Sparsified Distributed Adaptive Learning with Error Feedback

Anonymous Author(s)
Affiliation
Address
email

Abstract

To be completed...

1 Introduction

Some related work:

[17] develops variant of signSGD (as a biased compression schemes) for distributed optimization. Contributions are mainly on this error feedback variant. In [24], the authors provide theoretical results on the convergence of sparse Gradient SGD for distributed optimization (we want that for AMS here). [25] develops a variant of distributed SGD with sparse gradients too. Contributions include a memory term used while compressing the gradient (using top k for instance). Speeding up the convergence in $\frac{1}{T^3}$.

2 Preliminaries

Sparse Optimization Methods.

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

Decentralized optimization methods include methods such as ADMM [6], Distributed Subgradient Descent [22], Dual Averaging [10], Prox-PDA [13], GNSD [20], and Choco-SGD [19].

A recent work [7], which focuses on adaptive gradient methods, namely the Adam [18] annd the AMSGrad [23] optimization methods, develops a decentralized variant of gradient based and adaptive methods in the context of gossip protocols. To date, very few contributions provided attempt to efficiently run adaptive gradient method is such a distributed setting. Apart from [7], **?**] proposes a decentralized version of AMSGrad [23] which provably satisfies some non-standard regret. Though, no sparsified variants of them have been proposed for practical purposes nor been studied in the literature.

Sparse Distributed Optimization. While the capabilities of the compute powers is exploding, the communication complexity between either the central server and the decentralized workers or among workers is becoming ineffectively large [**?** 21]. Gradient sparsification constitutes one popular method to induce sparsity through the optimization procedure and reduce the number of bits transmitted at each iteration. Extensive works have studied this technique to improve the communication efficiency of SGD-based methods such as distributed SGD. This large class of sparsification techniques include gradient quantization leveraging quantized vector of gradients in the communication phase [2, 27, 15, 26, 12, 8, 14], gradient sparsificaiton generally selection top k components

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

of the vector to be communicated, see [25, 1], or variants of the particular SGD algorithm such as low-precision SGD [4, 17] proposing a trade-off between communication cost and precision, and signSGD [9, 28] where only the signs of the gradient vectors are communicated. Most of these works apply to the SGD method [5] as a prototype where a novel method and some convergence results are presented with a rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ where T denotes the total number of iterations, see [3], thus achieving the same rate as plain SGD, see [11, 16].

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

3 Method

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

3.1 TopK AMSGrad with Error Feedback

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

Algorithm 1 SPARS-AMS for Federated Learning

Input: parameter β_1, β_2, learning rate η_t.
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; $m_0 = 0$, $v_0 = 0$, $\hat{v}_0 = 0$
for $t = 1$ to T **do**
  parallel for worker $i \in [n]$ **do**:
    Receive model parameter θ_{t-1} from central server
    Compute stochastic gradient $g_{t,i}$ at θ_t
    Compute $\tilde{g}_{t,i} = TopK(g_{t,i} + e_{t,i}, k)$
    Update the error $e_{t+1,i} = e_{t,i} + g_{t,i} - \tilde{g}_{t,i}$
    Send $\tilde{g}_{t,i}$ back to central server
  end parallel
  Central server do:
  $\bar{g}_t = \frac{1}{N}\sum_{i=1}^{N} \tilde{g}_{t,i}$
  $m_t = \beta_1 m_{t-1} + (1 - \beta_1)\bar{g}_t$
  $v_t = \beta_2 v_{t-1} + (1 - \beta_2)\bar{g}_t^2$
  $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$
  Update global model $\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{\hat{v}_t}}$
end for

3.2 Convergence Analysis

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

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

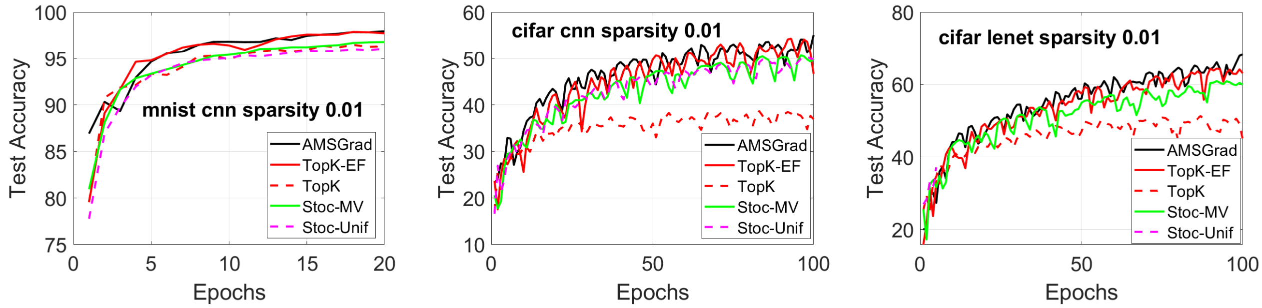

Figure 1: Test accuracy.

3.2.1 Single machine

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

We first define multiple auxiliary sequences. For the first moment, define

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

such that

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

3.2.2 Multiple machine

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.

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] 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.

[10] 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.

[11] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] 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.

[20] 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.

[21] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[22] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.

[23] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[24] 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.

[25] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

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

[27] 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.

[28] 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.

A Appendix