Indirect Stochastic Gradient Quantization and Its Application 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

Transmitting the gradients or model parameters is a critical bottleneck in distributed training of large models. To mitigate this issue, we propose an *indirect* quantization and compression of stochastic gradients (SG) via factorization. The gist of the idea is that, in contrast to the direct compression methods, we focus on the factors in SGs, i.e., the forward and backward signals in the backpropagation algorithm. We observe that these factors are correlated and generally sparse in most deep models. This gives rise to rethinking of the approaches for quantization and compression of gradients with the ultimate goal of minimizing the error in the final computed gradients subject to the desired communication constraints. We have proposed and theoretically analyzed different indirect SG quantization (ISGQ) methods. The proposed ISGQ reduces the reconstruction error in SGs compared to the direct quantization methods with the same number of quantization bits. Moreover, it can achieve compression gains of more than 100, while the existing traditional quantization schemes can achieve compression ratio of at most 32 (quantizing to 1 bit). Further, for a fixed total batch-size, the required transmission bit-rate per worker decreases in ISGQ as the number of workers increases.

1 Introduction

In recent years, the size of deep learning problems is increased significantly both in terms of the number of available training samples as well as the number of model's parameters. However scaling up of neural networks requires massive amounts of storage, memory and computational power for training. As such, large-scale distributed machine learning in which the training samples are distributed among different repository or processing units (referred to as workers) has started to be a viable approach for tackling the memory, storage and computational constraints.

Since most common deep learning algorithms are based on computing the gradients, in this paper, we focus on parallel (distributed) computation of stochastic gradients (SG). The requirement to exchange the locally computed SGs incurs significant communication overhead which is a major bottleneck in distributed learning. Variety of approaches have been proposed to mitigate the communication bottleneck by reducing the overall transmission rate such as:

1. *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 such as (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 different quantization levels and/or adaptive quantizers, one can alleviate such issues (Dryden et al. 2016). Stochastic quantizers such as QSGD (Alistarh et al. 2017), TernGrad (Wen et al. 2017) and Dithered qunatization (Abdi and Fekri 2019a; 2019b) are alternative unbiased quantization approaches with performance guarantees which provide a trade-off between the gradient precision and the model accuracy.
2. *Sparsification*. Another approach is 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 further 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 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) and random (stochastic) sparsification of the gradients (Wangni et al. 2018).

Contribution. In this paper, we focus on the quantization of SGs. The main drawbacks of the existing quantization methods are the limited compression gain of at most 32 (quantizing 32 bit floating point number to only 1 bit), and scalability as the total transmission bits increases almost linearly with the number of workers. To overcome these issues, we observe that the cost function of a neural network w.r.t. the parameters of a layer, $\boldsymbol{W}$, can be reformulated as $\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{W}\boldsymbol{x})]$ where $\boldsymbol{x}$ is the 'virtual' input of that layer. Therefore, we first consider the SG quantization of this class of functions and develop a new algorithm, *indirect stochastic*

gradient quantization via factorization (ISGQ). Then, we study its complexity and convergence properties and extend the algorithm to distributed training of deep neural networks. By analyzing the signals propagating in the neural networks, we observe that the forward and backward signals in neural networks are more compression-friendly than the stochastic gradients, themselves. Hence, ISGQ can achieve superior performance in terms of total transmission bits and quantization error compared to the traditional approaches.

Our proposed approach is different from the existing low rank matrix approximation methods such as (Konecný et al. 2016) in the sense that those methods enforce the updates of SG to be as $\boldsymbol{G} = \boldsymbol{AB}$ with generally $\boldsymbol{A}$ (or $\boldsymbol{B}$) being generated randomly and fixed at each iteration of training. Therefore, essentially the updates of the SG are forced to be in the subspace generated by $\boldsymbol{A}$. Another related line of research is optimizing quantizers as in ZipML (Zhang et al. 2017). However, it is primarily developed for linear regression with ℓ_2 loss and is based on double sampling strategy for random quantization of the input data. Moreover, optimizing the quantizer is based on a dynamic programming algorithm requiring access to the entire dataset or knowing its statistical properties. In contrast to these approaches, our proposed method does not enforce any constraint on the SGs and it exploits the natural decomposition inherent in the calculation of SG. Further, the ISGQ is developed for general (activation) functions, has negligible computational overhead and can be directly applied to distributed deep learning.

2 Problem Statement and Motivation

Consider the problem of learning the parametric model $\mathfrak{m}_{\boldsymbol{W}} : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of the inputs and outputs of the model, respectively, and $\boldsymbol{W}$ is the parameters of the model to be learned. Let the cost function associated with the model's output $\boldsymbol{y} = \mathfrak{m}_{\boldsymbol{W}}(\boldsymbol{x})$ and the desired output $\boldsymbol{t}$ be given by $\ell(\boldsymbol{y}, \boldsymbol{t})$. Hence, $\mathcal{L}(\boldsymbol{W}) = \mathbb{E}[\ell(\mathfrak{m}_{\boldsymbol{W}}(\boldsymbol{x}), \boldsymbol{t})]$ is the objective function to be minimized for learning $\boldsymbol{W}$.

In this paper, we consider efficient quantization of the stochastic gradients (SG) of $\mathcal{L}(\cdot)$. First we consider the class of generalized linear functions as the parametric model $\mathfrak{m}_{\boldsymbol{W}}(\cdot)$ and develop and theoretically analyze our proposed quantization. Then, we extend our algorithm and results to the distributed training of general deep models in § 4.

Let the parametric model be given as $\mathfrak{m}_{\boldsymbol{W}}(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{W}\boldsymbol{x})$, where $\boldsymbol{W} \in \mathcal{W} \subset \mathbb{R}^{m\times n}$, $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ and $\boldsymbol{\sigma}(\cdot)$ is an arbitrary function. Let $f(\boldsymbol{W}\boldsymbol{x}, t) = \ell(\boldsymbol{\sigma}(\boldsymbol{W}\boldsymbol{x}), t)$ be the cost for input $\boldsymbol{x}$ and target $\boldsymbol{t}$, assumed to be an arbitrary smooth differentiable function. To simplify the notations, since $\boldsymbol{t}$ is uniquely determined from $\boldsymbol{x}$ and the dataset, we ignore it in our notations and denote the cost function as $f(\boldsymbol{W}\boldsymbol{x})$. Hence, the objective is minimizing $\mathcal{L}(\boldsymbol{W}) = \mathbb{E}[f(\boldsymbol{W}\boldsymbol{x})]$.

A stochastic gradient (SG) of $\mathcal{L}(\boldsymbol{W})$ is an unbiased random estimator of the gradient, i.e., the SG $\boldsymbol{G}(\boldsymbol{W})$ is a random function such that $\mathbb{E}[\boldsymbol{G}(\boldsymbol{W})] = \nabla_{\boldsymbol{W}}\mathcal{L}$ for all $\boldsymbol{W}$. $\boldsymbol{G}$ has bounded variance if there exists a finite B such that $\mathbb{E}\big[\|\boldsymbol{G}(\boldsymbol{W}) - \nabla\mathcal{L}\|_F^2\big] \le B$. For an arbitrary $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{G} = \nabla_{\boldsymbol{W}} f(\boldsymbol{W}\boldsymbol{x}) = \nabla_{\boldsymbol{y}} f(\boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{W}\boldsymbol{x}}\, \boldsymbol{x}^\mathsf{T}$ is a SG of $\mathcal{L}$. It is common to compute and average the SG over a mini-batch to reduce its variance. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L] \in \mathbb{R}^{n\times L}$ be a training batch of size L, $\boldsymbol{\delta}_k = \nabla_{\boldsymbol{y}} f(\boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{W}\boldsymbol{x}_k}$ for $k = 1, \ldots, L$ and $\boldsymbol{\Delta} = [\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_L] \in \mathbb{R}^{m\times L}$. Therefore,

$$\boldsymbol{G} = \frac{1}{L}\sum_{k=1}^{L} \boldsymbol{G}_k = \frac{1}{L}\sum_{k=1}^{L} \boldsymbol{\delta}_k \boldsymbol{x}_k^\mathsf{T} = \frac{1}{L}\boldsymbol{\Delta}\boldsymbol{X}^\mathsf{T}. \tag{1}$$

Our proposed method for quantization and compression of the stochastic gradients, computed via (1), is motivated by the following observation:

Instead of computing the gradients and then compressing them, our idea aims at compressing the intermediate signals, $\boldsymbol{\Delta}$ and $\boldsymbol{X}$, and transmitting them. We refer to this approach as indirect *compression, in contrast to the* direct *quantization and compression of the stochastic gradients $\boldsymbol{G}$. This is specially helpful when the number of parameters is large relative to the batch size; since the dimension of SG is $m \times n$, direct method requires transmission of mn values for $\boldsymbol{G}$. On the other hand, the indirect method requires transmitting only $L(m + n)$ values for a batch of size L. Moreover, as it will be investigated later, these signals are more compression-friendly, i.e., they tend to be sparser and having less entropy than the stochastic gradients.*

3 Indirect SG Quantization via Factorization

Here, we introduce and analyze the proposed indirect quantization of SG. Let $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{\Delta}}$ be the quantized values of $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, respectively. Then the indirect SG quantization (ISGQ) is defined as

$$\widetilde{\boldsymbol{G}} = \frac{1}{L}\widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{X}}^\mathsf{T}. \tag{2}$$

In this paper, we focus on unbiased indirect quantizers, i.e., $\mathbb{E}\big[\boldsymbol{G} - \frac{1}{L}\widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{X}}^\mathsf{T}\big] = \boldsymbol{0}$. We consider two classes of quantizers for $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, namely, *deterministic* and *random dithered* quantization.

Deterministic Indirect SG Quantization

We call a quantizer $Q(\cdot)$ *deterministic* if for any v, repeated application of the quantizer to v results in the same quantized value. A quantizer $Q(\cdot)$ is statistically optimized for random variable z if it is unbiased and has the minimum mean squared error (MSE) (Max 1960; Lloyd 1982), hence

$$\mathbb{E}_z[z - Q(z)] = 0, \quad \mathbb{E}_z[(Q(z) - z)\, Q(z)] = 0. \tag{3}$$

Obviously, designing such a quantizer requires knowledge about the probability distribution of data or accessing the entire dataset.

Let $g = G_{i,j}$ be an arbitrary element of the SG, $\boldsymbol{x} := (\boldsymbol{X}_{j,\cdot})^\mathsf{T}$ and $\boldsymbol{\delta} := (\boldsymbol{\Delta}_{i,\cdot})^\mathsf{T}$ be the j-th and i-th row of $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, respectively. Hence, $g = \frac{1}{L}\boldsymbol{\delta}^\mathsf{T}\boldsymbol{x} = \frac{1}{L}\sum_k x_k\, \delta_k$. Further, assume that the signals have bounded joint second moment, i.e., $\mathbb{E}\big[\|\boldsymbol{x}\|^2\|\boldsymbol{\delta}\|^2\big] < \infty$.

One may hope that if the quantizers for $\boldsymbol{x}$ and $\boldsymbol{\delta}$ are designed optimally w.r.t. each individual signal, then the resulting indirect quantization of SG becomes almost optimal as well. We refer to this quantization approach as *naïve ISGQ*.

Lemma 1. *Assume that the quantizers $\boldsymbol{x}$ and $\boldsymbol{\delta}$ are designed optimally and $\tilde{g}$ is the naïve indirect quantization of g.*

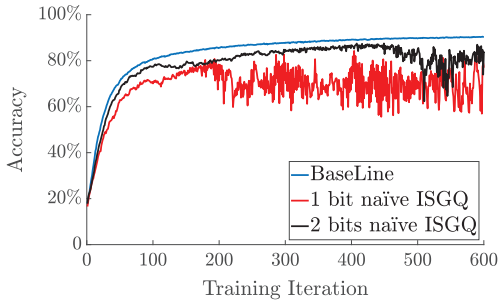


Figure 1: Performance of naïve-ISGQ w.r.t. the baseline (non-quantized) for training a fully connected model over MNIST.

– *If $\boldsymbol{x}$ and $\boldsymbol{\delta}$ are independent random variables, then $\tilde{g}$ is an unbiased and bounded-variance SG. Moreover, in 1-bit quantization, if x_k's are i.i.d. Folded Normal and δ_k's are Normal random variables, then the MSE gap with the optimum direct quantizer is less than* 4%.

– *If $\boldsymbol{x}$ and $\boldsymbol{\delta}$ are correlated random variables, the naïve ISGQ is not necessarily unbiased.*

Unfortunately, designing optimum individual quantizers for $\boldsymbol{x}$ and $\boldsymbol{\delta}$ is not feasible in many applications. Further, the independence assumption between $\boldsymbol{x}$ and $\boldsymbol{\delta}$ is not generally satisfied in practice and by Lemma 1, the naïve ISGQ is likely to become biased. These shortcomings limit the effectiveness of naïve ISGQ in many applications such as distributed deep learning (see Fig. 1).

The drawbacks of naïve-ISGQ are mainly due to the fact that the quantizers for the signals are designed independently, i.e., the quantized signals $\tilde{x}$ and $\tilde{\delta}$ are obtained by minimizing $\mathbb{E}\big[(x-\tilde{x})^2\big]$ and $\mathbb{E}\Big[(\delta-\tilde{\delta})^2\Big]$ separately without considering their joint effect on the computed SG. To overcome the problems of naïve ISGQ, we propose jointly optimizing the individual quantizers for $\boldsymbol{X}$ and $\boldsymbol{\Delta}$ such that the MSE of the resulting ISGQ is minimized. If the joint statistical properties of $\boldsymbol{X}$ and $\boldsymbol{\Delta}$ are available, one can aim at analytically finding optimum individual quantizers for unbiased minimum MSE ISGQ (please refer to the supplementary document). Here, we focus on empirical methods (using data of each training mini-batch) to approximately find good indirect quantizers.

Note that the quantization of $\boldsymbol{X}$ can be written as $\widetilde{\boldsymbol{X}} = \sum_{k=1}^{K} \boldsymbol{A}_k \alpha_k$, where K is the number of quantization bins, $[\boldsymbol{A}_k]_{i,j} = 1$ if $[\boldsymbol{X}]_{i,j}$ is in the k-th quantization bin (and $[\boldsymbol{A}_k]_{i,j} = 0$, otherwise) and α_k is the reconstruction point associated with the k-th bin. Similarly, we can represent quantization of $\boldsymbol{\Delta}$ as $\widetilde{\boldsymbol{\Delta}} = \sum_k \boldsymbol{B}_k \beta_k$. Therefore, ISGQ can be computed as

$$\widetilde{\boldsymbol{G}} = \frac{1}{L}\widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{X}}^{\mathsf{T}} = \frac{1}{L}\sum_{k,l} \boldsymbol{B}_l \boldsymbol{A}_k^{\mathsf{T}}\, \alpha_k\beta_l = \sum_{k,l} \boldsymbol{C}_{k,l}\, \alpha_k\beta_l. \quad (4)$$

where $\boldsymbol{C}_{k,l} = \frac{1}{L}\boldsymbol{B}_l\boldsymbol{A}_k^{\mathsf{T}}$. Define the empirical bias as

Algorithm 1 Empirical MSE-ISGQ

1: Initialize $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
2: **for** few iterations **do**
3: Fix $\boldsymbol{\alpha}$ and solve (6) to update $\boldsymbol{\beta}$.
4: Fix $\boldsymbol{\beta}$ and solve (6) to update $\boldsymbol{\alpha}$.
5: **return** Quantizers for $\boldsymbol{X}$ and $\boldsymbol{\Delta}$.

$$\begin{aligned}\text{bias} &:= \sum_{i,j}\big(G_{i,j} - \frac{1}{L}[\widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{X}}^{\mathsf{T}}]_{i,j}\big)\\ &= \sum_{i,j}\big[\boldsymbol{G} - \sum_{k,l}\boldsymbol{C}_{k,l}\alpha_k\beta_l\big]_{i,j} = \bar{G} - \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{\alpha}, \quad (5)\end{aligned}$$

where $\bar{G} = \sum_{i,j} G_{i,j}$ and $P_{k,l} = \sum_{i,j}[\boldsymbol{C}_{k,l}]_{i,j}$. Since the problem of optimizing the quantization bins for ISGQ is non-convex and computationally complex, we decide to fix them and only adjust the reconstruction points of each quantizer. Hence, the mappings $\boldsymbol{X} \mapsto \boldsymbol{A}_k$ and $\boldsymbol{\Delta} \mapsto \boldsymbol{B}_k$ are known. For example, in 1-bit ISGQ for correlated normal $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, the quantization threshold is set to zero and only the reconstruction values for positive and negative $\boldsymbol{X}$ and $\boldsymbol{\Delta}$ are adjusted. We propose to adjust the quantizers for the empirical MSE-ISGQ via the optimization problem

$$\begin{aligned}&\min_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ \|\boldsymbol{G} - \tilde{\boldsymbol{G}}\|_F^2 + \lambda(\text{bias})^2\\ =&\min_{\boldsymbol{\alpha},\boldsymbol{\beta}}\ \|\boldsymbol{G} - \sum_{k,l}\boldsymbol{C}_{k,l}\alpha_k\beta_l\|_F^2 + \lambda(\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{P}\boldsymbol{\alpha} - \bar{G})^2, \quad (6)\end{aligned}$$

where λ controls the trade-off between the MSE and empirical bias of MSE-ISGQ.

Computational Complexity. Since, the optimization problem (6) is bi-convex, we suggest the iterative approach summarized in Alg. 1 to solve it. The quantizers for $\boldsymbol{X}$ and $\boldsymbol{\Delta}$ can be initialized approximately based on the expected properties of the signals or as uniform quantizer. It can be easily verified that by fixing $\boldsymbol{\alpha}$, (6) becomes a quadratic problem w.r.t. $\boldsymbol{\beta}$ which has a closed form solution and can be computed efficiently. Similar arguments hold for fixing $\boldsymbol{\beta}$ and updating $\boldsymbol{\alpha}$. Moreover, in out experiments, we found out that only 1-2 iterations of Alg. 1 yields satisfactory results. Hence, the computational complexity of finding MSE-ISGQ quantizers is insignificant. For example, for the special case of 1-bit quantization, the *total* computational complexity of MSE-ISGQ to solve (6) is less than 100 FLOPs. (see supp. document for more detailed analysis.)

Dithered Indirect SG Quantization

The main drawback of using the deterministic approach for the quantization is the dependency of the quantization noise to the signal. Since $\boldsymbol{x}$ and $\boldsymbol{\delta}$ are generally correlated, this forced us in §3 to adjust the individual quantizers for each batch of data ($\boldsymbol{X}$ and $\boldsymbol{\Delta}$) to minimize the MSE and bias of ISGQ. Here, we pursue a different approach and develop a *simple and fixed* quantization scheme whose noise is independent of the signals. Our proposed algorithm is based on dithered quantization.

Definition (Dithered Quantization)**.** Let ϱ be the quantization step size. For an input signal x, assume that u is a random dither signal, generated independently of x. The dithered quantization of x is defined as $\tilde{x} = \varrho(\lfloor x/\varrho + u \rceil - u)$, where $\lfloor \alpha \rceil$ is the nearest integer to α.

Remark 1. To transmit the dithered quantization of x, it is sufficient to send the index of the quantization bin that $x/\varrho + u$ resides in, i.e., $q = \lfloor x/\varrho + u \rceil$. The receiver can reproduce the (pseudo-)random sequence u using the same random number generator algorithm and seed number and then compute the quantized value as $\tilde{x} = \varrho(q - u)$.

Characteristics of the dither signal has a major impact on the properties of the quantization noise. It is known that if the dither signal is generated uniformly over $(-1/2, 1/2)$, i.e. $u \sim \mathcal{U}(-1/2, 1/2)$, then the quantization noise $e = x - \tilde{x}$ is independent of the signal x and $e \sim \mathcal{U}(-\varrho/2, \varrho/2)$.

We consider the dithered indirect quantization of SG as follows: Let K_x and K_d be the number of desired quantization levels for $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, respectively. $\boldsymbol{X}$ is quantized as

$$\boldsymbol{Q}_x = \lfloor \boldsymbol{X}/\kappa_x + \boldsymbol{U} \rceil, \quad \widetilde{\boldsymbol{X}} = \kappa_x \left(\boldsymbol{Q}_x - \boldsymbol{U}\right), \qquad (7)$$

where the *scale factor* $\kappa_x = \|\boldsymbol{X}\|_\infty / K_x$ maps the signal into the range $[-K_x, K_x]$ prior to quantization and $\boldsymbol{U} \sim \mathcal{U}(-1/2, 1/2)$ is an independently generated random dither signal. It can be easily verified that the scaled quantization noise $\boldsymbol{E}_x = (\boldsymbol{X} - \widetilde{\boldsymbol{X}})/\kappa_x$ is independent of the signals $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, and uniformly distributed over $(-1/2, 1/2)$. The dithered quantization of $\boldsymbol{\Delta}$ is defined similarly.

Theorem 2. *Let $\boldsymbol{G} = \frac{1}{L}\boldsymbol{\Delta}\,\boldsymbol{X}^\mathsf{T}$ be a stochastic gradient of $\mathcal{L}(\boldsymbol{W})$. Then, the Dithered-ISGQ, $\widetilde{\boldsymbol{G}} = \frac{1}{L}\widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{X}}^\mathsf{T}$ with number of quantization levels K_x and K_d has the following properties:*

- P1. *$\widetilde{\boldsymbol{G}}$ is unbiased, i.e., $\mathbb{E}[\widetilde{\boldsymbol{G}}] = \nabla\mathcal{L}$,*
- P2. *Its variance is bounded as $\mathbb{E}\big[\|\widetilde{\boldsymbol{G}} - \nabla\mathcal{L}\|_F^2\big] \leq \frac{mn}{L}\gamma\,\mathbb{E}\big[\|\boldsymbol{X}\|_\infty^2\,\|\boldsymbol{\Delta}\|_\infty^2\big] + \mathbb{E}\big[\|\boldsymbol{G} - \nabla\mathcal{L}\|_F^2\big]$, where γ is a constant depending only on the number of quantization levels.*

Especially, if we assume that $\boldsymbol{X}$ follows a Normal or Folded-Normal distribution with variance σ_x^2 and $\boldsymbol{\Delta} \sim \mathcal{N}(0, \sigma_d^2)$, independent of each other, then

$$\mathbb{E}\big[\|\widetilde{\boldsymbol{G}} - \nabla_{\boldsymbol{W}}\mathcal{L}\|_F^2\big] \leq \frac{mn}{L}\sigma_x^2\sigma_d^2\,\big(\frac{\ln(\sqrt{2}nL)}{3K_x^2} + 1\big)\times \big(\frac{\ln(\sqrt{2}mL)}{3K_d^2} + 1\big). \qquad (8)$$

Note that although the Dithered-ISGQ may have higher variance than MSE-ISGQ in some applications, it has the advantages of having fixed quantizers and not requiring joint-optimization of the individual factorized quantizer.

Rate-Distortion Analysis. It is worth exploring the relation between the variance of Dithered-ISGQ (i.e., the distortion) and the total number of bits (i.e., the transmission rate). Since the quantizer index of $\boldsymbol{X}$, $\boldsymbol{Q}_x \in \{-K_x, \ldots, K_x\}$ can take at most $2K_x + 1$ distinct values and $\boldsymbol{X}$ has nL elements, the *total* number of bits for quantized $\boldsymbol{X}$ would be $nL\log(2K_x + 1)$. Similarly, total number of bits for quantized $\boldsymbol{\Delta}$ would be $mL\log(2K_d + 1)$. Hence, the total number of bits for dithered ISGQ is $R = L\big(n\log(2K_x + 1) + m\log(2K_d + 1)\big)$ per training iteration.

Considering the rate-distortion with respect to the batch-size L, we realize that $R = \mathcal{O}(L)$ while from (8) MSE$= \mathcal{O}(\frac{\ln(L)^2}{L})$. Thus, the rate increases linearly w.r.t. the batch-size but the decrease in quantization noise is sublinear.

Similarly, to analyze the rate-distortion w.r.t. the number of quantization levels, we observe that doubling the number of quantization levels increases the number of bits by 1 per sample. For sufficiently large nL and mL (relative to K_x and K_d), the MSE would be reduced approximately by a factor of 16. However, when $\ln(nL) \ll K_x^2$ and $\ln(mL) \ll K_d^2$, which corresponds to more quantization levels (i.e. finer quantization of $\boldsymbol{X}$ and $\boldsymbol{\Delta}$), the MSE of Dithered-ISGQ would be the same as the non-quantized SG and any further increase in the number of bits would not improve the accuracy anymore.

Computational Complexity. It is worth mentioning that only the intermediate signals, $\boldsymbol{X}$ and $\boldsymbol{\Delta}$, are required to be available for Dithered-ISGQ and there is no need to compute the SG via (1). Moreover, quantizing $\boldsymbol{X}$ can be done in parallel while performing the forward and backward computations. Hence, generally the computation time of Dithered-ISGQ is less than other direct quantization methods.

Convergence Analysis. The convergence analysis of the Dithered-ISGQ relies on the fact that the proposed quantization method is unbiased and has bounded variance. Consider stochastic gradient descent learning algorithm with ISGQ in which at the t-th iteration, the parameters are updated as

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta_t\widetilde{\boldsymbol{G}}_t, \qquad (9)$$

where η_t is the learning rate. Convergence of the learning algorithm can be easily verified under almost the same assumptions as in (Bottou 1998, §5.1), i.e.,

- **A1.** $f(\cdot)$ is lower bounded and 3-times differentiable with continuous derivatives.
- **A2.** Learning rates satisfy $\sum \eta_t = +\infty$ and $\sum \eta_t^2 < \infty$.
- **A3.** Over the support of $f(\cdot)$, the signals have bounded joint 4^{th} moment $\mathbb{E}\big[\|\boldsymbol{X}\|_F^4 . \|\boldsymbol{\Delta}\|_F^4\big] < \infty$.
- **A4.** If $\boldsymbol{W}$ grows too large, the gradient descent direction points towards zero.

Theorem 3. *Assume that conditions (A1) to (A4) hold. Then gradient descent with Dithered-ISGQ* (9) *converges almost surely to a local extremum, i.e., $\nabla_{\boldsymbol{W}_t}\mathcal{L} \overset{a.s.}{\to} 0$ as $t \to +\infty$.*

4 Application to Distributed Training of Neural Networks

In this section, we show how ISGQ can be employed for efficient communication of stochastic gradients in distributed training of deep neural networks. Consider the l-th layer of a neural network, whose input signal is $\boldsymbol{x}^{(l-1)}$ and the weights and biases are $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$, respectively. By concatenating

$\boldsymbol{b}^{(l)}$ to $\boldsymbol{W}^{(l)}$ and appending 1 to $\boldsymbol{x}^{(l-1)}$ [1], the input signal into the nodes and the output of the l-th layer are given by

$$\boldsymbol{y}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{x}^{(l-1)}, \quad \boldsymbol{x}^{(l)} = \boldsymbol{\sigma}(\boldsymbol{y}^{(l)}), \tag{10}$$

where $\boldsymbol{\sigma}(\cdot)$ is the activation function, applied element-wise. It is worth noting that the convolutional layers can be represented similarly by appropriate reshaping and reformulation of the input and the parameters.

There exists a function $g(\cdot)$ such that the final output of the neural network, $\boldsymbol{y}$, can be represented as $\boldsymbol{y} = g(\boldsymbol{x}^{(l)})$, where $g(\cdot)$ may depend on other signals and parameters of the neural network. Hence the loss function w.r.t. $\boldsymbol{x}^{(l)}$ and desired output $\boldsymbol{t}$ is given by $\ell(g(\boldsymbol{x}^{(l)}), \boldsymbol{t})$. By defining $f(\boldsymbol{v}) = \ell(g(\boldsymbol{\sigma}(\boldsymbol{v})), \boldsymbol{t})$, the training loss function with respect to the parameters of the l-th layer would be $\mathcal{L} = \mathbb{E}\big[f(\boldsymbol{W}^{(l)}x^{(l-1)})\big]$, where $x^{(l-1)}$ can be considered as the *virtual input* of the l-th layer.

Moreover, it is worth mentioning that the backpropagation algorithm, widely used in deep learning (Rojas 1996; Goodfellow et al. 2016), is indeed a realization of (1) and chain-rule. It is well-known that gradient of the cost function for an input $\boldsymbol{x}$ w.r.t. the parameters of the l-th layer can be computed as

$$\nabla_{\boldsymbol{W}^{(l)}}\mathcal{L} = \boldsymbol{\delta}^{(l)}\big(\boldsymbol{x}^{(l-1)}\big)^{\mathsf{T}}, \tag{11}$$

$$\boldsymbol{\delta}^{(l)} = \boldsymbol{\sigma}'(\boldsymbol{y}^{(l)}) \odot \big((\boldsymbol{W}^{(l+1)})^{\mathsf{T}}\boldsymbol{\delta}^{(l+1)}\big). \tag{12}$$

where $\delta_j^{(l)} := \frac{\partial f}{\partial y_j^{(l)}}$ is the partial derivative of the cost function w.r.t. input signal of the j-th node of the l-th layer, i.e., $\boldsymbol{\delta}^{(l)} = \nabla_{\boldsymbol{y}} f(\boldsymbol{y})|_{\boldsymbol{y}=\boldsymbol{W}^{(l)}\boldsymbol{x}^{(l-1)}}$. These observations imply the potential application of the ISGQ algorithms developed in §3 for the compression of SG and distributed training of deep models. Using ISGQ in distributed learning can provide the following benefits:

- Since calculating SGs at the workers is generally done via backpropagation algorithm, computing forward and backward signals does not incur extra computational complexity. On the other hand, in Dithered-ISGQ, there is no need to compute the SG via (11) and having access to $\boldsymbol{X}$ and $\boldsymbol{\Delta}$ (computed via (12)) is sufficient. Since the complexity of quantizing individual signals is less than matrix multiplication, we argue that Dithered-ISGQ can slightly reduce the computational load at the workers in addition to reducing the total transmission bits.
- As the majority of signals are sparse due to the structure of neural networks and the forward and backward signals have generally less entropy, they are more compressible than the gradients (please refer to the supplementary document and [Anonymized]). For example, with ReLU activation function, $\sigma(y) = \sigma'(y) = 0$ for $y < 0$. Hence, the forward and backward signals ($\boldsymbol{x}$, $\boldsymbol{\delta}$) in the hidden layers are mostly sparse, and because of (10) and (12) their sparsities are correlated which can be used to further reduce the communication bit rate.
- Since the quantization of the signals are performed separately, it can be potentially implemented in parallel, and some operations (such as generating random dither signal) can be executed simultaneous to the neural network's forward and backward computations.

Note that the proposed indirect quantization is more suitable when the batch-size is smaller then the number of parameters. For layers with weight sharing schemes such as convolutional layers which generally have fewer parameters for transmission, distributed training benefits more from direct compression and transmission of the stochastic gradients using methods such as (Alistarh et al. 2017; Wen et al. 2017).

[1] i.e., $\boldsymbol{W}^{(l)} \leftarrow [\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}]$ and $\boldsymbol{x}^{(l)} \leftarrow [\boldsymbol{x}^{(l)}; 1]$.

5 Experiments

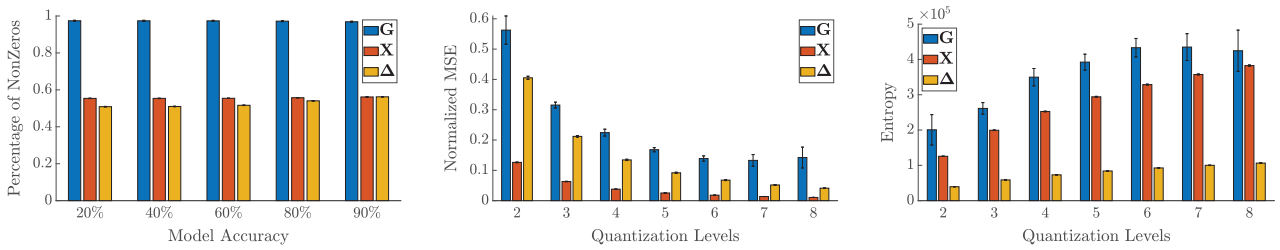
In this section, we evaluate the properties of the developed ISGQ algorithms and their performance in distributed training. For the simulations, we consider MNIST database with fully-connected (784-1000-300-100-10) neural network (hereafter referred to as FC) and Lenet model (LeCun et al. 1998), CIFAR-10 database using CifarNet (Krizhevsky, Sutskever, and Hinton 2012), and Imagenet (Russakovsky et al. 2015) using AlexNet deep model (Krizhevsky, Sutskever, and Hinton 2012). The considered deep models, FC, Lenet, CifarNet and AlexNet have approximately 1.16, 1.66, 1.07 and 62.4 million parameters, respectively. In all our experiments, we use SGD or Adam algorithm with initial learning rate 0.01, decay rate 0.98 per epoch and batch-sizes 256 or 128 per worker. To evaluate the reduction in the transmission bits as well as the performance loss of the trained model, we compared our proposed method against the baseline distributed training without any quantization (i.e., 32 bits used for the transmissions of values) and other direct quantization methods: 1-bit quantization of (Seide et al. 2014), TernGrad (Wen et al. 2017) and QSGD (Alistarh et al. 2017). For implementation details and the distributed learning algorithm, please refer to the supplementary document.

Quantizer Performance. First, we investigate how our proposed ISGQ methods are compared against the direct optimum Lloyd-Max quantization (Lloyd 1982; Max 1960). For this purpose, we consider different neural networks at various stages of training and repeated the experiments numerous times to compute the mean and variance of the desired metrics. Some of the results are presented in figures 2 and 3.

We observe that generally the forward and backward signals are sparser (Fig. 2a), and their optimum quantized values have less entropy and normalized MSE (defined as $\|\boldsymbol{v} - \tilde{\boldsymbol{v}}\|^2/\|\boldsymbol{v}\|^2$ for vector $\boldsymbol{v}$) than the SG (Fig. 2). Hence, quantization of the intermediate signals generally requires fewer number of bits and has smaller individual quantization noise than directly quantizing the signals, confirming that these signals are more compression-friendly.

Moreover, our proposed MSE-ISGQ (using only 1 iteration of Alg. 1) and Dithered-ISGQ usually performs comparable or better than the optimum (Lloyd-Max) direct quantization of the SG (see Fig. 3), showing the effectiveness of ISGQ.

Processing Time per Iteration. Next, we measure the complexity of the proposed SG compression technique by measuring the average time required to process (e.g., feed



(a) Percentage of Non-Zeros (b) Normalized MSE (c) Entropy

Figure 2: Sparsity at different stages of training, Normalized MSE and Entropy of quantized SG vs signals of the 2nd hidden layer of FC at accuracy=40% for various number of quantization levels.

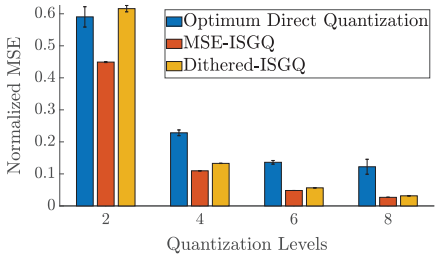


Figure 3: Comparing ISGQ with optimum direct SG quantization, 2nd hidden layer of FC at accuracy=40%.

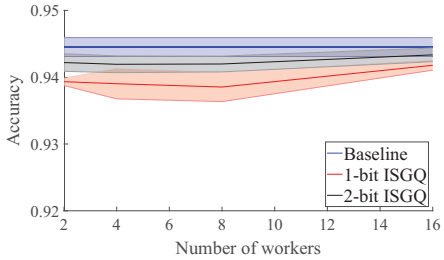


Figure 4: Final accuracy of the trained model, shaded areas represent 1 standard deviation.

mini-batch and compute the SG), quantize and communicate the SGs. Let T_p be the total processing and quantization time and T_c be the average communication time to transmit the raw parameters of the model. Obviously, if a worker compresses the gradients by a factor of k, its communication time would be reduced approximately by Tc/k, while on the other hand, its processing time might increase slightly. As a result, in a centralized synchronous distributed training with P identical workers, the total processing time would be $T_p + PT_c/k + T_u$, where T_u is the communication time to broadcast back the aggregated gradients to the workers by the server.

First, we compare the required total processing and quantization times of the proposed ISGQ with QSG (Alistarh et al. 2017) and baseline (no quantization) for different batch-sizes and different models using Intel Core i7 CPU and Nvidia Titan Xp GPU. Since, baseline transmission only computes the SGs, the total processing time is expected to be larger when quantization is added. Tables 1 and 2 show the results for processing 200 batches on CPU and GPU, respectively. Since the dithered-ISGQ does not require computing the SG via (11) and only relies on back-propagation calculations, when matrix multiplications are costly (e.g., on CPU or for large matrices), its computation time is significantly lower than other quantization techniques and comparable to the baseline.

Next, to find the effectiveness of different quantization schemes in terms of communication overhead, we calculated and compared compression gain of each scheme as

$$\text{compression gain} = \frac{32 \times (\text{\# model's parameters})}{\text{\# transmitted bits per worker}}.$$

Some of the results are presented in Tbl. 3 for different models, batch-sizes and various quantization schemes.

One can easily conclude that 1000 iterations of decentralized distributed training Alexnet with 4 workers, batch-size 128 per worker using Titan Xp GPUs connected via InfiniBand links would take approximately 3.9 minutes using ISGQ compared to 4.5 minutes by QSG and 9 minutes by Baseline (no SG compression), while centralized single node training with the same total batch-size takes approximately 14.8 minutes to execute.

Accuracy of the Distributed Training. 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 compare the effect of the quantization algorithms.

Through our simulations, we have found that distributed training of the considered deep models using either of the quantization schemes eventually converges to $\pm 1\%$ of the accuracy of the baseline model. However, the convergence speed of the 1-bit method (Seide et al. 2014) is considerably slower than the others for complex models, while ISGQ performs comparably well. For example, Fig. 4 compares the

Table 1: Computation time (sec.) with Core i7 CPU

	Batch size	256	128	64
FC	Baseline	1.2	0.78	0.63
	QSGD	1.85	1.43	1.23
	D-ISGQ	1.29	0.76	0.52
Lenet	Baseline	14.4	8.17	5.1
	QSGD	15.79	9.1	5.9
	D-ISGQ	15.12	8.45	4.98
Cifarnet	Baseline	30.33	16.31	9.2
	QSGD	31.59	17.1	9.92
	D-ISGQ	31.4	16.77	9.19
Alexnet	Baseline	66.4	34.5	18.9
	QSGD	70	37.6	21.8
	D-ISGQ	66.7	34.4	18.4

Table 2: Computation time (sec.) w/ Titan Xp GPU

	Batch size	256	128	64
FC	Baseline	0.29	0.26	0.25
	QSGD	0.34	0.32	0.31
	D-ISGQ	0.36	0.33	0.31
Lenet	Baseline	1.27	0.84	0.62
	QSGD	1.39	0.98	0.77
	D-ISGQ	1.41	1.0	0.79
Cifarnet	Baseline	3.27	1.62	0.92
	QSGD	3.34	1.69	0.99
	D-ISGQ	3.26	1.7	1.01
Alexnet	Baseline	83	45.2	25
	QSGD	86	46.1	25.5
	D-ISGQ	84	44.4	24.1

final accuracy of the trained model with ISGQ using different number of workers with the baseline. As seen, the accuracy loss due to the training with quantized SG is small (less than 0.2% most of the time for 2-bit ISGQ).

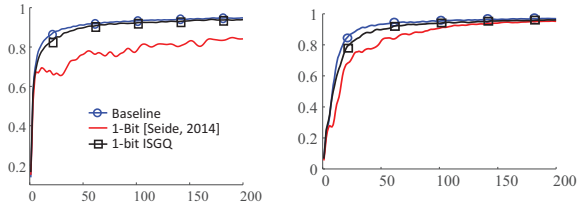
Figure 5 shows the test accuracy of the model at each iteration during training with stochastic gradient descent using baseline (no quantization), 1-bit quantization (Seide et al. 2014) and our proposed ISGQ. Note that here we omitted the time delays that is caused by more communication overhead in the baseline and 1-bit quantization and assumed that the speed of connection link is infinity. As shown in the figure, the convergence rate of ISGQ closely follows the baseline while it has the potential of achieving compression gains of beyond 32, much higher than the traditional direct quantization methods.

6 Conclusion

In this paper, we proposed a novel approach, *indirect stochastic gradient quantization via factorization*, instead of commonly used direct methods. Our method takes advantage of the characteristics of the backpropagation algorithm and

Table 3: Average compression gains of different quantization methods in distributed deep learning

	Batch size	256	128	64
FC	1-bit ISGQ	33	67	133
	1-bit (Seide et al. 2014)	28.4	28.4	28.4
	TernGrad / 1-bit QSGD	20.2	20.2	20.2
Lenet	1-bit ISGQ	56	105	180
	1-bit (Seide et al. 2014)	28	28	28
	TernGrad / 1-bit QSGD	20	20	20
Cifarnet	1-bit ISGQ	38	65	98
	1-bit (Seide et al. 2014)	28	28	28
	TernGrad / 1-bit QSGD	20.1	20.1	20.1
AlexNet	1-bit ISGQ	117	170	221
	2-bits ISGQ	80	118	153
	1-bit (Seide et al. 2014)	29	29	29
	TernGrad / 1-bit QSGD	19.4	19.4	19.4



(a) FC (b) Lenet

Figure 5: Convergence rate of distributed training with 8 workers using different quantization methods.

the statistical properties of the forward and backward signals during training. For the quantization of the forward and backward signals, we proposed different approaches whose objective is minimizing the error in the reconstructed SG. We showed that despite its simplicity, ISGQ can perform close to the optimum Lloyd-Max quantization algorithm in terms of reconstruction error while requiring much less bite-rate. Moreover, ISGQ leads to significant reduction in the communication overhead, achieving compression gain of more than 100, without sacrificing the training speed or accuracy. Especially for a fixed total batch-size, at each worker the required transmission bit-rate of the fully connected layers decreases as the number of workers increases. This results in the reduction of total bits for transmission of the parameters in ISGQ, in contrast to the existing direct approaches whose transmission bit-rate remains fixed regardless of the number of workers.

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.

Bottou, L. 1998. Online Learning and Stochastic Approximations, Revised 2018. *On-Line Learning in Neural Networks* 17(9):1–35.

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.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Okazaki, N. 2016. *Deep learning*. MIT press.

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

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.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28(2):129–137.

Max, J. 1960. Quantizing for minimum distortion. *IRE Transactions on Information Theory* 6(1):7–12.

Rojas, R. 1996. Neural Networks, Sringer-Verlag, Berlin. *Neural Networks* 7.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252.

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.

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

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.

Zhang, H.; Li, J.; Kara, K.; Alistarh, D.; Liu, J.; and Zhang, C. 2017. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In *International Conference on Machine Learning*, 4035–4043.