Accelerating Deep Neural Networks

Aumit Leon ’19.5
aleon@middlebury.edu

CS 702 – A Thesis Presented to the Computer Science Department at Middlebury College

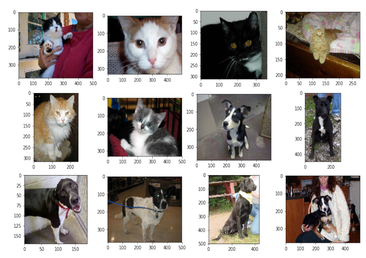
ABSTRACT

Deep neural networks (DNNs) are a class of machine learning methods that have been successfully applied to domains with large, labelled datasets. While access to data in domains such as image classification is increasing, the computational power required to train such large DNNs is not ubiquitous. Accelerating DNN training would increase the accessibility of deep learning methods across device topologies, which effectively increases scalability, reproducibility, and oversight. Parallelization methods have been developed to tackle this issue. In this thesis, I present a survey of general and expert designed parallelization methods, as well as generalizable frameworks that attempt to accelerate training times for AlexNet – a Convolutional Neural Network used in image classification tasks.

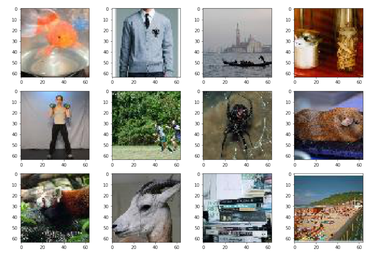
BENCHMARKS

Datasets:

Dogs vs. Cats (Kaggle)

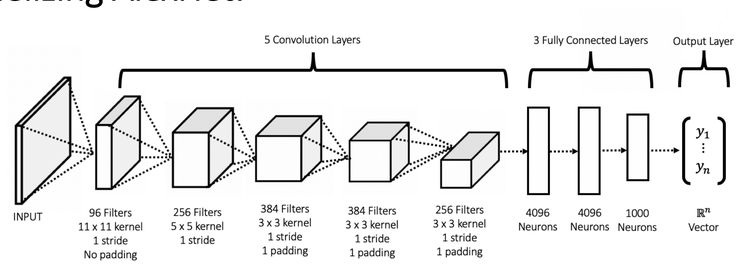


Tiny ImageNet (Stanford)



- Tiny ImageNet has 200 classes with 120,000 total images.
- Dogs vs. Cats has 2 classes, with 25,000 total images.

Parallelizing AlexNet:



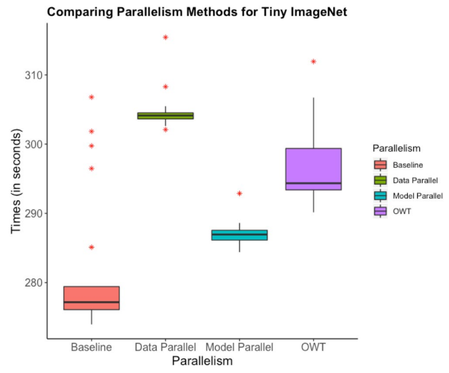
- 5 convolution, 3 max pooling, and 3 fully connected layers.
- Tiny ImageNet input images scaled to 56 x 56 x 3.
- Dogs vs. Cats input images scaled to 128 x 128 x 3.
- Trained over 20 epochs for each dataset using each parallelization method.

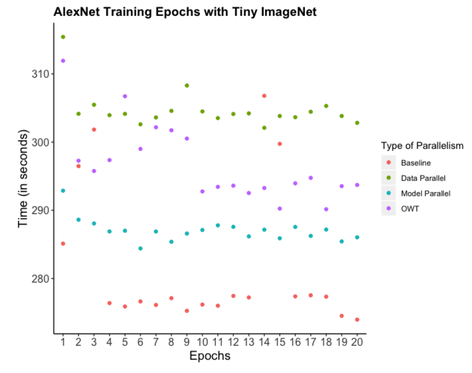
Parallelization Methods:

- ***Baseline:*** Training the model architecture with the default TensorFlow operation allocation strategy.
- ***Data Parallelism:*** Full AlexNet replica on each GPU, each GPU processes an equal sized subset of the data.
- ***Model Parallelism:*** Convolution, batch normalization, and fully connected layers on GPU, padding and max pooling on the CPU.
- ***One Weird Trick (OWT):*** Model parallelism for fully connected layers, data parallelism for convolution layers.

RESULTS

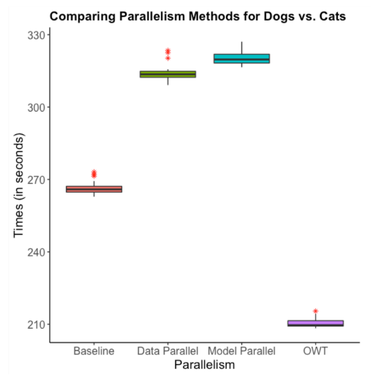
Tiny ImageNet Results:

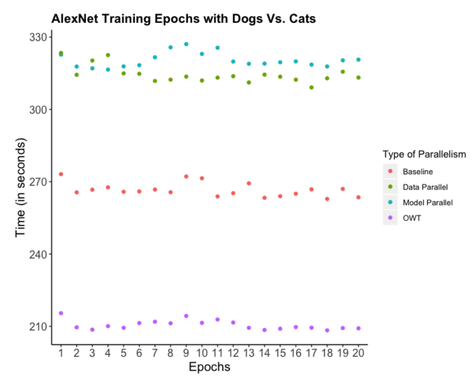




- Baseline method out performs parallelization methods as a result of smaller image sizes.
- Communication overhead associated with parallelization methods does not yield a speedup with these images and batch sizes.

Dogs vs. Cats Results:

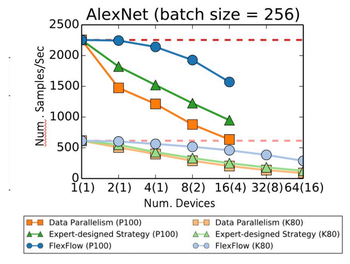




- OWT method has a 1.2x speedup over the baseline AlexNet implementation.
- Model and data parallelism applied in isolation are not entirely effective due to the communication overhead associated with these methods.

Comparing with FlexFlow:

Num. GPUs	AlexNet Full	AlexNet Delta	AlexNet Speedup
4	0.11	0.04	**2.9**×
8	0.40	0.13	**3.0**×
16	1.4	0.48	**2.9**×
32	5.3	1.8	**3.0**×
64	18	5.9	**3.0**×



Beyond Data and Model Parallelism for Deep Neural Networks, Jia et al.

- Trained on the full ImageNet dataset (14 million images), FlexFlow outperforms data parallel and expert designed methods (OWT) on P100 GPUs.
- FlexFlow is architecture agnostic, and yields up to a 2.9x speedup across 4 GPUs.

CONCLUSIONS

- Deep neural networks can be accelerated through parallelism.
- Expert designed methods are useful, but don’t scale.
- Efficient device usage makes deep learning more accessible.
- Parallelism should be treated as an abstraction, as in FlexFlow.