
Sparsified Distributed Adaptive Learning with Error Feedback

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 To be completed...

2 1 Introduction

3 Most modern machine learning tasks can be casted as a large finite-sum optimization problem writ-
4 ten as:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n f_i(\theta) \quad (1)$$

5 where n denotes the number of workers, f_i represents the average loss for worker i and θ the global
6 model parameter taking value in Θ , a subset of \mathbb{R}^d .

7 Some related work:

8 [18] develops variant of signSGD (as a biased compression schemes) for distributed optimization.
9 Contributions are mainly on this error feedback variant. In [26], the authors provide theoretical
10 results on the convergence of sparse Gradient SGD for distributed optimization (we want that for
11 AMS here). [27] develops a variant of distributed SGD with sparse gradients too. Contributions
12 include a memory term used while compressing the gradient (using top k for instance). Speeding up
13 the convergence in $\frac{1}{T^3}$.

14 2 Preliminaries

15 Sparse Optimization Methods.

16 **Distributed Learning.** When a large number of compute engines is available, being able to
17 train global machine learning models while mutualizing the available and *decentralized* source of
18 computation has been a growing focus for the community.

19 Decentralized optimization methods include methods such as ADMM [6], Distributed Subgradient
20 Descent [24], Dual Averaging [11], Prox-PDA [14], GNSD [21], and Choco-SGD [20].

21 A recent work [7], which focuses on adaptive gradient methods, namely the Adam [19] and the
22 AMSGrad [25] optimization methods, develops a decentralized variant of gradient based and adap-
23 tive methods in the context of gossip protocols. To date, very few contributions provided attempt
24 to efficiently run adaptive gradient method in such a distributed setting. Apart from [7], (author?)
25 [23] proposes a decentralized version of AMSGrad [25] which provably satisfies some non-standard
26 regret. Though, no sparsified variants of them have been proposed for practical purposes nor been
27 studied in the literature.

28 **Compression-Based Distributed Optimization.** While the capabilities of the compute powers
 29 is exploding, the communication complexity between either the central server and the decentralized
 30 workers or among workers is becoming ineffectively large [9, 22]. Gradient sparsification con-
 31 stitutes one popular method to induce sparsity through the optimization procedure and reduce the
 32 number of bits transmitted at each iteration. Extensive works have studied this technique to improve
 33 the communication efficiency of SGD-based methods such as distributed SGD. This large class of
 34 sparsification techniques include gradient quantization leveraging quantized vector of gradients in
 35 the communication phase [2, 29, 16, 28, 13, 8, 15], gradient sparsification generally selection top
 36 k components of the vector to be communicated, see [27, 1], or variants of the particular SGD al-
 37 gorithm such as low-precision SGD [4, 18] proposing a trade-off between communication cost and
 38 precision, and signSGD [10, 30] where only the signs of the gradient vectors are communicated.
 39 Most of these works apply to the SGD method [5] as a prototype where a novel method and some
 40 convergence results are presented with a rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ where T denotes the total number of itera-
 41 tions, see [3], thus achieving the same rate as plain SGD, see [12, 17].

42 Yet these communication reduction techniques, still presents a negative dependence on the number
 43 of workers, typically a linear dependence. Hence the need for even more efficient techniques which
 44 constitutes the object of our paper.

45 3 Method

46 Consider standard synchronous distributed optimization setting. AMSGrad is used as the prototype,
 47 and the local workers is only in charge of gradient computation.

48 3.1 TopK AMSGrad with Error Feedback

49 The key difference (and interesting part) of our TopK AMSGrad compared with the following arxiv
 50 paper “Quantized Adam” <https://arxiv.org/pdf/2004.14180.pdf> is that, in our model only
 51 gradients are transmitted. In “QAdam”, each local worker keeps a local copy of moment estimator
 52 m and v , and compresses and transmits m/v as a whole. Thus, that method is very much like the
 53 sparsified distributed SGD, except that g is changed into m/v . In our model, the moment estimates
 54 m and v are computed only at the central server, with the compressed gradients instead of the full
 55 gradient. This would be the key (and difficulty) in convergence analysis.

Algorithm 1 SPARS-AMS for Distributed Learning

```

1: Input: parameter  $\beta_1, \beta_2$ , learning rate  $\eta_t$ .
2: Initialize: central server parameter  $\theta_0 \in \Theta \subseteq \mathbb{R}^d$ ,  $e_{t,i} = 0$  the error accumulator for each
   worker; sparsity parameter  $k$ ;  $N$  local workers;  $m_0 = 0, v_0 = 0, \hat{v}_0 = 0$ 
3: for  $t = 1$  to  $T$  do
4:   parallel for worker  $i \in [n]$  do:
5:     Receive model parameter  $\theta_t$  from central server
6:     Compute stochastic gradient  $g_{t,i}$  at  $\theta_t$ 
7:     Compute  $\tilde{g}_{t,i} = \text{TopK}(g_{t,i} + e_{t,i}, k)$ 
8:     Update the error  $e_{t+1,i} = e_{t,i} + g_{t,i} - \tilde{g}_{t,i}$ 
9:     Send  $\tilde{g}_{t,i}$  back to central server
10:  end parallel
11:  Central server do:
12:     $\bar{g}_t = \frac{1}{N} \sum_{i=1}^N \tilde{g}_{t,i}$ 
13:     $m_t = \beta_1 m_{t-1} + (1 - \beta_1) \bar{g}_t$ 
14:     $v_t = \beta_2 v_{t-1} + (1 - \beta_2) \bar{g}_t^2$ 
15:     $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$ 
16:    Update global model  $\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{\hat{v}_t + \epsilon}}$ 
17: end for

```

56 3.2 Convergence Analysis

57 Several mild assumptions to make: Nonconvex and smooth loss function, unbiased stochastic gradi-
 58 ent, bounded variance of the gradient, bounded norm of the gradient, control of the distance between
 59 the true gradient and its sparse variant.

60 Check [7] starting with single machine and extending to distributed settings (several machines).

61 Under the distributed setting, the goal is to derive an upper bound to the second order moment of
 62 the gradient of the objective function at some iteration $T_f \in [1, T]$.

63 3.3 Mild Assumptions

64 We begin by making the following assumptions.

65 **Assumption 1.** (Smoothness) For $i \in \llbracket n \rrbracket$, f_i is L -smooth: $\|\nabla f_i(\theta) - \nabla f_i(\vartheta)\| \leq L \|\theta - \vartheta\|$.

66 **Assumption 2.** (Unbiased and Bounded gradient *per worker*) For any iteration index $t > 0$ and
 67 worker index $i \in \llbracket n \rrbracket$, the stochastic gradient is unbiased and bounded from above: $\mathbb{E}[g_{t,i}] =$
 68 $\nabla f_i(\theta_t)$ and $\|g_{t,i}\| \leq G_i$.

69 **Assumption 3.** (Bounded variance *per worker*) For any iteration index $t > 0$ and worker index
 70 $i \in \llbracket n \rrbracket$, the variance of the noisy gradient is bounded: $\mathbb{E}[|g_{t,i} - \nabla f_i(\theta_t)|^2] < \sigma_i^2$.

71 Denote by $Q(\cdot)$ the quantization operator Line 7 of Algorithm 1, which takes as input a gradient
 72 vector and returns a quantized version of it, and note $\tilde{g} := Q(g)$. Assume that

73 **Assumption 4.** (Bounded Quantization) For any iteration $t > 0$, there exists a constant $q > 0$ such
 74 that $\|g_t - \tilde{g}_t\| \leq q \|g_t\|$, where g_t is the stochastic gradient computed at iteration t .

75 Denote for all $\theta \in \Theta$:

$$f(\theta) := \frac{1}{n} \sum_{i=1}^n f_i(\theta), \quad (2)$$

76 where n denotes the number of workers.

77 3.4 Intermediary Lemmas

78 **Lemma 1.** Under Assumption 1 to Assumption 4, with a decreasing sequence of stepsize $\{\eta_t\}_{t>0}$, we
 79 have:

$$\mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} \bar{g}_t \rangle] \leq \textcolor{red}{TBD} \quad (3)$$

80 where \mathbf{I}_d is the identity matrix, \hat{V}_t the diagonal matrix which diagonal entries are $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$
 81 defined Line 15 of Algorithm 1 and \bar{g}_t is the aggregation of all **quantized** gradients from the workers.

82 *Proof.* Decompose \bar{g}_t as the unbiased stochastic gradients and quantized versions. \square

83 **Lemma 2.** Under Assumption 1 to Assumption 4, with a decreasing sequence of stepsize $\{\eta_t\}_{t>0}$, we
 84 have:

$$\begin{aligned} \mathbb{E}[f(\theta_{t+1}) - f(\theta_t)] &\leq \textcolor{red}{bound of Lemma 1} - \eta_{t+1} \beta_1 \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle] \\ &\quad + \left(\frac{L}{2} + \beta_1 L \right) \|\theta_t - \theta_t\|^2 \\ &\quad + \eta_{t+1} G^2 \mathbb{E} \left[\sum_{j=1}^d \left[(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2} \right] \right] \end{aligned} \quad (4)$$

85 where d denotes the dimension of the parameter vector

86 *Proof.* We first define multiple auxiliary sequences. For the first moment, define

$$\begin{aligned}\bar{m}_t &= m_t + \mathcal{E}_t, \\ \mathcal{E}_t &= \beta_1 \mathcal{E}_{t-1} + (1 - \beta_1)(e_{t+1} - e_t),\end{aligned}$$

87 such that

$$\begin{aligned}\bar{m}_t &= \bar{m}_t + \mathcal{E}_t \\ &= \beta_1(m_t + \mathcal{E}_t) + (1 - \beta_1)(\bar{g}_t + e_{t+1} - e_1) \\ &= \beta_1 \bar{m}_{t-1} + (1 - \beta_1)g_t.\end{aligned}$$

88 Denote the following auxiliary variables at iteration $t + 1$

$$z_{t+1} = \theta_{t+1} + \frac{\beta_1}{1 - \beta_1}(\theta_{t+1} - \theta_t) \quad (5)$$

89 By assumption Assumption 1, we can write the smoothness condition on the overall objective (2),
90 between iteration t and $t + 1$:

$$f(\theta_{t+1}) \leq f(\theta_t) + \langle \nabla f(\theta_t) | \theta_{t+1} - \theta_t \rangle + \frac{L}{2} \|\theta_{t+1} - \theta_t\| \quad (6)$$

91 Denote by \hat{V}_t the diagonal matrix which diagonal entries are $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$ defined Line 15 of
92 Algorithm 1. Hence, we obtain,

$$f(\theta_{t+1}) \leq f(\theta_t) - \eta_{t+1} \left\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} \right\rangle + \frac{L}{2} \|\theta_{t+1} - \theta_t\| \quad (7)$$

93 where \mathbf{I}_d denotes the identity matrix.

94 We now take the expectation of those various terms conditioned on the filtration \mathcal{F}_t of the total
95 randomness up to iteration t .

$$\mathbb{E}[f(\theta_{t+1}) - f(\theta_t)] \leq -\eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} \rangle] + \frac{L}{2} \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2] \quad (8)$$

96 We now focus on the computation of the inner product obtained in the equation above. We have

$$\begin{aligned}& \eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} \rangle] \\ &= \eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} + (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} - (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} \rangle] \\ &= \eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_{t+1} \rangle] + \eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | [(\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} - (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2}] m_{t+1} \rangle] \\ &= \eta_{t+1} \beta_1 \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle] + \eta_{t+1} (1 - \beta_1) \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} \bar{g}_t \rangle] \\ &\quad + \eta_{t+1} \mathbb{E}[\langle \nabla f(\theta_t) | [(\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} - (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2}] m_{t+1} \rangle] \quad (10)\end{aligned}$$

97 where \bar{g}_t is the aggregated gradients from all workers.

98 Plugging the above in (8) yields:

$$\begin{aligned}& \mathbb{E}[f(\theta_{t+1}) - f(\theta_t)] \\ & \leq \underbrace{-\beta_1 \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle]}_{A_t} \eta_{t+1} \\ & \quad \underbrace{-\mathbb{E}[\langle \nabla f(\theta_t) | [(\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} - (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2}] m_{t+1} \rangle]}_{B_t} \eta_{t+1} \\ & \quad \underbrace{-(1 - \beta_1) \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} \bar{g}_t \rangle]}_{C_t} \eta_{t+1} + \frac{L}{2} \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2] \quad (11)\end{aligned}$$

99 To begin with, by the tower rule, we have that

$$A_t = -\beta_1 \mathbb{E}[\mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle | \mathcal{F}_t]] \quad (12)$$

$$= -\beta_1 \langle \nabla f(\theta_{t-1}) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle - \beta_1 \langle \nabla f(\theta_t) - \nabla f(\theta_{t-1}) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle \quad (13)$$

$$(14)$$

where we recognize the first term as the term in (9), at iteration $t - 1$ and hence apply the same decomposition as in (10). Coupling with the smoothness of f , which gives that

$$-\beta_1 \langle \nabla f(\theta_t) - \nabla f(\theta_{t-1}) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle \leq \frac{\beta_1 L}{\eta_{t-1}} \|\theta_t - \theta_{t-1}\|^2$$

100 we obtain,

$$A_t = -\beta_1 \mathbb{E}[\mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle | \mathcal{F}_t]] \quad (15)$$

$$\leq \eta_{t+1} \beta_1 (A_{t-1} + B_{t-1} + C_{t-1}) + \eta_{t+1} \frac{\beta_1 L}{\eta_{t-1}} \|\theta_t - \theta_{t-1}\|^2 \quad (16)$$

101 Then,

$$B_t = -\mathbb{E}[\langle \nabla f(\theta_t) | [(\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} - (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2}] m_{t+1} \rangle] \quad (17)$$

$$= \mathbb{E}[\sum_{j=1}^d \nabla^j f(\theta_t) m_{t+1}^j [(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2}]] \quad (18)$$

$$\stackrel{(a)}{\leq} \mathbb{E}[\|\nabla f(\theta_t)\| \|m_{t+1}\| \sum_{j=1}^d [(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2}]] \quad (19)$$

$$\stackrel{(b)}{\leq} G^2 \mathbb{E}[\sum_{j=1}^d [(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2}]] \quad (20)$$

102 where $\nabla^j f(\theta_t)$ denotes the j -th component of the gradient vector $\nabla f(\theta_t)$, (a) uses of the Cauchy-
 103 Schwartz inequality and (b) boils down from the norm of the gradient vector boundedness assump-
 104 tion 2, denoting $G := \frac{1}{n} \sum_{i=1}^n G_i$.

105 Plugging the above into (11) yields

$$\begin{aligned} \mathbb{E}[f(\theta_{t+1}) - f(\theta_t)] &\leq \eta_{t+1} (A_t + B_t + C_t) + \frac{L}{2} \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2] \\ &\leq -\eta_{t+1} \beta_1 \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle] \\ &\quad + \eta_{t+1} G^2 \mathbb{E}[\sum_{j=1}^d [(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2}]] \\ &\quad + \left(\frac{L}{2} + \eta_{t+1} \frac{\beta_1 L}{\eta_{t-1}} \right) \|\theta_t - \theta_{t-1}\|^2 \\ &\quad - \eta_{t+1} (1 - \beta_1) \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} \bar{g}_t \rangle] \end{aligned} \quad (21)$$

106 **Bound $T_1 = \mathbb{E}[\langle \nabla f(\theta_t) | (\hat{V}_{t+1} + \epsilon \mathbf{I}_d)^{-1/2} \bar{g}_t \rangle]$ with an intermediate Lemma**

107 Under the assumption that we use a decreasing stepsize such that $\eta_{t+1} \leq \eta_t$, and given that according
 108 to Line 15 we have that $\hat{v}_{t+1} \geq \hat{v}_t$ by construction, we obtain

$$\begin{aligned}
 \mathbb{E}[f(\theta_{t+1}) - f(\theta_t)] &\leq \textcolor{red}{bound of T1} - \eta_{t+1} \beta_1 \mathbb{E}[\langle \nabla f(\theta_t) \mid (\hat{V}_t + \epsilon \mathbf{I}_d)^{-1/2} m_t \rangle] \\
 &\quad + \left(\frac{L}{2} + \beta_1 L \right) \|\theta_t - \theta_t\|^2 \\
 &\quad + \eta_{t+1} G^2 \mathbb{E} \left[\sum_{j=1}^d \left[(\hat{v}_{t+1}^j + \epsilon)^{-1/2} - (\hat{v}_t^j + \epsilon)^{-1/2} \right] \right]
 \end{aligned} \tag{22}$$

109 Finally, using Lemma ??, we obtain the desired result.

110

□

4 Experiments

Our proposed TopK-EF with AMSGrad matches that of full AMSGrad, in distributed learning. Number of local workers is 20. Error feedback fixes the convergence issue of using solely the TopK gradient.

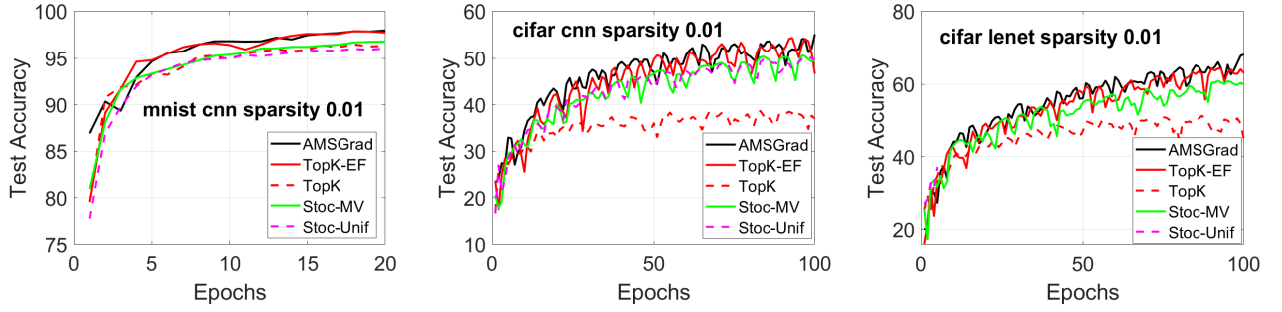


Figure 1: Test accuracy.

5 Conclusion

References

- [1] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.
- [2] 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*, pages 1709–1720, 2017.
- [3] 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.
- [4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.
- [5] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 161–168. Curran Associates, Inc., 2008.
- [6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [7] Congliang Chen, Li Shen, Haozhi Huang, Qi Wu, and Wei Liu. Quantized adam with error feedback. *arXiv preprint arXiv:2004.14180*, 2020.
- [8] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. *Sensors*, 10(12):11259–11273, 2010.
- [9] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *Symposium on Operating Systems Design and Implementation*, pages 571–582, 2014.
- [10] Christopher De Sa, Matthew Feldman, Christopher Ré, and Kunle Olukotun. Understanding and optimizing asynchronous low-precision stochastic gradient descent. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 561–574, 2017.
- [11] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.
- [12] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [13] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization. In *International Conference on Machine Learning*, pages 2545–2554. PMLR, 2019.
- [14] Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pages 1529–1538, 2017.
- [15] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.
- [16] Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2530–2541, 2018.

- [17] Belhal Karimi, Blazej Miasojedow, Eric Moulines, and Hoi-To Wai. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974. PMLR, 2019.
- [18] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. *arXiv preprint arXiv:1901.09847*, 2019.
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [20] Anastasia Koloskova, Sebastian U Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487, 2019.
- [21] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321, 2019.
- [22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [23] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.
- [24] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.
- [25] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.
- [26] Shaohuai Shi, Kaiyong Zhao, Qiang Wang, Zhenheng Tang, and Xiaowen Chu. A convergence analysis of distributed sgd with communication-efficient gradient sparsification. In *IJCAI*, pages 3411–3417, 2019.
- [27] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.
- [28] Jianqiao Wangni, Jiale Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*, 2017.
- [29] 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. *arXiv preprint arXiv:1705.07878*, 2017.
- [30] Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Chris De Sa. Swalp: Stochastic weight averaging in low precision training. In *International Conference on Machine Learning*, pages 7015–7024. PMLR, 2019.

