Gradient Quantization for Data-Parallel DNN Training

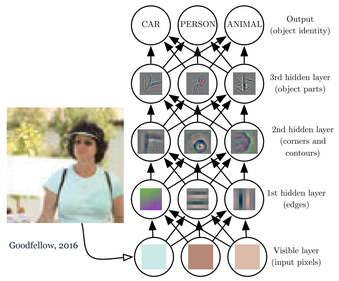
Nikoli Dryden[1,2], Brian Van Essen[2], and Marc Snir[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign [2]Lawrence Livermore National Laboratory

Motivation

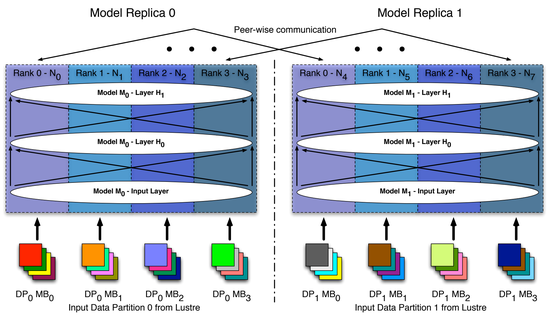
Deep Neural Networks

- Responsible for state-of-the-art results in object detection and recognition, translation, speech recognition, genomics, etc.



- DNNs use many layers of neurons to build hierarchical representations to learn complex data
- Training DNNs requires repeatedly presenting the network with many examples

HPC for DNN training



- Make use of HPC resources to train large DNNs fast and efficiently
- Existing work focuses on heterogeneous cloud computing or small clusters
- Update involves an allreduce of every replica's updates
- Training is **bandwidth bound** for large models

Solution: Quantized allreduce

- DNNs are very robust to noise
- We can **approximate** the gradient signal from each model's updates by quantizing it
  - Reduces amount of data transferred
  - If done well, this does not impact the final model accuracy
- **Trade increased computation for reduced communication**
- Approaches:
  - Onebit quantization (Seide et al.)
  - **Adaptive quantization (new, Dryden et al.)**
  - Baseline: MPI_Allreduce, no quantization

Adaptive quantization

- Gradients are computed and stored as a *nxm* matrix of 32-bit floats
- That much precision is not necessary!
- Instead:
  - Don't send unnecessary gradients
  - Use 1 bit to send the remaining gradients
- Work on each column of the gradient matrix separately (helps reduce error)
- Draw on rate-distortion theory for guidance
  - Select 2 regions (as we have 1 bit)
  - Determine reconstruction values for each region
- Input parameter: π, the *proportion* of gradients to send
  - Enables tuning how much data is sent to problem characteristics

Which gradients to send? (Regions)

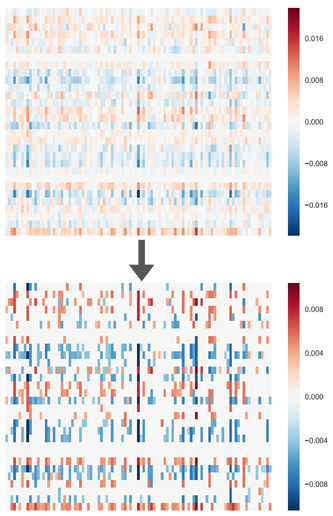
- Key idea: Large gradients are most important
- Choose thresholds τ^+ and τ^- to describe the regions:
  - Send every gradient $\geq\tau^+$ or $\leq\tau^-$
  - The resulting gradients are now *sparse*
- Use a selection algorithm to find the $(n\text{-}n/\pi)$th largest gradient
- Choose these over the *entire* gradient matrix to maximize the gradients sent

What values should the sent gradients take? (Reconstruction)

- Once regions are selected, the mean-squared error is minimized by choosing the mean of each region

Representation

- Need to send the chosen gradients plus the reconstruction values for each column
- Use a variant of the compressed sparse column format
- Metadata kept in a header
- Use a 15-bit row index and 1-bit gradient for each value
- Data volume is $\sim 1/(2\pi)$ less

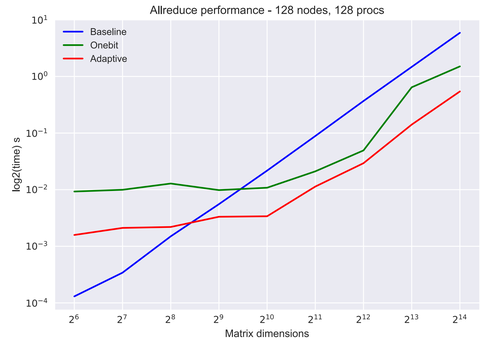


Example gradient and adaptively-quantized gradient matrix (π=4). Note each column now has two different values and some contain no values at all.

Results

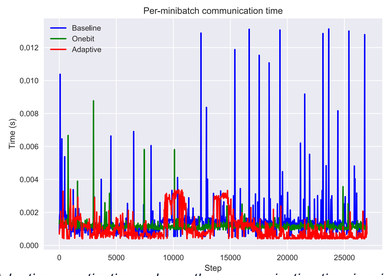
Synthetic allreduce benchmark

- Simulate large-scale data-parallelism
- Adaptive quantization is superior once matrices are moderately large, **~10.9x faster** at largest scale

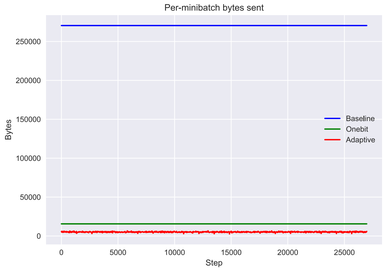


MNIST and ImageNet

- Evaluate using a simple benchmark on MNIST and ImageNet datasets
- 3 4096-neuron dense layers, ReLU activations, Adam optimizer (Adagrad for onebit quantization)
- 4 model replicas
  - Used 16 nodes total for MNIST, 64 for ImageNet
  - 24 cores/node
- No hyperparameter tuning



Above: Adaptive quantization reduces the communication time in minibatches. Below: Onebit and adaptive quantization both significantly reduce data sent.



Below: Average total time for each minibatch for the MNIST and ImageNet models, and the speedup of adaptive quantization over the baseline. Typical training involves tens of thousands to millions of minibatches.

	MNIST	ImageNet
Baseline	0.053 s	2.69 s
Adaptive	0.038 s	2.00 s
Speedup	1.39x	1.34x

Accuracy

- **Quantization does not impact final model accuracy**
- Better models are in-progress
- Onebit quantization can work well, but did not converge with our hyperparameter settings and does not support the Adam optimizer
- See paper for some additional (older) results

	MNIST	ImageNet
Baseline	98.31%	36.4%
Adaptive	98.32%	36.4%

Above: Test accuracies of our model on the MNIST and ImageNet datasets after 20 epochs of training. ImageNet accuracy is top-1.

Conclusions and future work

- Adaptive quantization successfully trades communication for increased computation to help data-parallel training scale
- Future work:
  - GPU support
  - Better allreduce algorithms
  - More complex DNN models
  - More scalable training (SGD doesn't scale)
- Code available as part of LBANN: https://github.com/LLNL/lbann

Acknowledgements

- This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-POST-729415)
- Experiments were performed using Livermore Computing facility resources
- The LBANN Team

References

Nikoli Dryden, Tim Moon, Sam Ade Jacobs, and Brian Van Essen. "Communication Quantization for Data-parallel Training of Deep Neural Networks." MLHPC, 2016.
Frank Seide, et al. "1-bit Stochastic Gradient Descent and its Application to Data-parallel Distributed Training of Speech DNNs." INTERSPEECH, 2014.
Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep Learning." MIT Press, 2016.
Thomas Cover and Joy Thomas. "Elements of Information Theory," 2ed. John Wiley & Sons, 2012.

Lawrence Livermore National Laboratory

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN