

Deep Learning Environment Setup

1. ubuntu 16.04 64bit

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install build-essential ssh git cmake dkms
```

2. Nvidia Driver

```
sudo add-apt-repository ppa:graphics-drivers/ppa  
sudo apt-get update  
sudo apt-get install nvidia-387  
# sudo apt-get install nvidia-opengl-icd-384 nvidia-opengl-dev opengl-headers  
# sudo mv /usr/lib/nvidia-384/libEGL.so.1 /usr/lib/nvidia-384/libEGL.so.1.org  
# sudo mv /usr/lib32/nvidia-384/libEGL.so.1 /usr/lib32/nvidia-384/libEGL.so.1.org  
# sudo ln -s /usr/lib/nvidia-384/libEGL.so.384.98 /usr/lib/nvidia-384/libEGL.so.1  
# sudo ln -s /usr/lib32/nvidia-384/libEGL.so.384.98 /usr/lib32/nvidia-384/libEGL.so.1  
# sudo ldconfig  
sudo reboot
```

3. install python lib

```
sudo apt-get install python2.7-dev python3.5-dev  
wget https://bootstrap.pypa.io/get-pip.py  
sudo python3 get-pip.py  
sudo python get-pip.py  
sudo pip3 install numpy  
sudo pip install numpy  
vi ~/.bashrc  
add:  
    export C_INCLUDE_PATH=/usr/local/lib/python3.5/dist-packages/numpy/core/include  
source ~/.bashrc  
  
sudo apt-get install swig libopenblas-dev liblapack-dev liblapacke-dev libhdf5-serial-dev python3-tk graphviz  
sudo pip3 install scipy
```

```
sudo pip3 install matplotlib
sudo pip3 install pytest
sudo pip3 install cython
sudo pip3 install pandas
sudo pip3 install scikit-learn
sudo pip3 install scikit-image
sudo pip3 install ipython
sudo pip3 install ipdb
sudo pip3 install notebook
sudo pip3 install h5py
sudo pip3 install pyyaml
sudo pip3 install graphviz
sudo pip3 install tqdm
sudo pip3 install joblib
sudo pip3 install beautifulsoup4
sudo pip3 install Polygon3
sudo pip3 install mlens
sudo pip3 install future
sudo pip3 install pydotplus
```

4. OpenSlide:

```
tar zxvf zlib-1.2.11.tar.gz
cd zlib-1.2.11
./configure
make
sudo make install
cd ..

tar zxvf jpegsrc.v9b.tar.gz
cd jpeg-9b
./configure
make
sudo make install
cd ..
```

```
tar zxvf openjpeg-2.3.0.tar.gz
cd openjpeg-2.3.0
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make
sudo make install
sudo ldconfig
cd ..
cd ..

sudo apt-get install libperl-dev libgtk2.0-dev libsqlite3-dev
sudo apt-get install autoconf libtool automake libtool pkg-config libxml2-dev libtiff5-dev
tar zxvf openslide-3.4.1.tar.gz
cd openslide-3.4.1
autoreconf -i
./configure
make
sudo make install
sudo ldconfig
sudo pip3 install openslide-python
cd ..
```

5. VIPS:

```
sudo apt-get install build-essential pkg-config glib2.0-dev libexpat1-dev gobject-introspection
sudo apt-get install python-gi-dev libgirepository1.0-dev libtiff5-dev libjpeg-turbo8-dev libgsf-1-dev
tar zxvf vips-8.5.8.tar.gz
cd vips-8.5.8
./configure
make -j4
sudo make install
sudo ldconfig
sudo cp /usr/local/lib/python2.7/site-packages/gi/overrides/Vips.* /usr/lib/python3/dist-packages/gi/overrides
vi ~/.bashrc
add:
```

```
export GI_TYPELIB_PATH=/usr/local/lib/girepository-1.0  
source ~/.bashrc  
  
python3  
from gi.repository import Vips  
  
sudo pip3 install pyvips
```

6. install opencv

```
sudo apt-get install libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev  
sudo apt-get install libxvidcore-dev libx264-dev  
sudo apt-get install libatlas-base-dev gfortran  
tar zxvf opencv-3.4.0.tar.gz tar  
tar zxvf opencv_contrib-3.4.0.tar.gz  
cd opencv-3.2.0  
mkdir build  
cd build  
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D  
INSTALL_PYTHON_EXAMPLES=ON -D INSTALL_C_EXAMPLES=OFF -D  
OPENCV_EXTRA_MODULES_PATH=~/Downloads/opencv_contrib-3.2.0/modules -D  
BUILD_EXAMPLES=ON -DBUILD_opencv_dnn=OFF ..  
make -j4  
sudo make install  
sudo ldconfig  
cd /usr/local/lib/python2.7/site-packages  
sudo ln -s /usr/local/lib/python2.7/dist-packages/cv2.so cv2.so  
cd /usr/local/lib/python3.5/dist-packages  
sudo ln -s cv2.cpython-35m-x86_64-linux-gnu.so cv2.so  
  
python3  
import cv2
```

7. install cuda, cudnn and nccl

```
sudo ./cuda_9.0.176_384.81_linux.run  
During install:
```

Accept EULA

Say No to installing the NVIDIA driver

Say Yes to installing CUDA Toolkit

Say Yes to installing CUDA Samples

```
vi ~/.bashrc
```

add:

```
export PATH=/usr/local/cuda/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

```
source ~/.bashrc
```

```
nvcc -V
```

```
sudo tar -xzf cudnn-9.0-linux-x64-v7.tgz -C /usr/local
```

```
sudo ldconfig
```

```
cd ~/NVIDIA_CUDA-9.0_Samples
```

```
make -j4
```

```
cd ~/NVIDIA_CUDA-9.0_Samples/bin/x86_64/linux/release
```

```
./deviceQuery
```

```
./bandwidthTest
```

```
sudo dpkg -i nccl-repo-ubuntu1604-2.1.4-ga-cuda9.0_1-1_amd64.deb
```

```
sudo apt-get update
```

```
sudo apt-get install libnccl2 libnccl-dev
```

8. install tensorflow, keras and xgboost

```
sudo pip3 install tensorflow-gpu
```

```
sudo pip3 install keras
```

```
sudo pip3 install xgboost
```

9. install pytorch

```
sudo pip3 install torch-0.3.0.post4-cp35-cp35m-linux_x86_64.whl
```

```
sudo pip3 install torchvision
```

```
sudo pip3 install git+https://github.com/pytorch/tnt.git@master
```

```
sudo pip3 install cupy
```

```
sudo pip3 install visdom
```

```
# nohup python3 -m visdom.server &
```

10. install ONNX, ONNX-CAFFE2 and caffe2

```
sudo apt-get install libgoogle-glog-dev libprotobuf-dev protobuf-compiler
sudo pip3 install protobuf

sudo apt-get install libgtest-dev libiomp-dev libleveldb-dev liblmdb-dev libopenmpi-
dev libsnpappy-dev

sudo apt-get install libeigen3-dev

sudo pip3 install flask
sudo pip3 install future
sudo pip3 install hypothesis
sudo pip3 install python-nvd3

sudo pip3 install onnx
sudo pip3 install onnx-caffe2

git clone --recursive https://github.com/caffe2/caffe2.git
cd caffe2
mkdir build
cd build

cmake .. -DPYTHON_EXECUTABLE=/usr/bin/python3 -DPYTHON_LIBRARY=/usr/lib/x86_64-linux-
gnu/libpython3.5m.so -DPYTHON_INCLUDE_DIR=/usr/include/python3.5m
make -j4

sudo make install

python3 -m caffe2.python.operator_test.relu_op_test
```

install PyCuda, scikit-cuda and scrapy

```
tar zxvf pycuda-2016.1.2.tar.gz
cd pycuda-VERSION
python3 configure.py --cuda-root=/usr/local/cuda
sudo make install
cd test
python3 test_driver.py
sudo pip3 install scikit-cuda
sudo pip3 install pynvrtc
```

```
sudo apt-get install libssl-dev libffi-dev
sudo apt-get install -y libxml2-dev libxslt1-dev
sudo pip3 install scrapy
scrapy version -v
```

install lasagne, nolearn and pymc3, pgmpy

```
sudo pip3 install https://github.com/Lasagne/Lasagne/archive/master.zip
sudo pip3 install git+https://github.com/dnouri/nolearn.git@master#egg=nolearn==0.7.git
sudo pip3 install pymc3
sudo pip3 install https://github.com/pgmpy/pgmpy/archive/dev.zip
```

install mxnet

```
git clone https://github.com/dmlc/mxnet.git --recursive
cd mxnet
cp make/config.mk .
echo "USE_CUDA=1" >>config.mk
echo "USE_CUDA_PATH=/usr/local/cuda" >>config.mk
echo "USE_CUDNN=1" >>config.mk
sudo apt-get install graphviz
make -j4
cd python
sudo python3 setup.py install
vi ~/.bashrc
-----
export PYTHONPATH=~/.workspace/mxnet/python:$PYTHONPATH
-----
source ~/.bashrc
sudo pip3 install graphviz
ipython
import mxnet
exit
```

install xgboost

```
git clone --recursive https://github.com/dmlc/xgboost
cd xgboost
mkdir build
cd build
cmake .. -DUSE_CUDA=ON
make -j4
cd ../python-package
sudo python3 setup.py install
vi ~/.bashrc

-----

export PYTHONPATH=~/.workspace/xgboost/python-package:$PYTHONPATH

-----

source ~/.bashrc
python3
import xgboost
```

install LightGBM

```
git clone --recursive https://github.com/Microsoft/LightGBM
cd LightGBM
mkdir build
cd build
cmake ..
make -j4
sudo make install
cd ../python-package
sudo python3 setup.py install
ipython
import lightgbm
exit
```

compile tensorflow

```
sudo apt-get install openjdk-8-jdk swig
./bazel-0.3.0-installer-linux-x86_64.sh --user
vi ~/.bashrc
```


add:

```
export PATH=$PATH:$HOME/bin
```

reopen terminal

```
git clone https://github.com/tensorflow/tensorflow
```

```
cd tensorflow
```

```
./configure
```

Please specify the location of python. [Default is /usr/bin/python]: /usr/bin/python(3)

Do you wish to build TensorFlow with GPU support? [y/N] y

GPU support will be enabled for TensorFlow

Please specify which gcc nvcc should use as the host compiler. [Default is

/usr/bin/gcc]: /usr/bin/gcc

Please specify the Cuda SDK version you want to use, e.g. 7.0. [Leave

empty to use system default]: 8.0

Please specify the location where CUDA 7.5 toolkit is installed. Refer to

README.md for more details. [default is: /usr/local/cuda]: /usr/local/cuda

Please specify the Cudnn version you want to use. [Leave empty to use system

default]: 5.1

Please specify the location where the cuDNN 4.0.4 library is installed. Refer to

README.md for more details. [default is: /usr/local/cuda]: /usr/local/cuda

Please specify a list of comma-separated Cuda compute capabilities you want to
build with. You can find the compute capability of your device at:

<https://developer.nvidia.com/cuda-gpus>.

Please note that each additional compute capability significantly increases your
build time and binary size. [Default is: \"3.5,5.2\"]: 5.2

bazel build -c opt --config=cuda //tensorflow/cc/tutorials_example_trainer

```
bazel-bin/tensorflow/cc/tutorials_example_trainer --use_gpu
```

```
bazel build -c opt --config=cuda //tensorflow/tools/pip_package:build_pip_package
```

```
bazel-bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
```

```
sudo pip3 install /tmp/tensorflow_pkg/tensorflow-0.9.0-py3-none-any.whl
cd ~
python3 -m tensorflow.models.image.mnist.convolutional
```

compile serving

```
sudo pip install grpcio
git clone --recurse-submodules https://github.com/tensorflow/serving
cd serving
cd tensorflow
./configure
```

Please specify the location of python. [Default is /usr/bin/python]: /usr/bin/python(3)

Do you wish to build TensorFlow with GPU support? [y/N] y

GPU support will be enabled for TensorFlow

Please specify which gcc nvcc should use as the host compiler. [Default is /usr/bin/gcc]: /usr/bin/gcc

Please specify the Cuda SDK version you want to use, e.g. 7.0. [Leave empty to use system default]: 8.0

Please specify the location where CUDA 7.5 toolkit is installed. Refer to README.md for more details. [default is: /usr/local/cuda]: /usr/local/cuda

Please specify the Cudnn version you want to use. [Leave empty to use system default]: 5.1

Please specify the location where the cuDNN 4.0.4 library is installed. Refer to README.md for more details. [default is: /usr/local/cuda]: /usr/local/cuda

Please specify a list of comma-separated Cuda compute capabilities you want to build with. You can find the compute capability of your device at: <https://developer.nvidia.com/cuda-gpus>.

Please note that each additional compute capability significantly increases your build time and binary size. [Default is: \"3.5,5.2\"]: 5.2

```
-----  
cd..  
bazel build tensorflow_serving/...  
bazel test tensorflow_serving/...
```

Add New Hard Driver:

```
sudo fdisk -l  
sudo fdisk /dev/sda  
-----  
input: m, 可以看到帮助信息  
input: n, 添加新分区  
input: p, 选择添加主分区  
选择该分区的开始值和结束值（直接回车）  
input: w, 保存所有并退出  
-----  
sudo mkfs -t ext4 /dev/sda1  
  
sudo parted /dev/sda  
print  
mklabel gpt  
mkpart primary 2048s 100%  
print  
quit  
sudo mkfs -t ext4 /dev/sda1
```

Virtualbox:

```
sudo apt-get install build-essential dkms unzip wget  
sudo vi /etc/apt/sources.list  
add:  
deb http://download.virtualbox.org/virtualbox/debian xenial contrib  
wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -  
sudo apt-get update  
sudo apt-get install virtualbox-5.1  
sudo usermod -aG vboxusers terrypang
```

```
sudo systemctl status vboxdrv
```

```
wget http://download.virtualbox.org/virtualbox/5.1.28/Oracle_VM_VirtualBox_Extension_Pack-5.1.28-117968.vbox-extpack
```

```
sudo VBoxManage extpack install Oracle_VM_VirtualBox_Extension_Pack-5.1.28-117968.vbox-extpack
```

```
sudo vi /etc/fstab
```

```
# 512G
```

```
UUID=a7f6309d-065b-4618-b89f-8689518625ca /media/terrypang/a7f6309d-065b-4618-b89f-8689518625ca ext4 defaults O 1
```

```
# 4T
```

```
UUID=857803a4-007b-433a-a5f3-b17ce6b86fd9 /media/terrypang/857803a4-007b-433a-a5f3-b17ce6b86fd9 ext4 defaults O 1
```

```
# 4T-Backup
```

```
UUID=873ba382-6f29-43fe-85cd-3087afd1fca6 /media/terrypang/873ba382-6f29-43fe-85cd-3087afd1fca6 ext4 defaults O 1
```

```
sudo add-apt-repository ppa:hzwhuang/ss-qt5
```

```
sudo apt-get update
```

```
sudo apt-get install shadowsocks-qt5
```

```
sudo apt-get -o Acquire::http::proxy="http://127.0.0.1:1080/" update
```

```
sudo pip3 install numpy --proxy="http://127.0.0.1:1080"
```

```
wget -e use_proxy=yes -e https_proxy=http://127.0.0.1:1080 https://www.google.com
```

```
git -c "http.proxy=192.168.100.12:1080" clone --recursive https://github.com/caffe2/caffe2.git
```

```
env path set:
```

```
#在PATH中找到可执行文件程序的路径。
```

```
export PATH=$PATH:$HOME/bin
```

```
#gcc找到头文件的路径
```

```
C_INCLUDE_PATH=/usr/include/libxml2/MyLib
```

```
export C_INCLUDE_PATH
```

```
#g++找到头文件的路径
```

```
CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/include/libxml2:/MyLib
```

```
export CPLUS_INCLUDE_PATH
```

#找到动态链接库的路径

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/MyLib
```

```
export LD_LIBRARY_PATH
```

#找到静态库的路径

```
LIBRARY_PATH=$LIBRARY_PATH:/MyLib
```

```
export LIBRARY_PATH
```