Quantized Stochastic Gradient Descent: Communication versus Convergence

Anonymous Author(s)
Affiliation
Address
email

Abstract

Parallel implementations of stochastic gradient descent (SGD) have received significant research attention, thanks to excellent scalability properties of this algorithm, and to its efficiency in the context of training deep neural networks. A fundamental barrier for parallelizing large-scale SGD is the fact that the cost of communicating the gradient updates between nodes can be very large. Consequently, lossy compression heuristics have been proposed, by which nodes only communicate quantized gradients. Although effective in practice, these heuristics do not always provably converge, and it is not clear whether they are optimal. In this paper, we propose *Quantized SGD (QSGD)*, a family of compression schemes which allow the compression of gradient updates at each node, while guaranteeing convergence under standard assumptions. QSGD allows the user to trade off compression and convergence time: it can communicate a *sublinear* number of bits per iteration in the model dimension, and can achieve asymptotically optimal communication cost. We complement our theoretical results with empirical data, showing that QSGD can significantly reduce communication cost, while being competitive with standard uncompressed techniques on a variety of real tasks.

1 Introduction

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a known convex set, and consider stochastic gradient descent for a smooth function $f : \mathcal{X} \to \mathbb{R}$, in which we only have access to independent stochastic gradients of f. We assume that stochastic gradient $\tilde{g}(\boldsymbol{x})$ is unbiased $\mathbb{E}[\widetilde{g}(\boldsymbol{x})] = \nabla f(\boldsymbol{x})$ and satisfies the second moment condition $\mathbb{E}[\|\widetilde{g}(\boldsymbol{x})\|_2^2] \leq B$ for all $\boldsymbol{x} \in \mathcal{X}$.

Now consider a synchronous parallel stochastic gradient descent setting in which we have K workers, each of which have access to independent stochastic gradients of f. Each worker computes the stochastic gradient synchronously and communicates the gradients with each other. After the communication, each worker updates the parameter using the aggregated gradient as

$$\boldsymbol{x}_{t+1} = \Pi_{\mathcal{X}} \left(\boldsymbol{x}_t - \frac{\eta_t}{K} \sum_{\ell=1}^{K} \widetilde{g}^{\ell}(\boldsymbol{x}_t) \right) ,$$

where $\widetilde{g}^{\ell}(\boldsymbol{x}_t)$ is the stochastic gradient computed on the ℓth worker.

One can easily imagine that when the number of parameters n is large, the cost of communication can be significant. 1-Bit SGD [5] addresses this issue by introducing a quantization function that roughly speaking encodes a gradient vector into one bit for each coordinate corresponding to its sign and two float numbers corresponding to the mean of the positive coordinates and the mean of the

Submitted to 29th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

negative coordinates. On the receiver's side the gradient vector can be approximately recovered by setting all the positive coordinates to the positive mean and the negative coordinates as the negative mean. This heuristic requires roughly n bits per gradient, but does not guarantee convergence.

2 A Randomized Quantization Scheme

We propose the random quantization function $Q(\boldsymbol{v})$ defined as follows:

$$Q_i(\boldsymbol{v}) = \|\boldsymbol{v}\|_2 \cdot \mathrm{sgn}\,(\boldsymbol{v}_i)\,\xi_i(\boldsymbol{v})\ , \tag{1}$$

where $\xi_i(\boldsymbol{v})$'s are independent random variables such that $\xi_i(\boldsymbol{v}) = 1$ with probability $|\boldsymbol{v}_i|/\|\boldsymbol{v}\|_2$, and $\xi_i(\boldsymbol{v}) = 0$, otherwise. If $\boldsymbol{v} = \mathbf{0}$, we define $Q(\boldsymbol{v}) = \mathbf{0}$.

The key properties of $Q[\tilde{g}(\boldsymbol{x})]$ are sparsity, unbiasedness, and bounded second moment as shown in the following lemma:

Lemma 2.1. *For any $\boldsymbol{v} \in \mathbb{R}^n$, we have $\mathbb{E}[\|Q(\boldsymbol{v})\|_0] \leq \sqrt{n}$ (sparsity), $\mathbb{E}[Q(\boldsymbol{v})] = \boldsymbol{v}$ (unbiasedness), and $\mathbb{E}[\|Q(\boldsymbol{v})\|^2] \leq \sqrt{n}\|\boldsymbol{v}\|_2^2$ (second moment bound).*

The sparsity allows us to succinctly encode $Q(\boldsymbol{x})$, for any $\boldsymbol{x}$, in expectation. The information contained in $Q(\boldsymbol{v})$ can be expressed by (1) a float variable that encodes the value of $\|\boldsymbol{v}\|_2$, (2) identities of the vector coordinates i for which $\xi_i(\boldsymbol{v}) = 1$, and (3) the values of signs $\mathrm{sgn}\,(v_i)$ for these coordinates. Let $\mathrm{Code}(Q(\boldsymbol{v}))$ denote a binary representation of such a tuple representation of $Q(\boldsymbol{v})$. Then, one can show the following bound, whose proof is deferred to the full version of our paper.

Lemma 2.2. *For every vector $\boldsymbol{v} \in \mathbb{R}^n$, we have $\mathbb{E}[|\mathrm{Code}(Q(\boldsymbol{v}))|] \leq \sqrt{n}(\log(n) + \log(2e)) + F$, where F is the number of bits for representing one floating point number.*

These two lemmas together imply the following theorem.

Theorem 2.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be fixed, and let $\boldsymbol{x} \in \mathbb{R}^n$ be arbitrary. If $\widetilde{g}(\boldsymbol{x})$ is a stochastic gradient for f at $\boldsymbol{x}$ with second moment bound B, then $Q(\widetilde{g}(\boldsymbol{x}))$ is a stochastic gradient for f at $\boldsymbol{x}$ with second moment bound $\sqrt{n}B$. Moreover, in expectation $Q(\widetilde{g}(\boldsymbol{x}))$ can be communicated using $\sqrt{n}(\log n + \log 2e) + F$ bits.*

Note that the above communication cost is *sublinear* in the dimension n compared to the linear cost when the gradients are communicated without compression or using 1-bit SGD. Using standard convergence results (e.g., [1]), we obtain an stochastic gradient algorithm that requires only $O(\sqrt{n}\log n)$ communication per round and converges to the same precision in $O(\sqrt{n})$ times more iterations.

We can control the trade-off between communication and convergence by introducing *bucketing*. More precisely, we partition the gradient vector into n/d buckets, each of which containing d consecutive coordinates, and apply the quantization and encoding to each bucket. Then a simple extension of Theorem 2.3 predicts that the second moment bound then becomes $\sqrt{d}B$. Setting $d = 1$, we recover no quantization (vanilla SGD), and $d = n$ corresponds to full quantization. However, since quantization increases the second moment bound, there is no improvement in terms of the total communication cost.

3 A Generalized Randomized Quantization Scheme

In order to explore the trade-off between communication and convergence more carefully, we propose a more general lossy-compression scheme defined as follows:

$$Q_i(\boldsymbol{v}, s) = \|\boldsymbol{v}\|_2 \cdot \mathrm{sgn}\,(v_i)\,\xi_i(\boldsymbol{v}, s)\ , \tag{2}$$

where $s \geq 1$ is a tuning parameter, $\xi_i(\boldsymbol{v}, s)$'s are independent random variables with distributions defined as follows. Let $0 \leq \ell < s$ be an integer such that $|v_i|/\|\boldsymbol{v}\|_2 \in [\ell/s, (\ell+1)/s]$. Then

$$\xi_i(\boldsymbol{v}, s) = \begin{cases} \ell/s & \text{with probability } 1 - p\left(\frac{|v_i|}{\|\boldsymbol{v}\|_2}, s\right); \\ (\ell+1)/s & \text{otherwise.} \end{cases}$$

Here, $p(a,s) = as - \ell$ for any $a \in [0,1]$. If $\boldsymbol{v} = \mathbf{0}$, then we define $Q(\boldsymbol{v}, s) = \mathbf{0}$.

The random quantization function (1) corresponds to the special case $s = 1$. We obtain the three key properties as we show in the following lemma.

Lemma 3.1. *For any $\boldsymbol{v} \in \mathbb{R}^n$, we have that $\mathbb{E}[\|Q(\boldsymbol{v}, s)\|_0] \le s^2 + \sqrt{n}$ (sparsity), $\mathbb{E}[Q(\boldsymbol{v}, s)] = \boldsymbol{v}$ (unbiasedness), and $\mathbb{E}[\|Q(\boldsymbol{v}, s)\|_2^2] \le (1 + \min(n/s^2, \sqrt{n}/s))\|\boldsymbol{v}\|_2^2$ (second moment bound).*

Note that the factor $c_{n,s} := 1 + \min(n/s^2, \sqrt{n}/s)$ in the second-moment bound is parameterized with the dimension n and the tuning parameter s. For the special case $s = 1$, we have $c_{n,s} = \Theta(\sqrt{n})$, which is consistent with the result in Lemma 2.1. By varying the value of the parameter s between 1 and $\sqrt{n}$, we can smoothly vary $c_{n,s}$ between $\Theta(\sqrt{n})$ and $\Theta(1)$. We can also see that as we increase s, the quantized gradient becomes less sparse. The sparsity bound is $O(\sqrt{n})$ at $s = 1$ and $O(n)$ at $s = \sqrt{n}$. We also note that the distribution of $\xi_i(\boldsymbol{v}, s)$ is a unique distribution that has minimal variance over distributions that have support $\{0, 1/s, \ldots, 1\}$ and unbiased.

In the sparse regime where we expect the quantized gradient to contain at most $n/2$ non-zero coordinates, we have the following theorem.

Theorem 3.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be fixed, and let $\boldsymbol{x} \in \mathbb{R}^n$ be arbitrary. If $\widetilde{g}(\boldsymbol{x})$ is a stochastic gradient for f at $\boldsymbol{x}$ with second moment bound B, then $Q_s(\widetilde{g}(\boldsymbol{x}))$ is a stochastic gradient for f at $\boldsymbol{x}$ with second moment bound $\left(1 + \min\left(n/s^2, \sqrt{n}/s\right)\right) B$. Moreover, there is an encoding scheme so that in expectation, the number of bits needed to communicate $Q_s(\widetilde{g}(\boldsymbol{x}))$ is upper bounded by*

$$F + \left(3 + \frac{3}{2} \cdot (1 + o(1)) \log\left(\frac{2(s^2+n)}{s^2 + \sqrt{n}}\right)\right)(s^2 + \sqrt{n}) \, .$$

The communication cost can be, roughly speaking, broken down into one float number representing the norm and $s^2 + \sqrt{n}$ (in expectation) bits and integers representing the signs, magnitudes, and positions of the non-zero coordinates. We use the *recursive Elias coding*, which is favorable for small integers, to achieve the above bound.

For large s, the quantized gradient becomes dense, and we no longer need to communicate the positions of the non-zero coordinates.

Theorem 3.3. *Let $f, \boldsymbol{x}$, and $\widetilde{g}(\boldsymbol{x})$ be as in Theorem 3.2. There is an encoding scheme for $Q_s(\widetilde{g}(\boldsymbol{x}))$ which in expectation has length*

$$F + \left(\frac{1+o(1)}{2}\left(\log\left(1 + \frac{s^2 + \min(n, s\sqrt{n})}{n}\right) + 1\right) + 2\right) n \, .$$

In particular, if $s = \sqrt{n}$, then this encoding requires $\le F + 2.8n$ bits in expectation.

In fact, for $s = \sqrt{n}$, the second moment bound is only 2 times worse than no quantization and the communication cost is only 2.8 bits per coordinate. We defer the description of the quantization schemes in Theorems 3.2 and 3.3, and their use in the context of SVRG [2], to the full version.

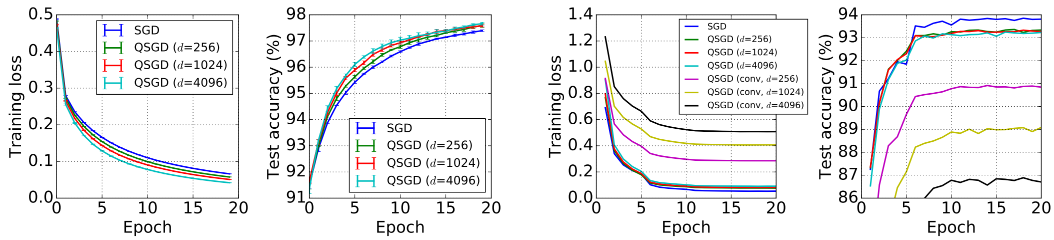
4 Experiments

We now empirically validate our approach, using experiments aimed at data-parallel and model-parallel settings. We have implemented QSGD on GPUs using deep learning framework Chainer [7].

Quantization vs. Accuracy. In the first set of experiments, we explore the relation between performance and the granularity at which quantization is applied to the gradient vector.

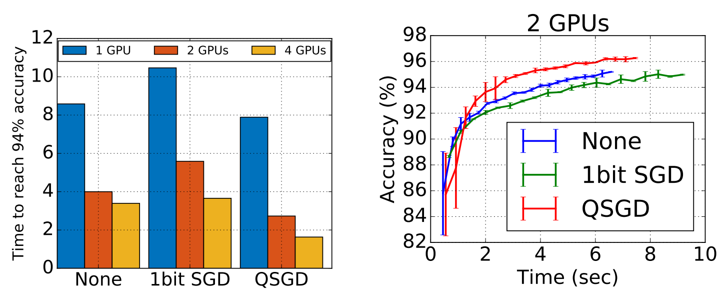
Here, our experiments deviate from the theory, as we use a deep network, with non-convex objective.

MNIST dataset. The first dataset is the MNIST dataset of handwritten digits. The training set consists of 60,000 28 x 28 single digit images. The test set consists of 10,000 images. We train a two-layer perceptron with 4096 hidden units and ReLU activation with a minibatch size of 256 and step size of 0.1. Results are shown in Figure 1(a). Rather surprisingly, in terms of both training negative log-likelihood loss and the test accuracy, QSGD *improves* performance. This is consistent with recent work [4] suggesting benefits of added noise in training deep networks. We observed no such improvement for a linear model on the same dataset.



(a) MNIST (b) CIFAR-10

Figure 1: Training on a single machine on MNIST and CIFAR-10. SGD corresponds to bucket size of $d = 1$. QSGD performs better in terms of both training loss and test accuracy on the MNIST dataset.



(a) Data parallel training time (b) Data parallel accuracy

Figure 2: Multi GPU experiment.

The total number of parameters of this model is 3.3 million, most of them lying in the first layer. Using Theorem 2.2, we can approximate the effective number of floats communicated by QSGD. Assuming $F = 32$, we get roughly 88k, 49k, and 29k effective floats for bucket sizes $d = 256, 1024$, and 4096, respectively. There is a massive reduction in communication since for each bucket we only need to communicate one float and the positions and signs of $\tilde{O}(\sqrt{d})$ entries, each of which only requires $O(\log d)$ bits, which is typically much smaller than 32 (e.g., 11 bits for $d = 256$).

CIFAR-10 dataset. Next, we consider the CIFAR-10 object classification dataset [3]. The original training set consists of 50,000 32×32 color images, augmented by translating, cropping with window size 28×28, and horizontal flipping. The augmented training set contains 1.8 million images.

We use a small VGG model [6] consisting of nine 2D convolution layers and three fully connected layers. The total number of parameters is roughly 22 million. All methods used momentum of 0.9. See the full paper for the details. When we only quantized the fully connected layers, we have found that the bucket size can be increased without much loss in accuracy (see Fig. 1(b)). The effective number of floats to be communicated are 1.5 million, 1.3 million, and 1.2 million for bucket sizes $d = 256, 1024$, and 4096, respectively. On the other hand, when we also applied the quantization to the convolutional layers, we observed a noticeable increase in the training objective as well as reduction in the test accuracy. The effective number of floats to be communicated are 580k, 312k, 176k, respectively.

Parallelization. In Figure 2 (a) and (b), we show preliminary scalability experiments on MNIST, using up to 4 GPUs, compared with vanilla SGD and 1-Bit SGD [5]. The setup is the same as in the previous section, and we use double buffering [5] to perform communication and quantization concurrently with the computation. Experiments are preliminary in the sense that we did not fully optimize either 1-Bit SGD or QSGD to their full potential; in particular, quantized gradients are communicated in raw floats instead of using more efficient encoding.

References

[1] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[2] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, 2013.

[3] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009.

[4] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[5] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *INTERSPEECH*, 2014.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] S. Tokui, K. Oono, S. Hido, C. San Mateo, and J. Clayton. Chainer: a next-generation open source framework for deep learning.