Q-GADMM: QUANTIZED GROUP ADMM FOR COMMUNICATION EFFICIENT DECENTRALIZED MACHINE LEARNING

Anis Elgabli, Jihong Park, Amrit S. Bedi†, Mehdi Bennis, and Vaneet Aggarwal‡

ABSTRACT

In this paper, we propose a communication-efficient decentralized machine learning (ML) algorithm, coined *quantized group ADMM (Q-GADMM)*. Every worker in Q-GADMM communicates only with two neighbors, and updates its model via the group alternating direct method of multiplier (GADMM), thereby ensuring fast convergence while reducing the number of communication rounds. Furthermore, each worker quantizes its model updates before transmissions, thereby decreasing the communication payload sizes. We prove that Q-GADMM converges to the optimal solution for convex loss functions, and numerically show that Q-GADMM yields 7x less communication cost while achieving almost the same accuracy and convergence speed compared to GADMM without quantization.

Index Terms— Communication-efficient decentralized machine learning, GADMM, ADMM, quantization.

1. INTRODUCTION

Recently, distributed machine learning (ML) at the network edge has received significant attention [1–3]. In contrast to classical cloud-based ML, distributed ML hinges on wireless communication and network dynamics, whereby communication may hinder its performance. To mitigate this bottleneck, one can decrease the communication cost of distributed ML by reducing the number of communication *rounds* until convergence, communication *links* per round, and/or the *payload size* per link.

Specifically, to reduce the communication payload sizes, arithmetic precision of the model parameters can be decreased by for instance 1-bit gradient quantization [4], multi-bit gradient quantization [5], or weight quantization with random rotation [6]. Alternatively, instead of model parameters, model outputs can be exchanged for large models via knowledge distillation [7, 8]. To reduce communication links, model updates can be sparsified by collecting the updates until a time deadline [9], upon the values sufficiently changed from the preceding updates [5, 10], or based on channel conditions [11–13]. To reduce the number of communication rounds, convergence speed can be accelerated via collaboratively adjusting the training momentum [14, 15]. However, these methods commonly postulate a central server that collects model update information, which may not be scalable.

A. Elgabli, J. Park, and M. Bennis are with the Center of Wireless Communication, University of Oulu, Finland (email: {anis.elgabli, jihong.park, mehdi.bennis}@oulu.fi). †A. Bedi is with the Department of Electrical Engineering, IIT Kanpur (email: amritbd@iitk.ac.in). ‡V. Aggarwal is with the School of Industrial Engineering and the School of Electrical and Computer Engineering, Purdue University, USA (email: vaneet@purdue.edu).

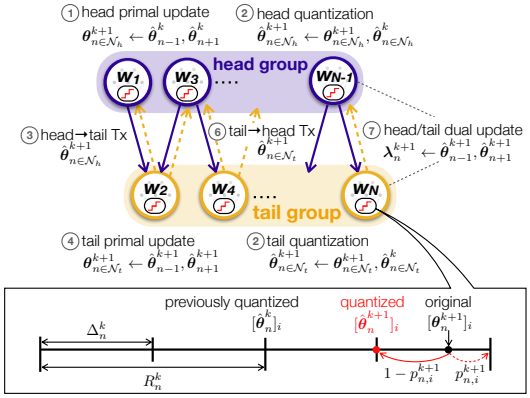

Fig. 1: Operational structure of quantized GADMM (Q-GADMM), in which every worker communicates with two neighbors, after quantizing its local model $\boldsymbol{\theta}_n^k$ with the radius R_n^k (for 4 quantization levels).

In decentralized ML architectures without any central entity, communication payload sizes can be reduced by a quantized weight gossiping algorithm [16]. On the other hand, both communication links and rounds can be decreased using the group ADMM (GADMM) algorithm proposed in our prior work [17]. Spurred by these preceeding works, in this article we further integrate quantization into GADMM, and propose a communication-efficient decentralized ML algorithm, coined *quantized group ADMM (Q-GADMM)*, thereby reducing communication rounds, links, and payload sizes altogether.

Quantized Group GADMM (Q-GADMM). As shown in Fig. 1, Q-GADMM divides workers into head and tail groups as done in GADMM. For less communication rounds via faster convergence, the workers in the same group update their models in parallel, whereas the workers in different groups update the models in an alternating way. Every alternation entails a single communication round, in which each worker communicates only with two neighboring workers in the opposite group, reducing communication links. Lastly, before every transmission, the workers quantize their model parameters, decreasing communication payload sizes.

Contributions. It is non-trivial to prove the convergence of Q-GADMM, in which quantization errors may lead to high unintended variance in model parameter updates. Its neighbor-based communications aggravate this problem, since the errors may easily propagate across iterations. To mitigate these problems, we propose a stochastic quantization scheme that ensures unbiased error and non-increasing quantization step sizes, by adjusting the quantization levels and probabilities over iterations. We thereby prove that with exchanging the maximum range of quantization and the number of quantization levels, Q-GADMM achieves the convergence and the optimality of GADMM for convex functions. Numerical results show that Q-GADMM converges as fast as GADMM with 7x less communication cost.

2. PROBLEM FORMULATION AND PROPOSED ALGORITHM

We consider a set of N workers storing their local batch of input samples. The n-th worker has its model vector $\boldsymbol{\theta}_n \in \mathbf{R}^d$, and aims to solve the following decentralized learning problem:

$$\underset{\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\cdots,\boldsymbol{\theta}_N\}}{\text{Minimize}} \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n)$$
$$\text{subject to } \boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1}, \forall n = 1, \cdots, N-1. \quad (1)$$

GADMM was proposed to solve (1) where workers are split into two groups, *heads* and *tails*, such that each worker in the head (or tail) group is communicating with two tail (or head) workers. As Fig. 1 illustrates, the primal variables of head workers are updated in parallel, and then downloaded by their neighboring tail workers. Likewise, the primal variables of tail workers are updated in parallel, and downloaded by their neighboring head workers. Lastly, the dual variables are updated locally at each worker. To improve the communication efficiency further, we solve (1) using Q-GADMM.

We now describe the overall operational procedure of Q-GADMM . Following [5], worker n in Q-GADMM at iteration k quantizes its model vector $\boldsymbol{\theta}_n^k$ as $\hat{\boldsymbol{\theta}}_n^k = Q_n(\boldsymbol{\theta}_n^k, \hat{\boldsymbol{\theta}}_n^{k-1})$, based on its previously quantized model vector $\hat{\boldsymbol{\theta}}_n^{k-1}$. The function $Q_n(\cdot)$ is a stochastic quantization operator that depends on the quantization probability $p_{n,i}^k$ for each model vector's dimension $i \in \{1, 2, \cdots, d\}$, and on b_n^k bits used for representing each model vector dimension. To ensure the convergence of Q-GADMM, $p_{n,i}^k$ and b_n^k should be properly chosen as detailed next.

Stochastic Quantization. As Fig. 1 shows, the i-th dimensional element $[\hat{\boldsymbol{\theta}}_n^{k-1}]_i$ of the previously quantized model vector is centered at the quantization range $2R_n^k$ that are equally divided into $2^{b_n^k} - 1$ quantization levels, yielding the quantization step size $\Delta_n^k = 2R_n^k/(2^{b_n^k} - 1)$. In this coordinate, the difference between the i-th dimensional element $[\boldsymbol{\theta}_n^k]_i$ of the current model vector and $[\hat{\boldsymbol{\theta}}_n^{k-1}]_i$ is drawn at

$$[c_n(\boldsymbol{\theta}_n^k)]_i = \frac{1}{\Delta_n^k}\left([\boldsymbol{\theta}_n^k]_i - [\hat{\boldsymbol{\theta}}_n^{k-1}]_i + R_n^k\right). \quad (2)$$

Here, adding R_n^k ensures the non-negativity of the quantized value. Then, $[c_n(\boldsymbol{\theta}_n^k)]_i$ is mapped to:

$$[q_n(\boldsymbol{\theta}_n^k)]_i = \begin{cases} \lceil [c_n(\boldsymbol{\theta}_n^k)]_i \rceil & \text{with probability } p_{n,i}^k \\ \lfloor [c_n(\boldsymbol{\theta}_n^k)]_i \rfloor & \text{otherwise,} \end{cases} \quad (3)$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are ceiling and floor functions, respectively.

Next, to maintain the quantization error unbiased after quantization, $p_{n,i}^k$ is chosen such that the expected quantization error $\mathbb{E}\left[\boldsymbol{\epsilon}_{n,i}^k\right]$ becomes zero, *i.e.,*

$$p_{n,i}^k\Big([c_n(\boldsymbol{\theta}_n^k)]_i - \lfloor [c_n(\boldsymbol{\theta}_n^k)]_i \rfloor\Big) + (1-p_{n,i}^k)\Big(\lceil [c_n(\boldsymbol{\theta}_n^k)]_i \rceil - [c_n(\boldsymbol{\theta}_n^k)]_i\Big) = 0, \quad (4)$$

Consequently, $p_{n,i}^k$ is obtained as:

$$p_{n,i}^k = \Big(\lceil [c_n(\boldsymbol{\theta}_n^k)]_i \rceil - [c_n(\boldsymbol{\theta}_n^k)]_i\Big)/\Delta_n^k, \quad (5)$$

where the denominator follows from $\Delta_n^k = \lceil [c_n(\boldsymbol{\theta}_n^k)]_i \rceil - \lfloor [c_n(\boldsymbol{\theta}_n^k)]_i \rfloor$. With $p_{n,i}^k$in (5), quantization error variance is bounded, *i.e.,* $\mathbb{E}\left[\left(\boldsymbol{\epsilon}_{n,i}^k\right)^2\right] \leq (\Delta_n^k)^2/4$.

In addition to the above, the convergence of Q-GADMM requires non-increasing quantization step sizes over iterations, *i.e.,* $\Delta_n^k \leq \Delta_n^{k-1}\ \forall k$. To satisfy this condition, b_n^k is chosen as:

$$b_n^k \geq \left\lceil \log_2\left(1 + (2^{b_n^{k-1}} - 1)R_n^k/R_n^{k-1}\right)\right\rceil. \quad (6)$$

Given $p_{n,i}^k$ in (5) and b_n^k in (6), the convergence of Q-GADMM is to be proven in Sec. 4. Note that in our numerical simulations in Sec. 5, we observe that R_n^k decreases over iterations, and thus $\Delta_n^k \leq \Delta_n^{k-1}$ holds even for b_n^k fixed as a constant.

With the aforementioned stochastic quantization procedure, b_n^k, R_n^k, and $q_n(\boldsymbol{\theta}_n^k) = ([q_n(\boldsymbol{\theta}_n^k)]_1, [q_n(\boldsymbol{\theta}_n^k)]_2, \cdots, [q_n(\boldsymbol{\theta}_n^k)]_d)^\intercal$ suffice to represent $\hat{\boldsymbol{\theta}}_n^k$, which are transmitted to neighbors. After receiving these values, $\hat{\boldsymbol{\theta}}_n^k$ can be reconstructed as follows:

$$\hat{\boldsymbol{\theta}}_n^k = \hat{\boldsymbol{\theta}}_n^{k-1} + \Delta_n^k q_n(\boldsymbol{\theta}_n^k) - R_n^k \mathbf{1}. \quad (7)$$

Consequently, when the full arithmetic precision uses 32bit, every transmission payload size of Q-GADMM is $b_n^k d + (b_R + b_b)$ bits, where $b_R \leq 32$ and $b_b \leq 32$ are the required bits to represent R_n^k and b_n^k, respectively. Compared to GADMM whose payload size is $32d$ bits, Q-GADMM can achieve a huge reduction in communication overhead by setting $b_n^k \ll 32$, particularly for large models, *i.e.,* large d.

Q-GADMM Operations. For the sake of explanation, we consider an even N number of workers. We start by writing the augmented Lagrangian as:

$$\mathcal{L}_\rho = \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n) + \sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_n, \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}\rangle + \frac{\rho}{2}\sum_{n=1}^{N-1}\|\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}\|_2^2. \quad (11)$$

Let $\mathcal{N}_h = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \cdots, \boldsymbol{\theta}_{N-1}\}$, and $\mathcal{N}_t = \{\boldsymbol{\theta}_2, \boldsymbol{\theta}_4, \cdots, \boldsymbol{\theta}_N\}$ denotes the sets of head and tail workers, respectively. At iteration $k+1$, the workers' primal and dual variables are updated as follows. First, head worker's primal variables are updated as:

$$\boldsymbol{\theta}_n^{k+1} = \underset{\theta_n}{\text{argmin}}\big\{ f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \hat{\boldsymbol{\theta}}_{n-1}^k - \boldsymbol{\theta}_n\rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^k\rangle$$
$$+\frac{\rho}{2}\|\hat{\boldsymbol{\theta}}_{n-1}^k - \boldsymbol{\theta}_n\|_2^2 + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^k\|_2^2\big\}, n \in \mathcal{N}_h \setminus \{1\}. \quad (12)$$

Algorithm 1 Quantized Group ADMM (Q-GADMM)

Input: $N, f_n(\boldsymbol{\theta}_n) \forall n, \rho, K$, **Output**: $\boldsymbol{\theta}_n, \forall n$
Initialization: $\boldsymbol{\theta}_n^{(0)} = 0, \boldsymbol{\lambda}_n^{(0)} = 0, \forall n$
$\mathcal{N}_h = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \cdots, \boldsymbol{\theta}_{n-1}\}, \mathcal{N}_t = \{\boldsymbol{\theta}_2, \boldsymbol{\theta}_4, \cdots, \boldsymbol{\theta}_N\}$
while $k \leq K$ **do**
  Head workers ($n \in \mathcal{N}_h$)**: in Parallel**
    Reconstruct $\hat{\boldsymbol{\theta}}_{n-1}$ and $\hat{\boldsymbol{\theta}}_{n+1}$ via (7), and update $\boldsymbol{\theta}_n$ via (12)
    Choose $p_{n,i}^k$ via (5) and b_n^k via (6)
    Quantize $\boldsymbol{\theta}_n$ via (3)
    Transmit $b_n^k, R_n^k, q_n(\boldsymbol{\theta}_n)$ to its two tail neighbors
  Tail workers ($n \in \mathcal{N}_t$)**: in Parallel**
    Reconstruct $\hat{\boldsymbol{\theta}}_{n-1}$ and $\hat{\boldsymbol{\theta}}_{n+1}$ via (7), and update $\boldsymbol{\theta}_n$ via (14)
    Choose $p_{n,i}^k$ via (5) and b_n^k via (6)
    Quantize $\boldsymbol{\theta}_n$ via (3)
    Transmit $b_n^k, R_n^k, q_n(\boldsymbol{\theta}_n)$ to its two head neighbors
  All workers ($n \in \{1, \cdots, N\}$)**: in Parallel**
    Update $\boldsymbol{\lambda}_{n-1}^k$ and $\boldsymbol{\lambda}_n^k$ locally via (16)
  $k \leftarrow k+1$
end while

Since $\boldsymbol{\theta}_{n-1}$ is not defined for $n = 1$ (the first head worked does not have a left neighbor), the update is done as follows

$$\boldsymbol{\theta}_n^{k+1} = \underset{\theta_n}{\operatorname{argmin}} \{ f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^k \rangle + \frac{\rho}{2} \| \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^k \|_2^2 \}. \tag{13}$$

Next, each head worker transmits its quantized model to its two tail neighbors. Then, the tail workers' primal variables are updated as:

$$\boldsymbol{\theta}_n^{k+1} = \underset{\boldsymbol{\theta}_n}{\operatorname{argmin}} \{ f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \boldsymbol{\theta}_n \rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^{k+1} \rangle + \frac{\rho}{2} \| \hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \boldsymbol{\theta}_n \|_2^2 + \frac{\rho}{2} \| \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n+1}^{k+1} \|_2^2 \}, n \in \mathcal{N}_t \setminus \{N\}. \tag{14}$$

The update of the last tail worker ($n = N$) is given by

$$\boldsymbol{\theta}_n^{k+1} = \underset{\boldsymbol{\theta}_n}{\operatorname{argmin}} \{ f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \boldsymbol{\theta}_n \rangle + \frac{\rho}{2} \| \hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \boldsymbol{\theta}_n \|_2^2 \tag{15}$$

Finally, every worker locally updates the dual variables $\boldsymbol{\lambda}_{n-1}$ and $\boldsymbol{\lambda}_n$ as follows:

$$\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\hat{\boldsymbol{\theta}}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^{k+1}), \ n = 1, \ldots, N-1. \tag{16}$$

3. CONVERGENCE ANALYSIS

In this section, we prove the optimality and convergence of Q-GADMM for convex functions. The necessary and sufficient optimality conditions are the primal and dual feasibility which are defined by

$$\boldsymbol{\theta}_n^* = \boldsymbol{\theta}_{n-1}^*, \forall n > 1, \quad \mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^*) - \boldsymbol{\lambda}_{n-1}^* + \boldsymbol{\lambda}_n^*, \forall n, \tag{17}$$

where $\boldsymbol{\lambda}_0^* = \boldsymbol{\lambda}_N^* = 0$.
First, at iteration $k+1$, every $\boldsymbol{\theta}_n^{k+1}$ such that $n \in \mathcal{N}_t \setminus \{N\}$ minimizes (14), which implies that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n-1}^{k+1}) + \rho(\boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^{k+1}). \tag{18}$$

Let $\boldsymbol{\epsilon}_n^{k+1}$ be the quantization error at iteration $k+1$. Hence, $\boldsymbol{\epsilon}_n^{k+1} = \boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_n^{k+1}$. Moreover, given that $\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\hat{\boldsymbol{\theta}}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^{k+1})$, (18) can be re-written as

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + 2\rho\boldsymbol{\epsilon}_n^{k+1}, n \in \mathcal{N}_t \setminus \{N\}. \tag{19}$$

Similarly, we can write for the last tail worker ($n = N$)

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + 2\rho\boldsymbol{\epsilon}_n^{k+1}, n = N \tag{20}$$

Second, every $\boldsymbol{\theta}_n^{k+1}$ such that $n \in \mathcal{N}_h \setminus \{1\}$ minimizes (12) at iteration $k+1$. Therefore

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n-1}^k) + \rho(\boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^k). \tag{21}$$

Using $\boldsymbol{\epsilon}_n^{k+1} = \boldsymbol{\theta}_n^{k+1} - \hat{\boldsymbol{\theta}}_n^{k+1}$ and $\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\hat{\boldsymbol{\theta}}_n^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^{k+1})$, (21) can be re-written as

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + 2\rho\boldsymbol{\epsilon}_n^{k+1} + \rho(\hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \hat{\boldsymbol{\theta}}_{n-1}^k) + \rho(\hat{\boldsymbol{\theta}}_{n+1}^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^k), n \in \mathcal{N}_h \setminus \{1\}. \tag{22}$$

Using similar steps, we can write for the first head worker ($n = 1$)

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) + \boldsymbol{\lambda}_n^{k+1} + 2\rho\boldsymbol{\epsilon}_n^{k+1} + \rho(\hat{\boldsymbol{\theta}}_{n+1}^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^k), n = 1. \tag{23}$$

Let $\boldsymbol{r}_{n-1,n}^{k+1} = \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n^{k+1}$ and $\boldsymbol{r}_{n,n+1}^{k+1} = \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1}$ be the primal residual of each agent n, and we define the dual residual of worker $n \in \mathcal{N}_h$ at iteration $k+1$ as

$$\boldsymbol{s}_n^{k+1} = \begin{cases} \rho(\hat{\boldsymbol{\theta}}_{n-1}^{k+1} - \hat{\boldsymbol{\theta}}_{n-1}^k) + \rho(\hat{\boldsymbol{\theta}}_{n+1}^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^k), & \text{if } n \in \mathcal{N}_h \setminus \{1\}, \\ \rho(\hat{\boldsymbol{\theta}}_{n+1}^{k+1} - \hat{\boldsymbol{\theta}}_{n+1}^k), & \text{if } n = 1. \end{cases} \tag{24}$$

To prove the convergence of the proposed algorithm, we first provide the upper and lower bounds on the optimality gap in Lemma 1.

Lemma 1 At $k+1$ iteration of Q-GADMM, the optimality gap satisfies $\mathsf{LB}_1 \leq \mathbb{E}\left[\left\{\sum_{n=1}^N f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^N f_n(\boldsymbol{\theta}_n^*)\right\}\right] \leq \mathsf{UB}_1$, where LB_1 and UB_1 are given in (20), at the bottom of this page.

$$\mathsf{LB}_1 = \mathbb{E}\left[\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^*, \boldsymbol{r}_{n,n+1}^{k+1} \rangle\right], \mathsf{UB}_1 = \mathbb{E}\left[-\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{r}_{n,n+1}^{k+1} \rangle + \sum_{n=1}^{N-1} 2\rho \langle \boldsymbol{\epsilon}_n^{k+1}, \boldsymbol{\theta}_n^* - \boldsymbol{\theta}_n^{k+1} \rangle + \sum_{n \in \mathcal{N}_h} \langle \boldsymbol{s}_n^{k+1}, \boldsymbol{\theta}_n^* - \boldsymbol{\theta}_n^{k+1} \rangle\right] \tag{20}$$

$$H_v = \rho \sum_{n \in \mathcal{N}_h \setminus \{1\}} \mathbb{E}\left[\| (\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^k) - (\boldsymbol{\epsilon}_{n-1}^{k+1} - \boldsymbol{\epsilon}_{n-1}^k) - \boldsymbol{r}_{n-1,n}^{k+1} \|_2^2\right] + \rho \sum_{n \in \mathcal{N}_h} \mathbb{E}\left[\| (\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k) - (\boldsymbol{\epsilon}_{n+1}^{k+1} - \boldsymbol{\epsilon}_{n+1}^k) + \boldsymbol{r}_{n,n+1}^{k+1} \|_2^2\right] + 2\rho\left(\mathbb{E}\left[\| \boldsymbol{\epsilon}_1^{k+1} \|_2^2\right] + \mathbb{E}\left[\| \boldsymbol{\epsilon}_N^{k+1} \|_2^2\right]\right) + 2\rho \sum_{n \in \mathcal{N}_t} \mathbb{E}\left[\| \boldsymbol{\epsilon}_n^k \|_2^2\right] \tag{21}$$

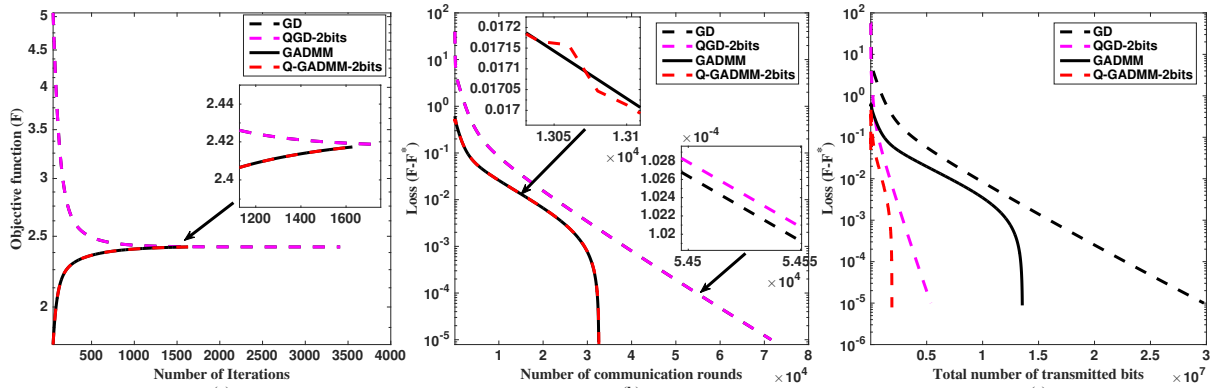

Fig. 2: Performance comparison of Q-GADMM-2bits, with GADMM, GD, and Q-GD-2bits For linear regression task using real dataset, (a) Convergence in the objective value, (b) Loss $(F - F^*)$ vs number of communication rounds, (c) Loss $(F - F^*)$ vs number of transmitted bits.

Before we introduce the next lemma, we define a Lyapunov function $V^k = \frac{1}{\rho}\sum_{n=1}^{N-1} \|\boldsymbol{\lambda}_n^k - \boldsymbol{\lambda}^*\|_2^2 + \rho \sum_{n\in\mathcal{N}_h\setminus\{1\}} \|\boldsymbol{\theta}_{n-1}^k - \boldsymbol{\theta}^*\| + \rho \sum_{n\in\mathcal{N}_h} \|\boldsymbol{\theta}_{n+1}^k - \boldsymbol{\theta}^*\|$. Next lemma shows that $\mathbb{E}\left[V^k - V^{k+1}\right]$ is bounded above. This property is then used in Theorem 1 to prove that V^k is monotonically decreasing at each iteration k and the primal residual goes to zero as $k \to \infty$ when the quantization step size is reduced as k increases.

Lemma 2 When $f_n(\boldsymbol{\theta}_n)$ is closed, proper, and convex, and the Lagrangian $\boldsymbol{\mathcal{L}}_0$ has a saddle point, then the following inequality holds true at the $(k+1)$-th iteration of Q-GADMM:

$$\mathbb{E}\left[V^k - V^{k+1}\right] \geq \rho \sum_{n} \mathbb{E}\left[\| \boldsymbol{\epsilon}_n^{k+1} + \boldsymbol{\epsilon}_{n+1}^{k+1} \|_2^2\right] - \frac{\rho d}{2} \sum_{n\in\mathcal{N}_t} (\Delta_n^{k+1})^2 + H_v, \tag{23}$$

where H_v is given in (21) at the bottom of the previous page. Using Lemmas 1 and 2, we derive our main theorem stating the convergence and the optimality of Q-GADMM in solving (1).

Theorem 1 For non-increasing quantization step sizes, *i.e.,* $\Delta_n^k \leq \Delta_n^{k-1}$ $\forall k$, and under the assumption of Lemma 2, as $k \to \infty$, the primal and dual residual converges to 0 with probability 1, *i.e.,* $\lim_{k\to\infty} \boldsymbol{r}_{n,n+1}^k \overset{a.s.}{=} \mathbf{0}$ and $\lim_{k\to\infty} \boldsymbol{s}_n^k \overset{a.s.}{=} \mathbf{0}$. Furthermore, the optimality gap converges to 0 with probability 1, *i.e.,* $\lim_{k\to\infty} \sum_{n=1}^N f_n(\boldsymbol{\theta}_n^k) = \sum_{n=1}^N f_n(\boldsymbol{\theta}^*)$.

Intuitively, when Δ_n^k is non-increasing, the RHS of (23) is always positive. For such positive $\mathbb{E}\left[V^k - V^{k+1}\right]$, iteration $k+1$ is one more step towards the optimal solution. Therefore, following the same proof of theorem 1 for GADMM [17], as $k \to \infty$, Q-GADMM converges to the optimal solution.

4. NUMERICAL RESULTS

We evaluated the performance of Q-GADMM for decentralized linear regression using California Housing dataset [19]. We used 4000 samples for training, and we uniformly distributed them across 10 workers. To benchmark Q-GADMM, we compare it with GADMM [17]($\rho = 1$), GD, and quantized GD (QGD) [5]. In GD, each worker computes its gradient and then sends it to a parameter server. The server updates the global model using aggregated gradient descent and broadcasts it to all workers. In QGD, each worker sends a quantized version of its local model. Finally, for quantized versions of GADMM and GD, we assume that $b_n^k = 2$ for all n and k, so the number of quantization levels is 4, and it remains constant over iterations and across workers.

Figure 2(a) verifies Theorem 1, and shows the convergence of Q-GADMM for convex loss function with the same speed as GADMM. Moreover, Figure 2(b), shows that both GADMM and Q-GADMM-2bits achieves the loss of 10^{-5} with almost the same number of communication rounds (32600 for Q-GADMM-2bits vs 32560 for GADMM) which is around 50% less compared to GD and QGD-2bits. However, as shown in Figure 2(c), Q-GADMM-2bits requires significantly less number of bits as compared to GADMM (7x less number of bits). Moreover, compared to GD and QGD-2bits, Q-GADMM-2bits requires around 15x and 3x less number of transmitted bits to achieve the loss of 10^{-5}. Thanks to the fast convergence inherited from GADMM, and stochastic quantization, the number of transmitted bits at every iteration are significantly reduced while ensuring unbiased and zero mean quantization error.

5. CONCLUSIONS

This article proposed a communication-efficient decentralized ML algorithm, Q-GADMM. Compared to the original GADMM, Q-GADMM enjoys the same convergence rate, but at significantly lower communication overhead. Numerical tests in a convex linear regression task corroborate the advantages of Q-GADMM over GADMM, GD, and QGD.

6. REFERENCES

[1] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *to appear in Proceedings of the IEEE [Online]. Early access is available at: https://ieeexplore.ieee.org/document/8865093*, November 2019.

[2] J. Park, S. Wang, A. Elgabli, S. Oh, E. Jeong, H. Cha, H. Kim, S.-L. Kim, and M. Bennis, "Distilling on-device intelligence at the network edge," *Arxiv preprint*, vol. abs/1908.05895, August 2019.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," [Online]. ArXiv preprint: https://arxiv.org/abs/1908.07873, August 2019.

[4] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed optimisation for non-convex problems," *In Proc. Intl. Conf. Machine Learn., Stockholm, Sweden*, July 2018.

[5] J. Sun, T. Chen, G. B. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," *arXiv preprint arXiv:1909.07588*, 2019.

[6] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," in *Proc. of NIPS Wksp. PMPML*, Barcelona, Spain, December 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[7] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *presented at Neural Information Processing Systems Workshop on Machine Learning on the Phone and other Consumer Devices (MLPCD), Montréal, Canada*, 2018. [Online]. Available: http://arxiv.org/abs/1811.11479

[8] J. H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," [Online]. Arxiv preprint: http://arxiv.org/abs/1907.02745, May 2019.

[9] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, June 2019.

[10] T. Chen, G. Giannakis, T. Sun, and W. Yin, "Lag: Lazily aggregated gradient for communication-efficient distributed learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5055–5065.

[11] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," *In Proc. Int'l Conf. Commun. (ICC), Shanghai, China*, May 2019. [Online]. Available: http://arxiv.org/abs/1804.08333

[12] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *arXiv preprint arXiv: 1908.06287*, 2019.

[13] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv preprint arXiv: 1909.07972*, 20019.

[14] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *arXiv preprint arXiv: 1910.03197*, 2019.

[15] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," *In Proc. Intl. Conf. Machine Learn., Long Beach, CA, USA*, June 2019.

[16] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," *Proc. International Conference on Machine Learning (ICML), Long Beach, CA, USA*, July 2019.

[17] A. Elgabli, J. Park, A. S. Bedi, M. Bennis, and V. Aggarwal, "Gadmm: Fast and communication efficient framework for distributed machine learning," *arXiv preprint arXiv:1909.00047*, 2019.

[18] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[19] L. Torgo, "Regression datasets," 2014. [Online]. Available: https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html