Generalized Distributed-Memory Convolutional Neural Networks for Large-Scale Parallel Systems

Naoya Maruyama¹, Nikoli Dryden¹,², Tim Moon¹, Brian Van Essen¹, and Mark Snir² (1: LLNL, 2: UIUC)

Abstract: Large-scale machines such as LLNL’s Sierra present a tremendous amount of compute capacity, and are considered an ideal platform for training deep neural networks. We present a new generalized distributed training framework that aims to exploit such large scale systems more effectively.

COMPUTATIONAL CHALLENGES IN CNN TRAINING

Limited Parallel Scalability
• CNN training is a compute-intensive problem, yet, its distributed memory parallelism is not fully exploited
• State-of-the-art parallel training typically uses data parallelism, which is limited by mini-batch sizes (O(100)-O(1000))

Limited Model Scalability
• Memory capacity, esp. that of fast stacked memories, has not been growing fast enough
• The demand for larger memory capacity is growing very rapidly
• Simple mesh tiling model would require O(10) GB just for one sample \(\rightarrow\) Unlikely to fit device memory on Sierra
• Higher resolution input/output with deeper networks

APPROACH: GENERALIZED PARALLELIZATION

• Parallels along any of sample, channel, filter and spatial domains.
• Supports halo exchanges in spatial domains, implemented with a custom GPU-centric communication library within a node and with MPI across nodes

METHOD

Distributed Multidimensional Tensors
• Allows partitioning along any of sample, channel, filter and spatial domains.
• Supports halo exchanges in spatial domains. Implemented with a custom GPU-centric communication library within a node and with MPI across nodes

Distributed GPU Convolutions
• Communicates halo data when spatial domains are partitioned
• Uses cuDNN for local sub tensors

An Extended LBANN Training Framework
• LBANN is an MPI-based distributed deep learning framework supporting data-parallel convolutions with parallel CPUs/GPUs

PRELIMINARY RESULTS

• Compares training throughputs of the extended LBANN on an IBM P8 node with 4 Tesla P100 GPUs
• Uses a Resnet-like model, consisting of a series of convolutions, batch normalization, and ReLU, with the ImageNet dataset
• Measurement only includes forward propagation through the above layers, and does not include back prop and I/O
• Hybrid parallel partitions the sample and height dimensions into half, respectively

Convolution performance with spatial partitioning
• Up to 4 Tesla P100 GPUs on an IBM Power8 node
• Partitioned only along the second slowest-changing dimension

HxW: 32x32, N: 8, C=F=64

HxW: 64x64, N: 16, C=F=64

HxW: 128x128, N: 16, C=F=64

HxW: 128x128, N: 32, C=F=64

CONCLUSION

• A new CNN training approach that aims to exploit all dimensions of parallelism
• Preliminary evaluation confirms expected performance characteristics

Ongoing work:
• Full-model performance evaluation
• Spatial parallelization over multi-node GPUs
• Channel/filter parallelization
• Performance modeling

FURTHER INFORMATION

• Contact: Naoya Maruyama (maruyama3@llnl.gov)
• Livermore Big Artificial Neural Networks (LBANN): https://github.com/llnl/lbann