m. The transmission power is denoted as p_t^m, and the small scale fading is denoted as h_t^m, which is assumed to be complex normal Gaussian distributed, i.e., $h_t^m \sim \mathcal{CN}(0, 1)$. The distance between the PS and client m is represented by d^m, and the path-loss exponent is denoted as α. The additive noise is given as $n^t \sim \mathcal{CN}(0, \sigma^2)$. PS has the knowledge of the channel gain, i.e., $c_t^m = \sqrt{p_t^m (d^m)^{-\alpha}} h_t^m$, and only the noise serves as an error source.

The entire transmission process can be described as follows. First, the gradients are converted from decimal format to binary format.

The bits are then mapped to symbols using a QAM modulation scheme. The symbols are then transmitted through the wireless fading channel. At the receiver end, the signal is decoded with maximum likelihood estimation and then demodulated to the closest point in the constellation,

$$\hat{g}_t^m = \arg_{\bar{g}_t^m \in \mathcal{G}} \min ||r_t^m - \sqrt{p_t^m (d^m)^{-\alpha}} h_t^m \bar{g}_t^m||^2, \tag{8}$$

where $\mathcal{G}$ is the symbol points set of the constellation diagram.

3 BOUNDED GRADIENTS UNDER CONSTRAINTS - A SKETCH OF PROOF

3.1 Gradient Backpropagation

In machine learning, particularly in deep neural networks, back-propagation is widely used for calculating gradients in each layer. In a fully connected neural network, the feed-forward equation at each neuron can be expressed as

$$\begin{aligned} z_j^l &= b_j^l + \sum_k w_{jk}^l a_k^{l-1}, \\ a_j^l &= \sigma(z_j^l). \end{aligned} \tag{9}$$

Here, b is the bias, w is the weights, z is the intermediate output, and a is the final output after the activation function $\sigma(\cdot)$. l represents l-th layer and j, k are indices. The corresponding four fundamental equations in back-propagation for a fully connected network is

$$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L), \tag{10a}$$

$$\delta_j^l = \frac{\partial C}{\partial z_j^l} = \sum_k \frac{\partial C}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial a_j^l}\frac{\partial a_j^l}{\partial z_j^l} = \sum_k \delta_k^{l+1} w_{kj}^{l+1}\sigma'(z_j^l), \tag{10b}$$

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l}\frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l, \tag{10c}$$

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l a_k^{l-1}. \tag{10d}$$

Here, L is the final layer index in the neural network, and δ_j^l is the defined "error" in l-th layer at node j.

To ensure that the gradient $\nabla C = \frac{\partial C}{\partial w_{jk}^l}$ is bounded, it is necessary to limit δ_j^l and a_k^{l-1} based on equation (10d). These two terms are discussed separately. a_k^{l-1} is the activation function output of neuron k at the $(l-1)$-th layer. It depends on the activation function being used. For example, the Sigmoid function ensures that a_k^{l-1} is in the range $(0, 1)$ regardless of the input z_k^{l-1}, while the ReLU activation function requires the input to be bounded. Further discussion and mathematical expressions on activation functions can be found in [8].

The calculation of δ_j^l is described in equation (10b), which involves a summation of products of next layer errors δ_k^{l+1}, weights from node j in the l-th layer to next layer w_{kj}^{l+1}, and the derivative of activation function $\sigma'(z_j^l)$. There are three terms in this equation, and the summation requires the number of neurons in each layer to be finite. The derivative of the activation function $\sigma'(z_j^l)$ also depends on the activation function used, with the derivative being in the range $(0, 0.25)$ for the Sigmoid function and $\{0, 1\}$ for ReLU. The weight w_{kj}^{l+1} depends on model initialization, learning rate η, and last round gradient based on equation (6). Weight initialization methods typically generate random weight values in the range $(-1, 1)$ or even smaller, and there are newer initialization methods such as [3] and [4]. Without loss of generality, we assume that the weight value w_{kj}^{l+1} is bounded. The error δ_k^{l+1} can be written in the same way as in equation (10b) with elements in the $(l+2)$-th layer, and this process continues all the way back to the final layer. In classification problems, the softmax function is commonly used as the activation function in the final layer to normalize the output class probabilities. When the cross-entropy loss function is used, it can be combined with the softmax function. For cross-entropy loss function,

$$C = -\sum_i y_i log(p_i), \tag{11}$$

where y_i is the input truth label, p_i is the softmax probability for the i-th class

$$p_i = \sigma(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}}, \tag{12}$$

and the derivative is

$$\frac{\partial p_i}{\partial z_j} = \begin{cases} p_i(1-p_j), & \text{if } i = j; \\ -p_j \cdot p_i, & \text{if } i \neq j. \end{cases} \tag{13}$$

Equation (10a) can be written as

$$\begin{aligned} \delta_j^L = \frac{\partial C}{\partial p_i^L}\frac{\partial p_i^L}{\partial z_j^L} &= -\sum_i y_i \frac{\partial log(p_i)}{\partial p_i}\frac{\partial p_i}{\partial z_j}, \\ &= -\sum_i y_i \frac{1}{p_i}\frac{\partial p_i}{\partial z_j}, \\ &= -y_j(1-p_j) - \sum_{i \neq j} y_i \frac{1}{p_i}(-p_j \cdot p_i), \\ &= p_j \cdot \sum_i y_i - y_j. \end{aligned} \tag{14}$$

Since y is a one-hot encoded label vector, so $\sum_i y_i = 1$, that is

$$\delta_j^L = p_j - y_j. \tag{15}$$

As p_j takes values between 0 and 1 and y_j is either 0 or 1, δ_j^L lies in the interval $(-1, 1)$.

To summarize, in a fully connected neural network with cross-entropy as the cost function and softmax function as the activation function in the final layer, the final layer error δ_j^L is in the range $(-1, 1)$. In addition, if the weights are assumed in the range $(-1, 1)$ and Sigmoid functions are used as activation functions in other layers, the gradient $\frac{\partial C}{\partial w_{jk}^l}$ is bounded by the sum of the number of neurons after l-th layer, denoted as B^l.

3.2 Gradient in Convolutional Neural Network

Modern image recognition tasks often use convolutional neural networks (CNNs) as an advanced technique. CNNs are a special variant of feedforward networks that consist of three types of layers:

convolutional layers, pooling layers, and fully connected layers. The feedforward process of a CNN can be written as:

$$z^1_{j,k} = b^1_{j,k} + \sum_p \sum_q w^1_{p,q} x^0_{j+p,k+q}, \tag{16a}$$

$$a^1_{j,k} = \sigma(z^1_{j,k}), \tag{16b}$$

$$a^2_{j,k} = \max(a^1_{2j,2k}, a^1_{2j+1,2k}, a^1_{2j,2k+1}, a^1_{2j+1,2k+1}), \tag{16c}$$

$$z^3_i = b^3_i + \sum_{j,k} w^3_{i;j,k} a^2_{j,k}, \tag{16d}$$

$$a^3_i = \sigma(z^3_i). \tag{16e}$$

For the sake of simplicity, we assume that this CNN network comprises only three layers. Equation (16a) and (16b) represent the convolutional layer, equation (16c) represents the max pooling layer with a 2×2 kernel, and equations (16d) and (16e) represent the fully connected layer. Here, p and q denote the indices of convolutional kernels.

Now the backpropagation for the CNN network becomes

$$\delta^3_i = \frac{\partial C}{\partial z^3_i} = \frac{\partial C}{\partial a^3_i}\frac{\partial a^3_i}{\partial z^3_i} = \frac{\partial C}{\partial a^3_i}\sigma'(z^3_i), \tag{17a}$$

$$\begin{aligned}\delta^1_{j,k} = \frac{\partial C}{\partial z^1_{j,k}} &= \sum_i \frac{\partial C}{\partial z^3_i}\frac{\partial z^3_i}{\partial a^2_{s,t}}\frac{\partial a^2_{s,t}}{\partial z^1_{j,k}}, \\ &= \sum_i \delta^3_i w^3_{i;s,t}\frac{\partial a^2_{s,t}}{\partial a^1_{j,k}}\frac{\partial a^1_{j,k}}{\partial z^1_{j,k}}, \\ &= \sum_i \delta^3_i w^3_{i;s,t}\frac{\partial a^2_{s,t}}{\partial a^1_{j,k}}\sigma'(z^1_{j,k}), \\ &= \begin{cases}\sum_i \delta^3_i w^3_{i;s,t}\sigma'(z^1_{j,k}), & \text{if case 1;}\\ 0, & \text{otherwise;}\end{cases}\end{aligned} \tag{17b}$$

$$\frac{\partial C}{\partial w^3_{i;j,k}} = \frac{\partial C}{\partial z^3_i}\frac{\partial z^3_i}{\partial w^3_{i;j,k}} = \delta^3_i a^2_{j,k}, \tag{17c}$$

$$\frac{\partial C}{\partial w^1_{p,q}} = \frac{\partial C}{\partial z^1_{j,k}}\frac{\partial z^1_{j,k}}{\partial w^1_{p,q}} = \delta^1_{j,k} x^0_{j+p,\,k+q}. \tag{17d}$$

Here, case 1 is $a^2_{j,k} = \max(a^1_{2s,2t},\ a^1_{2s+1,2t},\ a^1_{2s,2t+1},\ a^1_{2s+1,2t+1})$ in equation (17b), and $s = \frac{j}{2}, n = \frac{k}{2}$.

Similarly, when the cross entropy serves as the loss and the activation function in the last layer is the softmax function, δ^3_i lies in the range $(-1, 1)$. In order to bound the gradient $\frac{\partial C}{\partial w^3_{i;j,k}}$ in the fully connected layer, it is necessary to ensure that $a^2_{j,k}$ or $\max(a^1_{2j,2k}, a^1_{2j+1,2k}, a^1_{2j,2k+1}, a^1_{2j+1,2k+1})$ is also bounded. If the Sigmoid function is used as the activation function in the first layer, then $a^2_{j,k}$ is bounded within the range of $(0, 1)$, which results in $\frac{\partial C}{\partial w^3_{i;j,k}}$ being bounded within the range of $(-1, 1)$. The gradient $\frac{\partial C}{\partial w^1_{p,q}}$ in equation (17d) can be bounded by considering the two terms involved. Firstly, $x^0_{j+p,k+q}$ is the input and is bounded. Secondly, $\delta^1_{j,k}$ can be expressed as either 0 or $\sum_i \delta^3_i w^3_{i;s,t}\sigma'(z^1_{j,k})$. We know that δ^3_i is in the range $(-1, 1)$ and $\sigma'(z^1_{j,k})$ is in the range $(0, 0.25)$ if the activation function is Sigmoid. For $w^3_{i;s,t}$, its value depends on the model initialization, learning rate η, and the last round gradient based on equation (6), as we discussed above. If we assume $w^3_{i;s,t}$ is bounded in $(-1, 1)$, then $\frac{\partial C}{\partial w^1_{p,q}}$ can be bounded by the number of neurons in the last layer, also denoted as B^l.

4 PROPOSED METHOD

In section 3, we have presented mathematical proofs that under certain conditions, the gradients are bounded by B^l. Empirically, it has been shown in [6, 14] that the gradients are not only bounded but also bounded within the range of $(-1, 1)$ or even a smaller range. This allows us to have an expected value (statistically) for the received gradient at the PS. Correspondingly, we first design a QAM encoding scheme for the bounded gradients,

4.1 QAM Encoding

In ML, gradients are commonly expressed using 32-bit floating-point numbers. These numbers follow the format defined by the IEEE-754 standard, which assigns the first bit to the sign and the next 8 bits to the exponent, leaving the final 23 bits for the fraction. Bits in different locations have varying importance. The sign bit controls the sign of the gradient value, while the exponent part defines the integer and decimal values. The fraction part only controls the decimal value, and thus, the exponent bits are more important than the fraction bits. Furthermore, the bits located on the left side of the exponent are more important than those on the right.

During transmission, each bit is susceptible to noise, which can cause corruption. To avoid block corruption, we employ interleaving at the transmitter and de-interleaving at the receiver, reducing the likelihood of multiple error bits taking place together. In the bit representation, when the second bit in the 32-bit representation, i.e., the first bit in the exponent part, is 1 and all other 31 bits are 0s, the decimal value is 2. Conversely, when the second bit in the 32-bit representation is 0, and all other 31 bits are 1s, the magnitude is less than 2. When assuming a magnitude threshold of 1 for the gradient value, the first bit in the exponent part is always 0. This motivated us, on the receiver side, regardless of the value decoded in the second-bit location of the gradient, it will be set to 0, as shown in Figure 1.

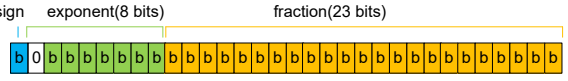


Figure 1: Received Gradient Bit Representation

Moreover, we have also observed that different modulation schemes have varying effects on bits located at different positions [11]. This is important not only in media message transmission but also in ML model parameter transmission. In wireless transmission, the transmission system is not aware of the relative importance of data bits and treats all the bits equally. When using QPSK as the modulation scheme, each symbol consists of 2 bits, with the bit combinations being 00, 01, 11, 10. The error probability for the first and second bits in QPSK is the same. In contrast, 16-QAM has 4 bits per symbol, and the constellation map with gray code is shown below.

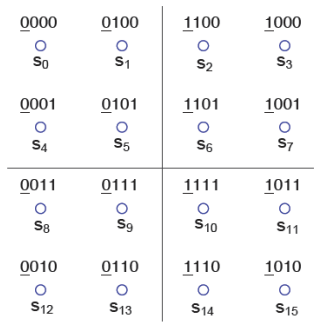


Figure 2: 16-QAM with Gray Coding Constellation Map

The bits with underlines in Figure 2 correspond to the first bit in each symbol, which are the most significant bits (MSB), while the fourth or last bits are the least significant bits (LSB). When the transmission probability for each symbol is the same, the error probability for the MSB is higher than for the LSB. For example, if the symbol s_0 is decoded with an error, it is most likely to be decoded as s_1, s_4, or s_5. The MSB bit remains the same, while the LSB changes twice. This is summarized in Table 1. The symbols in the other quadrants are symmetric to the first quadrant, so the results are identical. High-order modulation schemes with gray coding provide built-in protection for the MSB bits of gradient values in bit representation.

Table 1: 16-QAM MSB/LSB Error Count

Symbol	Potential Error Symbol	MSB Error Count	LSB Error Count
s_0	s_1, s_4, s_5	0	2
s_1	s_0, s_2, s_4, s_5, s_6	2	3
s_4	s_0, s_1, s_5, s_8, s_9	0	2
s_5	$s_0, s_1, s_2, s_4, s_6, s_8, s_9, s_{10}$	3	3

4.2 Approximate Wireless Transmission

To further improve gradient exchange efficiency over wireless networks, we propose an approximate wireless transmission scheme. Essentially, since the gradient is resilient to errors, as witnessed in the existing gradient compression methods, the delivered messages (gradient) do not have to be accurate. Hence we eliminate FEC and re-transmission when channel SNR is satisfactory. While the exact SNR value is to be determined, our empirical results have shown that at around 10-20 dB, the BER is acceptable for FL. Notice that our approach is different from user datagram protocol (UDP), where retransmission is not required either. The difference is that UDP works at a higher level, and the CRC is used only to check the UDP payload. When the error happens at the physical or MAC layer, re-transmission is still issued. Our approach eliminates both FEC and re-transmission at lower layers, including physical and MAC layers. The benefit is three-fold: 1) it reduces the communication overhead, so more data bits can be transmitted; 2) it reduces the computation overhead for FEC, and this is very appealing for edge devices; 3) it improves latency performance since no re-transmission is required.

5 SIMULATION RESULTS

In this section, we first present the parameter settings for the simulation. We then provide the empirical probability of bit error versus signal-to-noise ratio (SNR) under the wireless channel mentioned earlier. Among the modulation schemes tested, QPSK achieves a better bit error rate (BER) than 16-QAM and 256-QAM at the same SNR level. Next, we compare the FL performance under three scenarios: ECRT transmission with error correction and retransmission, naive erroneous transmission, and erroneous transmission with our proposed scheme. The naive error transmission is the transmission in wireless networks with errors without extra operation. Compared to naive erroneous transmission, our proposed scheme achieves a high testing accuracy. Furthermore, compared to ECRT transmission, our proposed scheme for erroneous transmission saves much more time. Finally, we discuss different modulation schemes with gray coding to show the built-in bit protection for MSB in the bit representation.

We consider a typical FL setting in our simulation, where $M = 100$ LCs are connected to the PS, and all LCs participate in the learning process in each communication round. The LCs perform image classification tasks using the MNIST dataset, which consists of handwritten digits 0-9. The training set contains $60,000$ images, and the test set contains $10,000$ images, with each digit having approximately $6,000$ images in the training set and $1,000$ images in the test set. To simulate a realistic scenario where data is collected from the environment, we distribute the data in a non-iid way, with each LC having 2 digits and each digit having around 300 images for training. We use a convolutional neural network (CNN) as the ML model, with 2 convolutional layers, each having a kernel size of 5, 2 max-pooling layers with size 2, and 2 fully connected layers. ReLU is used as the activation function in all layers except the last one, which uses the log softmax function. The learning rate is set to $\eta = 0.01$.

We set the path loss exponent for the wireless channel as $\alpha = 3$, and consider a distance of 10m between the PS and LCs. The transmission power at the LCs is normalized to 1. We use QPSK as the modulation scheme, and the receiver SNR is set at $\gamma = 10$ dB unless otherwise specified.

Under the specific fading channel, QPSK achieves a lower BER compared to 16-QAM and 256-QAM at the same SNR level. For QPSK, at SNR=10 dB, the BER is approximately 4×10^{-2} while the BER is 5×10^{-3} when SNR is 20 dB.

In the ECRT scheme with error correction and retransmission, all the bits are received correctly by the PS, which incurs a cost for forward error correction (FEC) and possible retransmission when the error exceeds the FEC capability. In contrast, the naive error transmission scheme involves transmitting bits with errors without prior knowledge of the gradients, where the test accuracy remains flat at around 10%, similar to random guessing as shown in Figure 3. This occurs because the model cannot learn anything due to transmission errors. Our proposed method, however, takes into account prior knowledge of the gradient values, which are expected to be in the range of (-1, 1). This makes the proposed scheme achieves much better results than naive error transmission.

To quantify the transmission time saved by our proposed method compared to ECRT transmission, we employ a practical IEEE 802.11

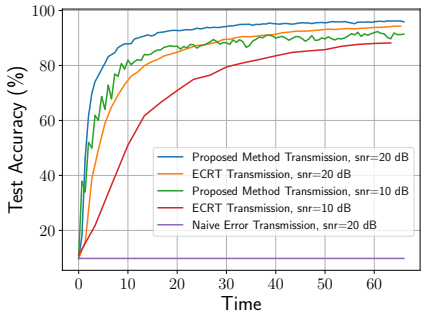
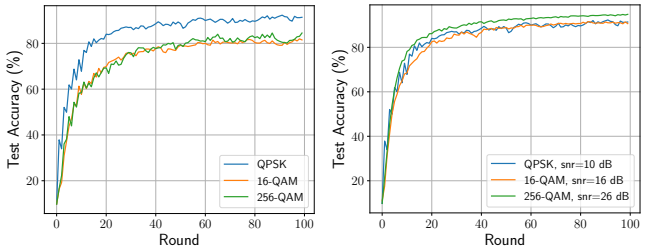


Figure 3: Test Accuracy v.s. Communication Time

protocol with LDPC error correction coding. LDPC is a promising ECC that can approach the Shannon limit. For different coding rates, there exists a trade-off between error correction capability and transmission overhead. Lower coding rate results in high transmission overhead but comes with high error correction capability. Here, we use a coding rate of $1/2$ to enhance error correction. According to [2], the minimum Hamming distance is 15 for a code rate of $1/2$ when the code length is 648, and we search using the parity check matrix. This results in an error correction capability of 7 bits. In Figure 3, the transmission with LDPC coding with retransmission takes 2× time than the proposed scheme to achieve 80% accuracy at SNR=20 dB while it takes more than 3× for SNR=10 dB for the LDPC coding with retransmission scheme to achieve that performance.



(a) Test Accuracy with the Same SNR=10 dB **(b) Test Accuracy with the Same BER** $\approx 4 \times 10^{-2}$

Figure 4: Test Accuracy with the Same SNR/BER

To demonstrate the effectiveness of built-in MSB bit protection of high-order modulation with gray coding, we begin by presenting the test accuracy of different modulations at the same SNR in Figure 4(a). At an SNR of 10 dB, the BER for QPSK, 16-QAM, and 256-QAM is roughly 4×10^{-2}, 10^{-1}, and 3×10^{-1}, respectively. Because QPSK results in fewer errors, its learning performance is better than in 16-QAM and 256-QAM.

In Figure 4(b), we present a scenario where the BER is made the same for different modulations. To accomplish this, we increase the SNR for 16-QAM to 16 dB and the SNR for 256-QAM to 26 dB. Consequently, the BER for all three modulation schemes is 4×10^{-2}. In this scenario, 256-QAM achieves significantly better learning performance than QPSK, with smaller transmission errors in 256-QAM than QPSK.

6 CONCLUSIONS

In this paper, we proposed a federated learning parameter transmission scheme in wireless networks. Unlike existing transmission methods that rely on forward error correction and retransmission, we proposed gradient transmission with errors based on prior knowledge of gradient values. The gradient value is mathematically proven to be within a small range under certain constraints, so the received gradient value is expected to be within that range. This approach achieves learning performance with errors much better than naive error transmission and saves at least half time to achieve the same learning performance as ECRT transmission. Additionally, we explored high-order modulation and demonstrated improved learning results. In the future, our plan is to quantify the impact of communication errors on FL performance.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under grants CNS-2007995, CNS-2008145, ECCS-2139508, and ECCS-2139520. The work of H. Sun is supported in part by the US National Science Foundation grant CNS-2236449.

REFERENCES

[1] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021* (2017).
[2] Brian K Butler. 2016. Minimum distances of the QC-LDPC Codes in IEEE 802 Communication Standards. *arXiv preprint arXiv:1602.02831* (2016).
[3] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.
[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.
[5] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
[6] Ahmed M Abdelmoniem, Ahmed Elzanaty, Mohamed-Slim Alouini, and Marco Canini. 2021. An efficient statistical-based gradient compression technique for distributed training systems. *Proceedings of Machine Learning and Systems* 3 (2021), 297–322.
[7] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
[8] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378* (2018).
[9] Benjamin Ransford and Luis Ceze. 2015. SAP: an architecture for selectively approximate wireless communication. *arXiv preprint arXiv:1510.03955* (2015).
[10] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 2014. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth annual conference of the international speech communication association*.
[11] Sayandeep Sen, Syed Gilani, Shreesha Srinath, Stephen Schmitt, and Suman Banerjee. 2010. Design and implementation of an" approximate" communication system for wireless media applications. In *Proceedings of the ACM SIGCOMM 2010 Conference*. 15–26.
[12] Mahyar Shirvanimoghaddam, Ayoob Salari, Yifeng Gao, and Aradhika Guha. 2022. Federated learning with erroneous communication links. *IEEE Communications Letters* 26, 6 (2022), 1293–1297.
[13] Haijian Sun, Xiang Ma, and Rose Qingyang Hu. 2020. Adaptive Federated Learning With Gradient Compression in Uplink NOMA. *IEEE Transactions on Vehicular Technology* 69, 12 (2020), 16325–16329. https://doi.org/10.1109/TVT.2020.3027306
[14] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2017. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *Advances in neural information processing systems* 30 (2017).