2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)

Analog Compression and Communication for Federated Learning over Wireless MAC

Afshin Abdi, Yashas Malur Saidutta, Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology
{abdi, ysaidutta3, fekri}@gatech.edu

***Abstract*—In this paper, we consider federated learning in wireless edge networks. Transmitting stochastic gradients (SG) or deep model's parameters over a limited-bandwidth wireless channel can incur large training latency and excessive power consumption. Hence, data compressing is often used to reduce the communication overhead. However, efficient communication requires the compression algorithm to satisfy the constraints imposed by the communication medium and take advantage of its characteristics, such as over-the-air computations inherent in wireless multiple-access channels (MAC), unreliable transmission and idle nodes in the edge network, limited transmission power, and preserving the privacy of data. To achieve these goals, we propose a novel framework based on Random Linear Coding (RLC) and develop efficient power management and channel usage techniques to manage the trade-offs between power consumption, communication bit-rate and convergence rate of federated learning over wireless MAC. We show that the proposed encoding/decoding results in an unbiased compression of SG, hence guaranteeing the convergence of the training algorithm without requiring error-feedback. Finally, through simulations, we show the superior performance of the proposed method over other existing techniques.**

***Index Terms*—Federated Learning, Analog Compression, Machine Learning, Edge Network**

I. Introduction

The training data in a wireless edge network is generally unevenly distributed over a large number of nodes with limited resources such as communication bandwidth and battery power. Transferring data from edge nodes to a central server to train a deep model is often infeasible due to the limited wireless bandwidth and battery power as well as privacy concerns in some applications. Hence, it is desired to train the deep model over an edge network in a distributed manner. Federated learning [1], [2], [3], [4] enables such networks to collaboratively learn a unified deep model without transmitting the training data to a central server.

Federated learning differs from traditional distributed machine learning as 1) the number of edge nodes is generally very large, and 2) the data observed by the nodes are usually unbalanced and non-iid. Hence, distributed optimization algorithms which are often developed for high performance computing clusters are not readily applicable to training deep models over edge networks. The core idea is that each node uses its own dataset to locally compute the gradients or updates the model's parameters. Then the Stochastic Gradients (SGs) or parameters are globally aggregated to improve the deep model. However, the requirement to transmit the gradients or updates can put a huge burden on the network especially for state-of-the-art deep models with millions of parameters. There exist two possible approaches to mitigate these shortcomings: (1) reducing the frequency at which the nodes transmit their data [1], [5], [6], [7], [8], and (2) compress the SGs or parameters to reduce number of transmitted values. In this work, we will consider the second approach with focus on SGs. The majority of existing methods rely on quantizing the SGs [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], sparsification [19], [20], [21], [22], [23], [24] or a combination of both. However, direct application of these compression methods requires transmission of the compressed values without any interference from other nodes in the wireless channel. Therefore, such approaches require channel assignments to individual nodes (e.g., through TDMA or FDMA), which increases the latency. The majority of past works in federated learning over wireless MAC are restricted to the transmission of raw (uncompressed) SGs or parameter updates [25], [26], [27], [28]. The exceptions are [29], [30] which implicitly require SGs to have almost the same sparsity patterns, and thus limiting their use to the iid datasets where the SGs computed by the edge nodes have similar sparsity pattern. In contrast, we seek to develop a framework that incorporates the requirements of ML in wireless networks, and exploits properties such as over-the-air computation over wireless-MAC.

In section II, we overview the problem and the constraints and characteristics of the wireless MAC. Section III presents the background and preliminaries. The proposed framework is introduced and analyzed in section IV. Finally, in section V, we empirically validate our proposed method and the theoretical results.

Notations

Bold lowercase letters represent vectors and the i-th element of the vector $\boldsymbol{x}$ is denoted as x_i or $\left[\boldsymbol{x}\right]_i$. Matrices are denoted by bold capital letters such as $\boldsymbol{X}$, with the (i,j)-th element is represented by $X_{i,j}$ or $\left[\boldsymbol{X}\right]_{i,j}$. $\boldsymbol{A} \odot \boldsymbol{B}$ is the Hadamard product of $\boldsymbol{A}$ and $\boldsymbol{B}$. $\mathbf{1}$ is a vector or matrix of all ones, whose size would be clear from the context.

Throughout the paper, for notational convenience, we often omit the dependency of variables and parameters on time t. d

This work is supported Jointly by Intel and National Science Foundation under NSF-Intel MLWINS award ID **2003002**.

978-1-7281-5478-7/20/$31.00 ©2020 IEEE

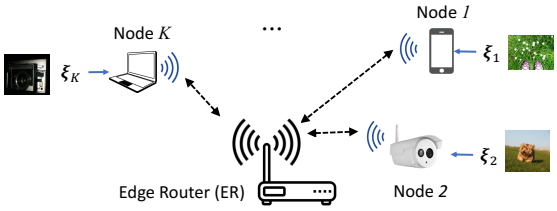


Fig. 1: Wireless Edge Network

usually refers to the number of parameters of the deep model and K is the number of nodes in the edge network.

II. Problem Statement

Figure 1 illustrates the wireless edge network considered throughout this paper. *We will refer to the edge device as edge node or simply a node throughout*. The uplink communication is over a wireless Multiple Access Channel (MAC), which naturally performs an analog over-the-air addition on incoming signals from the edge nodes to the router. However, the downlink communication from the edge router to the edge nodes is wireless broadcast. Like edge nodes, the edge router is also assumed to have some memory and computing power.

For the communication between edge nodes and the edge router (ER), we assume symbol level synchronization (e.g., via a synchronization channel or synchronized clocks). During the uplink transmission, let $\boldsymbol{x}_i \in \mathbb{R}^m$ be the symbols transmitted by the i-th node. The received signal at the ER is given by

$$\boldsymbol{y} = \sum_i \boldsymbol{h}_i \odot \boldsymbol{x}_i + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{h}_i \in \mathbb{C}^m$ is subchannels' gains from node i to ER, and $\boldsymbol{\eta} \sim \mathcal{CN}(0, \sigma^2 \boldsymbol{I})$ is the MAC channel noise, assumed to be complex Gaussian and independent across subchannels. In the downlink, if ER broadcasts $\boldsymbol{y}$ to the edge network, each node receives a noisy scaled replica of $\boldsymbol{y}$. For simplicity, we assume the channel state information is available at the nodes. Hence, by compensating for the downlink channel gains, the reconstructed value at the i-th node is given by $\widehat{\boldsymbol{y}}_i = \boldsymbol{y} + \boldsymbol{\eta}_i'$, where $\boldsymbol{\eta}_i' \sim \mathcal{N}(\boldsymbol{0}, \sigma_i^2 \boldsymbol{I})$.

Consider training a deep model with a cost function $F(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\xi}}[\ell(\boldsymbol{\xi}; \boldsymbol{\theta})] \approx \frac{1}{n} \sum_{\boldsymbol{\xi} \in \mathcal{X}} \ell(\boldsymbol{\xi}; \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is the parameters of the deep model, $\ell(\boldsymbol{\xi}; \boldsymbol{\theta})$ is the loss function of the model corresponding to input data $\boldsymbol{\xi}$, $\mathcal{X}$ is the training dataset and $n = |\mathcal{X}|$ is the number of training samples. Assume that node i observes only subset $\mathcal{X}_i \subset \mathcal{X}$, $|\mathcal{X}_i| = n_i$. Hence, its local objective function is $f_i(\boldsymbol{\theta}) = \frac{1}{n_i} \sum_{\boldsymbol{\xi} \in \mathcal{X}_i} \ell(\boldsymbol{\xi}; \boldsymbol{\theta})$, and the total cost function can be reformulated as $F(\boldsymbol{\theta}) = \sum_i \alpha_i f_i(\boldsymbol{\theta})$, where $\alpha_i = n_i/n$ is introduced to compensate for unbalanced training data sets among edge nodes.

In this paper, we focus on federated learning over wireless edge, with the focus on compressing SGs to reduce communication overhead. However, the framework and its convergence analysis can be extended to the compression and transmission of parameter updates. Further, we require the compression

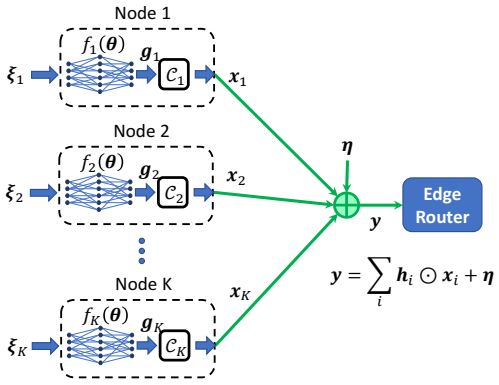


Fig. 2: Federated learning over Wireless MAC. Node i observes data $\boldsymbol{\xi}_i$ and based on its local model, computes the model's stochastic gradient $\boldsymbol{g}_i$. Compression engine $\mathcal{C}_i$ compresses $\alpha_i \boldsymbol{g}_i$ to $\boldsymbol{x}_i$ and transmits over MAC. The edge router receives the noisy aggregated data $\boldsymbol{y} = \sum_i \boldsymbol{h}_i \odot \boldsymbol{x}_i + \boldsymbol{\eta}$ and broadcasts it back to the edge nodes.

algorithm to be tailored to satisfy the constraints imposed by the communication medium and take advantage of its characteristics, i.e.,

- **P1** The MAC channel (1) can naturally compute weighted average of the transmitted values.
- **P2** The transmission power of each individual node is bounded, i.e., $\mathbb{E}\left[\|\boldsymbol{x}_i\|^2\right] \leq P_i$.
- **P3** All edge nodes may not transmit at every round of communication.
- **P4** Edge-node's private information should not leak to ER.

III. Preliminaries

The high level diagram of federated learning over wireless MAC is shown in Fig. 2. Let $\boldsymbol{g}_i \in \mathbb{R}^d$ be the stochastic gradient computed at node i, such that $\mathbb{E}[\boldsymbol{g}_i] = \nabla f_i(\boldsymbol{\theta})$. Therefore, the SG of $F(\boldsymbol{\theta})$ would be given as $\boldsymbol{g} = \sum_i \alpha_i \boldsymbol{g}_i$. For each node, our goal is to design an efficient encoding algorithm $\mathcal{C}_i(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ to compress scaled SGs, where $m \ll d$ and will be selected to control the trade offs among the wireless bandwidth requirement, the communication latency, and the training convergence rate.

For simplicity, we assume that the channel state information and hence $\boldsymbol{h}_i$ is known at node i. After compensating for the channel loss[1], $\boldsymbol{x}_i = \boldsymbol{h}_i^{-1} \odot \mathcal{C}_i(\alpha_i \boldsymbol{g}_i)$ would be the transmitted signal at node i. Then the received signal at the ER is given by

$$\boldsymbol{y} = \sum_{i \in \mathcal{K}} \mathcal{C}_i(\alpha_i \boldsymbol{g}_i) + \boldsymbol{\eta}, \tag{2}$$

where $\mathcal{K} \subset \{1, 2 \ldots, K\}$ is the subset of nodes transmitting their data. The aggregated signal $\boldsymbol{y}$ is then broadcasted back to the nodes to estimate an SG of $F(\boldsymbol{\theta})$. Ideally, at each node, we wish to be able to compute $\boldsymbol{g} = \sum_i \alpha_i \boldsymbol{g}_i$, i.e., the stochastic gradient of the objective function $F(\boldsymbol{\theta})$. However, due to the

[1]Note that here, for the presentation simplicity, we did not ignore subchannels with huge losses. However, in practice, those poor channels can be discarded during data transmission.

limited bandwidth, channel noise and the loss at the decoder of $\mathcal{C}_i(\cdot)$, the estimated SG may not be the same as $\boldsymbol{g}$. We consider two additional criteria in developing the encoders $\mathcal{C}_i(\cdot)$'s:

- **C1** For privacy, given $\boldsymbol{y}$, the ER should not be able to infer any information about individual $\boldsymbol{g}_i$'s.
- **C2** Each participating node should be able to estimate an *unbiased* stochastic gradient of $F(\boldsymbol{\theta})$ from $\boldsymbol{y}$. This ensures the convergence of the SG-based learning algorithms. Otherwise, the training procedure can drift away from converging to the optimum (or good) solution, unless the bias in SG is compensated by error-feedback [31], [32], [17]. This, in turn, increases the memory footprint of the compression algorithm.

Lemma 1. *The condition in* **C1** *imposes a Homomorphic property on the encoder. As such to satisfy* **C1***, it is necessary that the encoder $\mathcal{C}_i(\cdot)$ be identical linear transforms for all i.*

As a result of Lemma 1, we focus on the encoders given by $\mathcal{C}_i(\boldsymbol{z}) = \boldsymbol{A}\boldsymbol{z}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times d}$ to be designed. On the other hand, note that if $\boldsymbol{A}$ is chosen to be fixed and deterministic, the information in the SGs residing in the Null space of $\boldsymbol{A}$ would be lost, hindering the learning algorithm from exploring the entire space of parameters while trying to minimize the objective function. As such, it is crucial to change $\boldsymbol{A}$ every few iterations of training to allow the SGs to navigate different directions in the parameter space.

One possible approach is generating elements of $\boldsymbol{A}$, a_{ij}, iid according to a zero-mean distribution such as Gaussian, Rademacher or $a_{ij} \in \{-1, 0, +1\}$. However, in the proposed *Random Linear Coding*, we restrict $\boldsymbol{A}$ to be of the form $\boldsymbol{A} = \frac{1}{\sqrt{m}}\boldsymbol{H}\boldsymbol{R}$ where $\boldsymbol{H} \in \{\pm 1\}^{m\times d}$ is a partial Hadamard matrix, $\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}} = d\boldsymbol{I}$, and $\boldsymbol{R}$ is a random diagonal Rademacher matrix, i.e., $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{r})$, $\mathrm{P}(r_i = 1) = \mathrm{P}(r_i = -1) = 0.5$. Hence, the encoding at the i-th node is given as

$$\mathcal{C}_i(\alpha_i\boldsymbol{g}_i) = \alpha_i\boldsymbol{A}\boldsymbol{g}_i = \frac{\alpha_i}{\sqrt{m}}\boldsymbol{H}(\boldsymbol{r}\odot\boldsymbol{g}_i), \tag{3}$$

where fast Walsh-Hadamard algorithms can be used to perform multiplication by $\boldsymbol{H}$. Note that the edge nodes must use a common seed and follow the same random number generation protocol to generate a common random matrix for encoding.

IV. Proposed Method: Random Linear Coding

To develop the proposed RLC, first assume that all edge nodes transmit their SG. Hence, the received signal over wireless-MAC at ER would be $\boldsymbol{y} = \sum_i \boldsymbol{A}(\alpha_i\boldsymbol{g}_i) + \boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{g} + \boldsymbol{\eta}$. The node i estimates SG from received $\boldsymbol{y}$ (or its noisy version $\boldsymbol{y} + \boldsymbol{\eta}_i$) from ER via

$$\widehat{\boldsymbol{g}} = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}. \tag{4}$$

Lemma 2. *$\widehat{\boldsymbol{g}}$ is an unbiased SG estimator with mean squared error*

$$\mathbb{E}\big[\|\boldsymbol{g} - \widehat{\boldsymbol{g}}\|_2^2\big] = (\frac{d}{m} - 1)\|\boldsymbol{g}\|_2^2 + d\sigma^2, \tag{5}$$

where σ^2 is the variance of the communication noise.[2]

[2]Note that throughout the paper, the expectation is generally taken w.r.t. randomness in the coding, i.e., random matrix $\boldsymbol{A}$.

We have thus far incorporated **P1** and privacy **P4** into the proposed RLC framework. Now, we take into account the constraints **P2** and **P3**, while ensuring that the estimated values at the edge nodes be an unbiased SG of $F(\cdot)$. Specifically, the developed RLC and the estimation algorithm ((4) or its variants) should be insensitive to the *local decisions* made at each individual node, as will be explained later.

A. Power Constraint

One major challenge in federated learning in wireless edge networks is the limited transmission power. Note that the average transmission power at node i can be computed as

$$\mathbb{E}\big[\|\boldsymbol{x}_i\|_2^2\big] = \mathbb{E}\big[\|\boldsymbol{h}_i^{-1}\odot(\alpha_i\boldsymbol{A}\boldsymbol{g}_i)\|_2^2\big] = \alpha_i^2\|\boldsymbol{g}_i\|_2^2\ \frac{\|\boldsymbol{h}_i^{-1}\|_2^2}{m}.$$

To control the transmission power, $\boldsymbol{x}_i$'s of all nodes can be scaled appropriately by the same value such that the transmission power constraint of all nodes are satisfied. Moreover, since the contribution of sub-channels with huge losses (small entries in $\boldsymbol{h}_i$) is remarkable in the transmission power, those sub-channels might be ignored to preserve energy at the expense of lower transmission rate. Note that the channel selection of each node in the network is performed locally and might not be known by others. Hence, it is desirable to have SG estimation at the edge nodes be independent of those local decisions. Let

$$\big[\boldsymbol{q}_i\big]_l = \begin{cases} [\boldsymbol{h}_i^{-1}]_l & \text{if sub-channel } l \text{ is being used,} \\ 0 & \text{o.w.} \end{cases} \tag{6}$$

To have an unbiased SG estimation given by (4) or its variants, we suggest scaling the transmitted signal inversely proportional to the number of channels as

$$\boldsymbol{x}_i = c\alpha_i\ \frac{m}{m_i}\big(\boldsymbol{q}_i\odot(\boldsymbol{A}\boldsymbol{g}_i)\big), \tag{7}$$

where m_i is the number of sub-channels being selected for data transmission by node i (i.e., $m_i = \|\boldsymbol{q}_i\|_0$ the number of non-zero entries of $\boldsymbol{q}_i$), and c is a *global* parameter shared by all nodes to control all nodes' transmission powers and may vary at different transmission rounds. It can be easily verified that the average transmitted power at node i is

$$\mathbb{E}\big[\|\boldsymbol{x}_i\|_2^2\big] = mc^2\alpha_i^2\ \|\boldsymbol{g}_i\|_2^2\ \frac{\|\boldsymbol{q}_i\|_2^2}{\|\boldsymbol{q}_i\|_0^2}. \tag{8}$$

Lemma 3. *Let the transmitted signals by each edge node be given as* (7)*. The reconstruction given via $\widehat{\boldsymbol{g}} = \frac{1}{c}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}$ provides an unbiased SG estimator. Moreover, the variance of error is bounded as*

$$\mathbb{E}\big[\|\boldsymbol{g} - \widehat{\boldsymbol{g}}\|_2^2\big] \le (\sum_i \frac{d}{m_i}\alpha_i\|\boldsymbol{g}_i\|)(\sum_i \alpha_i\|\boldsymbol{g}_i\|) - \|\boldsymbol{g}\|_2^2 + \frac{d}{c^2}\sigma^2. \tag{9}$$

In summary, the proposed RLC framework controls the transmitted power by appropriately adjusting c and choosing "good" sub-channels. Specifically, for a given c, to satisfy the power constraint **P2** while minimizing the MSE (9), it suffices to select the most number of elements from $\boldsymbol{h}_i$ with the largest magnitude such that $\mathbb{E}\big[\|\boldsymbol{x}_i\|_2^2\big]$ given via (8) is at most P_i. Similarly, for given m_i's (and hence $\boldsymbol{q}_i$'s), maximizing global c under the given power constraints, $\mathbb{E}\big[\|\boldsymbol{x}_i\|^2\big] \le P_i$ for all i, results in minimum MSE (9).

B. Transmission by a Subset of Nodes

In the wireless network, due to nodes being idle and unreliability in transmission, some nodes may not transmit their data. Let $b_i \in \{0,1\}$ be a random variable denoting whether node i is transmitting its data at the current iteration of training or not. We assume an iid probabilistic transmission, i.e., node i transfers its data with probability π_i at each round of training, independent of other nodes, hence $b_i \sim \text{Bernoulli}(\pi_i)$. To compensate for this random behavior and still be able to recover an unbiased SG estimate, we propose to scale the transmitted signals by $1/\pi_i$, i.e.,

$$\boldsymbol{x}_i = c\alpha_i \, \frac{b_i}{\pi_i} \frac{m}{m_i} \big(\boldsymbol{q}_i \odot (\boldsymbol{A}\boldsymbol{g}_i)\big), \tag{10}$$

where $b_i = 0$ corresponds to node i not transmitting any data. Intuitively, if a node does not transmit for $\tau - 1$ round of training, at the τ-th round, its effect on the computed SG should be scaled proportionate to τ to compensate for the missing contribution in the previous rounds of training. Similar to Lemma 3, it can be shown that the reconstruction given via $\widehat{\boldsymbol{g}} = \frac{1}{c}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}$ provides an unbiased and bounded-variance SG estimator. However, the average transmission power would be scaled by $1/\pi_i$.

Remark 1. As shown in [17], using local weighted error feedback at individual nodes can improve the convergence rate at the expense of larger memory usage at edge nodes, even for biased SG compression. Hence, by relaxing the unbiasedness constraint on RLC, for example, we can easily control the transmission power by

$$\boldsymbol{x}_i = s_i \left(\boldsymbol{q}_i \odot (\boldsymbol{A}\boldsymbol{g}_i)\right), \tag{11}$$

where s_i is an appropriately chosen constant, optimized locally at node i. However, to ensure convergence, the remaining portion of $\boldsymbol{g}_i$, given as $\boldsymbol{e}_i = \boldsymbol{g}_i - \frac{1}{c}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{x}_i$ should be stored for transmission at later rounds of training. The details of this approach are omitted here due to the lack of space and left as future work. ■

V. Experiments and Discussions

To evaluate the performance of the proposed RLC framework, we considered training various deep models over networks of 32 and 50 nodes at different channel signal to noise ratios. Further, we assume that all nodes have the same power constraint P, and they may transmit their data with probability $\pi_i = 0.5$. For comparison, we also implemented the digital communication scheme which first compresses and encodes the stochastic gradients and then transmits the compressed values of each node one at a time. For digital data compression, we used quantized compressive sampling (QCS) [17] which provides state-of-the-art performance in terms of compression gain and convergence rate. To have the same number of channel uses (hence, the same latency per training iteration) for a network of K nodes, if the compression gain of RLC is set to be γ, the digital communication scheme have to achieve a compression gain of K_e times larger, $K_e\gamma$, where K_e is the number of non-idle transmitting nodes. Further, we optimize the parameters of QCS to achieve the minimum MSE while having the desired compression gain. We also consider baseline transmission (no SG compression and assuming infinite channel bandwidth). Due to the large number of nodes in the network and unbalanced distributed dataset over nodes, analog compression based on sparsity such as [30] causes large amount of distortion in the reconstructed SG, hindering the convergence of the learning algorithm.

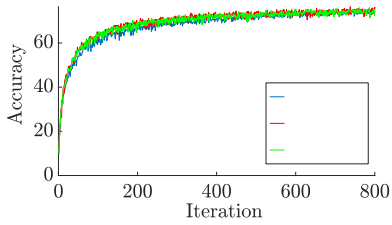
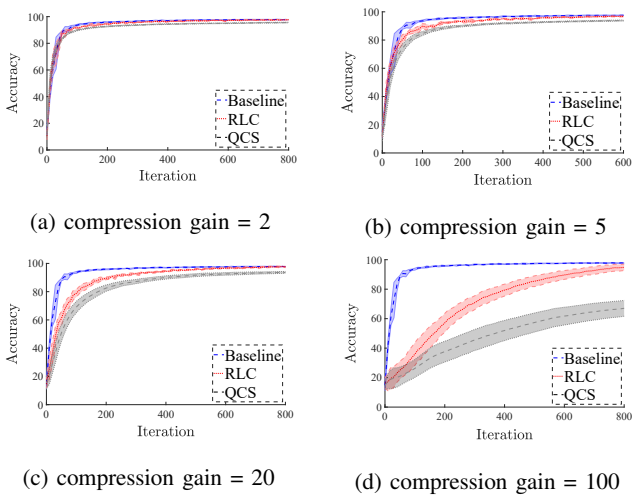


Fig. 3: Convergence rate vs training iteration for Cifarnet. QCS has approximately 350 times more channel uses than RLC.

First, we consider a network of 50 edge nodes, communicating to the ER with channel signal to noise ratio $\texttt{SNR} = 18$dB. Hence, $P/\sigma^2 \approx 63$ and the capacity of end-to-end channel is $C = 3$ bits per symbol. We then consider training Cifarnet, a deep convolutional model with approximately one million parameters, over Cifar10 dataset using stochastic gradient descent (SGD) algorithm. Traditional communication of SGs using QCS with a compression gain of 30 requires total transmission of approximately $53e6$ symbols, which results in $17.8e6$ channel uses. On the other hand, the proposed RLC framework with compression gain of 20 achieves the same performance with only $50e3$ channels uses, reducing the communication latency by a factor of at least 350. Moreover, as shown in Fig. 3, the convergence rate of the proposed algorithm follows that of the QCS and baseline (no SG compression) closely, *in terms of accuracy vs. number of iterations*. But since the communication latency of RLC is much lower, the *training time* using RLC is orders of magnitude smaller than digital communication.

Next, we consider training a Lenet-5 like convolutional network [33] over MNIST dataset using SGD with step-size $\mu = 0.05$. We consider different compression gains $\gamma = 2, 5, 20$ and 100 over a network of 32 nodes (with unbalanced datasets). The experiments are ran several times with different initial points to derive the mean and variance of the performance during federated learning, and are compared against QCS with the same communication requirements and Baseline (no compression and infinite communication bandwidth). As shown in 4, for low compression gains, the performance of training with compressed SGs are close to the baseline, although RLC slightly performs better than digital communication with QCS. However, for large compression gains, RLC outperforms QCS significantly. we have observed similar results with different SGD step-sizes, different channel SNRs and different neural networks.



(a) compression gain = 2

(b) compression gain = 5

(c) compression gain = 20

(d) compression gain = 100

Fig. 4: Convergence rate vs training iteration for Lenet over a network of 32 nodes. Baseline (blue) represents the ideal case of no SG compression and infinite communication resources.

Comparing results of analog compression via RLC for federated learning over wireless-MAC with those of digital communication methods confirms that designing a compression method that utilizes the characteristics and constraints of the wireless-MAC, **P1**–**P4**, can significantly improve the convergence rate and reduce the training latency.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint, arXiv:1602.05629v1*, 2016.

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

[3] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, jan 2019.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[6] S. U. Stich, "Local SGD Converges Fast and Communicates Little," in *ICLR*, 2019, pp. 1–12.

[7] H. Yu, S. Yang, and S. Zhu, "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5693–5700.

[8] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *arXiv preprint arXiv:1903.02891*, 2019.

[9] F. Seide, H. Fu, J. Droppo, G. Li, D. Yu, M. Stevenson, R. Winter, and B. Widrow, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Interspeech*, 2014, pp. 1058–1062.

[10] A. Øland and B. Raj, "Reducing communication overhead in distributed learning by an order of magnitude (almost)," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, apr 2015, pp. 2219–2223.

[11] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1707–1718.

[12] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed optimisation for non-convex problems," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research. PMLR, 2018, pp. 560–569.

[13] N. Dryden, S. A. Jacobs, T. Moon, and B. Van Essen, "Communication quantization for data-parallel training of deep neural networks," in *Proceedings of the Workshop on Machine Learning in High Performance Computing Environments*, ser. MLHPC '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 1–8.

[14] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast convergence rates of distributed subgradient methods with adaptive quantization," *arXiv preprint arXiv:1810.13245*, 2018.

[15] A. Abdi and F. Fekri, "Nested dithered quantization for communication reduction in distributed training," *arXiv preprint arXiv:1904.01197*, 2019.

[16] ——, "Reducing communication overhead via ceo in distributed training," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[17] ——, "Quantized compressive sampling of stochastic gradients for efficient communication in distributed deep learning," in *AAAI conference on Artificial Intelligence*, 2020.

[18] ——, "Indirect stochastic gradient quantization and its application in distributed deep learning," in *AAAI conference on Artificial Intelligence*, 2020.

[19] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing." in *INTERSPEECH*, vol. 7, 2015, p. 10.

[20] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[21] C. Renggli, D. Alistarh, T. Hoefler, and M. Aghagolzadeh, "SparCML: High-performance sparse communication for machine learning," *arXiv preprint arXiv:1802.08021*, 2018.

[22] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Advances in Neural Information Processing Systems*, no. NeurIPS, 2018, pp. 4452–4463.

[23] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems*, 2018, pp. 1306–1316.

[24] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Advances in Neural Information Processing Systems*, 2018, pp. 5977–5987.

[25] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *arXiv preprint arXiv:1812.11494 v3*, 2018.

[26] T. Sery and K. Cohen, "On analog gradient descent learning over multiple access fading channels," *arXiv preprint arXiv:1908.07463*, 2019.

[27] W. Liu and X. Zang, "Over-the-air computation systems: Optimization, analysis and scaling laws," *arXiv preprint arXiv:1909.00329*, 2019.

[28] N. H. Tran, W. Bao, A. Zomaya, N. M. N.H., and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. IEEE, apr 2019.

[29] M. M. Amiri and D. Gunduz, "Over-the-air machine learning at the wireless edge," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, jul 2019.

[30] ——, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *arXiv preprint arXiv:1901.00844*, 2019.

[31] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error Feedback Fixes SignSGD and other Gradient Compression Schemes," *arXiv preprint arXiv:1901.09847*, 2019.

[32] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error Compensated Quantized SGD and its Applications to Large-scale Distributed Optimization," *ICML*, 2018.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.