How to Install Environment for FpointNet Code

Peiyan Gong

6/9/2018

Here is a series of command I used to install necessary environment for running FpointNet code. I choose to use the combination of Ubuntu 16.04 LTS + CUDA 8.0.61 + CUDNN 6.0.21 + tensorflow 1.4. This setup has been proofed to work perfectly. Note that, to run Ubuntu 16.04 LTS on GCP VM instance, you need to select Ubuntu 16.04LTS image when you create the VM instance. Please follow the instruction accordingly.

Check ubuntu version:

```
lsb_release -a
```

Install git:

```
sudo apt-get install git
```

Install pip:

```
sudo apt-get install python-pip python-dev build-essential
```

Install nvcc :

```
sudo apt-get install nvidia-cuda-toolkit
```

(**Check nvidia driver:** nvidia-smi **Check CUDA version:** nvcc --version)
(Check the CUDA version after you set environment variable)

(CUDA version: 8.0.61)(command to check:

```
cat /usr/local/cuda/version.txt)
```

Download CUDA:

```
curl -O
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
```

Unpack CUDA:

```
sudo dpkg -i ./cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
```

Update apt-get:

```
sudo apt-get update
```

Install CUDA8.0:

```
sudo apt-get install cuda-8-0
```

Set up environment variables:

```
echo 'export CUDA_HOME=/usr/local/cuda' >> ~/.bashrc
echo 'export PATH=$PATH:$CUDA_HOME/bin' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=$CUDA_HOME/lib64' >> ~/.bashrc
echo 'export PATH=/usr/local/cuda-8.0/bin:$PATH' >> ~/.bashrc
```

Remove the downloaded file:

```
rm cuda-repo-ubuntu1604_8.0.61-1_amd64.deb
```

(cudnn version: 6.0.21)(command to check:

```
cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2)
```

Set up environment variables:

```
CUDNN_TAR_FILE="cudnn-8.0-linux-x64-v6.0.tgz"
```

Download cudnn 6.0:

```
wget
http://developer.download.nvidia.com/compute/redist/cudnn/v6.0/${CUDNN_TAR_FILE}
```

Unpack cudnn:

```
wget
http://developer.download.nvidia.com/compute/redist/cudnn/v6.0/${CUDNN_TAR_FILE}
```

Copy cudnn to local directory:

```
sudo cp cuda/lib64/* /usr/local/cuda/lib64/
sudo cp cuda/include/cudnn.h /usr/local/cuda/include/
```

Remove the unpacked file:

```
rm -rf ~/cuda
```

Remove the packed file:

```
rm cudnn-8.0-linux-x64-v5.1.tgz
```

(tensorflow version: 1.4)(python code to check version:

```
import tensorflow as tf; print(tf.__version__))
```

Install basic tools:

```
sudo apt-get install python-dev python-pip libcupti-dev
```

Install tensorflow:

```
sudo pip install --upgrade tensorflow-gpu==1.4.0
```

Here is a test script for tensorflow:

```
import tensorflow as tf

with tf.device('/cpu:0'):

    a_c = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a-cpu')

    b_c = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b-cpu')

    c_c = tf.matmul(a_c, b_c, name='c-cpu')

with tf.device('/gpu:0'):

    a_g = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a-gpu')

    b_g = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b-gpu')

    c_g = tf.matmul(a_g, b_g, name='c-gpu')

with tf.Session(config=tf.ConfigProto(log_device_placement=True)) as sess:

    print (sess.run(c_c))

    print (sess.run(c_g))

print 'DONE!'
```

Make directory:

```
mkdir FpointNet
```

Go to the new folder:

```
cd FpointNet
```

Initiate git:

```
git init
```

Clone FpointNet code:

```
git clone https://github.com/PeiyanGong/frustum-pointnets.git
```

Go to mayavi folder:

```
cd /frustum-pointnets/mayavi
```

Get permission for shell script:

```
chmod +x mayavi_install.sh
```

run installation script:

```
./mayavi_install.sh
```