Lab Exercise #0

Luyu Liu

1. Setup and Login Login to Owens (OSC) cluster with your accounts and learn how to allocate nodes.

After login in: the bash returned:

As of 2019-09-29T14:00:01.000000 project/group **PAS1588** on /fs/project used 0 GiB of quota 5120 GiB and 4 files of quota 100000 files

As of 2019-09-29T14:01:29.000000 userid **liuluyu0378** on /users/PAS1588 used 20.00 KiB of quota 500 GiB and 13 files of quota 1000000 files

To allocate a node with GPU, qsub -I -l nodes=1:ppn=2:gpus=1 -A PAS1588

The shell returned: nodes=1:ppn=2:gpus=1 defaulting to mem=9142MB

And:

qsub: waiting for job 8084726.owens-batch.ten.osc.edu to start

After 30 minutes it returns:

qsub: job 8084726.owens-batch.ten.osc.edu ready

The allocation process take a long time to finish. Compared with my former experience with the OSC shell a year ago, the speed is significantly slower, even when I was doing this in a Sunday.

To allocate a CPU-only node, qsub -I -l nodes=1:ppn=28 -A PAS1588

However, compared with GPU, CPU allocation is significantly faster, which finished within 10 seconds.

The bash returned:

qsub: waiting for job 8084738.owens-batch.ten.osc.edu to start

qsub: job 8084738.owens-batch.ten.osc.edu ready

1. Load and Verify your TensorFlow installation

-bash-4.2$ module load python/3.6-conda5.2

-bash-4.2$ pip install --user tensorflow-gpu

The bash returned typical pip installation process:

twisted 18.7.0 requires PyHamcrest>=1.9.0, which is not installed.

tensorflow 1.9.0 has requirement setuptools<=39.1.0, but you'll have setuptools 41.2.0 which is incompatible.

tensorflow 1.9.0 has requirement tensorboard<1.10.0,>=1.9.0, but you'll have tensorboard 1.14.0 which is incompatible.

And:

Installing collected packages: absl-py, setuptools, protobuf, tensorboard, tensorflow-estimator, wrapt, keras-applications, keras-preprocessing, google-pasta, tensorflow-gpu

Test with Python shell:

[liuluyu0378@o0779 ~]$ python

Python 3.6.6 |Anaconda custom (64-bit)| (default, Jun 28 2018, 17:14:51) [GCC 7.2.0] on linuxType "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>> tf.VERSION

'1.14.0'

Test GPU:

[liuluyu0378@o0779 ~]$ module load cuda/10.0.130

[liuluyu0378@o0779 ~]$ python

Python 3.6.6 |Anaconda custom (64-bit)| (default, Jun 28 2018, 17:14:51) [GCC 7.2.0] on linuxType "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>> tf.test.gpu\_device\_name()

2019-09-29 14:46:39.946370: I

tensorflow/core/platform/cpu\_feature\_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA

2019-09-29 14:46:39.994525: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcuda.so.1

2019-09-29 14:46:40.168754: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x562662522f70 executing computations on platform CUDA. Devices:

2019-09-29 14:46:40.168841: I tensorflow/compiler/xla/service/service.cc:175] StreamExecutor device (0): Tesla P100-PCIE-16GB, Compute Capability 6.0

2019-09-29 14:46:40.173075: I tensorflow/core/platform/profile\_utils/cpu\_utils.cc:94] CPU Frequency: 2399710000 Hz

2019-09-29 14:46:40.173283: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x562662593fe0 executing computations on platform Host. Devices:

2019-09-29 14:46:40.173378: I tensorflow/compiler/xla/service/service.cc:175] StreamExecutor device (0): <undefined>, <undefined>

2019-09-29 14:46:40.174568: I tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1640] Found device 0 with properties: name: Tesla P100-PCIE-16GB major: 6 minor: 0 memoryClockRate(GHz): 1.3285pciBusID: 0000:04:00.0

2019-09-29 14:46:40.210530: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcudart.so.10.0

2019-09-29 14:46:40.320545: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcublas.so.10.0

2019-09-29 14:46:40.383067: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcufft.so.10.0

2019-09-29 14:46:40.426523: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcurand.so.10.0

2019-09-29 14:46:40.548050: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcusolver.so.10.0

2019-09-29 14:46:40.621121: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcusparse.so.10.0

2019-09-29 14:46:40.627145: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcudnn.so.7

2019-09-29 14:46:40.629103: I tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1763] Adding visible gpu devices: 0

2019-09-29 14:46:40.629174: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:42] Successfully opened dynamic library libcudart.so.10.0

2019-09-29 14:46:40.630902: I tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1181] Device interconnect StreamExecutor with strength 1 edge matrix:

2019-09-29 14:46:40.630945: I tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1187] 0

2019-09-29 14:46:40.630975: I tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1200] 0: N

**2019-09-29 14:46:40.633271: I Tensorflow/core/common\_runtime/gpu/gpu\_device.cc:1326] Created**

**TensorFlow device (/device:GPU:0 with 15189 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:04:00.0, compute capability: 6.0)'/device:GPU:0'**

The tensorflow is able to be run on a GPU node.

1. Experiment with the Iris Classification Problem
2. What is an Estimator? What are three main capabilities of an Estimator object?

According to the documentation[[1]](#footnote-1), an estimator is a model-level abstraction, which belongs to a high-level tensorflow API.

The three major capabilities of Estimator is: training, evaluation, prediction.

1. How many features and labels are there for the Iris dataset?

There are four features: SepalLength, SepalWidth, PetalLength, PetalWidth.

There are three lables: 'Setosa', 'Versicolor', 'Virginica'

1. What is an input function?

According to the documentation[[2]](#footnote-2), an input function is a function that returns a tf.data.Dataset object which outputs the following two-element tuple:

* 1. [features](https://developers.google.com/machine-learning/glossary/#feature) - A Python dictionary whose key is the name of a feature and whose value is an array containing all of that feature's values.
  2. label - An array containing the values of the [label](https://developers.google.com/machine-learning/glossary/#label) for every example.

1. What is the difference between features and feature columns?

Features are the field or dimension of the input data; while a [feature column](https://developers.google.com/machine-learning/glossary/#feature_columns) is an object describing how the model should use raw input data from the features dictionary.

1. Write the name of three Premade Estimators that TensorFlow provides

tf.estimator.DNNClassifier for deep models that perform multi-class classification.

tf.estimator.DNNLinearCombinedClassifier for wide & deep models.

tf.estimator.LinearClassifier for classifiers based on linear models.

Results:

Training and test parameters: model: DNNClassifier; two hidden layers, one with 30 nodes and one with 10 nodes; step: 5000;

|  |  |  |
| --- | --- | --- |
|  | GPU | CPU |
| Accuracy | 0.967 | 0.967 |
| Average loss | 0.07202509 | 0.11378458 |
| Loss | 2.1607528 | 3.4135375 |
| Overall time | 13.77 | 11.76 |

The loss’s changing trend:

The final accuracy is the same. Although GPU is faster than CPU during the training, it takes a lot of time to start up, which make GPU more suitable for larger and more complicated work.

1. <https://www.tensorflow.org/guide/estimator> [↑](#footnote-ref-1)
2. <https://www.tensorflow.org/tutorials/estimator/premade#create_input_functions> [↑](#footnote-ref-2)