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:

https://arxiv.org/pdf/1901.09847.pdf https://proceedings.neurips.cc/paper/2018/file/b440509a0106086a6 pdf https://pdfs.semanticscholar.org/8728/dee89906022c1d4f5c1de1233c3f65ab92f2.pdf?_ga=2.152244026.2027005181.1606271153-15127215.1603945483

The key difference (and interesting part) of our TopK AMSGrad comprared 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 L&D Local 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 **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 $e_{t+1,i} = e_{t,i} + g_{t,i} - \tilde{g}_{t,i}$
    Send $\tilde{g}_{t,i}$ back to central server
  end parallel
  Central server do:
  $\bar{g}_t = \frac{1}{N}\sum_{i=1}^{N}\tilde{g}_{t,i}$
  $m_t = \beta_1 m_{t-1} + (1-\beta_1)\bar{g}_t$
  $v_t = \beta_2 v_{t-1} + (1-\beta_2)\bar{g}_v^2$
  $\hat{v}_t = \max(v_t, \hat{v}_{t-1})$
  Update model$\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{\hat{v}_t}}$
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

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

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.

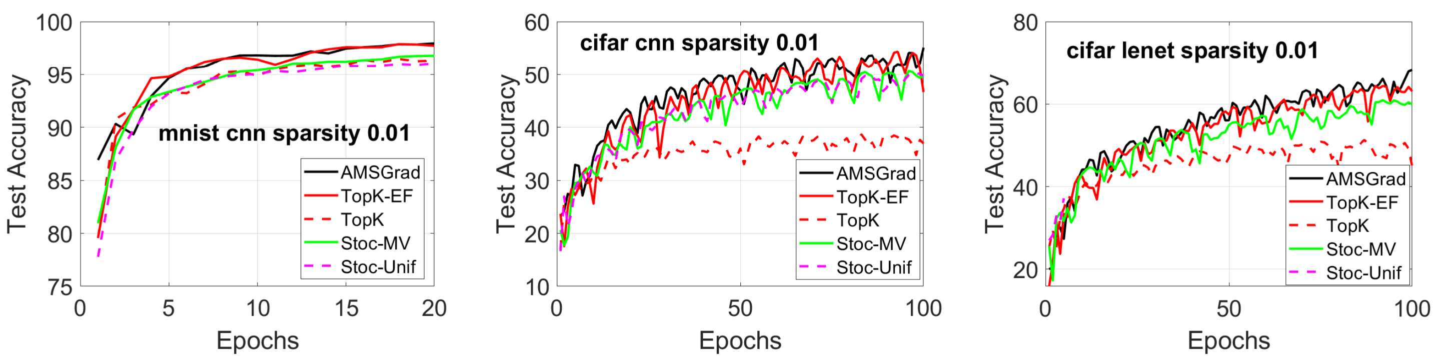

Figure 1: Test accuracy.

4 Conclusion

A Appendix