Project Proposal: **Image Classification using Multi-threading and ResNet-50**.

Durkesh Kumaaran Jevanandem | Goutham Saravanan

Nov 1, 2021

1. Methodology:

Image Segmentation from Aerial Imagery is a challenging task. Obstruction from nearby trees, shadows of adjacent buildings, varying texture and color of roads, road class imbalance (due to relatively few road image pixels) are among other challenges that hinder present day models in segmenting sharp road boundaries that extend from one end of the image to the other. High-quality aerial imagery datasets facilitate comparisons of existing methods and lead to increased interest in aerial imagery applications in the machine learning and computer vision communities.

For this project, we are interested in using TensorFlow/ PyTorch to train ResNet-50 on the ImageNet data set to be used to classify images. I hope this project will serve as an introduction to deep learning and an exploration to scaling the training process.

CUDA: As the deep learning problems we are trying to solve continue to grow and grow, training these deep learning models on becomes incredibly expensive in terms of computation and time. While TensorFlow offers some built in support for multi-node/multi-gpu, benchmarks run by Uber show that the marginal speedup gained from adding more GPUs plateaus for larger numbers of GPUs

2. Input Dataset:

Kaggle Dataset with .png/.csv files with RGB pixel values in GB size.

3. Platform & ML Algo's used:

- Pytorch/Tensorflow
- CUDA
- Jupyter Notebook
- Keras
- Numpy
- OpenCV

4. Experiments:

For experimentation, we will compare the amount of time it takes to train a ResNet-50 ImageNet Model through TensorFlow/ Pytorch on various compute configurations using CUDA to achieve scalability in the training process. In particular, we plan to collect timing data for 2,4 GPU's. In these experiments, all configurations are trained through to the end so we'd expect to see similar performance from the resulting weights. The major difference between the configurations is how quickly (wall time), did the model take to train. If the training times for the neural nets take too long to run a reasonable number of experiments, I will instead change models to a fixed wall time run, stopping the training process epoch after 1 the wall time elapses. Under different hardware configurations, I would expect more powerful configurations to reach further epochs and as a result, perform better in training.

5. Respective grades to results we achieve:

A: If we show an increase in the training speed as the number of GPUs increase between configurations.

A- Multi-node benchmarking. The expected training time for 4 GPUs is on the order hours but lesser than Single node.

B+: Single-node benchmarking, expected training time for training on a single node, which could be in order of hours.

6. A try !!!:

If we are lucky enough and get the 2020 MacBook pro from Snell, we could try this project with 8 GPU and contrast the performance between the previous experiments.