CONVERGENCE BOUNDS FOR COMPRESSED GRADIENT METHODS WITH MEMORY BASED ERROR COMPENSATION

Sarit Khirirat, Sindri Magnússon, Mikael Johansson

ABSTRACT

The veritable scale of modern data necessitates information compression in parallel/distributed big-data optimization. Compression schemes using memory-based error compensation have displayed superior performance in practice, however, to date there are no theoretical explanations for these observed advantages. This paper provides the first theoretical support for why such compression schemes yields higher accuracy solutions in optimization. Our results cover both gradient and incremental gradient algorithms for quadratic optimization. Unlike previous works, our theoretical results explicitly quantify the accuracy gains from error compensation, especially for ill-conditioned problems. Finally, the numerical results on linear least-squares problems validate the benefit of error compensation and demonstrate tightness of our convergence guarantees.

Index Terms— Quadratic optimization, quantization, gradient descent, incremental gradient methods.

1. INTRODUCTION

Parallel and distributed optimization algorithms play an important role in large-scale signal processing and machine learning. In essence, these algorithms are based on splitting large-scale problems among many processors that coordinate their computations to cooperatively find an optimal solution. Standard algorithms, which exchange full precision information among computing nodes, can easily run into communication bottlenecks that slow down convergence speed, especially when decision vectors are large and dense [1]. To alleviate this problem, several heuristic gradient compression strategies have recently been proposed [2, 3, 4, 5, 6], and optimization algorithms operating on compressed data have been developed [1, 7, 8, 9, 10, 11]. These studies show that while gradient compression can reduce the communication load significantly, both convergence speed and solution accuracy deteriorate if the quantization is too coarse.

To mitigate such adverse effects, one recent idea is to adopt error compensation, where the current gradient is combined with information on the accumulated quantization errors from previous iterations. It has been observed empirically that gradient algorithms with aggressive compression can benefit significantly from error compensation [12, 13]. Motivated by these encouraging experiments, a number of recent works analyzed the convergence of different optimization algorithms with error compensation, and confirmed their practical benefits in numerical experiments (*e.g.* in, [14, 15, 16]). However, as of today, no theoretical justification for the increased solution accuracy of error compensation has been published. In this paper, we provide the first theoretical support for how error compensation can improve performance of optimization algorithms which operate on compressed gradients.

Specifically, this paper analyzes convergence rates and solution accuracy of compressed gradient descent and compressed incremental gradient methods. Our theoretical results indicate that the error compensation scheme reduces the residual by approximately a factor of a condition number for gradient descent; numerical results confirm that this bound is reasonably tight. We also quantify the impact of gradient compression and error compensation on incremental gradient methods. In particular, the error compensation can decrease the residual error by the condition number in some cases. Due to page limitations, this paper only considers minimization of strongly convex quadratic functions. However, most of the results can be extended to general strongly convex functions.

1.1. Notations

We let $\mathbb{N}, \mathbb{N}_0, \mathbb{Z}$ be the set of natural numbers, the set of natural numbers including zero, and the set of integers, respectively. The set $\{0, 1, \dots, T\}$ is denoted $[0, T]$. For $x \in \mathbb{R}^d$, $\|x\|$ and $\|x\|_1$ are the ℓ_2 norm and ℓ_1 norm of x, respectively, while x_i is the i^{th} coordinate of x. The d-dimensional identity matrix is denoted I_d. A matrix $A \in \mathbb{R}^{d\times d}$ has eigenvalues $\lambda_1(A), \dots, \lambda_d(A)$ and spectral norm is $\|A\| = \max_{x\neq 0} \|Ax\|/\|x\| = \max_{i\in[1,d]} |\lambda_i(A)|$. We define $\mu = \min_{i\in[1,d]} \lambda_i(A)$ and $L = \max_{i\in[1,d]} \lambda_i(A)$; note that A is positive definite if and only if $\mu > 0$. Fi-

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. This work was also supported in parts by the AFOSR YIP. S. Khirirat and M. Johansson are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, and ACCESS Linnaeus Center, Royal Institute of Technology (KTH), Stockholm, Sweden. Emails: {*sarit@kth.se, mikaelj@kth.se*}. S. Magnússon is with the School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA. Email: *sindrim@seas.harvard.edu*.

nally, $\mathcal{U}(a,b)$ is the uniform distribution on the interval $[a,b]$, and $\mathcal{N}(\mu,\sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2.

2. COMPRESSION

In this paper, we focus on the following family of gradient compression schemes.

Definition 1. *The mapping $Q:\mathbb{R}^d\to\mathbb{R}^d$ is a* deterministic bounded error compressor *(BEC) if there exists a positive constant ϵ such that*

$$\|Q(z)-z\|\leq\epsilon,\quad\forall z\in\mathbb{R}^d.$$

According to Definition 1, the lower ϵ is, the higher compression accuracy is. Even though Definition 1 is abstract, it covers many quantizations of practical interest. One such example is the rounding quantizer:

Definition 2 ([10, 11])**.** *For a given quantization resolution $\Delta>0$, the* rounding quantizer $Q_r:\mathbb{R}^d\to\Lambda$ *is defined as*

$$[Q_r(z)]_i=t\Delta,\quad \textit{if}\quad (t-0.5)\Delta\leq z_i<(t+0.5)\Delta,$$

where $z\in\mathbb{R}^d$ and the quatization lattice is defined by

$$\Lambda=\{t\Delta:t\in\mathbb{Z}\}.$$

The rounding quantizer was proposed for incremental gradient algorithms in [10] and subsequently used for ADMM in [11]. It was shown in [11] that the rounding quantizer is a BEC with $\epsilon=\Delta\sqrt{d}/2$.

Another family of BECs arises when gradients are compressed using a bounded lattice set (i.e. t is bounded between two finite values) as proposed by, *e.g.*, [5, 17].

3. COMPRESSED GD

To build intuition for how error compensation benefits solution accuracy, we start by studying the compressed gradient descent algorithm. The result is of significance for distributed optimization using dual decomposition methods, where the dual function is typically optimized using gradient descent techniques [7, 18, 19].

Consider the quadratic optimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\quad f(x)=\frac{1}{2}x^TAx+b^Tx, \tag{1}$$

where $A\in\mathbb{R}^{d\times d}$ is a symmetric matrix and $b\in\mathbb{R}^d$ is a column vector.

The classical *gradient descent* (GD) algorithm for solving (1) forms a sequence $\{x_k\}_{k\in\mathbb{N}}$ via

$$x_{k+1}=x_k-\gamma\nabla f(x_k),$$

from a given initial point x_0 and some fixed positive step-size γ. It is well-known (see, *e.g.*, [20, 21]) that GD is guaranteed to converge toward the optimum with linear rate when A is positive definite.

The most straightforward way to reduce communication is to simply compress the full gradient. This leads to the compressed gradient descent (CGD) recursion

$$x_{k+1}=x_k-\gamma Q(\nabla f(x_k)). \tag{2}$$

We will assume that $Q(\cdot)$ is a BEC according to Definition 1. It is sometimes convenient to re-write this recursion as

$$x_{k+1}=x_k-\gamma\left(\nabla f(x_k)+e_k\right), \tag{3}$$

where $e_k=Q(\nabla f(x_k))-\nabla f(x_k)$.

Our first result characterizes the convergence of the iterates produced by the compressed gradient descent recursion.

Theorem 1. *Consider the quadratic optimization problem* (1) *where $\mu\cdot I_d\preceq A\preceq L\cdot I_d$ and $\mu<L$ for positive real numbers μ, L. Then, the iterates $\{x_k\}_{k\in\mathbb{N}}$ generated by* (3) *satisfy*

$$\|x_k-x^\star\|\leq\rho^k\|x_0-x^\star\|+\frac{1}{\mu}\epsilon,$$

where

$$\rho=\begin{cases}1-1/\kappa & \textit{if}\quad \gamma=1/L\\ 1-2/(\kappa+1) & \textit{if}\quad \gamma=2/(\mu+L)\end{cases},$$

and $\kappa=L/\mu$.

Note that we recover the classical results of GD [21, 20] when we let $\epsilon=0$ in Theorem 1. In addition, the guaranteed bound on the residual error of CGD is ϵ/μ, which increases as the strong convexity modulus μ decreases (*i.e.* as the function becomes "less" strongly convex).

This dependence on μ can be removed by adopting error compensation. We consider the following error-compensated compressed GD (EC-CGD) scheme

$$\begin{aligned}x_{k+1}&=x_k-\gamma Q(\nabla f(x_k)+Bm_k),\quad\text{and}\\ m_{k+1}&=\nabla f(x_k)+Bm_k-Q(\nabla f(x_k)+Bm_k),\end{aligned} \tag{4}$$

where $B\in\mathbb{R}^{d\times d}$. Unlike CGD, EC-CGD keeps the memory of the sequence $\{m_k\}_{k\in\mathbb{N}}$ to correct the gradient information in each iteration. Our update is similar to that proposed by [16] and [14] if we let $B=\beta I_d$ and $B=(1/\gamma)I_d$, respectively. For analysis purposes, we let $c_k=-m_k$ and re-write the EC-CGD updates (4) as

$$\begin{aligned}x_{k+1}&=x_k-\gamma\left(\nabla f(x_k)+e_k\right)\\ e_k&=Q(\nabla f(x_k)-Bc_k)-\nabla f(x_k),\quad\text{and}\\ c_{k+1}&=Q(\nabla f(x_k)-Bc_k)-\nabla f(x_k)+Bc_k.\end{aligned} \tag{5}$$

The next result characterizes the convergenceof EC-CGD:

Theorem 2 (Strongly convex case)**.** *Consider the quadratic optimization problem* (1) *where $\mu \cdot I_d \preceq A \preceq L \cdot I_d$ for some positive real numbers μ, L. Assume that $B = I_d - \gamma A$ and $c_0 = \mathbf{0}$. Then, the iterates $\{x_k\}_{k\in\mathbb{N}}$ generated by* (5) *satisfy*

$$\|x_k - x^\star\| \leq \rho^k \|x_0 - x^\star\| + \gamma\epsilon,$$

where

$$\rho = \begin{cases} 1 - 1/\kappa & \text{if} \quad \gamma = 1/L \\ 1 - 2/(\kappa+1) & \text{if} \quad \gamma = 2/(\mu+L) \end{cases},$$

and $\kappa = L/\mu$.

Like Theorem 1, Theorem 2 with $\epsilon = 0$ recovers the classical results of GD. More importantly, Theorem 2 implies that EC-CGD has the same convergence rate as CGD, but lower residual error. In particular, EC-CGD with $\gamma = 1/L$ and $\gamma = 2/(\mu + L)$ reduces the quantization error ϵ by κ and $(\kappa + 1)/2$, respectively. Thus, the error compensation technique improves the solution accuracy significantly when the problem is ill-conditioned (L/μ is large). In addition, two step-size choices reveal a trade-off between convergence speed and quantization error; the first step-size choice in Theorem 2 results in slower convergence but smaller residual error than the second step-size choice.

Also, the bound for EC-CGD from Theorem 2 is shown to be tight in Section 5.

4. COMPRESSED IGM

This section considers minimization problems with separable quadratic loss functions

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x) = \sum_{i=1}^{m} f_i(x). \tag{6}$$

Here, each f_i is on the form $f_i(x) = (1/2)x^T A_i x + b_i^T x$ where $A_i \in \mathbb{R}^{d\times d}$ is positive definite and $b_i \in \mathbb{R}^d$. Problem (6) arises in several machine learning and signal processing applications. The simplest instance may be standard least-squares. Due to the explosive scale of datasets, modern applications focus on solving (6) when m is extremely large.

The *incremental gradient* method (IGM) is a popular first-order method due to its low per-iteration cost and its convergence guarantee toward the sub-optimum, [22]. IGM updates the iterate x_k^i as

$$x_k^{i+1} = x_k^i - \gamma \nabla f_i(x_k^i), \quad \text{for} \quad i = 1, 2, \ldots, m$$

with a fixed positive step-size γ. We set $x_{k+1}^1 = x_k^{m+1}$ and refer to $\{x_k^1\}_{k\in\mathbb{N}}$ as the outer iterates.

To study the effect of using lossy gradient information, we consider the convergence of compressed incremental gradient methods (CIGM), which updates the iterate x_k^i according to

$$x_k^{i+1} = x_k^i - \gamma Q(\nabla f_i(x_k^i)) \quad \text{for} \quad i = 1, 2, \ldots, m \tag{7}$$

given the initial point x_0 and a fixed positive step-size γ. Here, we initialize $x_0^1 = x_0$ and set $x_{k+1}^1 = x_k^{m+1}$. Unlike [14], we consider the deterministic version of CIGM. It is easy to verify that the equivalent update of (7) is

$$x_k^{i+1} = x_k^i - \gamma \left(\nabla f_i(x_k^i) + e_k^i\right), \tag{8}$$

where $e_k^i = Q(\nabla f_i(x_k^i)) - \nabla f_i(x_k^i)$.

Theorem 3. *Consider the quadratic optimization problem* (1) *where $\bar{\mu} \cdot I_d \preceq A_i \preceq \bar{L} \cdot I_d$ and $\bar{\mu} > 0$. Assume that $\|\nabla f_i(x^\star)\| \leq \sigma$ for some finite σ. Then, the iterates $\{x_k^1\}_{k\in\mathbb{N}}$ generated by* (8) *with $\gamma = 1/(\theta\bar{L})$ and $0 < \theta < \bar{L}/\bar{\mu}$ satisfy*

$$\|x_k^1 - x^\star\| \leq \rho^{m\cdot k}\|x_0 - x^\star\| + e,$$

where

$$\rho = 1 - \frac{\bar{\mu}}{\theta\bar{L}}, \quad \text{and} \quad e = \frac{1}{1-\rho^m}\frac{\gamma}{1-\rho}(\sigma + \epsilon).$$

To show the benefit of error compensation scheme, we consider the error-compensated compressed IGM (EC-CIGM) which forms the following recursion

$$\begin{aligned} x_k^{i+1} &= x_k^i - \gamma Q(\nabla f_i(x_k^i) + B_i m_k^i) \quad \text{and} \\ m_k^{i+1} &= \nabla f_i(x_k^i) + B_i m_k^i - Q(\nabla f_i(x_k^i) + B_i m_k^i) \end{aligned} \tag{9}$$

for $i = 1, 2, \ldots, m$. We let $x_{k+1}^1 = x_k^{m+1}$ and set $m_k^1 = \mathbf{0}$ in each cycle i. In the special case that $B_i = (1/\gamma) \cdot I_d$, the updates reduce to the deterministic version of SGD with memory proposed in [14]. In our analysis, we consider the equivalent update

$$\begin{aligned} x_k^{i+1} &= x_k^i - \gamma \left(\nabla f_i(x_k^i) + e_k^i\right) \\ e_k^i &= Q(\nabla f_i(x_k^i) - B_i c_k^i) - \nabla f_i(x_k^i), \quad \text{and} \\ c_k^{i+1} &= Q(\nabla f_i(x_k^i) - B_i c_k^i) - \nabla f_i(x_k^i) + B_i c_k^i. \end{aligned} \tag{10}$$

obtained by letting $c_k^i = -m_k^i$ in (9).

Theorem 4. *Consider the quadratic optimization problem* (1) *where $\bar{\mu} \cdot I_d \preceq A_i \preceq \bar{L} \cdot I_d$ and $\bar{\mu} > 0$.. Assume that $\|\nabla f_i(x^\star)\| \leq \sigma$ with σ finite, let $B_i = I_d - \gamma A_i$ and set $c_k^1 = \mathbf{0}$ for each k. Then, the iterates $\{x_k^1\}_{k\in\mathbb{N}}$ generated by* (10) *with $\gamma = 1/(\theta\bar{L})$ and $0 < \theta < \bar{L}/\bar{\mu}$ satisfy*

$$\|x_k^1 - x^\star\| \leq \rho^{m\cdot k}\|x_0 - x^\star\| + e.$$

where

$$\rho = 1 - \frac{\bar{\mu}}{\theta\bar{L}}, \quad \text{and} \quad e = \frac{1}{1-\rho^m}\frac{\gamma}{1-\rho}\left(\sigma + (1-\rho)\epsilon\right).$$

In contrast to the results in [14, 16, 15], Theorem 3 and 4 demonstrate how the error compensation improves the solution accuracy of CIGM while maintaining the same linear convergence rate. Since the residual error results from σ and

the quatization accuracy ϵ, we consider two extreme cases. If $\sigma \gg \epsilon$, then error compensation does not have any significant impact on the solution accuracy. On the other hand, if $\epsilon \geq \sigma \cdot \min(1, \theta\bar{L}/\bar{\mu})$, then ϵ is the main contributor to the residual error in Theorems 3 and 4. In this scenario, the error compensation is able to decrease the residual error by a factor $1/(1-\rho) = \theta\bar{L}/\bar{\mu}$.

5. SIMULATION RESULTS

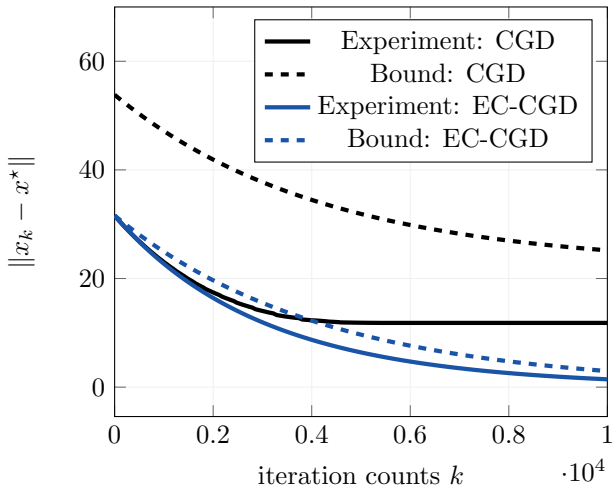

Fig. 1. The performance of CGD and EC-CGD with their theoretical bounds for least-squares problems.

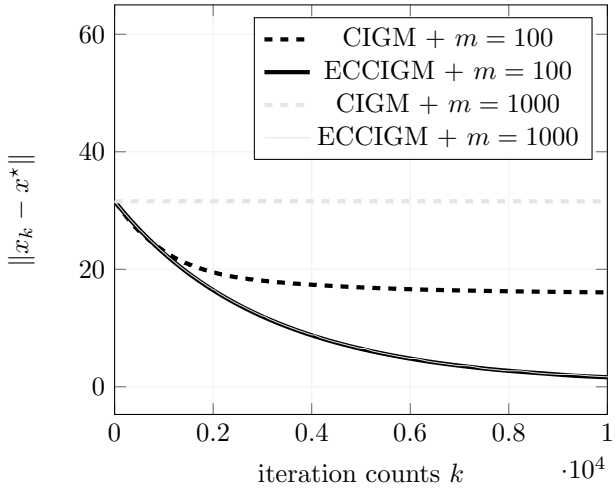

Fig. 2. The performance of CIGM and EC-CIGM for least-squares problems when the number of mini-batch groups m varies.

To validate our theoretical results, we consider the least-squares problem, which is the minimization problem over the objective function

$$f(x) = \frac{1}{2}\sum_{j=1}^{n}(a_j^T x - c_j)^2,$$

where we are given n training samples $(a_1, c_1), \ldots, (a_n, c_n)$ where $a_i \in \mathbb{R}^d$ is the training input with its associated output $c_i \in \mathbb{R}$.

The least-squares problem can be cast into the quadratic optimization problem (1) with

$$A = \sum_{j=1}^{n} a_j a_j^T, \text{and} \quad b = -\sum_{j=1}^{n} a_j c_j.$$

Alternatively, it can be formulated as the minimization problem with separable quadratic loss functions (6) with

$$A_i = \sum_{j=(i-1)\cdot B+1}^{i\cdot B} a_j a_j^T, \text{and} \quad b_i = \sum_{j=(i-1)\cdot B+1}^{i\cdot B} a_j c_j,$$

for $i = 1, 2, \ldots, m$, $B = m \cdot n$ is the mini-batch size, and m is the number of mini-batch groups. Clearly, $\mu = \lambda_{\min}(A)$, $L = \lambda_{\max}(A)$, $\bar{\mu} = \min_{i\in[1,m]} \lambda_{\min}(A_i)$ and $\bar{L} = \max_{i\in[1,m]} \lambda_{\max}(A_i)$.

We implemented the rounding quantizer, and all compressed gradient-based optimization algorithms in MATLAB. Each element of a_i is randomly drawn from $\mathcal{U}(0,1)$ and each element of $x^\star$ is drawn from $\mathcal{N}(0,1)$, and we set $b_i = a_i^T x^\star$. Therefore, $x^\star$ is the optimum to the least-squares problem. We set $n = 40000, d = 1000, x_0 = \mathbf{0}$, $\gamma = 1/L$ for CGD, EC-CGD, and $\gamma = 1/\bar{L}$ for CIGM and EC-CIGM.

From Figure 1, our theoretical bound for EC-CGD in Theorem 2 is shown to be tight, and confirms that EC-CGD produces significantly higher accurate solution than CGD. Figure 2 indicates that EC-CIGM improves convergence speed and solution accuracy especially when the number of mini-batch groups m increases.

6. CONCLUSIONS

Motivated by how error compensation improves performance of compressed optimization algorithms, this paper is the first in the literature which shows such theoretical supports. We analyze the convergence rates of compressed gradient descent and incremental gradient algorithms for strongly convex quadratic optimization. Our theoretical bounds explicitly show that the error compensation strategy significantly reduces the compression error especially for ill-conditioned problems, and have been validated to be tight in the numerical simulations on the least-squares problems.

7. REFERENCES

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic, "Qsgd: Communication-efficient

sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[2] Hongyi Wang, Scott Sievert, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright, "Atomo: Communication-efficient learning via atomic sparsification," *arXiv preprint arXiv:1806.04090*, 2018.

[3] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang, "Gradient sparsification for communication-efficient distributed optimization," *arXiv preprint arXiv:1710.09854*, 2017.

[4] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[5] Christopher De Sa, Megan Leszczynski, Jian Zhang, Alana Marzoev, Christopher R Aberger, Kunle Olukotun, and Christopher Ré, "High-accuracy low-precision training," *arXiv preprint arXiv:1803.03383*, 2018.

[6] Yusuke Tsuzuku, Hiroto Imachi, and Takuya Akiba, "Variance-based gradient compression for efficient distributed deep learning," *arXiv preprint arXiv:1802.06058*, 2018.

[7] Sindri Magnússon, Chinwendu Enyioha, Na Li, Carlo Fischione, and Vahid Tarokh, "Convergence of limited communications gradient methods," *IEEE Transactions on Automatic Control*, 2017.

[8] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson, "Distributed learning with compressed gradients," *arXiv preprint arXiv:1806.06573*, 2018.

[9] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing systems*, 2017, pp. 1509–1519.

[10] Michael G Rabbat and Robert D Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[11] Shengyu Zhu, Mingyi Hong, and Biao Chen, "Quantized consensus admm for multi-agent distributed optimization," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4134–4138.

[12] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[13] Nikko Strom, "Scalable distributed dnn training using commodity gpu cloud computing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[14] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with Memory," *ArXiv e-prints*, 2018.

[15] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, Nikola Konstantinov, and Cédric Renggli, "The convergence of sparsified gradient methods," *arXiv preprint arXiv:1809.10505*, 2018.

[16] Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang, "Error compensated quantized sgd and its applications to large-scale distributed optimization," *arXiv preprint arXiv:1806.08054*, 2018.

[17] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee, "Learning low precision deep neural networks through regularization," *arXiv preprint arXiv:1809.00095*, 2018.

[18] Steven H Low and David E Lapsley, "Optimization flow control—i: basic algorithm and convergence," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 6, pp. 861–874, 1999.

[19] Sindri Magnússon, Chinwendu Enyioha, Na Li, Carlo Fischione, and Vahid Tarokh, "Communication complexity of dual decomposition methods for distributed resource allocation optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 717–732, 2018.

[20] Yurii Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[21] Boris T Polyak, "Introduction to optimization. translations series in mathematics and engineering," *Optimization Software*, 1987.

[22] Dimitri P Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning*, vol. 2010, no. 1-38, pp. 3, 2011.