COMPRESSING GRADIENT OPTIMIZERS VIA COUNT-SKETCHES

Ryan Spring, Anastasios Kyrillidis, Vijai Mohan, Anshumali Shrivastava
Rice University, Amazon Search

ICML 2019

Deep Learning is Resource Intensive

Training deep learning models requires large amounts of time and resources

Data-Parallelism for faster training!

A key tool for reducing training time is to increase the batch size

Data Parallelism – Memory Limitations

Increasing the batch size requires significant amounts of memory

Faster Training vs. Expressive Model

Sacrifice batch size for a larger, more expressive model

Pesky Popular Optimizers

- The auxiliary parameters used by popular optimizers aggravate the memory issue
- i.e. Adam, RMSProp, Adagrad, Momentum

Optimizers – A Concrete Example

- Training BERT Transformer on Nvidia V100 16GB*
- SGD: 10,800 MB, Adam: 13,362 MB
- Auxiliary variables require **2,562 MB extra memory!**

*Using Activation Checkpointing and Mixed Precision Training

Our Goal

- Compress the auxiliary variables
- Maintain convergence rate and accuracy of the full-sized optimizer

Count-Sketches to the Rescue!

- **Solution:** Compress the auxiliary variables with count-sketches
- **Intuition:** Map multiple model parameters to the same parameter in the count-sketch
- **Outcome:** Free memory for more expressive model and/or larger batch size

Highlighted Result - LSTM – LM1B

Metric	**Adam**	**Count-Sketch**
Time (Hrs)	**5.28**	5.42
Size (MB)	10,813	**7,693**
Test Perplexity	**39.90**	40.55

- Count-Sketch optimizer used 5x fewer parameters
- **Upshot:** Reduced memory usage with minimal accuracy or performance loss

Please visit the poster today!

6:30pm @ Pacific Ballroom #83

GitHub: https://github.com/rdspring1/Count-Sketch-Optimizers