Quantized Compressive Sampling of Stochastic Gradients for Efficient Communication in Distributed Deep Learning

Afshin Abdi, Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA
{abdi,fekri}@gatech.edu

Abstract

In distributed training of deep models, the transmission volume of stochastic gradients (SG) imposes a bottleneck in scaling up the number of processing nodes. On the other hand, the existing methods for compression of SGs have two major drawbacks. First, due to the increase in the overall variance of the compressed SG, the hyperparameters of the learning algorithm must be readjusted to ensure the convergence of the training. Further, the convergence rate of the resulting algorithm still would be adversely affected. Second, for those approaches for which the compressed SG values are biased, there is no guarantee for the learning convergence and thus an error feedback is often required. We propose *Quantized Compressive Sampling* (QCS) of SG that addresses the above two issues while achieving an arbitrarily large compression gain. We introduce two variants of the algorithm: Unbiased-QCS and MMSE-QCS and show their superior performance w.r.t. other approaches. Specifically, we show that for the same number of communication bits, the convergence rate is improved by a factor of 2 relative to state of the art. Next, we propose to improve the convergence rate of the distributed training algorithm via a *weighted error feedback*. Specifically, we develop and analyze a method to both control the overall variance of the compressed SG and prevent the staleness of the updates. Finally, through simulations, we validate our theoretical results and establish the superior performance of the proposed SG compression in the distributed training of deep models. Our simulations also demonstrate that our proposed compression method expands substantially the region of step-size values for which the learning algorithm converges.

1 Introduction

In recent years, the size of deep learning problems has increased significantly both in terms of the number of available training samples as well as the complexity of the model. Hence, training deep models on a single processing node is unappealing or nearly impossible. One viable approach to overcome the memory, storage and computational constraints is distributing the training over multiple processing units (a.k.a. workers). However, exchanging the gradients or the parameters of the model and synchronizing the workers' models incur significant communication overhead which is a major bottleneck in distributed deep learning. In recent years, there has been a great amount of effort on reducing the communication overhead. The majority of existing methods rely on either relaxing the synchronization among workers (Dean et al. 2012; Niu et al. 2011; Zhang, Hsieh, and Akella 2016; Ho et al. 2013; Stich 2019) or reducing the overall transmission rate via sparsification or quantization of the gradients.

Sparsification- This approach is based on transmitting only the *important* or a small subset of the gradients. (Strom 2015) was among the early works to use sparsification in conjunction with thresholded quantization to compress the gradients. As choosing the right threshold for gradient sparsification is difficult in practice and to improve the performance of distributed learning, other sparsification methods have also been proposed such as transmitting only a fixed portion of the gradients (Dryden et al. 2016; Aji and Heafield 2017), TopK SGD (Alistarh et al. 2018), deep gradient compression (Lin et al. 2018), random (stochastic) sparsification of the gradients (Wangni et al. 2018) and sparsification of the gradients in the transform domain (Wang et al. 2018).

Quantization- Reducing the number of bits in representing SG is a well-known technique to decrease the communication bit-rate. For example, quantizing the gradients to one-bit (Seide et al. 2014) or SignSGD (Bernstein et al. 2018) can significantly reduce the communication overhead. However, the reduced accuracy of gradients and quantization bias may impair the convergence rate. Using more quantization levels, adaptive quantizers (Dryden et al. 2016) or exploiting error-feedback (Wu et al. 2018; Karimireddy et al. 2019), one can alleviate such issues. Alternatively, stochastic quantizers such as QSGD (Alistarh et al. 2017), TernGrad (Wen et al. 2017) and Dithered Quantization (Abdi and Fekri 2019a; 2019b) provide unbiased quantization with performance and convergence guarantees.

It is worth mentioning that these approaches can be applied simultaneously as complement of each other to reduce the communication overhead significantly in distributed deep learning.

Our Contribution In this paper, we claim that the existing methods for compressing stochastic gradients (or model's

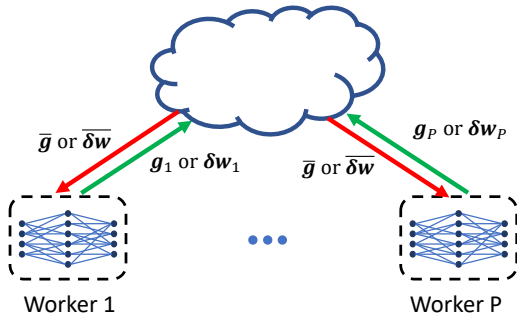


Figure 1: Schematic overview of the distributed training.

updates) suffer from few drawbacks such as increase in the total variance of SG, limited compression gains or added bias to the SG. These issues can adversely affect the convergence of the distributed learning algorithm. We propose a novel approach, Quantized Compressive Sampling (QCS) for the compression of stochastic gradients or parameters of deep models. The algorithm employs both a dither quantization and compressive sensing to achieve arbitrarily large compression gains while at the same time has desired properties such as unbiasedness and having small compression error. To further improve the convergence rate of QCS, we suggest using *weighted* error feedback. We theoretically analyze the effect of error feedback on the convergence rate and show that how the introduction of a 'decaying factor' in the feedback can improve the stability of the training via 1) controlling the variance of the residual signal in the error feedback, and 2) forgetting the outdated gradients.

Notations

Bold lowercase letters represent vectors and the i-th element of the vector $\boldsymbol{x}$ is denoted as x_i. Matrices are denoted by bold capital letters such as $\boldsymbol{X}$, with the (i,j)-th element represented by $X_{i,j}$ or $[\boldsymbol{X}]_{i,j}$. $\boldsymbol{A} \odot \boldsymbol{B}$ is the Hadamard product of $\boldsymbol{A}$ and $\boldsymbol{B}$. $\boldsymbol{A} \odot \boldsymbol{v}$ for vector $\boldsymbol{v}$ is computed by expanding the dimension of $\boldsymbol{v}$ appropriately to make it the same size as $\boldsymbol{A}$. Given a real number $x \in \mathbb{R}$, $\lfloor x \rceil$ is the nearest integer to x and $\text{sign}(x)$ is the sign of x defined as $+1$ for $x > 0$ and -1 for $x \leq 0$. $\log$ and $\log_2$ denote the natural and base 2 logarithms, respectively.

For a random variable u, $u \sim \mathcal{U}(a,b)$ if its probability distribution is uniform over interval (a,b).

Throughout the paper, usually n refers to the number of parameters, $\boldsymbol{g}$ stochastic gradient of the parameters, P is the number of processing nodes or workers and Q is the range of the quantizer, i.e., the output of the quantizer would be in $\{-Q,\ldots,0,\ldots,Q\}$.

2 Problem Statement and Motivation

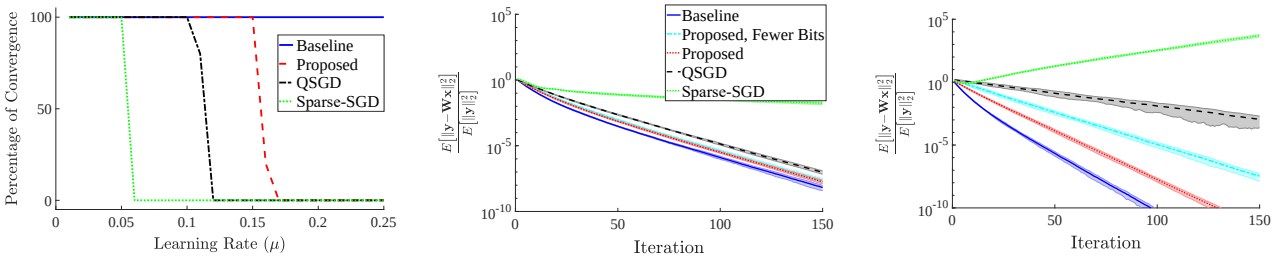
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable objective function to be minimized. We consider the distributed optimization of $f(\cdot)$ as shown in Fig. 1. There are P separate workers which have their own copy of the model to be trained. At each iteration of training, each worker computes a stochastic gradient (SG) of the parameters ($\boldsymbol{g}_p$) based on its own available data. It is then transmitted to a server (in the centralized training) or communicated with other workers (in the decentralized topology) to compute the average. The average of all gradients or the updates ($\bar{\boldsymbol{g}}$) is then broadcasted back to all workers to update their local copy of the model.

Quantizing the gradients ($\boldsymbol{g}_k$'s) is a well-known approach to reduce the number of transmitted bits and mitigate the communication bottleneck in distributed training. However, the existing quantization methods have few drawbacks;

- Due to the quantization noise, the total variance of the SG would be increased, and as illustrated later in the following example, the learning algorithm with quantized SG may not converge with the same set of training hyper-parameters as the baseline (non-quantized) algorithm. Hence, the hyper-parameters must be adjusted to ensure the convergence of the learning algorithms, which in turn can increase the required *number of training iterations* for the convergence of the model.
- If the quantizer is biased (e.g., signSGD), the training algorithm is not guaranteed to converge (see, e.g., (Karimireddy et al. 2019)).
- Since the small gradients are suppressed by the larger ones and thus would be most likely quantized to zero, the parameters whose gradients are relatively small may not be updated even if their gradients point to the same direction in multiple consecutive iterations of training.

Although using error-feedback (Wu et al. 2018; Karimireddy et al. 2019) can alleviate these issues to some degrees, the requirement to store the residual of quantization at each worker increases the memory footprint of the training algorithm significantly which is undesirable in many applications esp. for large deep models.

Example (Linear Regression)**.** Consider learning a linear regression model $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}$ with mean squared error (MSE) cost function $f = 0.5\,\mathbb{E}\big[\|\boldsymbol{y} - \boldsymbol{W}\boldsymbol{x}\|_2^2\big]$, where $\boldsymbol{y} \in \mathbb{R}^m$ is the desired (target) signal and $\boldsymbol{x} \in \mathbb{R}^n$ is the input. Assume that $\boldsymbol{x}$ is a zero-mean multivariate Gaussian random vector with correlation matrix $\boldsymbol{R}$ whose maximum and minimum eigenvalues are $\lambda_{\max}(\boldsymbol{R}) = 4$ and $\lambda_{\min}(\boldsymbol{R}) = 1$, respectively. It is known that gradient descent with step-size $\mu < 1/\lambda_{\max} = 1/4$ converges to the optimal solution. To investigate the impact of compressing SG on the convergence rate, we consider learning $\boldsymbol{W}$ via stochastic gradient descent algorithm with batch-size 32 and using no quantization (baseline), QSGD (Alistarh et al. 2017), Sparse-SGD (Wangni et al. 2018), and our proposed method (presented later in the paper). The parameters are adjusted such that compression gains of all methods are approximately 21, except the method labeled as '*Proposed, Fewer Bits*' in Fig. 2b which uses approximately 40% fewer bits. We set $n = 64$, $m = 50$ and repeated the experiments several times with different values of learning rate to obtain the range of μ that the training algorithm converges and the corresponding convergence rate. Figure 2a shows the percentage of times different learning algorithms converge vs. step-size μ. We nose that quantization or sparsification reduces the range of μ for which SGD converges. However, our proposed method significantly increases that range compared to existing methods. Although



(a) Convergence region for step-size μ

(b) Convergence rate, $\mu = 0.05$

(c) Convergence rate, $\mu = 0.1$

Figure 2: Effect of different quantization techniques on the convergence of SGD on learning simple linear regression model. The shaded region represents variations of ± 1.5 standard deviation. Note that the scale of convergence plots is logarithmic.

using smaller μ ensures the convergence for QSGD and Sparsified SGD (see Fig. 2b), it sacrifices the potential of higher convergence rates that can be achieved by using larger step-sizes (Fig. 2c). In this example, our proposed method consistently outperforms the other existing algorithms. Even by step-size $\mu = 0.10$, the convergence time can be reduced by a factor of 2 compared to the QSGD with $\mu = 0.05$.

Next, we introduce our proposed Quantized Compressive Sampling (QCS) method for the compression of stochastic gradients or parameters of deep models. The algorithm, employs both a dither quantization and compressed sampling to achieve arbitrarily large compression gains. We introduce two variants of the algorithm: 1. unbiased compression of SG (Unbiased-QCS), and 2. compression with the minimum mean squared error (MMSE-QCS). In section 4, we introduce weighted error feedback to improve the convergence rate of the learning algorithm and show that the introduction of a 'decaying factor' in the feedback greatly improves the stability of the training. In section 5, we theoretically prove the convergence for the QCS-SGD, analyze the convergence rates of the proposed methods and compare them with the baseline (uncompressed) SGD. We theoretically show that the weighted error feedback can improve the convergence rate and close the gap from the baseline (uncompressed) SGD. Finally, in section 6, we evaluate the theoretical results and compare our approach against the state-of-the-art and the baseline.

3 Quantized Compressive Sampling of Stochastic Gradient

Let $\boldsymbol{g} \in \mathbb{R}^n$ be the stochastic gradient of the model. Instead of directly compressing $\boldsymbol{g}$, our proposed method is based on mapping $\boldsymbol{g}$ onto $\mathbb{R}^k$, $k \leq n$, via $\boldsymbol{v} = \boldsymbol{T}\boldsymbol{g}$ and then compressing $\boldsymbol{v}$. Here, $\boldsymbol{T}$ is a random mixing matrix chosen from a class of appropriate transforms $\mathcal{T}$. Inspired by the work on structured measurement matrix in compressed sensing, we consider the following class of random mixing matrices

$$\boldsymbol{T} = \frac{1}{\sqrt{k}} \boldsymbol{H}\boldsymbol{R}, \tag{1}$$

where $\boldsymbol{R}$ is a random Rademacher diagonal matrix, i.e., $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{r})$, $\mathrm{P}(r_i = 1) = \mathrm{P}(r_i = -1) = 0.5$, and $\boldsymbol{H}$ is constructed by picking up the first k rows[1] from the Hadamard matrix $\boldsymbol{H}_n \in \mathbb{R}^{n \times n}$. Fore more detailed analysis of the proposed class of transforms, please refer to §2 of the Supp. document. Note that the random transformation can be alternatively applied as

$$\boldsymbol{v} = \frac{1}{\sqrt{k}} \boldsymbol{H}(\boldsymbol{r} \odot \boldsymbol{g}). \tag{2}$$

Lemma 1. *The random mixing matrix $\boldsymbol{T}$ has the following properties:*

$$\boldsymbol{T}\boldsymbol{T}^{\mathsf{T}} = \frac{n}{k}\boldsymbol{I}, \quad \mathbb{E}\left[\boldsymbol{T}^{\mathsf{T}}\boldsymbol{T}\right] = \boldsymbol{I}. \tag{3}$$

The quantization and compression of $\boldsymbol{v}$ is based on dithered quantization (Schuchman 1964; Gray and Stockham 1993) (see §1 of Supp. document). Let Q be the desired range of quantization levels and $\boldsymbol{u} \sim \mathcal{U}(-1/2, 1/2)$ be the random dither signal, independent of $\boldsymbol{v}$. The dithered quantization of $\boldsymbol{v}$ is computed as

$$\boldsymbol{q} = \lfloor \boldsymbol{v}/\varsigma + \boldsymbol{u} \rceil, \tag{4}$$

where the *scale factor* $\varsigma = \|\boldsymbol{v}\|_\infty / Q$ maps the elements of $\boldsymbol{v}$ into the range $[-Q, Q]$. For 1-bit dithered quantization (see remark 2 of §1 in Supp. document), set $\varsigma = 2\|\boldsymbol{v}\|_\infty$ and

$$\boldsymbol{q} = \mathrm{sign}(\boldsymbol{v}/\varsigma + \boldsymbol{u}). \tag{5}$$

The *Quantized Compressive Sampling* (QCS) of $\boldsymbol{g}$ is then computed via first dequantizing $\boldsymbol{v}$ as

$$\widehat{\boldsymbol{v}} = \varsigma\,(\boldsymbol{q} - \boldsymbol{u}), \tag{6}$$

and then estimating $\boldsymbol{g}$ from $\widehat{\boldsymbol{v}}$. Note that the quantization of $\boldsymbol{v}$ can be written as

$$\widehat{\boldsymbol{v}} = \boldsymbol{v} + \varsigma\boldsymbol{\varepsilon}, \tag{7}$$

where, as a result of Thm. I in Supp. document, the scaled quantization noise $\boldsymbol{\varepsilon}$ is independent of the signals and $\boldsymbol{\varepsilon} \sim \mathcal{U}(-1/2, 1/2)$.[2] Note that although $\boldsymbol{T}$ is a random matrix, the

[1] It is possible to choose any arbitrary or random subset of k rows from $\boldsymbol{H}_n$, but the performance and analysis would be the same.

[2] Note that this is not the case for ordinary quantization or stochastic quantization (QSG) in (Alistarh et al. 2017).

server can reproduce it by using the same random number generators and seed numbers. We consider two different criteria for reconstructing $\boldsymbol{g}$; 1) minimizing the mean squared error $\mathbb{E}\left[\|\boldsymbol{g}-\widehat{\boldsymbol{g}}\|_2^2\right]$ and 2) finding an unbiased estimator. To have simple yet efficient estimation of $\boldsymbol{g}$ from $\widehat{\boldsymbol{v}}$, we restrict ourselves to the class of linear estimators given by

$$\widehat{\boldsymbol{g}} = \alpha \boldsymbol{T}^{\mathsf{T}} \widehat{\boldsymbol{v}}, \tag{8}$$

where α is a scalar which may depend on ς but is independent of $\boldsymbol{g}$.

Remark 1. Note that $\widehat{\boldsymbol{g}}$ lies in the subspace of $\mathbb{R}^n$ spanned by the rows of $\boldsymbol{T}$. Therefore, if $\boldsymbol{T}$ was deterministic, the gradients would always be projected into a fixed k-dimensional subspace of $\mathbb{R}^n$, preventing the training algorithm to converge in general. The randomness added to $\boldsymbol{T}$ makes these subspaces change at each iteration of training, helping the training algorithm to explore 'almost' all directions in $\mathbb{R}^n$ in $\mathcal{O}(\frac{n}{k})$ iterations (see Supp. document).

Unbiased Estimator

We constraint the reconstruction matrix such that the resulting quantizer be unbiased, $\mathbb{E}[\widehat{\boldsymbol{g}}] = \boldsymbol{g}$, for any arbitrary $\boldsymbol{g}$. Using Lemma 1, it can be easily verified that for an *unbiased QCS*, the reconstruction matrix is given by

$$\alpha = 1. \tag{9}$$

The following theorem summarizes the properties of the proposed QCS.

Theorem 2. *The QCS with $\alpha = 1$ is unbiased and has bounded variance error. More specifically, for an arbitrary $\boldsymbol{g} \in \mathbb{R}^n$, let $\widehat{\boldsymbol{g}} = \boldsymbol{T}^{\mathsf{T}}\widehat{\boldsymbol{v}}$ be the QCS of $\boldsymbol{g}$ and $\boldsymbol{e} = \boldsymbol{g} - \widehat{\boldsymbol{g}}$. Then,*

P1. *The quantizer is unbiased, i.e., $\mathbb{E}[\boldsymbol{e}] = \boldsymbol{0}$.*

P2. *The variance of error is bounded as $\mathbb{E}\left[\|\boldsymbol{e}\|_2^2\right] \le \gamma \|\boldsymbol{g}\|_2^2$ where γ is a constant given by*

$$\gamma = \begin{cases} \frac{n}{k} - 1 + \frac{n}{4Q^2}\,\frac{\log(k)}{k-1} & k \ge 2 \\ n-1 & k = 1 \end{cases} \tag{10}$$

Thm. 2 provides a trade-off between the number of transmission bits per value and the variance of QCS. Assuming that the overhead to transmit scale factor ς is negligible, the total transmission bits would be $k\log(2Q+1)$ and hence the compression gain is

$$\text{gain} = \frac{nb}{k\log_2(2Q+1)}, \tag{11}$$

where b is the number of bits used in representing each parameter (generally, in floating point computations $b = 32$). For a fixed compression gain, minimizing (10) would result in the optimum number of quantization levels Q and k. Figure 3 shows the minimum achievable γ using the proposed unbiased QCS and compares it with QSG (Lemma 3.1 of (Alistarh et al. 2017)) and the *lower bound* of the expected compression gain of the unbiased sparsification (Wangni et al. 2018). Note that the compression gain of (Alistarh et al. 2017) is at most 32. As shown in the figure, the variance bound of our proposed unbiased QCS is orders of magnitude lower than both other approaches.

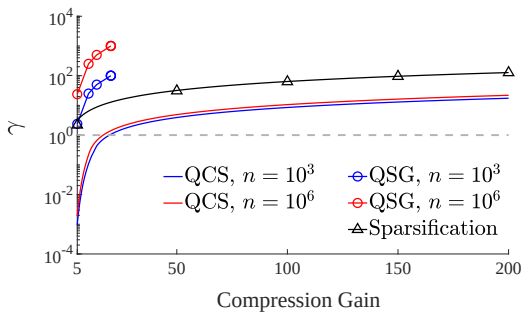


Figure 3: Variance bound γ vs. compression gain.

Minimum Mean Squared Error Estimator

In *MMSE-QCS*, the objective is finding the reconstruction matrix such that $\mathbb{E}\left[\|\widehat{\boldsymbol{g}}-\boldsymbol{g}\|_2^2\right]$ is minimized. However the quantizer is not necessarily unbiased. In this case, the reconstruction matrix is given by setting

$$\alpha = \frac{1}{\gamma+1}, \tag{12}$$

where γ is as in (10).

Lemma 3. *For an arbitrary $\boldsymbol{g} \in \mathbb{R}^n$, let $\widehat{\boldsymbol{g}} = \alpha \boldsymbol{T}^{\mathsf{T}}\widehat{\boldsymbol{v}}$ be the QCS of $\boldsymbol{g}$ and $\boldsymbol{e} = \boldsymbol{g} - \widehat{\boldsymbol{g}}$. Then, for α given by* (12)*, we have*

$$\mathbb{E}\left[\|\boldsymbol{e}\|_2^2\right] \le (1-\alpha)\|\boldsymbol{g}\|_2^2. \tag{13}$$

Algorithm 1 summarizes the proposed quantization and reconstruction for Unbiased-QCS and MMSE-QCS. Note that both QUANTIZE and DEQUATNIZE functions generate the same random Rademacher and uniform sequences via utilizing identical random number generation algorithms with the same seed values.

Algorithm 1 Quantized Compressive Sampling of SG

function QUANTIZE($\boldsymbol{g}, \boldsymbol{H}, Q$)
  Generate random Rademacher vector $\boldsymbol{r}$.
  Generate random dither $\boldsymbol{u} \sim \mathcal{U}(-1/2, 1/2)$.
  $\boldsymbol{v} \leftarrow \frac{1}{\sqrt{k}}\boldsymbol{H}(\boldsymbol{r} \odot \boldsymbol{g})$
  $\varsigma \leftarrow \|\boldsymbol{v}\|_\infty / Q$
  $\boldsymbol{q} \leftarrow \lfloor \boldsymbol{v}/\varsigma + \boldsymbol{u} \rceil$
  return $\boldsymbol{q}$ and ς

function DEQUANTIZE($\boldsymbol{q}, \varsigma, \boldsymbol{H}$)
  Set α. ▷ via (9) or (12)
  Reproduce random Rademacher vector $\boldsymbol{r}$.
  Reproduce random dither $\boldsymbol{u} \sim \mathcal{U}(-1/2, 1/2)$.
  $\widehat{\boldsymbol{v}} = \varsigma\,(\boldsymbol{q} - \boldsymbol{u})$
  $\widehat{\boldsymbol{g}} = \frac{\alpha}{\sqrt{k}}\boldsymbol{r} \odot (\boldsymbol{H}^{\mathsf{T}}\widehat{\boldsymbol{v}})$
  return $\widehat{\boldsymbol{g}}$

4 Weighted Error Feedback

Application of quantization or sparsification techniques in deep learning may introduce two major issues: (i) increase

in the variance of the aggregated gradients, and (ii) insertion of a bias to the stochastic gradient. These may degrade the convergence speed or even cause the learning algorithm fail to converge. A key component in tackling both of these issues is aggregating the compression residuals (i.e., quantization or sparsification errors) and carrying forward to the next mini-batch. This ensures that the true values of SG are eventually applied to the parameters of the deep model, although it may take several transmissions, i.e., it resembles stale (partial) gradient updates. Exploiting such a feedback can speed up the convergence rate or ensure the convergence of the learning algorithms such as stochastic gradient descents even in the presence of (biased) gradient compression (Stich, Cordonnier, and Jaggi 2018; Wu et al. 2018; Karimireddy et al. 2019).

Since adding quantization error from previous steps can potentially increase the overall variance of SG and the staleness of the gradients, we add a forgetting factor β in the error feedback which is a crucial part in bounding the variance of error feedback as we will show next. Let $\boldsymbol{r}_t$ be the running compression residue at the t-th iteration, with $\boldsymbol{r}_0 = \mathbf{0}$, and $\text{COMPRESS}(\cdot)$ denote quantizing and then dequantizing using Alg. 1. At the t-th iteration of training, the compression and residue update would be computed as

$$\boldsymbol{z}_t \leftarrow \boldsymbol{g}_t + \beta \boldsymbol{r}_t \tag{14a}$$
$$\widehat{\boldsymbol{z}}_t \leftarrow \text{COMPRESS}(\boldsymbol{z}_t) \tag{14b}$$
$$\boldsymbol{e}_t \leftarrow \boldsymbol{z}_t - \widehat{\boldsymbol{z}}_t \tag{14c}$$
$$\boldsymbol{r}_{t+1} \leftarrow (1-\beta)\boldsymbol{r}_t + \boldsymbol{e}_t \tag{14d}$$

and the parameters of the model are updated using $\widehat{\boldsymbol{z}}_t$ instead of SG $\boldsymbol{g}_t$. The next lemma states the sufficient conditions on β for the residual signal $\boldsymbol{r}_t$ be ℓ_2 bounded in expectation.

Lemma 4. *Assume that the SGs are ℓ_2 bounded, i.e., $\mathbb{E}\big[\|\boldsymbol{g}\|_2^2\big] \le B$. Then, $\mathbb{E}\big[\|\boldsymbol{r}_t\|_2^2\big] \le \eta B$, where*

- *for Unbiased-QCS and all $0 < \beta < \min(1, 2/(1+\gamma))$,*

$$\eta = \frac{\gamma}{1 - ((1-\beta)^2 + \beta^2\gamma)}. \tag{15}$$

- *For MMSE-QCS and $0 < \beta \le 1$,*

$$\eta = \frac{\gamma}{\left(\sqrt{\gamma+1-(1-\beta)^2} - \sqrt{\gamma}\right)^2}. \tag{16}$$

Note that for Unbiased-QCS, since γ might be greater than 1, the residual signal's magnitude may become unbounded for $\beta = 1$ (i.e., the traditional error feedback method), and hence the learning algorithm would not converge with error feedback. On the other hand, in MMSE-QCS all values of $0 \le \beta \le 1$ are viable choices for convergence with the error feedback.

Remark 2. We can choose β to minimize the upper bound on the ℓ_2 norm of the residual signal. In this case, the optimum values of β and the corresponding upper bound η for Unbiased-QCS and MMSE-QCS are given by (17) and (18), respectively;

$$\beta_u^* = \frac{1}{\gamma+1}, \quad \eta_u^* = \gamma(\gamma+1) \tag{17}$$
$$\beta_m^* = 1, \qquad \eta_m^* = \frac{\gamma}{\left(\sqrt{\gamma+1} - \sqrt{\gamma}\right)^2}. \tag{18}$$

Moreover, as it can be easily verified, $\eta_u^* < \eta_m^*$. Hence, the *theoretical* upper bound for the magnitude of the residual signal in Unbiased-QCS with weighted error feedback is smaller than MMSE-QCS.

Remark 3. Using Lemma 3 of (Karimireddy et al. 2019), by simple derivations and noting that δ in their notation is the same as $1/(\gamma+1)$ for MMSE-QCS, we realize that the upper bound in (Karimireddy et al. 2019) equals to $\eta_k = 4\gamma(\gamma+1)$ which can be easily verified that it is larger than η_m^* derived here.

5 Convergence Analysis

In this section, we show the convergence of the proposed SG compression algorithms with and without error feedback. In our analysis, we consider the gradient descent algorithm with the compressed stochastic gradients and we make the following assumptions;

Assumption 1. *The loss function is Lipschitz-smooth, i.e., there exists a constant L such that for all $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$*

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\|_2 \le L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2. \tag{19}$$

Assumption 2. *The stochastic gradients are ℓ_2 bounded in expectation, i.e., $\exists B > 0$ such that*

$$\mathbb{E}\big[\|\boldsymbol{g}\|_2^2\big] \le B. \tag{20}$$

Remark 4. Note that Assumption 2 can be relaxed to have bounded variance SG, i.e., $\mathbb{E}\big[\|\boldsymbol{g} - \nabla f\|_2^2\big] \le \sigma^2$ for some constant σ. The analysis would be slightly more involved, however the convergence results would be similar to the ones that are stated here (see supplementary document).

First, we consider training for T iterations of SGD with fixed step-size μ using Unbiased-QCS and MMSE-QCS without any error feedback, i.e., at the t-th iteration, the parameters are updated as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \mu \widehat{\boldsymbol{g}}_t, \tag{21}$$

where $\widehat{\boldsymbol{g}}_t$ is compressed SG from either Unbiased-QCS or MMSE-QCS.

Lemma 5. *Let f^* be the minimum of objective function $f(\cdot)$. Assuming* (19) *and* (20) *hold, in training with Unbiased-QCS, we get*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[\|\nabla f(\boldsymbol{w}_t)\|_2^2\big] \le \frac{f(\boldsymbol{w}_0) - f^*}{T\mu} + \frac{L}{2}\mu(1+\gamma)B.$$

Similarly, for MMSE-QCS we have

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[\|\nabla f(\boldsymbol{w}_t)\|_2^2\big] \le (1+\gamma)\frac{f(\boldsymbol{w}_0) - f^*}{T\mu} + \frac{L}{2}\mu B.$$

In both cases, by appropriate choice of step size, we can achieve $\mathcal{O}(1/\sqrt{T})$ convergence rate

$$\min_t \mathbb{E}\big[\|\nabla f(\boldsymbol{w}_t)\|_2^2\big] \le \frac{f(\boldsymbol{w}_0) - f^* + \frac{L}{2}(1+\gamma)B}{\sqrt{T}}. \tag{22}$$

Comparing the convergence rates of Unbiased-QCS and MMSE-QCS with that of the SGD with uncompressed gradients, we observe that both achieve asymptotically the same rate of convergence $\mathcal{O}(1/\sqrt{T})$, however the constant term in the rate is slightly larger due to the compression.

Next, we consider the effect of using weighted error feedback on the convergence of the training algorithm. At the t-th iteration of SGD learning algorithm with compressed gradients and weighted error feedback, the parameters are updated as

$$\begin{aligned}
\boldsymbol{z}_t &\leftarrow \boldsymbol{g}_t + \beta \boldsymbol{r}_t \\
\widehat{\boldsymbol{z}}_t &\leftarrow \textsc{Compress}(\boldsymbol{z}_t) \\
\boldsymbol{e}_t &\leftarrow \boldsymbol{z}_t - \widehat{\boldsymbol{z}}_t \\
\boldsymbol{r}_{t+1} &\leftarrow (1-\beta)\boldsymbol{r}_t + \boldsymbol{e}_t \\
\boldsymbol{w}_{t+1} &\leftarrow \boldsymbol{w}_t - \mu \widehat{\boldsymbol{z}}_t
\end{aligned}$$

The following lemma proves the convergence of the training algorithm.

Lemma 6. *Let f^* be the minimum of objective function $f(\cdot)$ and assume* (19) *and* (20) *hold. Then,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla f(\boldsymbol{w}_t)\|_2^2\big] \le \frac{f(\boldsymbol{w}_0)-f^*}{T\mu/2} + LB\big(\mu + 4L\eta\mu^2\big)$$

where η is given by (15) *for Unbiased-QCS and by* (16) *for MMSE-QCS.*

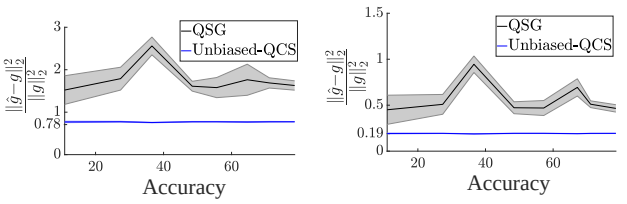
With a slightly tighter analysis and setting $\mu = \frac{\sqrt{1+\epsilon}}{\sqrt{T}}$ for arbitrary $\epsilon > 0$, we have

$$\min_t \mathbb{E}\big[\|\nabla f(\boldsymbol{w}_t)\|_2^2\big] \le \frac{f(\boldsymbol{w}_0) - f^* \; + \; \frac{L}{2}B(1+\epsilon)}{\sqrt{T}} + L^2 B \frac{(1+\epsilon)^2}{\sqrt{1+\epsilon}-1}\frac{\eta}{T}. \tag{23}$$

Comparing the convergence rates of (22) and (23) with that of SGD, we observe that the excess term in the convergence rate due to the compression of SG are proportional to $\gamma/\sqrt{T}$ and η/T, respectively, for training without and with feedback. When $\gamma \ll 1$, $\eta \approx \gamma$ and using error feedback dwarfs the effect of the compression on the convergence by an additional factor $1/\sqrt{T}$. On the other hand, for high compression gains and hence large γ, we have $\eta \approx \gamma^2$. Using error feedback reduces the term in (23) due to the compression of SG from $\mathcal{O}(1/\sqrt{T})$ to $\mathcal{O}(1/T)$, resulting in faster diminishing of the extra term and closing the gap with the SGD.

6 Experiments and Discussions

Our experiments are divided into three parts. First, we evaluate the performance of the proposed quantization methods. Next, we investigate the execution time of training with the proposed quantizers and finally, we evaluate the performance of distributed learning using different number of workers and various quantization parameters. To evaluate our algorithms, we considered a fully connected neural network with hidden layers of sizes $1000-300-100$ (herein, referred to as FC) and a Lenet-5 like convolutional network (LeCun et al. 1998) over MNIST, a convolutional network on Cifar10 (referred to as CifarNet) and Alexnet (Krizhevsky, Sutskever, and Hinton 2012) over Imagenet database. We compare QCS-SG with various communication bit-rates against the baseline (no quantization of gradients), 1-bit quantization (Seide et al. 2014), QSG (Alistarh et al. 2017) and Sparse-SGD (Wangni et al. 2018). In most cases, the experiments were repeated 10-100 times to obtain reliable results for mean and variance of the behavior of the desired quantities.



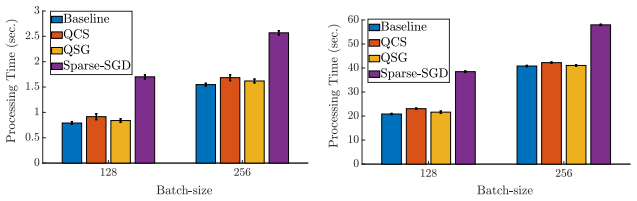
(a) compression gain = 21 (b) compression gain = 14

Figure 4: Relative quantization error vs. accuracy of model during training of Lenet over MNIST. Shaded areas represent 1σ variations.

In our implementation of QCS, we divided the gradients into partitions to reduce the complexity of the algorithm and improve its performance, similar to the approach suggested in (Alistarh et al. 2017). Depending on the size of each layer's parameters, the partition sizes were chosen to be a power of 2 or from the set $\{96, 100, 192, 200, 288, 320, 384\}$. For these choices, the Hadamard matrices are designed using Sylvester's, Payley's or Williamson's construction algorithm.

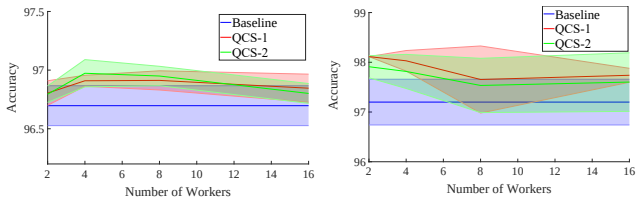
Quantizer Evaluation. To examine the effectiveness of the quantization scheme, we measured the relative quantization error, defined as $\frac{\|\boldsymbol{g}-\tilde{\boldsymbol{g}}\|_2^2}{\|\boldsymbol{g}\|_2^2}$, for different models, datasets and with different number of quantization levels. Figure 4 compares the relative quantization error of Unbiased-QCS against QSG (Alistarh et al. 2017) during training of Lenet over MNIST for different compression gains. We have observed similar behavior with other models and at different compression gains. The results confirm our findings in Thm. 2 and theoretical comparisons in Fig. 3. It is worth noting that unlike QSG, the relative quantization error of QCS is highly concentrated around the mean value. This suggest that *training with QCS-SG is similar to training with unquantized SG corrupted by a (Gaussian) noise with fixed signal to noise ratio.*

Processing Time. We measured the required time to compute and quantize the gradients for processing 100 batches of training data using different batch-sizes (not accounting for loading data from HDD or communicating among workers) and compared with the baseline (no quantization), QSG and Sparse-SGD over a Titan Xp GPU. Figure 5 shows the results for different batch-sizes per worker. We note that although the compression gain of our proposed QCS can become arbitrarily large, its processing time is only slightly higher that QSG and much lower than the random sparsification (Wangni et al. 2018).



(a) Cifarnet (b) Alexnet

Figure 5: Time to process and compress SG for 100 batches



(a) FC (b) Lenet

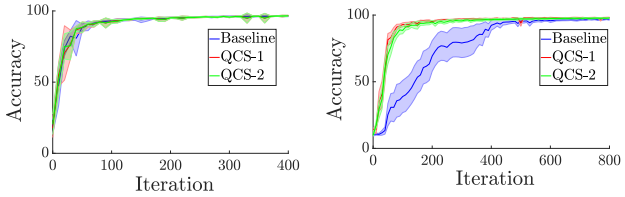
Figure 6: Accuracy of distributed training vs number of workers, using SGD learning algorithm

As an example, 100 iterations of decentralized distributed training of Alexnet with 4 workers, batchsize 128 per worker using Titan Xp GPUs connected via InfiniBand links would take approximately 22 seconds employing QCS with compression gain 100, compared to 27 seconds by QSG (compression gain of approximately 21), 42 seconds by Sparse-SGD (with compression gain of 100), and 55 seconds by Baseline (no SG compression), while centralized single node training with the same total batch-size takes approximately 90 seconds to execute.

It is worth noting that as the models become more complex and the number of parameters increases, the overhead of applying transforms to the partitions of SG, which have small size $d < 500$, becomes negligible relative to the computational complexity of the backpropagation algorithm. Hence, the more desirable properties of QCS and its relatively negligible overhead compared to QSG and other quantization or sparsification methods make QCS a favorable choice for distributed learning of large deep models.

Performance in Distributed Deep Learning. We evaluate the convergence and the number of communication bits in a distributed learning system with different number of workers. In our simulations, the batch-size per worker is fixed at 128. Hence, by increasing the number of workers, the effective total batch-size increases. Although it is possible to evaluate the performance of the quantization and compression schemes in both synchronous and asynchronous settings, here we assume that the workers and server are synchronous. The main reason for such a setting is to cancel-out the performance degradation (in terms of training accuracy or speed) that may be caused by the stale gradients in asynchronous updates, and to solely investigate the effect of the quantization/compression algorithms.

We consider two different settings: QCS-1 achieves compression gain of approximately 32 by optimally setting k and Q (see Thm. 2 and the discussion after), and in QCS-2 $k = n$ and $Q = 1$. Hence, QCS-2 achieves the same compression gain as QSG. We evaluate the performance of distributed training using the proposed compression method with both stochastic gradient descent and Adam learning algorithms. We use initial learning rate of 0.05 for SGD and 0.001 for Adam, both using a decay rate of 0.99 per epoch. Figure 6 shows the accuracy of the final trained model vs different number of workers for FC and Lenet models, using QCS-1 and QCS-2 and compares them with the baseline. Moreover, in Figures 7 and 8, we have compared the convergence rate of QCS w.r.t. baseline (no quantization) for different settings. It is interesting to note that QCS improves the convergence rate of the training as well as the final accuracy in some occasions compared to the baseline (no quantization). We believe this is mainly due to the characteristics of the quantization noise. Since the noise from the QCS behaves similar to a (Gaussian) noise with fixed signal to noise ratio, our method is likely to result in a better convergence property than the aforementioned techniques for complex training data (Neelakantan et al. 2015; Noh et al. 2017).



(a) FC (b) Lenet

Figure 7: Convergence rate of distributed training of FC and Lenet models, 4 workers with SGD learning algorithm

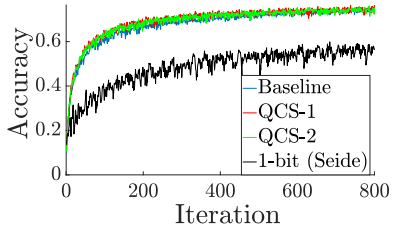


Figure 8: Convergence rate of distributed training of CifarNet, 4 workers with Adam learning algorithm

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number CPS-1837369 and by Sony Inc. under the Sony Faculty Research Award.

References

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

Abdi, A., and Fekri, F. 2019b. Reducing communication overhead via ceo in distributed training. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5. IEEE.

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

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

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

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

Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Senior, A.; Tucker, P.; Yang, K.; Le, Q. V.; and Others. 2012. Large scale distributed deep networks. In *Advances in neural information processing systems*, 1223–1231.

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

Gray, R. M., and Stockham, T. G. 1993. Dithered quantizers. *IEEE Transactions on Information Theory* 39(3):805–812.

Ho, Q.; Cipar, J.; Cui, H.; Lee, S.; Kim, J. K.; Gibbons, P. B.; Gibson, G. A.; Ganger, G.; and Xing, E. 2013. More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server. In *Advances in Neural Information Processing Systems 26*, 1223–1231.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, B.; and Dally, W. J. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations*, 1–13.

Neelakantan, A.; Vilnis, L.; Le, Q. V.; Sutskever, I.; Kaiser, L.; Kurach, K.; and Martens, J. 2015. Adding gradient noise improves learning for very deep networks. *arXiv preprint* 1–11.

Niu, F.; Recht, B.; Ré, C.; and Wright, S. 2011. Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems 24*, 693–701.

Noh, H.; You, T.; Mun, J.; and Han, B. 2017. Regularizing Deep Neural Networks by Noise: Its Interpretation and Optimization. In *Advances in Neural Information Processing Systems*, number Nips, 5109–5118.

Schuchman, L. 1964. Dither signals and their effect on quantization noise. *IEEE Transactions on Communication Technology* 12(4):162–165.

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

Stich, S. U.; Cordonnier, J.-B.; and Jaggi, M. 2018. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, number NeurIPS, 4452–4463.

Stich, S. U. 2019. Local SGD Converges Fast and Communicates Little. In *ICLR*, 1–12.

Strom, N. 2015. Scalable distributed DNN training using commodity GPU cloud computing. In *INTERSPEECH*, volume 7, 10.

Wang, H.; Sievert, S.; Charles, Z.; Liu, S.; Wright, S.; and Papailiopoulos, D. 2018. ATOMO: Communication-efficient Learning via Atomic Sparsification. In *Advances in Neural Information Processing Systems 31*, 9850–9861. Curran Associates, Inc.

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

Wen, W.; Xu, C.; Yan, F.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H. 2017. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. In *TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning*, 1509–1519. Curran Associates, Inc.

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

Zhang, H.; Hsieh, C. J.; and Akella, V. 2016. HogWild++: A New Mechanism for Decentralized Asynchronous Stochastic Gradient Descent. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 629–638.