Differentially Quantized Gradient Descent

Chung-Yi Lin, Victoria Kostina, and Babak Hassibi

***Abstract*—Consider the following distributed optimization scenario. A worker has access to training data that it uses to compute the gradients while a server decides when to stop iterative computation based on its target accuracy or delay constraints. The only information that the server knows about the problem instance is what it receives from the worker via a rate-limited noiseless communication channel. We introduce the technique we call *differential quantization* (DQ) that compensates past quantization errors to make the descent trajectory of a quantized algorithm follow that of its unquantized counterpart. Assuming that the objective function is smooth and strongly convex, we prove that *differentially quantized gradient descent* (DQ-GD) attains a linear convergence rate of $\max\{\sigma_{\mathrm{GD}}, \rho_n 2^{-R}\}$, where σ_{GD} is the convergence rate of unquantized gradient descent (GD), ρ_n is the covering efficiency of the quantizer, and R is the bitrate per problem dimension n. Thus at any $R \geq \log_2 \rho_n/\sigma_{\mathrm{GD}}$, the convergence rate of DQ-GD is the same as that of unquantized GD, i.e., there is no loss due to quantization. We show a converse demonstrating that no GD-like quantized algorithm can converge faster than $\max\{\sigma_{\mathrm{GD}}, 2^{-R}\}$. Since quantizers exist with $\rho_n \to 1$ as $n \to \infty$ (Rogers, 1963), this means that DQ-GD is asymptotically optimal. In contrast, naively quantized GD where the worker directly quantizes the gradient attains only $\sigma_{\mathrm{GD}} + \rho_n 2^{-R}$. The technique of differential quantization continues to apply to gradient methods with momentum such as Nesterov's accelerated gradient descent, and Polyak's heavy ball method. For these algorithms as well, if the rate is above a certain threshold, there is no loss in convergence rate obtained by the differentially quantized algorithm compared to its unquantized counterpart. Experimental results on both simulated and real-world least-squares problems validate our theoretical analysis.**

I. Introduction

A. Motivation and related work

Distributed optimization plays a central role in large-scale machine learning where gradient descent (GD) and its stochastic variant SGD are employed to minimize an objective function [1]–[8]. Despite the scalability of parallel gradient training, the frequent exchange of high-dimensional gradients has become a communication overhead that slows down the overall learning process [2], [5], [9]–[12].

To reduce the communication cost, one line of research focuses on gradient quantization with a fixed number of bits per problem dimension. Scalar quantizers, which quantize each coordinate of the input vector separately, are often used to address the communication bottleneck in distributed SGD algorithms. For each coordinate, the quantization levels are distributed either uniformly [9], [13]–[16] or non-uniformly [17] within the dynamic-range interval. Convergence of such quantized gradient methods is typically established via relating the variance of the quantized (thus noisy) gradient to the bit rate, and the communication cost is further reduced using an efficient integer coding scheme such as the Elias encoding [13], [17]. Due to scalar quantization, the obtained convergence rate would pay an extra dimension-dependent factor. On the other hand, vector quantizers are hard to implement in practice and hence are less considered in this context. In this direction, [18], [19] construct vector quantizers from the convex hull of specifically structured point sets. However, it has been observed in practice [9] as well as in theory [21] that gradient methods with fixed bit rate quantizers do not converge for a low bit rate.

Another common way to reduce the communication bandwidth in parallel SGD training is to sparsify the gradient vectors. For example, the top-k sparsifier (or *compressor*) preserves the k coordinates of the largest magnitude and sends them with full precision [11], [20], [22]–[25]. The compressors used are either biased [11], [23], [25], [25]–[30] or unbiased [31]–[33], and the compression error is controlled by a user-specific accuracy parameter (e.g. k for the top-k compressor). Similar to the failure of 1-bit SGD (without mini-batching) due to the biased quantization error [9], [21], it is observed in [30] that compressed GD with the top-1 compressor does not always converge. For an empirical risk minimization problem where the global objective function is the average of local objective functions, recent works [34]–[36] perform analog gradient compression and communication by taking the physical superposition nature of the underlying multiple-access channel into the account.

Most distributed SGD algorithms with biased compressors in the literature apply the idea of error compensation that can be traced back to the Σ-Δ modulation [37]. To form the compressor input at each iteration, there are various ways to add past compression errors back to the computed gradients. While [38] weights all the past errors in a time-decaying fashion, the *error-feedback scheme* (EF) uses only the very last error and provides convergence for the single-worker setting [21], [25], [39] as well as the multi-worker setting [30], [40].

Although convergence rates of quantized gradient methods depend on the bit rate R [13], [16]–[18], few existing works provide convergence lower bounds in terms of R that apply to any algorithm within a specified class. For quantized projected SGD, [16], [18] give lower bounds to a minimax expected estimation error (i.e. difference between the output function value and the optimal one), which is in the same order of convergence as that of the unquantized SGD over convex functions. However, the allowable quantizer input in [16], [18] is fixed to be the gradient of the current iterate. Besides, standard assumptions such as unbiasedness and boundedness

The authors are with California Institute of Technology (chungyi@caltech.edu, vkostina@caltech.edu, hassibi@caltech.edu). This work was supported in part by the National Science Foundation (NSF) under grants CCF-1751356, CCF-1956386, CNS-0932428, CCF-1018927, CCF-1423663 and CCF-1409204, by a grant from Qualcomm Inc., by NASAs Jet Propulsion Laboratory through the President and Directors Fund, and by King Abdullah University of Science and Technology.

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518254

for the stochastic gradients [41]–[43] are crucial for SGD and in fact simplify the analysis of the quantized gradient algorithms. In this paper, we show that the performance of quantized (non-stochastic) gradient descent can be improved if this restriction is relaxed.

B. Contributions

We consider the single-worker scenario of the parameter server framework [10], [13]–[15], [17], [44], [45] consisting of a worker that computes the gradients and a server that successively refines the model parameter (i.e. the iterate) and decides when to stop the distributed iterative algorithm based on its target accuracy or delay constraints. See Fig. 1.

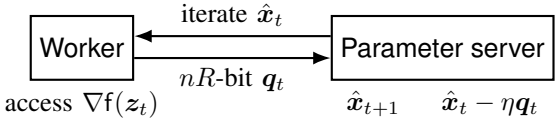

Fig. 1: Quantized gradient descent (QGD) in a single-worker remote training setting. At each iteration t, the server first sends the current iterate $\hat{\boldsymbol{x}}_t$ to the worker noiselessly, who computes the gradient at some point $\boldsymbol{z}_t$ that is a function of (but not necessarily equal to) $\hat{\boldsymbol{x}}_t$. Then, the worker forms a descent direction $\boldsymbol{q}_t$ and pushes it back to the server under the nR bits per iteration constraint.

We study the fundamental tradeoff between the convergence rate and the communication rate of quantized gradient descent. We focus on the class $\mathcal{F}_n$ of smooth and strongly convex objective functions $\mathsf{f}\colon \mathbb{R}^n \mapsto \mathbb{R}$ whose minimizers are bounded in the Euclidean norm. For a quantized iterative algorithm A, its worst-case linear *convergence rate* over $\mathcal{F}_n$ at rate R bits per problem dimension is defined as

$$\sigma_{\mathrm{A}}(n,R) \triangleq \inf_{R' \le R} \sup_{\mathsf{f}\in\mathcal{F}_n} \limsup_{T\to\infty} \|\hat{\boldsymbol{x}}_T(R') - \boldsymbol{x}^*_{\mathsf{f}}\|^{\frac{1}{T}} \tag{1}$$

where $\boldsymbol{x}^*_{\mathsf{f}}$ is the optimizer, and $\hat{\boldsymbol{x}}_0(R'), \hat{\boldsymbol{x}}_1(R'), \hat{\boldsymbol{x}}_2(R'), \ldots$ is the sequence of iterates generated by A in response to $\mathsf{f} \in \mathcal{F}_n$ when it operates at R' bits per problem dimension.

We consider three popular algorithms that converge linearly:[1] the classical gradient descent (GD) with fixed step size, the accelerated gradient descent (AGD) [46], and the heavy ball method (HB) [47]. We devise a novel technique for error feedback we call *differential quantization* (DQ) that compensates past quantization errors to guide the descent trajectory of a quantized algorithm to match the descent trajectory of its unquantized counterpart. By applying the DQ technique to the GD, AGD, and HB algorithms, we construct three new quantized iterative optimization algorithms: DQ-GD, DQ-AGD, and DQ-HB. By analyzing them, we show achievability bounds of the form[2]

$$\sigma_{\mathrm{A}}(n,R) \le \max\left\{\sigma_{\mathrm{A}}, \rho_n 2^{-R}\left(1+\phi_{\mathrm{A}}(n,R)\right)\right\}, \tag{2}$$

[1]Note that SGD converges only sub-linearly over smooth and strongly convex functions [41]–[43].

[2]The convergence result on DQ-HB in (2) requires that the function $\mathsf{f} \in \mathcal{F}_n$ is twice continuously differentiable.

where $\mathrm{A} \in \{\text{DQ-GD, DQ-AGD, DQ-HB}\}$, $\sigma_{\mathrm{A}} \triangleq \sigma_{\mathrm{A}}(\infty)$ is the linear convergence rate of the unquantized counterpart of A, ρ_n is the covering efficiency of the quantizer, and $\phi_{\mathrm{A}}(n,R) \ge 0$ is function that we specify; for example, $\phi_{\text{DQ-GD}}(n,R) \equiv 0$. As (2) indicates, each of the novel DQ algorithms achieves the corresponding σ_{A} for $R \ge R_{\mathrm{A}}(n)$, where

$$R_{\mathrm{A}}(n) \triangleq \min\left\{R\colon \phi_{\mathrm{A}}(n,R) = \sigma_{\mathrm{A}}\right\}. \tag{3}$$

In other words, there is no loss due to quantization once the rate surpasses $R_{\mathrm{A}}(n)$.

We show an information-theoretic converse of the form

$$\sigma_{\mathrm{A}}(n,R) \ge \max\left\{\sigma_{\mathrm{GD}}, 2^{-R}\right\}, \tag{4}$$

which applies to any "GD-like" algorithm A (the class of "GD-like" algorithms is formally defined in Definition IV.1 in Section IV below). Recalling the classical result of Rogers [48, Th. 3] that shows the existence of quantizers with covering efficiency $\rho_n \to 1$ as $n \to \infty$ and comparing (2) and (4), one can deduce the asymptotic optimality of DQ-GD within the class of "GD-like" algorithms. In contrast, the widely adopted method that quantizes the gradient of its current iterate directly [13]–[15], [49] referred to as naively quantized (NQ) GD in this paper, has linear convergence rate

$$\sigma_{\text{NQ-GD}}(n,R) \le \sigma_{\mathrm{GD}} + \rho_n 2^{-R}. \tag{5}$$

The rest of the paper is organized as follows. We present the DQ-GD algorithm in Section II. In Section III, we present its convergence analysis and an experimental validation on least-squares problems. The converse is presented in Section IV. Extensions to gradient methods with momentum and to the multiworker setting are discussed in Section V, which concludes the paper. Proofs and many details are in the extended version [50].

II. Differentially Quantized Gradient Descent

A *quantizer of dimension n and rate R* is a function $\mathsf{q}\colon \mathcal{D} \to \mathbb{R}^n$, where $\mathcal{D} \subseteq \mathbb{R}^n$ is the domain, such that its image satisfies

$$|\mathrm{Im}(\mathsf{q})| \le 2^{nR}. \tag{6}$$

We fix a dimension-n, rate-R quantizer q, and we set up quantizer q_t to be used at iteration t as

$$\mathsf{q}_t = r_t \mathsf{q}(\cdot/r_t) \tag{7}$$

for a properly chosen sequence of shrinkage factors $\{r_t\}$ (see (19), below). Therefore, each quantizer q_t has the same geometric structure but different resolution.

The (unquantized) gradient descent algorithm updates its iterate according to

$$\boldsymbol{x}_{t+1} \quad \boldsymbol{x}_t - \eta \nabla \mathsf{f}(\boldsymbol{x}_t), \tag{8}$$

where $\eta > 0$ is the constant stepsize chosen to minimize the function value along the search direction.

In Fig. 2, we illustrate an application of differential quantization (DQ) to GD (8), which yields the DQ–GD algorithm (Algorithm 1). At each iteration $t = 0, 1, 2, \ldots$, DQ–GD first guides its iterate $\hat{\boldsymbol{x}}_t$ back to the iterate $\boldsymbol{x}_t$ associated with the

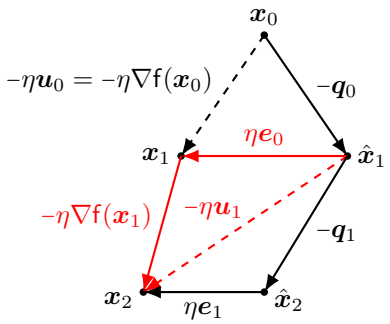

Fig. 2: Illustration of the DQ–GD algorithm (Algorithm 1).

corresponding unquantized algorithm, i.e., GD, by compensating previous scaled quantization error ηe_{t-1} (Line 4). It then computes the gradient at $z_t = x_t$ and sets the quantizer input as (Line 5)

$$u_t \quad \nabla \mathsf{f}\left(\hat{x}_t + \eta e_{t-1}\right) - e_{t-1}. \tag{9}$$

The recorded quantization error e_t captures exactly the difference between $\hat{x}_{t+1}$ and x_{t+1} for the next iteration.

Algorithm 1: DQ–GD

1 Initialize $e_{-1} = \hat{x}_0 = \mathbf{0}$
2 **for** $t = 0, 1, 2, \ldots$ **do**
3 **Worker:**
4 $z_t = \hat{x}_t + \eta e_{t-1}$
5 $u_t = \nabla \mathsf{f}(z_t) - e_{t-1}$
6 $q_t = \mathsf{q}_t(u_t)$
7 $e_t = q_t - u_t$
8 **Server:** $\hat{x}_{t+1} = \hat{x}_t - \eta q_t$
9 **end**

III. Convergence Rate

We denote by $\|\cdot\|$ the Euclidean norm, and by $\mathcal{B}(r) \triangleq \{u \in \mathbb{R}^n \colon \|u\| \le r\}$ the Euclidean ball $\mathbb{R}^n$ of radius r with center at $\mathbf{0}$.

We fix positive scalars L, and μ, and D, and we say that a continuously differentiable function $\mathsf{f} \colon \mathbb{R}^n \mapsto \mathbb{R}$ is in class $\mathcal{F}_n$ if

i) f is L-smooth, i.e.,

$$\|\nabla \mathsf{f}(v) - \nabla \mathsf{f}(w)\| \le L \|v - w\|; \tag{10}$$

ii) μ-strongly convex, i.e.,

$$\text{function } v \mapsto \mathsf{f}(v) - \frac{\mu}{2}\|v\|^2 \text{ is convex;} \tag{11}$$

iii) the minimizer $x_{\mathsf{f}}^* \triangleq \arg\min_{x \in \mathbb{R}^n} \mathsf{f}(x)$ satisfies

$$\|x_{\mathsf{f}}^*\| \le D. \tag{12}$$

We denote the *condition number* of an $\mathsf{f} \in \mathcal{F}_n$ as

$$\kappa \triangleq \frac{L}{\mu}. \tag{13}$$

Note that $\kappa \ge 1$ due to (10) and (11).

For a bounded-domain quantizer $\mathsf{q} \colon \mathcal{D} \to \mathbb{R}^n$, we refer to

$$r(\mathsf{q}) \triangleq \max\{\delta \colon \exists c \in \mathbb{R}^n \text{ s.t. } \mathcal{B}(\delta) \subseteq \mathcal{D}\} \tag{14}$$

as the *dynamic range* of q, to

$$d(\mathsf{q}) \triangleq \min\{d \colon \forall x \in \mathcal{D},\ \|x - \mathsf{q}(x)\| \le d\} \tag{15}$$

as its *covering radius*, and to

$$\rho(\mathsf{q}) \triangleq |\mathrm{Im}(\mathsf{q})|^{1/n} \frac{\mathsf{d}(\mathsf{q})}{r(\mathsf{q})}, \tag{16}$$

as its *covering efficiency*.[3] A scalar uniform quantizer q_u has domain $[-r(\mathsf{q}_\mathsf{u}), r(\mathsf{q}_\mathsf{u})]^n$ and covering efficiency $\sqrt{n}$. This is wasteful: the classical result of Rogers [48, Th. 3] implies that there exists a sequence of n-dimensional quantizers q_n with $\rho(\mathsf{q}_n) \to 1$ as $n \to \infty$, while definition (16) implies that $\rho(\mathsf{q}) \ge 1$ for any quantizer q.

Unquantized gradient descent achieves the convergence rate

$$\sigma_{\mathrm{GD}} = \frac{\kappa - 1}{\kappa + 1} \tag{17}$$

over $\mathcal{F}_n$ [46, Th. 2.1.15]. The following result provides a convergence guarantee for DQ–GD.

Theorem III.1 (Convergence of DQ–GD)**.** *Let $\mathsf{f} \in \mathcal{F}_n$. Fix a dimension-$n$, rate-$R$ quantizer q with dynamic range 1 and covering efficiency ρ_n. Then, Algorithm 1 with the stepsize*

$$\eta = \frac{2}{L + \mu} \tag{18}$$

implemented with the the shrinkage factors

$$r_t = DL \sum_{\tau=0}^{t} \sigma_{\mathrm{GD}}^{\tau} \left(\rho_n 2^{-R}\right)^{t-\tau} \tag{19}$$

in the definition of q_t (7) achieves the following convergence rate over $\mathcal{F}_n^1$ (1):

$$\sigma_{DQ\text{-}GD}(n, R) \le \max\left\{\sigma_{\mathrm{GD}}, \rho_n 2^{-R}\right\}, \tag{20}$$

Proof sketch. The path of DQ-GD and that of GD are related as (see Fig. 2)

$$\hat{x}_t = x_t - \eta e_{t-1} \tag{21}$$

Comparing (21) and Line 4 in Algorithm 1, we see that $z_t = x_t$, i.e., DQ-GD computes the gradient at the unquantized trajectory $\{x_t\}$. The convergence guarantee of GD [46, Theorem 2.1.15] controls the first term in the recursion (21). To bound the second term in (21), we observe using (16) that for any $r_t > 0$ in (7),

$$\max_{u \in \mathcal{B}(r_t)} \|\mathsf{q}_t(u) - u\| = r_t \max_{u \in \mathcal{B}(1)} \|\mathsf{q}(u) - u\| \tag{22}$$

$$= \frac{\rho_n}{2^R} r_t, \tag{23}$$

i.e. quantizer q_t used at iteration t has dynamic range r_t and covering radius (23). To complete the proof, we show by induction that with r_t in (19), the input u_t to the quantizer q_t generated by Algorithm 1 always lies within $\mathcal{B}(r_t)$, and as

[3] Covering efficiency introduced in (16) extends the notion of covering efficiency of an infinite lattice [51], which measures how well that lattice covers the whole space, to bounded-domain quantizers.

a result the quantization error decays exponentially fast. The stepsize (18) is optimal both for GD [46, Theorem 2.1.15] and for DQ-GD. See [50] for details. □

The bound in (20) exhibits a phase-transition behavior: at any $R \geq \log_2 \rho_n/\sigma_{\mathrm{GD}}$, achieving the convergence rate of unquantized GD is possible, while at any $R < \log_2 \rho_n/\sigma_{\mathrm{GD}}$, the achievable convergence rate is only 2^{-R}.

The Naively Quantized Gradient Descent (NQ–GD) is a common method of quantizing descent algorithms [13]–[17], [49] where the worker directly quantizes the gradient of its current iterate (cf. (9))

$$\boldsymbol{u}_t \leftarrow \nabla \mathsf{f}(\hat{\boldsymbol{x}}_t). \tag{24}$$

In the extended version [50], we show that (cf. (20))

$$\sigma_{\text{NQ-GD}}(n, R) \leq \sigma_{\mathrm{GD}} + \rho_n 2^{-R}. \tag{25}$$

In Figure 3, we numerically compare the linear convergence rate of DQ-GD (Algorithm 1), the NQ-GD, and the unquantized GD (8) on least-squares problems

$$\mathsf{f}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|^2 \tag{26}$$

where $\boldsymbol{y} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, with $m \geq n$. We use the uniform scalar quantizer for the ease of implementation and take as a consequence a space-filling loss of $\sqrt{n}$.

We observe that DQ-GD has a significantly faster convergence rate than NQ-GD, and that the empirical results closely track our analytical convergence bounds (20) and (25). The convergence rate of unquantized GD serves as a lower bound to both quantized algorithms.

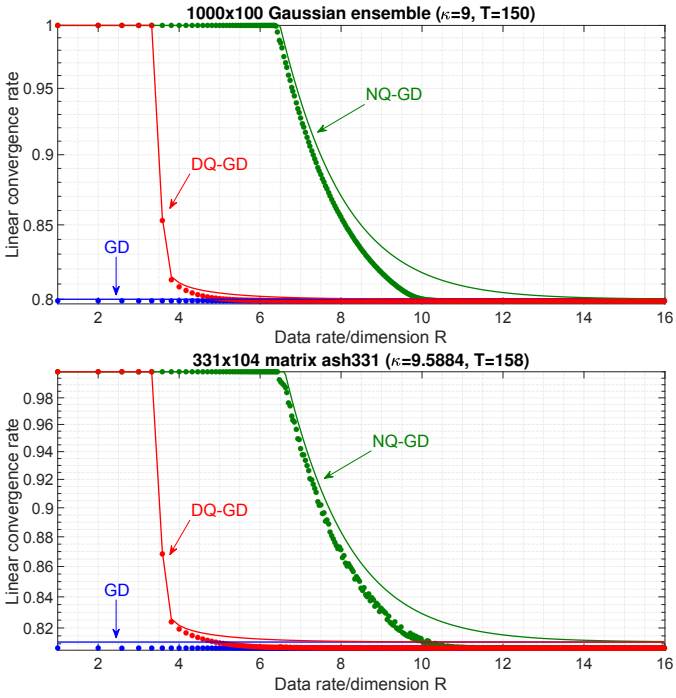

Fig. 3: Empirical convergence rates (as circles) and corresponding upper bounds (17), (20), and (25) (as lines). The real-world least-squares matrix `ash331` is extracted from the online repository *SuiteSpare* [52]. For smaller values of the data rate R, quantized GD may not even converge as $\sqrt{n}2^{-R} \geq 1$. In that case, we clip off the convergence rate at 1. For each per-dimension quantization rate $R \geq 1$, we generate 500 instances of the vector $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}_0$ with i.i.d. standard normal entries. In the case of Gaussian ensemble, we also generate 500 matrices $\mathbf{A}$'s with i.i.d. standard normal entries, one for each $\boldsymbol{y}$, and rescale the spectrum of $\mathbf{A}$ so that it has a prescribed condition number κ. We run the iterative algorithms for as many iterations T as possible until reaching the machine's finest precision.

IV. Converse

In this section, we characterize the optimal convergence rate achievable within class $\mathcal{A}_{\mathrm{GD}}$ of quantized gradient descent algorithms, formally defined next.

Definition IV.1 (Class $\mathcal{A}_{\mathrm{GD}}$ of quantized algorithms)**.** A quantized gradient descent algorithm $\mathsf{A} \in \mathcal{A}_{\mathrm{GD}}$ consists of a central server and an end worker. The worker is initialized with a sequence of quantizers q_t, and has access to the function f. We say that A is applied at dimension n and rate R if the objective function $\mathsf{f} \in \mathcal{F}_n$, and $|\mathrm{Im}(\mathsf{q}_t)| \leq 2^{nR}$. At each iteration t, the server first sends $\hat{\boldsymbol{x}}_t$ to the worker noiselessly, starting from some $\hat{\boldsymbol{x}}_0 \in \mathbb{R}^n$. The worker then determines its gradient-access point $\boldsymbol{z}_t$ and its quantizer input $\boldsymbol{u}_t$ under the structural constraints

$$\boldsymbol{z}_t \in \hat{\boldsymbol{x}}_t + \mathrm{span}\{\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{t-1}\} \tag{27}$$

$$\boldsymbol{u}_t \in \nabla \mathsf{f}(\boldsymbol{z}_t) + \mathrm{span}\{\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{t-1}\}, \tag{28}$$

where $\boldsymbol{e}_i \triangleq \boldsymbol{q}_i - \boldsymbol{u}_i, i = 0, \ldots, t-1$ are the past quantization errors before iteration t, and $+$ denotes Minkowski's sum. Upon receiving $\boldsymbol{q}_t = \mathsf{q}_t(\boldsymbol{u}_t)$ from the worker, the server performs the update

$$\hat{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{x}}_t - \eta \boldsymbol{q}_t \tag{29}$$

with a fixed stepsize $\eta > 0$.

Due to conditions (27) and (28), if there is no quantization error at each iteration (i.e., if $R = \infty$), then any quantized algorithm in $\mathcal{A}_{\mathrm{GD}}$ reduces to the unquantized gradient descent. Both DQ-GD and NQ-GD fall in the class $\mathcal{A}_{\mathrm{GD}}$.

Theorem IV.1 (Converse within class $\mathcal{A}_{\mathrm{GD}}$)**.** *The best linear convergence rate achievable within class $\mathcal{A}_{\mathrm{GD}}$ of algorithms satisfies*

$$\inf_{A \in \mathcal{A}_{\mathrm{GD}}} \sigma_A(n, R) \geq \max\left\{\sigma_{\mathrm{GD}}, 2^{-R}\right\} \tag{30}$$

Proof sketch. We fix an $\mathsf{A} \in \mathcal{A}_{\mathrm{GD}}$, and we lower-bound the convergence rate it achieves at rate R in two different ways. On one hand, we show that A cannot converge faster than the unquantized GD. Then, we use an argument similar to [53] to craft a worst-case problem instance $\mathsf{g} \in \mathcal{F}_n$ for which the iterates of the unquantized GD satisfy $\left\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*_{\mathsf{g}}\right\| = \sigma_{\mathrm{GD}}\left\|\boldsymbol{x}_t - \boldsymbol{x}^*_{\mathsf{g}}\right\|$, which ensures that $\inf_{\mathrm{A}\in\mathcal{A}_{\mathrm{GD}}} \sigma_{\mathrm{A}}(n, R) \geq \sigma_{\mathrm{GD}}$. On the other hand, we notice that if A is applied at dimension n and rate R, then the set $\mathcal{S}_{\mathrm{A}} \subseteq \mathbb{R}^n$ of all possible locations of the iterate $\hat{\boldsymbol{x}}_T$ after T iterations of A

has cardinality at most 2^{nRT}, and we apply a volume-division argument to claim that $\inf_{\mathrm{A}\in\mathcal{A}_{\mathrm{GD}}} \sigma_{\mathrm{A}}(n,R) \geq 2^R$. See [50] for details. □

Applying Theorem III.1 with Rogers-optimal quantizers with $\rho_n \to 1$ [48, Th. 3] and juxtaposing with Theorem IV.1, we characterize the optimal convergence rate achievable in the limit of large problem dimension as

$$\lim_{n\to\infty} \inf_{\mathrm{A}\in\mathcal{A}_{\mathrm{GD}}} \sigma_{\mathrm{A}}(n,R) = \max\left\{\sigma_{\mathrm{GD}}, 2^{-R}\right\}. \tag{31}$$

In other words, DQ-DG achieves the best possible convergence rate within $\mathcal{A}_{\mathrm{GD}}$, in the limit of large problem dimension. This is rather remarkable: it means not only that DQ-DG compensates previous quantization errors optimally so that no rate is wasted, but that our convergence analysis in Theorem III.1 is tight enough to capture this optimality. Furthermore, notice that the right side of (31) is < 1 at any $R > 0$. This means that at any $R > 0$ however small, DQ-DG with Rogers-optimal quantizers converges linearly at a large enough problem dimension n.

V. Extensions

A. Gradient methods with momentum

Gradient methods with momentum, such as Nesterov's accelerated gradient descent (AGD) [46], and Polyak's heavy ball (HB) method [47], rely on the memory of past iterations to smoothen the descent paths and to accelerate the convergence. Differential quantization applies to quantize these algorithms as well. See [50] for the details on DQ-AGD and DQ-HB, the differentially quantized versions of these algorithms. Where in DQ-GD (9) the quantizer input depends on the previous quantization error $\boldsymbol{e}_{t-1}$, the quantizer input $\boldsymbol{u}_t$ at iteration t of DQ-AGD and DQ-HB depends on the past two quantization errors $\boldsymbol{e}_{t-1}$ and $\boldsymbol{e}_{t-2}$:

$$\boldsymbol{u}_t \quad \nabla \mathrm{f}(\boldsymbol{z}_t) - [(1+\gamma)\boldsymbol{e}_{t-1} - \gamma \boldsymbol{e}_{t-2}] \tag{32}$$

for an appropriately chosen extrapolation coefficient. By solving a second-order linear non-homogeneous recurrence relation (the dynamic ranges' recursion is more straightforward in DQ-GD), we obtain the sequence of dynamic ranges that allows us to show achievability bounds of the form (2). For $\mathrm{A} \in \{\text{DQ-AGD, DQ-HB}\}$, the function $\phi_{\mathrm{A}}(n,R) > 0$ grows as $2^{R/2}$ with R. As (2) indicates, each of the novel DQ algorithms achieves the corresponding σ_{A} for $R \geq R_{\mathrm{A}}(n)$, where $R_{\mathrm{A}}(n)$ is defined in (3). In other words, there is no loss due to quantization once the rate surpasses $R_{\mathrm{A}}(n)$. Since $\sigma_{\mathrm{HB}} < \sigma_{\mathrm{AGD}} < \sigma_{\mathrm{GD}}$, this means that DQ-AGD, DQ-HB outperform DQ-DG at a large enough data rate R. This is no contradiction to the converse in Theorem IV.1 because these algorithms belong to the class $\mathcal{A}_{\mathrm{GMM}} \supset \mathcal{A}_{\mathrm{GD}}$, which relaxes (29) as

$$\hat{\boldsymbol{x}}_{t+1} \in \hat{\boldsymbol{x}}_0 + \operatorname{span}\{\boldsymbol{q}_0, \ldots, \boldsymbol{q}_t\}. \tag{33}$$

Unlike DQ-GD that enjoys linear convergence for any $R > 0$, DQ-AGD and DQ-HB exhibit a second phase transition: for $R \leq R'_{\mathrm{A}}$, where

$$R'_{\mathrm{A}} \triangleq \max\left\{R\colon 2^{-R}\left(1 + \lim_{n\to\infty} \phi_{\mathrm{A}}(n,R)\right) = 1\right\}, \tag{34}$$

these algorithms do not converge linearly even for an arbitrarily large n. Thus for small rates R, DQ–GD outperforms these momentum methods.

B. Multiworker gradient methods

The converse in Theorem IV.1 extends to quantized gradient descent on a sum of K smooth and strongly convex objective functions where each worker has access to only one of the summands and the sum rate is constrained by nR bits per iteration across the workers. Differential quantization does not directly apply to K-worker quantized gradient descent since each worker does not know the local quantization errors stored by the others, and thus cannot guide the descent trajectory back to the unquantized path. Thus, whether (30) is attainable in the multiworker setting, and how should each worker optimally compensate its own past quantization errors remains an open problem. In this direction, we show in [50] that the multiworker NQ-GD attains convergence rate $\sigma_{\mathrm{GD}} + c\,\rho_n 2^{-R}$, where c is a constant that is optimized by the rate allocation $R_k \sim \log_2 L_k$, where L_k is the smoothness parameter of the k-th local function. Although to the best of our knowledge this is the fastest convergence result for distributed quantized (non-stochastic) gradient descent, it approaches the converse only in the limit of large R.

Acknowledgment

Stimulating discussions with Dr. Himanshu Tyagi and Dr. Vincent Tan are gratefully acknowledged.

References

[1] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems 23*, Vancouver, British Columbia, Canada, Dec. 2010, pp. 2595–2603.

[2] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems 24*, Granada, Spain, Dec. 2011, pp. 693–701.

[3] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling Up Machine Learning: Parallel and Distributed Approaches*. New York, NY, USA: Cambridge University Press, Dec. 2011.

[4] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems 25*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1223–1231.

[5] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project Adam: Building an efficient and scalable deep learning training system," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, Broomfield, CO, Oct. 2014, pp. 571–582.

[6] C. M. De Sa, C. Zhang, K. Olukotun, C. Ré, and C. Ré, "Taming the wild: A unified analysis of hogwild-style algorithms," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., Dec. 2015, pp. 2674–2682.

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

[8] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3027–3036.

[9] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and application to data-parallel distributed training of speech DNNs," in *Interspeech 2014*, Sep. 2014.

[10] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Advances in Neural Information Processing Systems 27*, Montreal, QB, Canada, Dec. 2014, pp. 19–27.

[11] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *INTERSPEECH*, Sep. 2015.

[12] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Advances in Neural Information Processing Systems 28*, Montreal, QB, Canada, Dec. 2015, pp. 685–693.

[13] 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 30*, Long Beach, CA, USA, Dec. 2017, pp. 1709–1720.

[14] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in Neural Information Processing Systems 30*, Long Beach, CA, USA, Dec. 2017, pp. 1509–1519.

[15] 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*, Long Beach, CA, USA, Jul. 2018, pp. 560–569.

[16] P. Mayekar and H. Tyagi, "RATQ: A universal fixed-length quantizer for stochastic optimization," *arXiv*, vol. 1908.08200, Dec. 2019.

[17] A. Ramezani-Kebrya, F. Faghri, and D. M. Roy, "Nuqsgd: Improved communication efficiency for data-parallel sgd via nonuniform quantization," Aug. 2019.

[18] P. Mayekar and H. Tyagi, "Limits on gradient compression for stochastic optimization," *arXiv*, vol. 2001.09032, Jan. 2020.

[19] V. Gandikota, D. Kane, R. K. Maity, and A. Mazumdar, "vqsgd: Vector quantized stochastic gradient descent," Nov. 2019.

[20] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes SignSGD and other gradient compression schemes," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, CA, USA: PMLR, June 2019, pp. 3252–3261.

[21] 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, 2016, pp. 1–8.

[22] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Sep. 2017, pp. 440–445.

[23] 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 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., Dec. 2018, pp. 5973–5983.

[24] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems 31*, Montréal, Canada, Dec. 2018, pp. 4447–4458.

[25] M. Yu, Z. Lin, K. Narra, S. Li, Y. Li, N. S. Kim, A. Schwing, M. Annavaram, and S. Avestimehr, "GradiVeQ: Vector quantization for bandwidth-efficient gradient aggregation in distributed CNN training," *Advances in Neural Information Processing Systems*, vol. 31, pp. 5123–5133, 2018.

[26] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *International Conference on Learning Representations*, Vancouver, BC, Canada, Apr. 2018.

[27] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Advances in Neural Information Processing Systems 31*, Montréal, Canada, Dec. 2018, pp. 1299–1309.

[28] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "ATOMO: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems 31*, Montréal, Canada, Dec. 2018, pp. 9850–9861.

[29] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., Dec. 2019, pp. 14 695–14 706.

[30] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan, "On biased compression for distributed learning," Feb. 2020.

[31] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, "Distributed learning with compressed gradient differences," Jan. 2019.

[32] S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik, "Stochastic distributed learning with gradient quantization and variance reduction," Apr. 2019.

[33] S. Horváth, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtarik, "Natural compression for distributed deep learning," May. 2019.

[34] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 1432–1436.

[35] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2020.

[36] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[37] R. M. Gray, *Source Coding Theory*. Kluwer Academic Publishers, Oct. 1989.

[38] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul. 2018, pp. 5325–5333.

[39] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication," Sep. 2019.

[40] S. Zheng, Z. Huang, and J. Kwok, "Communication-efficient distributed blockwise momentum sgd with error-feedback," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., Dec. 2019, pp. 11 450–11 460.

[41] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[42] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3-4, pp. 231–357, Nov. 2015.

[43] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, May 2018.

[44] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, Broomfield, CO, Oct. 2014, pp. 583–598.

[45] S. Khirirat, H. R. Feyzmahdavian, and M. Johansson, "Distributed learning with compressed gradients," Jun. 2018.

[46] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, Dec. 2014.

[47] B. T. Polyak, *Introduction to optimization*, ser. Translations Series in Mathematics and Engineering. Optimization Software, Inc., Publications Division, New York, May 1987.

[48] C. A. Rogers, "Covering a sphere with spheres," *Mathematika*, vol. 10, no. 2, pp. 157—164, Dec. 1963.

[49] M. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1380–A1405, May 2012.

[50] C.-Y. Lin, V. Kostina, and B. Hassibi, "Differentially quantized gradient descent," *ArXiv preprint arXiv: 2002.02508*, 2021.

[51] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. USA: Cambridge University Press, Sep. 2014.

[52] S. Kolodziej, M. Aznaveh, M. Bullock, J. David, T. Davis, M. Henderson, Y. Hu, and R. Sandstrom, "The suitesparse matrix collection website interface," *Journal of Open Source Software*, vol. 4, no. 35, p. 1244, Mar. 2019.

[53] E. de Klerk, F. Glineur, and A. B. Taylor, "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions," *Optim. Lett.*, vol. 11, no. 7, pp. 1185–1199, Oct. 2017.