

# Sparsified Distributed Adaptive Learning with Error Feedback

## Abstract

To be completed...

## 1 Introduction

## 2 Method

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

### 2.1 TopK AMSGrad with Error Feedback

References:

[1] [2] [3] <https://arxiv.org/pdf/1901.09847.pdf> <https://proceedings.neurips.cc/paper/2018/file/b440509a0106086a67bc2ea9df0a1dab-Paper.pdf> [https://pdfs.semanticscholar.org/8728/dee89906022c1d4f5c.pdf?\\_ga=2.152244026.2027005181.1606271153-15127215.1603945483](https://pdfs.semanticscholar.org/8728/dee89906022c1d4f5c.pdf?_ga=2.152244026.2027005181.1606271153-15127215.1603945483)

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

---

- 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**  $i$  **do:**
  - 5:     Receive model parameter  $\theta_{t-1}$  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  $e_{t+1,i} = e_{t,i} + g_{t,i} - \tilde{g}_{t,i}$
  - 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 model  $\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{\hat{v}_t}}$
  - 17: **end for**
-

## 2.2 Convergence Analysis

Nonconvex smooth loss function. Bounded gradient variance.

### 2.2.1 Single machine

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

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

TBD...

### 2.2.2 Multiple machine

## 3 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.

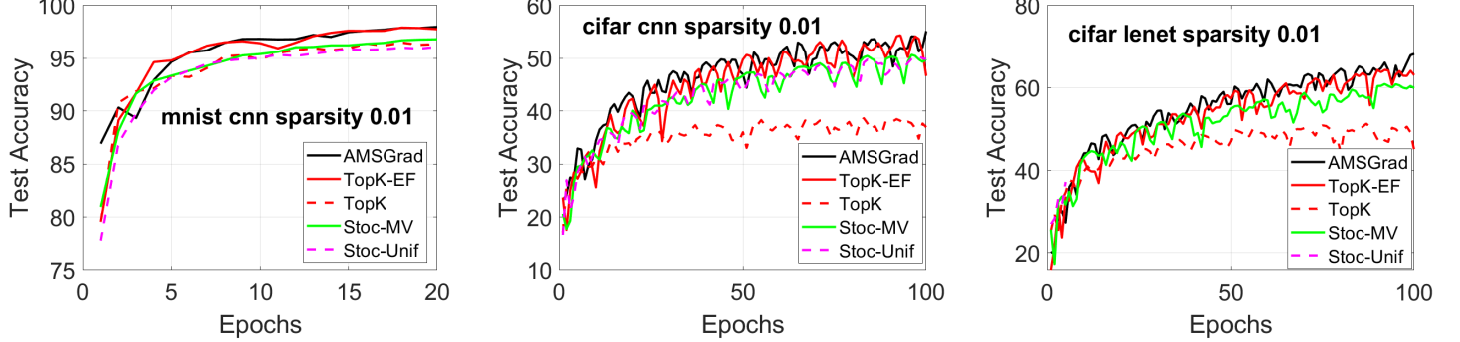


Figure 1: Test accuracy.

## 4 Conclusion

## References

- [1] 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.
- [2] 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.
- [3] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

## A Appendix