Machine Learning on Blue Waters Using TensorFlow with the Image Feature Detection Problem

Part Deux

Presented By: Dr. Aaron D. Saxton
Today's Topics

- Continuation of,
  - https://bluewaters.ncsa.illinois.edu/webinars/data-analytics/ml-tensorflow
- Review Blue Waters
- NN Review
- TensorFlow Basics
- Distributed TensorFlow Overview
- Parallelization Scheme
- Code Tour and Demos
Blue Waters Overview

Sonexion: 26 usable PB

4 XE Nodes

2 XK Nodes

Nvidia K20 (circa 2012-13)
ImageNet

- www.image-net.org
- Large High Quality Dataset
  - 14,197,122 Images
  - 21841 synsets
- Runs the Large Scale Visual Recognition Challenge (ILSVRC)
- Annotated
  - Bounding Boxes
  - synset
    - WordNet (http://wordnet.princeton.edu)
Neural Networks

- Parameterized function
  - \( Z_M = \sigma(\alpha_{0m} + \alpha_m X) \)
  - \( T_K = \beta_{0k} + \beta_k Z \)
  - \( f_K(X) = g_k(T) \)

- \( \beta_{0i}, \beta_i, \alpha_{0m}, \alpha_m \)
  - Weights to be optimized
### Neural Networks

- **Activation functions**
  - Logistic
    \[ \sigma(x) = \frac{1}{1 + e^{-x}} \]
  - Arctan
    \[ \sigma(x) = \frac{\tan^{-1}(x)}{\pi} \]
  - ReLU (Rectified Linear Unit)
    \[ \sigma(x) = \max(0, x) \]
  - Softmax
    \[ g_k(x_1, x_2, \ldots, x_N) = \frac{e^{x_k}}{\sum e^{x_i}} \]
TensorFlow Basics

- Python API
  - C++ under the hood
- Mediator Design Pattern
  - Uses python context manager (with)
- Workflow
  - Construct operations
  - Assign to name scope and or device
  - Enter Session context
    - run()
- www.tensorflow.org/programmers_guide/low_level_intro
Distributed TensorFlow: Parameter Sever/Worker

- Parameter Server contains and coordinates parameters
- Workers ask for and return computed results
- Resilient to Worker/PS failure
Parallelization Schemes: What we are trying to Parallelize

Input Data
- List of Images
- Annotation (1,000 classes)

Algorithm
- Inference
- Loss
- Optimize
Algorithm, Data

Inception V3

ImageNet
Distributed TensorFlow: Parameter Sever/Worker

- **ps:0**
  - Aggregate
  - Update Parameters

- **ps:1**
  - Aggregate
  - Update Parameters

- **worker:0**
  - Model
    - Loss (Cross Entropy)
    - Optimize (Gradient Decent)

- **worker:1**
  - Model
    - Loss (Cross Entropy)
    - Optimize (Gradient Decent)

- **worker:2**
  - Model
    - Loss (Cross Entropy)
    - Optimize (Gradient Decent)
Defining the Distributed Tensor Flow Cluster

- **Blue Waters Job Details**
  - `qsub -l nodes=8:ppn=16:xk inception_imagenet_distributed_train.pbs`
    - Makes 8*16 (128) processing elements available.
    - `aprun -n 8 -N 1 -- python ${RUN_CMD_WITH_ARGS}`
  - 1 processing element per XK node
    - Job will hang if multiple workers are competing for same GPU

- **Each Blue Waters node has a host name**
  - `nidxxxxx`
  - Construct list
    - `ps: nid00001:2222`
    - `worker: nid00002:2222, nid00003:2222, nid00004:2222`
Defining the Distributed Tensor Flow Cluster

- Get unique hostnames of job
  - `socket.gethostname()`
  - `mpi4py allgather(my_hostname)`
- Decide which hosts will be parameter servers and which will be works
  - `tf_ps_hosts_ports = ['nid25428:2222']`
  - `tf_worker_hosts_ports = ['nid25429:2222', 'nid25430:2222']`
- `tf.train.ClusterSpec()`
- `tf.train.Server()`
Defining the Distributed Tensor Flow Cluster

Demo
(Code Tour)
Defining the Distributed Tensor Flow Cluster

- `tf.device()` and `tf.train.replica_device_setter()`
  - `replica_device_setter()`
    - All the magic happens
  - Default round robin PS scheme
Defining the Distributed Tensor Flow Cluster

Demo
(Code Tour)
TensorFlow Basics

Demo
Simple Regression
Blue Waters TensorFlow Process

Demo
(Code Tour)