CHAPTER 2: TENSORFLOW INSTALLATION

Theory

In this chapter, we show how to install and configure the environment, and use the Tensorflow on various operating systems such as Windows and Ubuntu which are widely used as the machine learning framework, and how to create a simple code using Tensorflow.

Basic concepts

- Operation
  The nodes on the graph are called operations (abbreviation op). An operation can receive one or more tensors. Operations can perform calculations and return results in one or more tensors.
- Tensor
  Internally, all data is represented by tensors. Tensors are a kind of multidimensional array, in which only tensors are passed between operations in the graph. (Similar to Caffe's Blob.)
- Session
  A session object is required to run the graph. A session is an encapsulation of the execution environment of an operation.
- Variable
  Variables are used to store and update parameters when the graph is run. It acts as a buffer to store the tensor in memory.

Tensorflow

Tensorflow made by Google is quite famous among the packages for deep learning.

But Tensorflow is not the only alternative. The packages for deep learning are as follows.

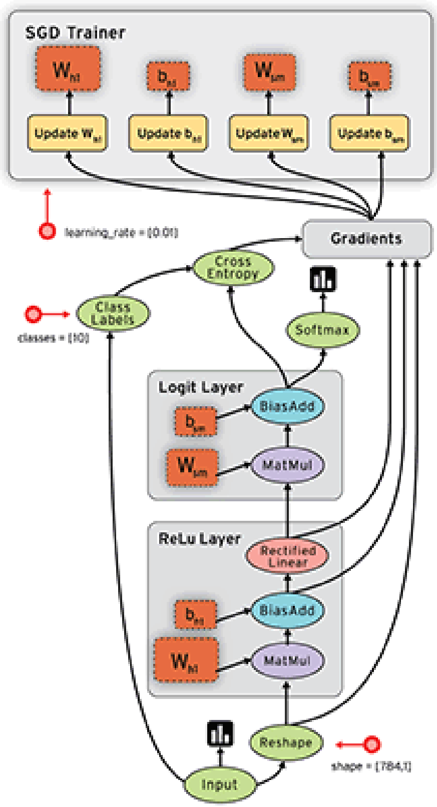
Famous among them are Facebook's Torch-based Torchnet, Google's Tensorflow, their ancestors Theano, and Berkeley's Caffe.

Among them, Google's Tensorflow is made easy to write in Python environment.

Software	Creator	Software license	Open Source	Platform
Caffe	Berkeley Vision and Learning Center, Community contributors	BSD 2-Clause License	Yes	Ubuntu, OS X, AWS, unofficial Android port, Windows supported by Microsoft Research, unofficial Windows port
CNTK	Microsoft	Free	Yes	Windows, Linux
Deeplearning4j	Various; Originally Adam Gibson	Apache 2.0	Yes	Linux, OSX, Windows, Android, CyanogenMod
MXNet	Distributed (Deep) Machine Learning Community	Apache 2.0	Yes	Ubuntu, OS X, Windows AWS, Android, iOS, Javascript
Neural Designer	Artelnics	Proprietary	No	Windows, OS X, Linux
OpenNN	Artelnics	GNU LGPL	Yes	Cross platform
SINGA	Apache Incubator	Apache 2.0	Yes	Linux
SystemML	IBM Research, Databricks, Netflix	Apache 2.0	Yes	Linux, Max OS, Windows
Tensorflow	Google Brain team	Apache 2.0	Yes	Linux, Mac OS X, Windows
Theano	University Montreal	BSD license	Yes	Cross-platform
Torch	Ronan Collobert, Koray, Kavukcuoglu, Clement Farabet	BSD license	Yes	Linux, Android, Mac OS X, iOS

Tensorflow is one of the mostly used framework and open source library for machine learning and deep learning developed by Google. Data flow graph (DFG) mode is used in Tensorflow.

Data flow graphs represent mathematical calculations and flow of data as directed graphs using nodes and edges.



Nodes perform operations such as mathematical computation, data input / output, and data read / save. And the edge represents the input / output relationship of data between nodes.

The edge carries a dynamically sized multidimensional data array (= tensor) and Tensorflow was named from there.

Tensor is a concept that has been used in various fields such as science and engineering. In mathematics, it is known as a notation for expressing arbitrary geometry in a coordinate-independent manner, but it is used in slightly different meanings for each field. In this case, you will understand the multidimensional array in which the training data is stored.

We can write operation codes using Python, a high-level programming language. Most other languages are supported, but most are Python related. Despite the fact that it was not so long ago, Tensorflow has been used in various fields.

Both the regular and GPU-accelerated versions are available. The generic version can run on any computer, and the GPU-accelerated version works much faster because it uses GPGPU to perform large-scale operations quickly.

However, since it uses NVIDIA's GPGPU language, CUDA, it can’t be used without an NVIDIA graphics card. There is also a version that Google uses internally for its services, which runs on Google's own AI acceleration hardware, the Tensor Processing Unit (TPU). It is 15 to 30 times faster than Intel Xeon or NVIDIA Tesla.

The tensor flow has the following characteristics.

- Rich expressiveness through data flow graph
- Works in CPU / GPU mode without code modification
- Available from idea test to service level
- Processes the differential calculation automatically when calculation structure and objective functions are defined
- Support Python / C++, and various languages via SWIG

Let's take a quick look at the basic concepts.

```
import tensorflow as tf

# Initialize variable to 0
state = tf.Variable(0, name="counter")

# Create an operation to add 1 to state
one = tf.constant(1)
new_value = tf.add(state, one)
update = tf.assign(state, new_value)

# The graph must first initialize the variable. Create an
init operation through the function below
init_op = tf.initialize_all_variables()
```

```
# Launch graphs and execute operations
with tf.Session() as sess:
  # Execute initialization operation
  sess.run(init_op)
  # Output the initial value of state
  print(sess.run(state))
  # Perform an operation to update state and output state
  for _ in range(3):
    sess.run(update)
    print(sess.run(state))
```

The results are displayed such as below.

```
0

1

2

3
```

AIM

The aim of the following lab exercise is to install Tensorflow on various operating systems and run, test it

Following steps are required.

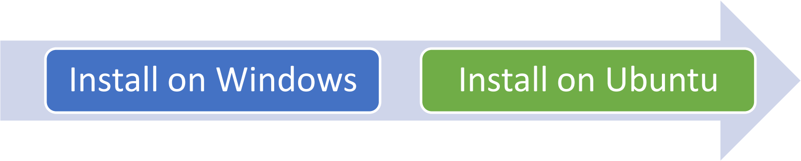
Task 1: Installing Tensorflow on Windows

Task 2: Installing Tensorflow on Ubuntu

We need following packages to perform the lab exercise:

- Python
- Tensorflow
- Anaconda
- Cuda
- cuDNN

LAB EXERCISE 2: INSTALLING TENSORFLOW



1. **Installing Tensorflow on Windows**
2. **Installing Tensorflow on Ubuntu**

Task 1: Installing Tensorflow on Windows

The process of installing the Tensorflow consists of the following four steps.

- Install Anaconda
- Install Tensorflow
- Install Jupyter Notebook (optional)
- Test Tensorflow

Install Anaconda:

To install Tensorflow, Python must be installed by default.

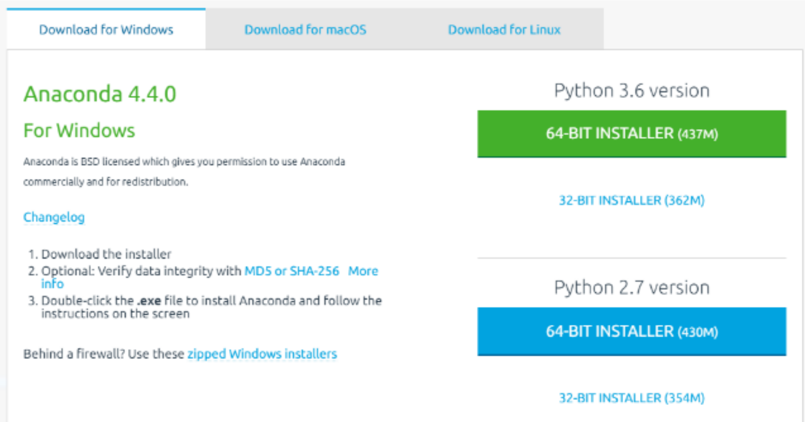
However, rather than just installing Tensorflow in Python, installing Anaconda, a distribution package that includes both Python's machine learning library, sklearn (scikit-learn), and Jupyter Notebook is an easy way to work for machine learning.

And more importantly, Anaconda also includes Python.

This means that if you install Anaconda then Python will be installed automatically. (It is unconditionally installed.)

The current latest version of Anaconda is 4.4.0, and Tensorflow is 1.3.0.

Anaconda for Windows could be downloaded from Anaconda site and could install it at once.



Anaconda 4.4.0

For Windows

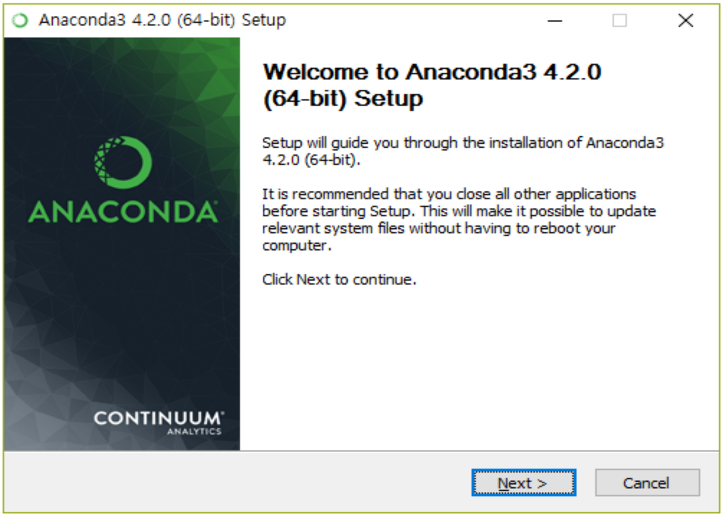
Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

1. Download the installer
2. Optional: Verify data integrity with MD5 or SHA-256 More info
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these zipped Windows installers

Download Anaconda 4.2 and install it.

The installation screens below are screens installed on Windows 10 64bit.



Welcome to Anaconda3 4.2.0 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 4.2.0 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Anaconda 4.2 is installed by default in c: \ Program files \ Anaconda3.

Anaconda 4.3 is installed by default at c: \ PrgramData \ Anaconda3.

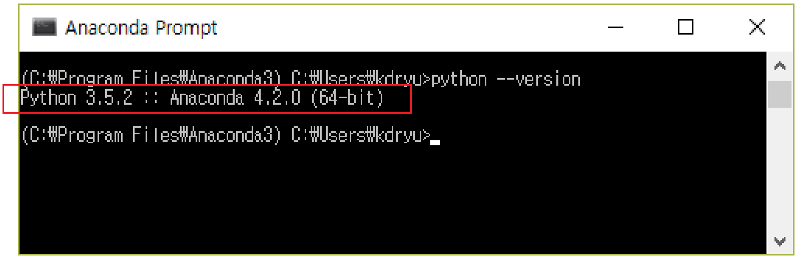
Installing anaconda takes a while. It seems like it took about 10 minutes.

After the installation is complete, you can confirm that Anaconda is installed in the Windows Start menu. I have installed only one Anaconda, but I can see that IPython, Jupyter Notebook, Spyder and so on are installed.

And you can confirm Python 3.5.2 is installed in Anaconda, too.

To run Python, first run Anaconda Prompt. When you run it, a command window displays.

If you check the version of Python installed, you can see that Python 3.5.2 is installed.



```
(C:₩Program Files₩Anaconda3) C:₩Users₩kdryu>python --version
Python 3.5.2 :: Anaconda 4.2.0 (64-bit)

(C:₩Program Files₩Anaconda3) C:₩Users₩kdryu>_
```

Install Tensorflow:

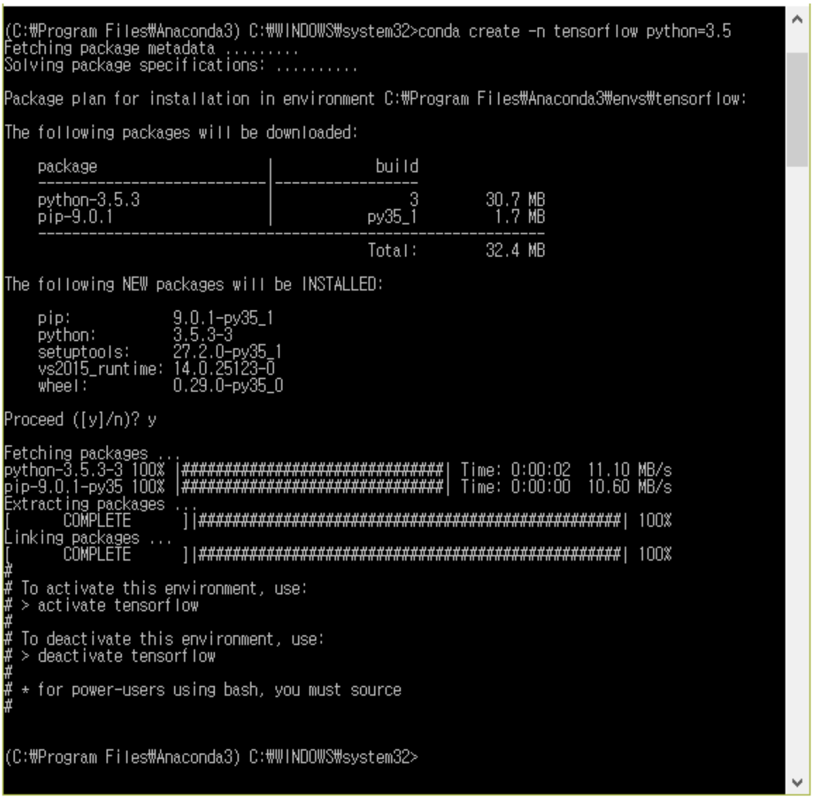
The packages or libraries are installed through the pip package in the Python, similar to Linux.

Once the Anaconda has been installed, you need to install the Tensorflow package. Anaconda provides a virtual execution environment for Python library management and so on.

So before installing the Tensorflow, first upgrade the pip package to the latest, create Anaconda's virtual development environment, and install Tensorflow there.

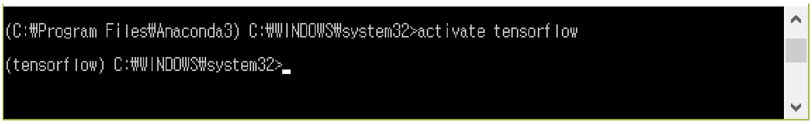
To install Tensorflow, first run Anaconda Prompt in administrator mode and enter the following command:

```
python -m pip install --upgrade pip
conda create -n tensorflow python=3.5
```



```
(C:₩Program Files₩Anaconda3) C:₩WINDOWS₩system32>conda create -n tensorflow python=3.5
Fetching package metadata .........
Solving package specifications: ..........

Package plan for installation in environment C:₩Program Files₩Anaconda3₩envs₩tensorflow:

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    python-3.5.3               |                3        30.7 MB
    pip-9.0.1                  |           py35_1         1.7 MB
    ------------------------------------------------------------
                                           Total:        32.4 MB

The following NEW packages will be INSTALLED:

    pip:            9.0.1-py35_1
    python:         3.5.3-3
    setuptools:     27.2.0-py35_1
    vs2015_runtime: 14.0.25123-0
    wheel:          0.29.0-py35_0

Proceed ([y]/n)? y

Fetching packages ...
python-3.5.3-3 100% |###############################| Time: 0:00:02  11.10 MB/s
pip-9.0.1-py35 100% |###############################| Time: 0:00:00  10.60 MB/s
Extracting packages ...
[      COMPLETE      ]|##################################################| 100%
Linking packages ...
[      COMPLETE      ]|##################################################| 100%
#
# To activate this environment, use:
# > activate tensorflow
#
# To deactivate this environment, use:
# > deactivate tensorflow
#
# * for power-users using bash, you must source
#


(C:₩Program Files₩Anaconda3) C:₩WINDOWS₩system32>
```

And then enter into environment.



```
(C:₩Program Files₩Anaconda3) C:₩WINDOWS₩system32>activate tensorflow
(tensorflow) C:₩WINDOWS₩system32>_
```

When run, the name of the virtual environment (tensorflow) is prepended to the prompt.

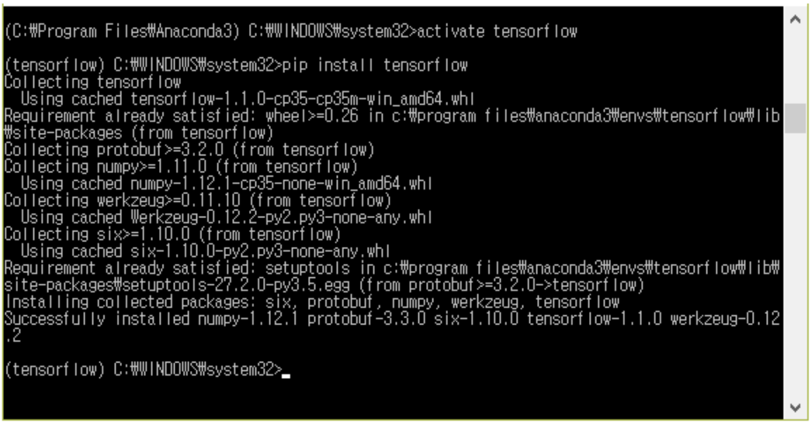
From now on, all packages installed in the (tensorflow) environment are managed separately under C:\Program Files\Anaconda3\envs\tensorflow.

Then install Tensorflow.

```
pip install tensorflow
```

Because Tensorflow is a Python package, you can install it with the pip install command.

If you use the source code of Tensorflow directly, you can also download it through github and compile it.



```
(C:₩Program Files₩Anaconda3) C:₩WINDOWS₩system32>activate tensorflow

(tensorflow) C:₩WINDOWS₩system32>pip install tensorflow
Collecting tensorflow
  Using cached tensorflow-1.1.0-cp35-cp35m-win_amd64.whl
Requirement already satisfied: wheel>=0.26 in c:₩program files₩anaconda3₩envs₩tensorflow₩lib
₩site-packages (from tensorflow)
Collecting protobuf>=3.2.0 (from tensorflow)
Collecting numpy>=1.11.0 (from tensorflow)
  Using cached numpy-1.12.1-cp35-none-win_amd64.whl
Collecting werkzeug>=0.11.10 (from tensorflow)
  Using cached Werkzeug-0.12.2-py2.py3-none-any.whl
Collecting six>=1.10.0 (from tensorflow)
  Using cached six-1.10.0-py2.py3-none-any.whl
Requirement already satisfied: setuptools in c:₩program files₩anaconda3₩envs₩tensorflow₩lib₩
site-packages₩setuptools-27.2.0-py3.5.egg (from protobuf>=3.2.0->tensorflow)
Installing collected packages: six, protobuf, numpy, werkzeug, tensorflow
Successfully installed numpy-1.12.1 protobuf-3.3.0 six-1.10.0 tensorflow-1.1.0 werkzeug-0.12
.2

(tensorflow) C:₩WINDOWS₩system32>_
```

After that you can check if tensorflow is installed properly by using python command as below.

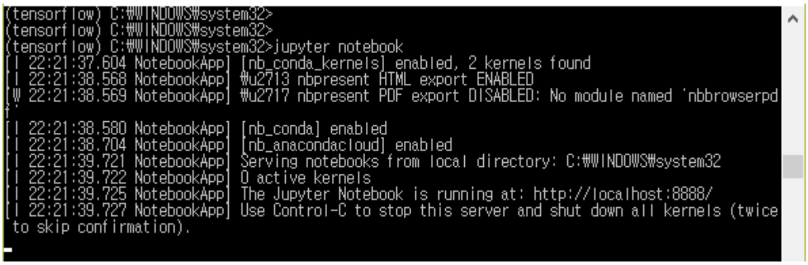
```
import tensorflow as tf
```

If there isn't any errors, we can know Tensorflow has installed successfully.

Install Jupyter Notebook:

Jupyter Notebook is a development environment for developing Python. This is used in Python development because it allows you to write and execute source code directly from a web browser without using a separate editor, so it provides a development environment that is easier to work with and share with github.

When you install Anaconda, Jupyter Notebook is installed by default. So when you type Jupyter Notebook at the anaconda prompt, the Jupyter notebook runs in your browser.

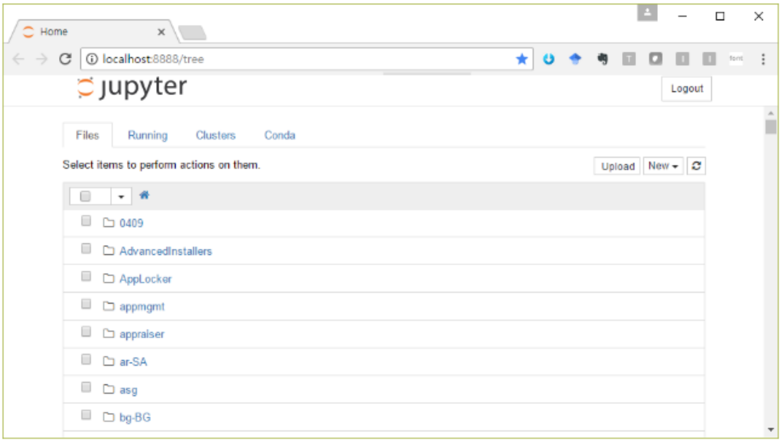


```
(tensorflow) C:₩WINDOWS₩system32>
(tensorflow) C:₩WINDOWS₩system32>
(tensorflow) C:₩WINDOWS₩system32>jupyter notebook
[I 22:21:37.604 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 22:21:38.568 NotebookApp] ₩u2713 nbpresent HTML export ENABLED
[W 22:21:38.569 NotebookApp] ₩u2717 nbpresent PDF export DISABLED: No module named 'nbbrowserpd
f'
[I 22:21:38.580 NotebookApp] [nb_conda] enabled
[I 22:21:38.704 NotebookApp] [nb_anacondacloud] enabled
[I 22:21:39.721 NotebookApp] Serving notebooks from local directory: C:₩WINDOWS₩system32
[I 22:21:39.722 NotebookApp] 0 active kernels
[I 22:21:39.725 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[I 22:21:39.727 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice
 to skip confirmation).
```

If you type the above command in the command window, Jupyter notebook will run in your browser as below.

Jupyter Notebook will run in the folder where Anaconda Prompt is executed because it runs in the directory where the command window is executed.

Pressing the New button at the top right opens an editor where you can enter Python code.



Select items to perform actions on them.

□ 0409
□ AdvancedInstallers
□ AppLocker
□ appmgmt
□ appraiser
□ ar-SA
□ asg
□ bg-BG

And you can use PyCharm instead of Jupyter Notebook.

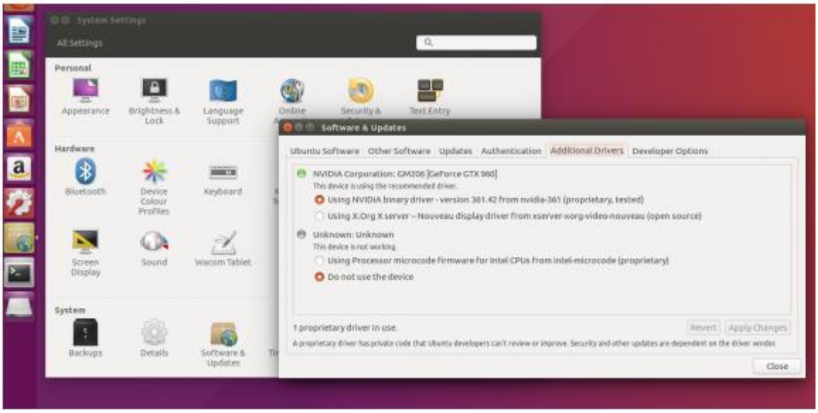
Task1 is complete.

Task 2: Install Tensorflow on Ubuntu

If you have GPU card, you can install Cuda and cuDNN for improve the performance.

Update Graphic card driver:

Go to System Settings> Software & Updates> Addition Drivers tab in Ubuntu 16 and select the NVIDIA binary driver to update it automatically. Once the update is complete, please reboot once.



NVIDIA Corporation: GM206 [GeForce GTX 960]
This device is using the recommended driver.
● Using NVIDIA binary driver - version 361.42 from nvidia-361 (proprietary, tested)
○ Using X.Org X server – Nouveau display driver from xserver-xorg-video-nouveau (open source)

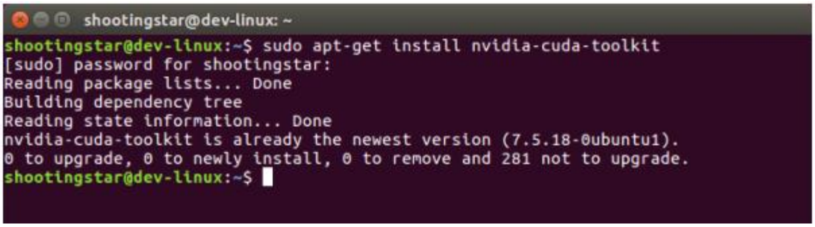
Unknown: Unknown
This device is not working.
○ Using Processor microcode firmware for Intel CPUs from intel-microcode (proprietary)
● Do not use the device

1 proprietary driver in use.
A proprietary driver has private code that Ubuntu developers can't review or improve. Security and other updates are dependent on the driver vendor.

Install Cuda Toolkit:

We can install cuda-toolkit using apt-get command,

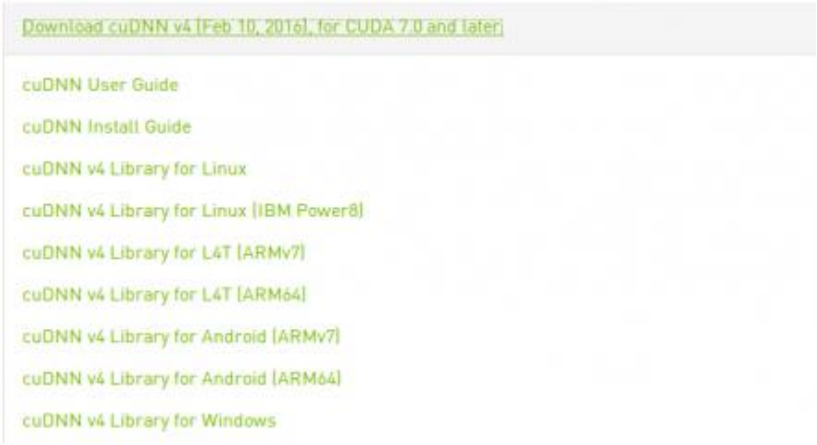
```
$ sudo apt-get install nvidia-cuda-toolkit
```



```
shootingstar@dev-linux:~$ sudo apt-get install nvidia-cuda-toolkit
[sudo] password for shootingstar:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nvidia-cuda-toolkit is already the newest version (7.5.18-0ubuntu1).
0 to upgrade, 0 to newly install, 0 to remove and 281 not to upgrade.
shootingstar@dev-linux:~$
```

Install cuDNN:

Download the cuDNN v4 or latest Library for Linux version from https://developer.nvidia.com/rdp/cudnn-download.



Download cuDNN v4 [Feb 10, 2016], for CUDA 7.0 and later

cuDNN User Guide

cuDNN Install Guide

cuDNN v4 Library for Linux

cuDNN v4 Library for Linux [IBM Power8]

cuDNN v4 Library for L4T [ARMv7]

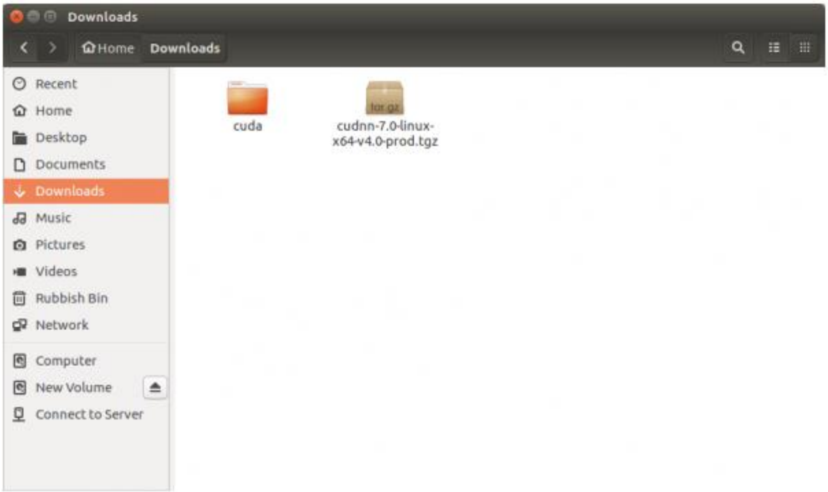
cuDNN v4 Library for L4T [ARM64]

cuDNN v4 Library for Android [ARMv7]

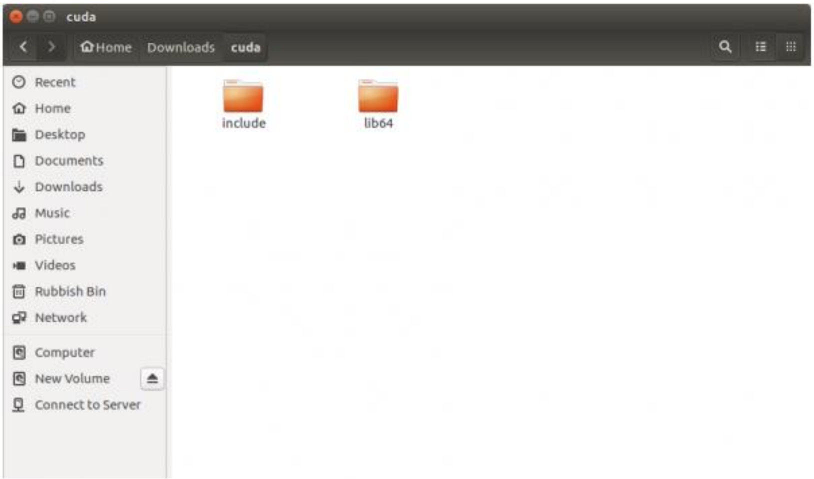
cuDNN v4 Library for Android [ARM64]

cuDNN v4 Library for Windows

Unzip the downloaded file and you will see the cuda folder. Go to the cuda folder.



You can see the include folder and the lib64 folder in the cuda folder.



Copy the cuda folder under /usr/local such as /usr/local/cuda.

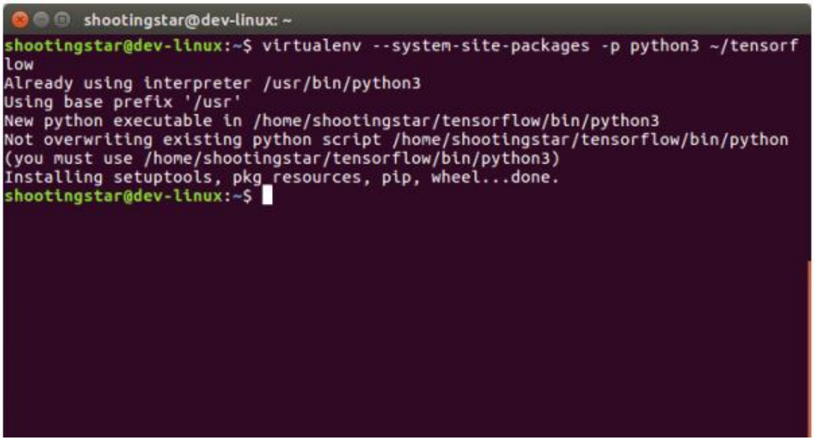
When copying is complete, run the bottom line to set permissions.

```
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/
local/cuda/lib64/libcudnn*
```

Install Virtualenv and Tensorflow:

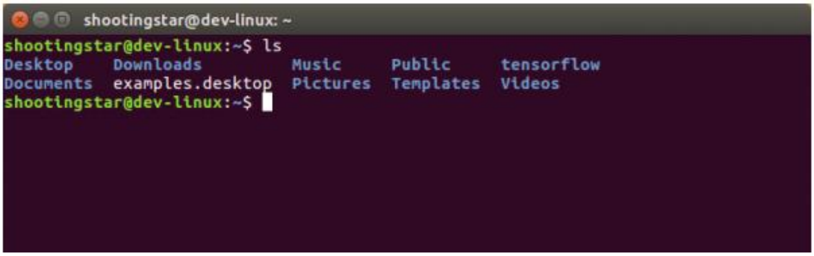
If you prefer Python 2, you can change python3 to python in the command below.

```
$ sudo apt-get install python3-pip python3-dev python-
virtualenv
```



```
shootingstar@dev-linux:~$ virtualenv --system-site-packages -p python3 ~/tensorf
low
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/shootingstar/tensorflow/bin/python3
Not overwriting existing python script /home/shootingstar/tensorflow/bin/python
(you must use /home/shootingstar/tensorflow/bin/python3)
Installing setuptools, pkg_resources, pip, wheel...done.
shootingstar@dev-linux:~$
```

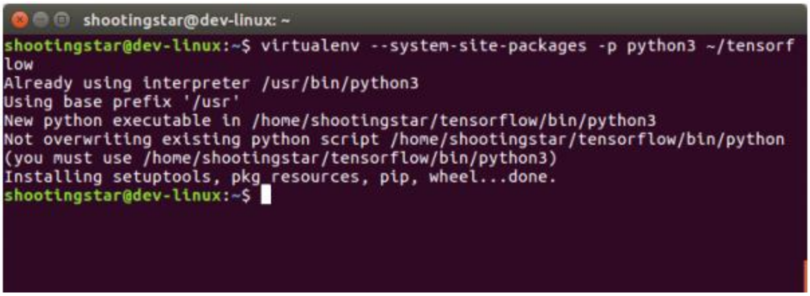
If the above command is executed successfully, tensorflow folder will be created.



```
shootingstar@dev-linux:~$ ls
Desktop    Downloads         Music     Public     tensorflow
Documents  examples.desktop  Pictures  Templates  Videos
shootingstar@dev-linux:~$ 
```

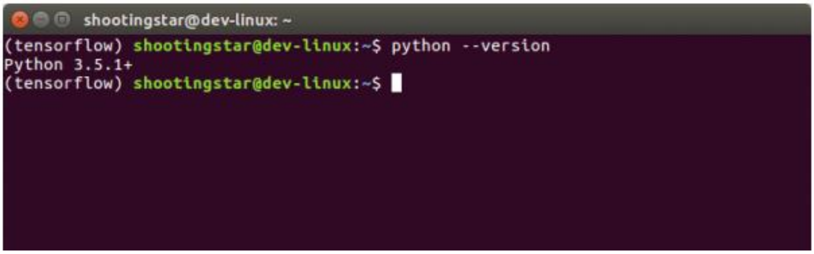
Then activate the virtualenv.

```
$ source ~/tensorflow/bin/activate
```



```
shootingstar@dev-linux:~$ virtualenv --system-site-packages -p python3 ~/tensorf
low
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/shootingstar/tensorflow/bin/python3
Not overwriting existing python script /home/shootingstar/tensorflow/bin/python
(you must use /home/shootingstar/tensorflow/bin/python3)
Installing setuptools, pkg_resources, pip, wheel...done.
shootingstar@dev-linux:~$ 
```

Then check the version of Python you are using.



```
(tensorflow) shootingstar@dev-linux:~$ python --version
Python 3.5.1+
(tensorflow) shootingstar@dev-linux:~$ 
```

Find the URL of the appropriate tensorflow binary version for Python version from the list below.

# Ubuntu/Linux 64-bit, CPU only, Python 2.7

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/cpu/tensorflow-0.9.0-cp27-none- linux_x86_64.whl
```

# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/gpu/tensorflow-0.9.0-cp27-none- linux_x86_64.whl
```

# Mac OS X, CPU only, Python 2.7:

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/mac/t
ensorflow-0.9.0-py2-none-any.whl
```

# Ubuntu/Linux 64-bit, CPU only, Python 3.4

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/cpu/tensorflow-0.9.0-cp34-cp34m -linux_x86_64.whl
```

# Ubuntu/Linux 64-bit, GPU enabled, Python 3.4
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/gpu/tensorflow-0.9.0-cp34-cp34m -linux_x86_64.whl
```

# Ubuntu/Linux 64-bit, CPU only, Python 3.5

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/cpu/tensorflow-0.9.0-cp35-cp35m -linux_x86_64.whl
```

# Ubuntu/Linux 64-bit, GPU enabled, Python 3.5
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/gpu/tensorflow-0.9.0-cp35-cp35m -linux_x86_64.whl
```

# Mac OS X, CPU only, Python 3.4 or 3.5:

```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/mac/t
ensorflow-0.9.0-py3-none-any.whl
```

Then execute export.

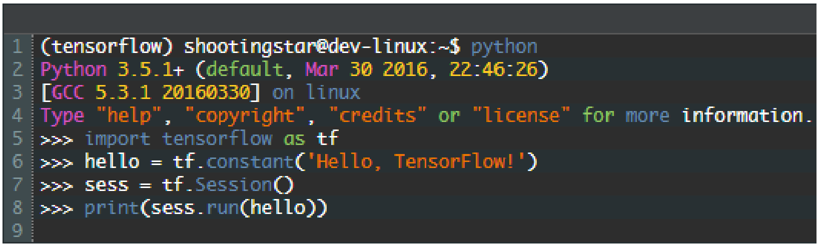
```
(tensorflow)$ export
TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux
/gpu/tensorflow-0.9.0-cp35-cp35m -linux_x86_64.whl
```

Of course you could select higher version Tensorflow and install it same method.

Install the Tensorflow using the following command.

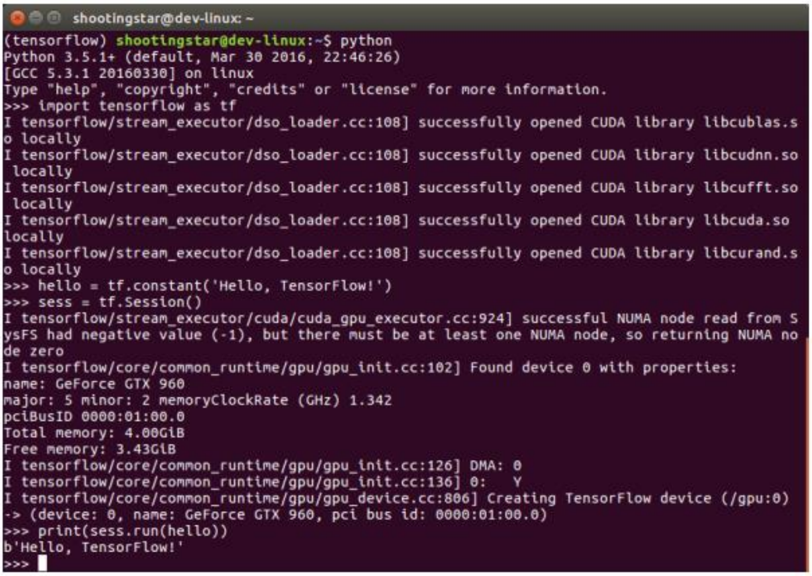
```
(tensorflow)$ pip3 install --upgrade $TF_BINARY_URL
```

Test:



```
(tensorflow) shootingstar@dev-linux:~$ python
Python 3.5.1+ (default, Mar 30 2016, 22:46:26)
[GCC 5.3.1 20160330] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))

```

You can see results similar to the following.



```
(tensorflow) shootingstar@dev-linux:~$ python
Python 3.5.1+ (default, Mar 30 2016, 22:46:26)
[GCC 5.3.1 20160330] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcublas.s
o locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcudnn.so
 locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcufft.so
 locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcuda.so
locally
I tensorflow/stream_executor/dso_loader.cc:108] successfully opened CUDA library libcurand.s
o locally
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:924] successful NUMA node read from S
ysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA no
de zero
I tensorflow/core/common_runtime/gpu/gpu_init.cc:102] Found device 0 with properties:
name: GeForce GTX 960
major: 5 minor: 2 memoryClockRate (GHz) 1.342
pciBusID 0000:01:00.0
Total memory: 4.00GiB
Free memory: 3.43GiB
I tensorflow/core/common_runtime/gpu/gpu_init.cc:126] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_init.cc:136] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:806] Creating TensorFlow device (/gpu:0)
-> (device: 0, name: GeForce GTX 960, pci bus id: 0000:01:00.0)
>>> print(sess.run(hello))
b'Hello, TensorFlow!'
>>> 
```

You can install Jupyter or PyCharm for edit Python code on Ubuntu similar with Windows.

And on Mac OS we could install Tensorflow similar with Ubuntu, too.

Task2 is complete.

LAB CHALLENGE

Challenge

In this task what is more important is select the version of packages such as Python, Tensorflow, and Anaconda and so on.

See below URLs for a more detailed description.

- https://www.tensorflow.org/install/
- https://www.tensorflow.org/versions/r0.12/get_started/os_setup

SUMMARY

Tensorflow is one of the mostly used framework and open source library for machine learning and deep learning developed by Google.

Tensorflow package is available for download along with github open source code or Tensorflow website.

REFERENCES

- https://en.wikipedia.org/wiki/TensorFlow
- https://www.tensorflow.org/install/

INDEX