**安装指南**

**1.前言**

由于这是我头一次接触Tensflow和深度学习的东西，所以在安装的时候遇到了不少的问题，非常感谢各种前辈们的博文，在经历了不停的搜索和挣扎后，我终于成功的安装了Tensorflow with GPU support。

好啦，废话不多说，下面就是详细的Tensorflow-Gpu安装详情了。

**2.准备工作**

在安装Tensorflow前我们首先要确定自己的电脑是否支持Gpu Support，以下是确认清单（如果已经核实完毕，可以跳过本章节）

* 显卡的品牌必须为**NVIDIA**
* 显卡必须要在**CUDA**支持列表中-[Support GPUs](https://developer.nvidia.com/cuda-gpus)
* 安装 **Python2.7** 或者 **Python3.x** (可以同时安装2.7和3.x)
* 确认你的 **pip** 或 **pip3** 版本 **>= 8.1**
* 安装Python版本对应的 **virtualenv**

**2.1 确认GPU是否支持CUDA**

1 访问[https://developer.nvidia.com/cuda-gpus](https://developer.nvidia.com/cuda-gpus" \t "_blank)

![](http://ox0sjjwt5.bkt.clouddn.com/17-9-29/35352778.jpg)

2 查找自己GPU的型号在下面的列表中(**Compute Capability 体现的是你的运算能力,和其他无关**)

![](http://ox0sjjwt5.bkt.clouddn.com/17-9-29/76156650.jpg)

**2.2 安装python**

**2.2.1 Pythone 2.7 的安装与验证**

# 安装 python2.7 发布版本，dev包必须安装，很多用pip安装包都需要编译

$ sudo apt-get install python2.7 python2.7-dev

# 校验安装的python 版本

$ python --version

输出结果: Python 2.7.12

**2.2.2 Pythone 3.x 的安装与验证**

# 安装 Python3.x 发布版本，dev包必须安装，很多用pip安装包都需要编译

$ sudo apt-get install python3.6 python3.6-dev

# 校验安装的python3 版本

$ python3 --version

输出结果: Python 3.6.2

**Topic : 安装常见错误**

* Question 1 : 无法定位软件包

# 一般是因为无法获取正确的源造成的，请更新Ubuntu的源，并检查输入软件名称是否有误

# 换源指南 http://blog.csdn.net/happywho250/article/details/52506321

$ sudo apt-get update

**2.3 安装pip**

**2.3.1 Python 2.x pip 安装与校验**

# 安装 pip

$ sudo apt-get install python-pip

# 校验安装的pip 版本

$ pip --version

输出结果: pip 9.0.1 from /usr/local/lib/python2.7

**2.4.2 Pythone 3.x pip3 安装与校验**

# 安装 pip3

$ sudo apt-get install python3-pip

# 校验安装的pip3 版本

$ pip3 --version

输出结果: 输出结果: pip 9.0.1 from /usr/local/lib/python3.6

**2.4.3 更新Pythone对应版本的pip(version >= 8.1 请跳过该步骤)**

# Pythone 2.7 对应 pip 的升级方法

$ sudo pip install --upgrade pip

# Pythone 3.x 对应 pip 的升级方法

$ sudo pip3 install --upgrade pip

**2.4 安装 virtualenv**

**2.4.1 virtualenv 安装与校验**

# 安装 virtualenv

$ sudo python-virtualenv

# Pythone 3.x 对应 pip 的升级方法

$ virtualenv --version

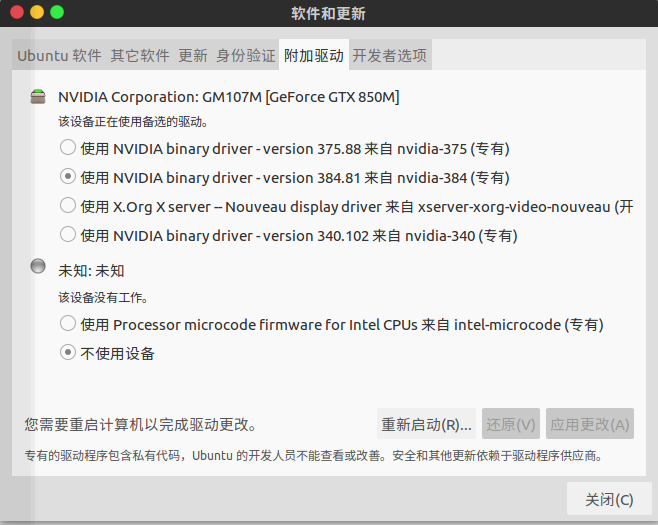
输出结果: 15.1.0

**3.安装CUDA和cuDNN支持**

**3.1 安装NVIDIA驱动**

# 从软件源更新软件列表

$ sudo apt-get update

系统设置->软件更新->附加驱动->选择nvidia最新驱动(384)->应用更改 

**3.2 安装 CUDA 和 cuDNN 依赖**

# 安装CUDA和cuDNN安装依赖

$ sudo apt-get install libcupti-dev

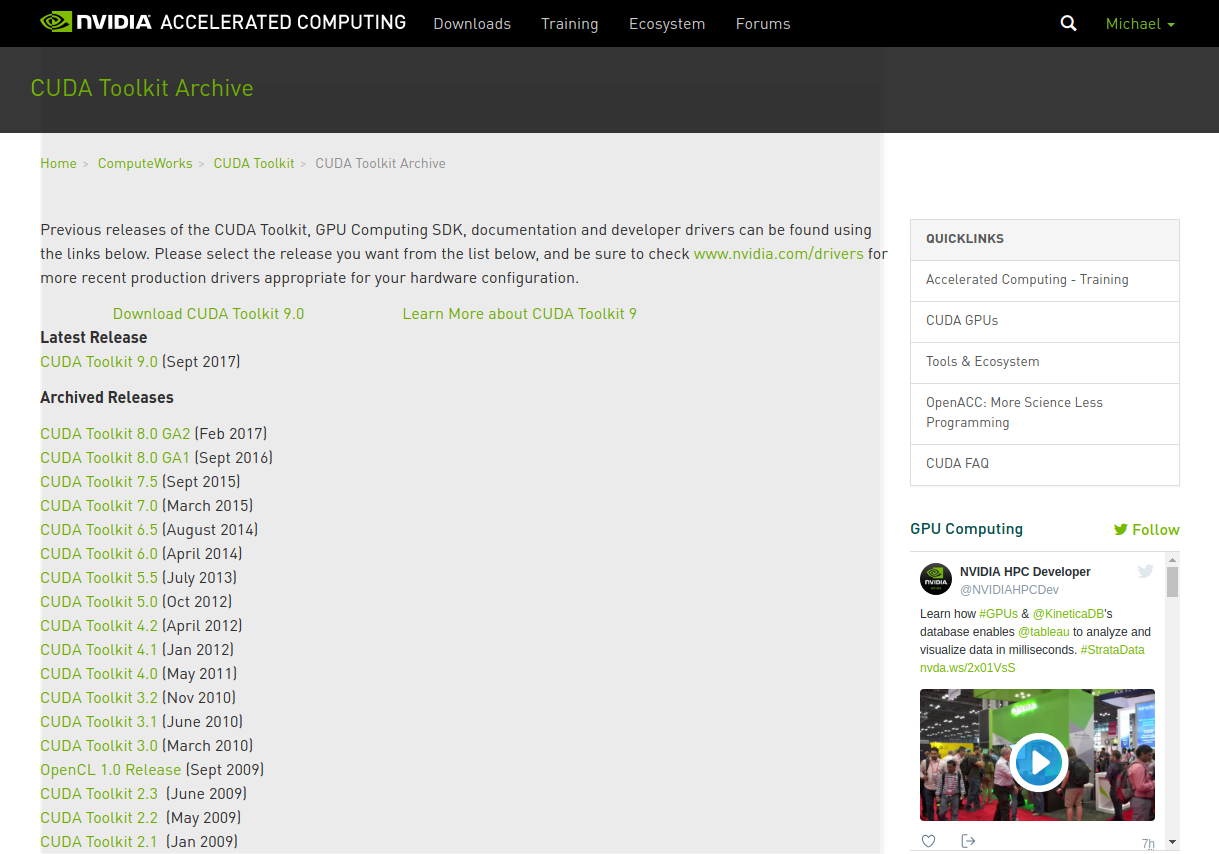
**3.3 下载安装 CUDA8.0 和 cuDNN 6.0**

**3.3.1 CUDA8.0 安装**

**Tip** : 为什么不下载 CUDA 9.0 和 cuDNN , 因为Tensflow暂时不支持这个版本 , 如果您安装的时间离 2017-09-29 有一段时间的话 ， 推荐您先搜索目前Tensflow支持CUDA的版本

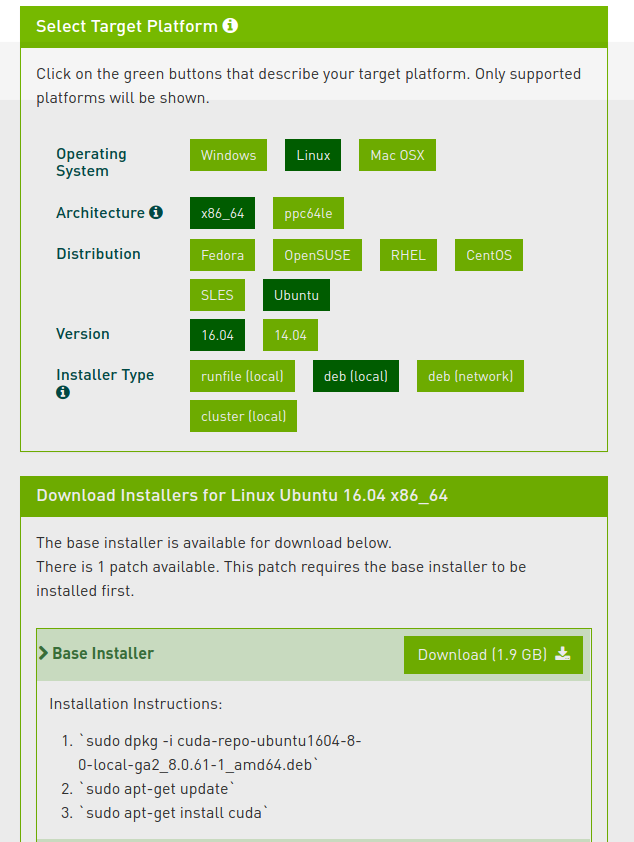
1 访问 <https://developer.nvidia.com/cuda-toolkit-archive>

2 选择对应版本 CUDA ToolKit 8.0 GA2 (Feb 2017)



3 下载对应版本的Deb包

* Case 1 ：Deb-Local版下载(推荐)



# 安装Deb的本地版本

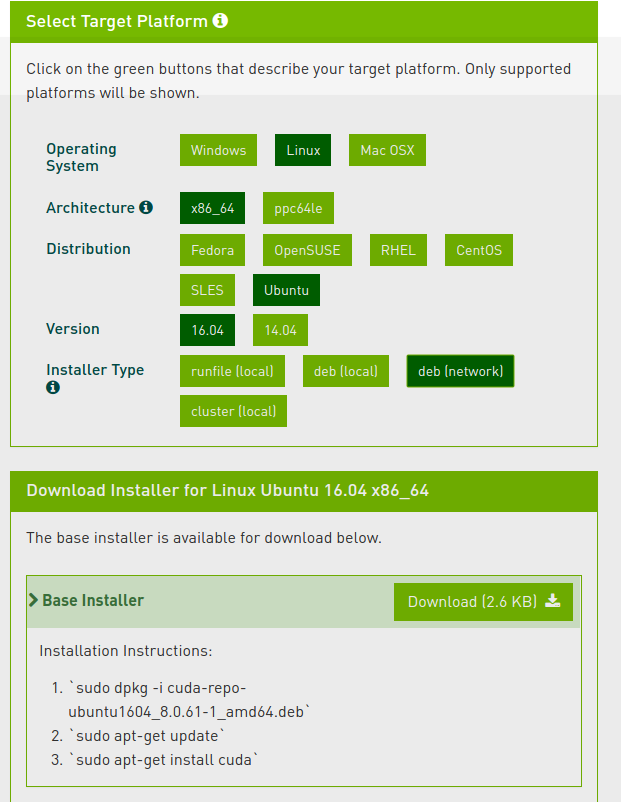
$ sudo dpkg -i cuda-repo-ubuntu1604-8-0-local-ga2\_8.0.61-1\_amd64.deb

$ sudo apt-get update

# 请注意必须要使用cuda-8.0，不然将会现在现在的最新版本

$ sudo apt-get install cuda-8.0

* Case 2 ：Deb-Network版下载(不推荐,仅当Local版本无法使用时作为代替)



# 安装Deb的网络版本

$ sudo dpkg -i cuda-repo-ubuntu1604\_8.0.61-1\_amd64.deb

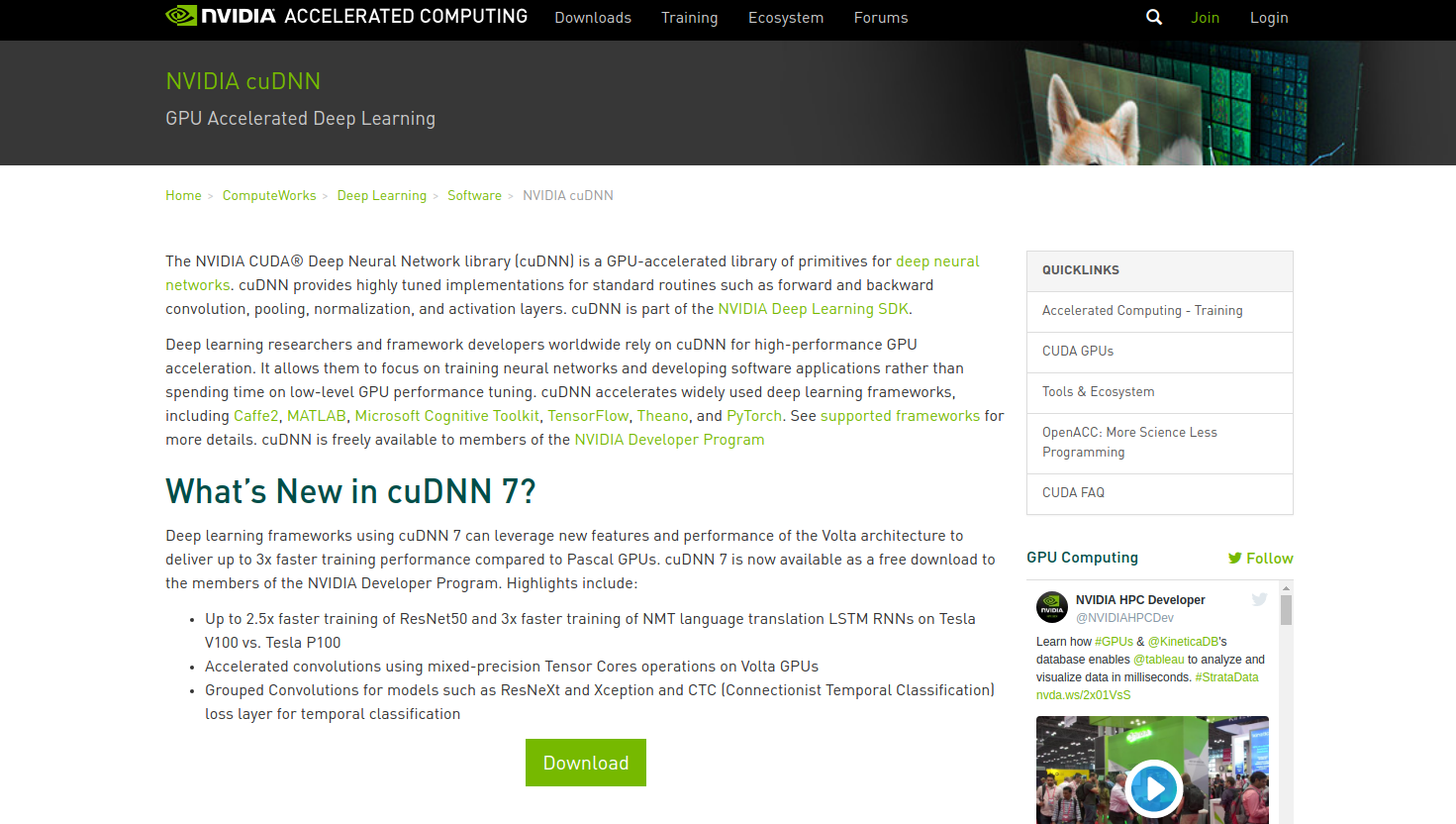
$ sudo apt-get update

# 请注意必须要使用cuda-8.0，不然将会现在现在的最新版本

$ sudo apt-get install cuda-8.0

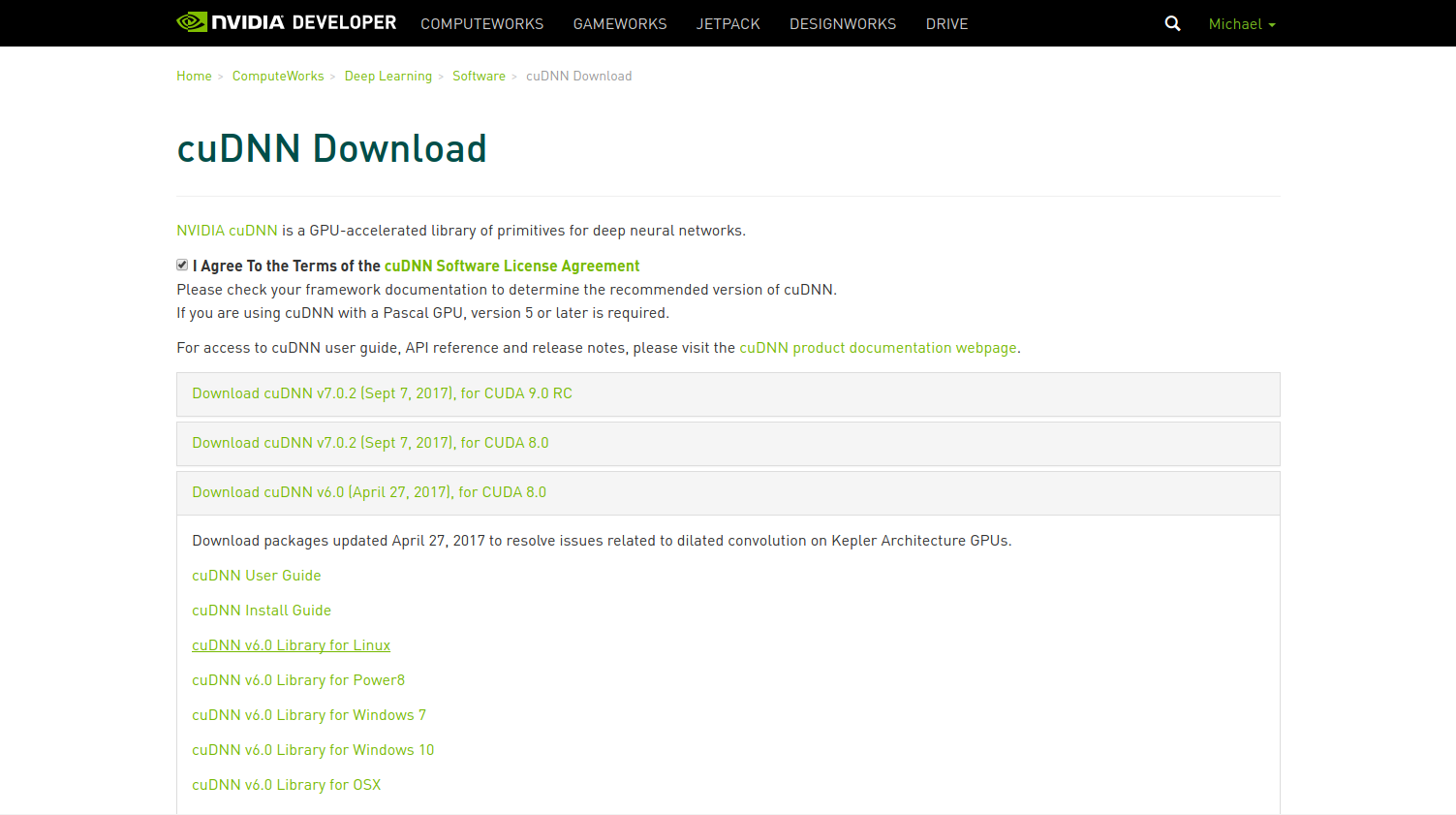
**3.3.2 cuDNN 安装**

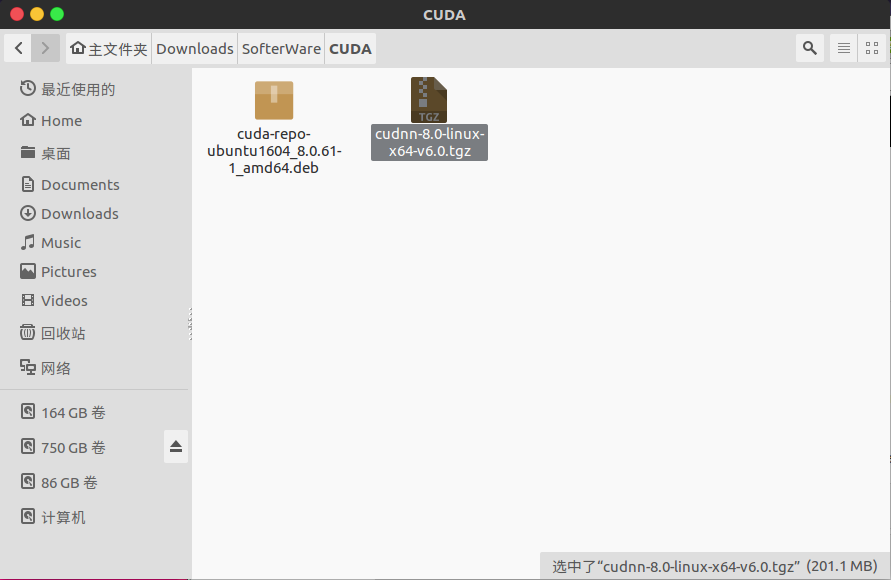
1 访问 <https://developer.nvidia.com/cudnn> 点击Download后注册登录



2 下载### 附录3 ： 参考文献

* [ubuntu16.04下安装TensorFlow(GPU加速)----详细图文教程](http://blog.csdn.net/zhaoyu106/article/details/52793183) by 蓝色荣誉
* [TensorFlow 官方文档中文版](http://wiki.jikexueyuan.com/project/tensorflow-zh/get_started/os_setup.html) by 极客学院
* [Installing TensorFlow on Ubuntu](https://www.tensorflow.org/install/install_linux#InstallingVirtualenv) by Tensorflow官网
* [NVIDIA CuDNN 安装说明](http://www.cnblogs.com/platero/p/4118139.html) by 普兒
* [NVIDIA CuDNN 安装说明](http://blog.csdn.net/wangkun1340378/article/details/72782593) by 无奈的小心酸 cuDNN v6.0 Library for Linux



3 查看下载后的文件(如图所示下载的内容应该是一个.tgz文件) 

4 在terminal中输入以下命令安装

# 安装指南 (请在安装完CUDA后再安装cuDNN)

$ tar xvzf cudnn-8.0-linux-x64-v6.0.tgz

$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include

$ sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64

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

**3.3.3 配置 GPU Support 的环境依赖**

以下两种方法你可以**任选一种**配置环境变量

* case one : **在.bash\_profile中配置环境变量**

#在terminal中输入以下命令：

$ sudo gedit ~/.bash\_profile

在文本中新增环境配置：

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"

export CUDA\_HOME=/usr/local/cuda

#保存后，在terminal中输入以下命令：

$ source ~/.bash\_profile

* case two : **在.bash\_profile中配置环境变量**

#在terminal中输入以下命令：

$ sudo gedit ~/.bashrc

在文本中新增环境配置：

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"

export CUDA\_HOME=/usr/local/cuda

#保存后，在terminal中输入以下命令：

$ source ~/.bashrc

**4.安装Tensorflow**

**4.1 Pythone 2.7 安装Tensorflow**

1 **首先找到你打算安装Tensorflow环境的位置(本教程中安装的位置为~目录下)**

2 **在terminal中输入以下命令：**

# Case 1 :安装Tensorflow With GPU

$ virtualenv --system-site-packages tensorflow-gpu //tensorflow-gpu 为文件夹名称

$ source ~/tensorflow-gpu/bin/activate

// 在这之后请注意观察 $ 前面的部分

(tensorflow-gpu) $ pip install --upgrade tensorflow-gpu

//在terminal中输入以下命令退出虚拟环境

(tensorflow-gpu) $ deactivate

# Case 2 :安装Tensorflow CPU-only support

$ virtualenv --system-site-packages tensorflow //tensorflow 为文件夹名称

$ source ~/tensorflow/bin/activate

// 在这之后请注意观察 $ 前面的部分

(tensorflow) $ pip install --upgrade tensorflow

//在terminal中输入以下命令退出虚拟环境

(tensorflow) $ deactivate

**4.2 Pythone 3.x 的Tensorflow安装**

1 **首先找到你打算安装Tensflow环境的位置(教程中安装的位置为~目录下)**

2 **在terminal中输入以下命令：**

# Case 1 :安装Tensorflow With GPU

$ virtualenv --system-site-packages -p python3 tensorflow-p3-gpu //tensorflow-p3-gpu 为文件夹名称

$ source ~/tensorflow-p3-gpu/bin/activate

// 在这之后请注意观察 $ 前面的部分

(tensorflow-p3-gpu) $ pip3 install tensorflow-gpu

//在terminal中输入以下命令退出虚拟环境

(tensorflow-p3-gpu) $ deactivate

# Case 2 :安装Tensorflow CPU-only support

$ virtualenv --system-site-packages -p python3 tensorflow-p3 //tensflow-gpu 为文件夹名称

$ source ~/tensorflow-p3/bin/activate

// 在这之后请注意观察 $ 前面的部分

(tensorflow-p3) $ pip3 install tensorflow

//在terminal中输入以下命令退出虚拟环境

(tensorflow-p3) $ deactivate

**4.3 测试Tensorflow安装结果**

# Case 1 : Python 2.7 Tensorflow 测试

$ source ~/tensorflow-gpu/bin/activate // 请将source后的地址换成您虚拟文件夹所在地址

$ python

>>> import tensorflow as tf

>>> hello = tf.constant('Hello, TensorFlow!')

>>> sess = tf.Session()

>>> print(sess.run(hello))

Hello, TensorFlow!

>>> a = tf.constant(10)

>>> b = tf.constant(32)

>>> print(sess.run(a + b))

42

# Case 2 : Python 3.x Tensorflow 测试

$ source ~/tensorflow-p3-gpu/bin/activate // 请将source后的地址换成您虚拟文件夹所在地址

$ python3

>>> import tensorflow as tf

>>> hello = tf.constant('Hello, TensorFlow!')

>>> sess = tf.Session()

>>> print(sess.run(hello))

Hello, TensorFlow!

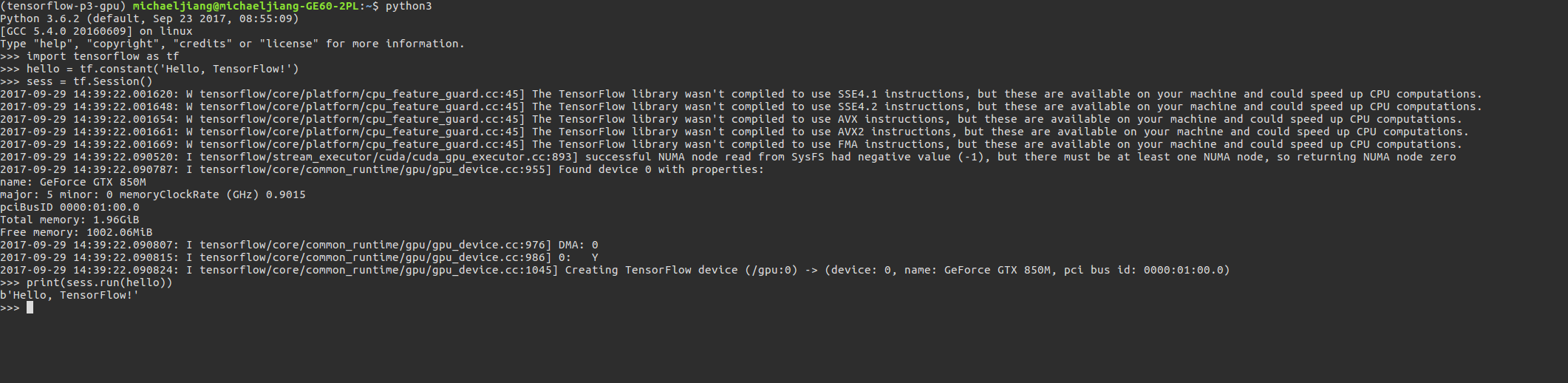
>>> a = tf.constant(10)

>>> b = tf.constant(32)

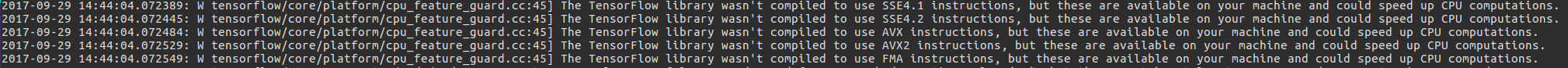
>>> print(sess.run(a + b))

42

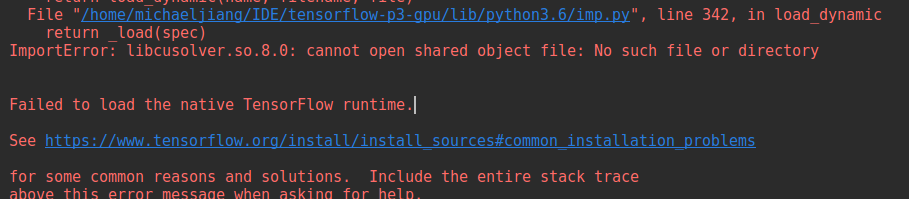
* Topic 1：如果是GPU支持版本的话，在Terminal中会输出GPU型号和训练时间



* Topic 2：细心的同学会发现除了成功信息外会有不少警告信息，这说明你的CPU并没有达到最高效率，如果希望提高速度去掉这几个Warning的话，你需要下载Tensorflow源码自己进行打包编译



* Topic 3：GPU版本如果在导入Tensorflow的时候出现bug，请参考以下方式修改CUDA文件权限



# 在terminal中输入以下命令：

$ cd /usr/local/cuda/lib64/

$ sudo rm -rf libcudnn.so libcudnn.so.6

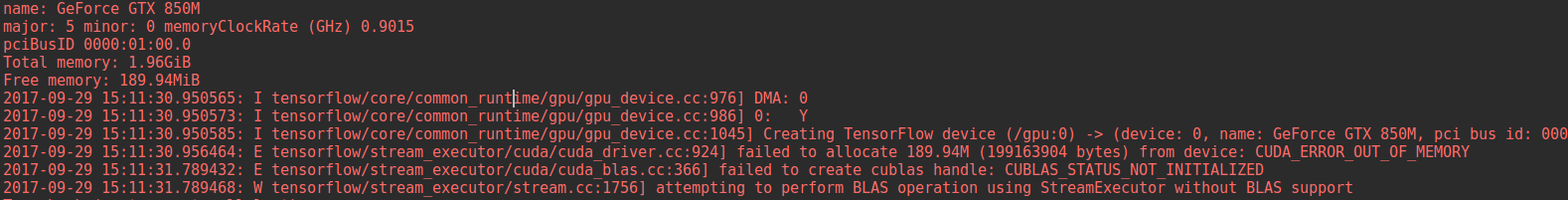
$ sudo chmod u=rwx,g=rx,o=rx libcudnn.so.6.0.21

$ sudo ln -s libcudnn.so.6.0.21 libcudnn.so.6

$ sudo ln -s libcudnn.so.6 libcudnn.so

$ sudo ldconfig

* Topic 4：如果在程序运行的过程始终出现CUDA out of memory，这是因为Tensorflow申请的GPU内存大于GPU能提供的内存



# 在代码session定义前新增配置：

>>> config = tf.ConfigProto(allow\_soft\_placement=True)

# 最多占gpu资源的70%

>>> gpu\_options = tf.GPUOptions(per\_process\_gpu\_memory\_fraction=0.7)

# 开始不会给tensorflow全部gpu资源 而是按需增加

>>> config.gpu\_options.allow\_growth = True

>>> sess = tf.Session(config=config)

**附录**

**附录1 ： Ubuntu常见问题解决方法**

* Question 1： E 无法定位软件包

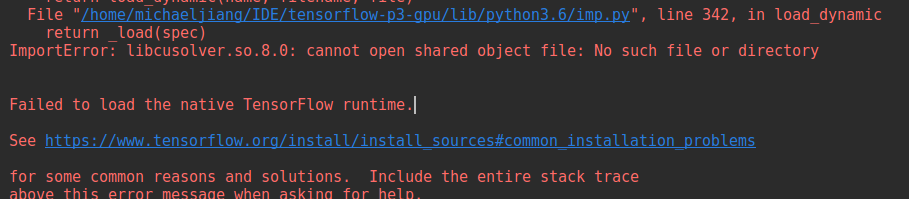
# 更换Ubuntu的软件源，并检查是否有输错软件名称

$ sudo apt-get update

[Ubuntu换源帮助](http://blog.csdn.net/happywho250/article/details/52506321)

**附录2 ： CUDA 和 cuDNN 安装常见问题解决方法**

* Question 1： CUDA文件权限不足,无法被链接



#在terminal中输入以下命令：

$ cd /usr/local/cuda/lib64/

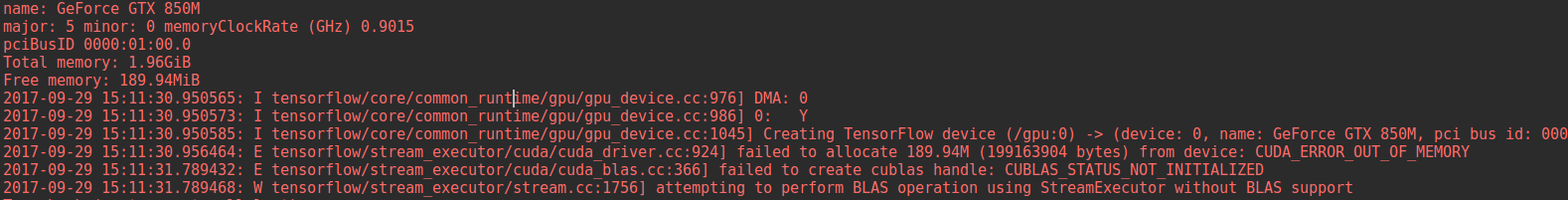
$ sudo rm -rf libcudnn.so libcudnn.so.6

$ sudo chmod u=rwx,g=rx,o=rx libcudnn.so.6.0.21

$ sudo ln -s libcudnn.so.6.0.21 libcudnn.so.6

$ sudo ln -s libcudnn.so.6 libcudnn.so

* Question 2： CUDA out of memory



# 在代码session定义前新增配置：

>>> config = tf.ConfigProto(allow\_soft\_placement=True)

# 最多占gpu资源的70%

>>> gpu\_options = tf.GPUOptions(per\_process\_gpu\_memory\_fraction=0.7)

# 开始不会给tensorflow全部gpu资源 而是按需增加

>>> config.gpu\_options.allow\_growth = True

>>> sess = tf.Session(config=config)

**附录3 ： 参考文献**

* [ubuntu16.04下安装TensorFlow(GPU加速)----详细图文教程](http://blog.csdn.net/zhaoyu106/article/details/52793183) by 蓝色荣誉
* [TensorFlow 官方文档中文版](http://wiki.jikexueyuan.com/project/tensorflow-zh/get_started/os_setup.html) by 极客学院
* [Installing TensorFlow on Ubuntu](https://www.tensorflow.org/install/install_linux#InstallingVirtualenv) by Tensorflow官网
* [NVIDIA CuDNN 安装说明](http://www.cnblogs.com/platero/p/4118139.html) by 普兒
* [NVIDIA CuDNN 安装说明](http://blog.csdn.net/wangkun1340378/article/details/72782593) by 无奈的小心酸