



Ubuntu 16.04 LTS+GTX1070 环境下安装 tensorflow GPU 详细流程

2018-03-21 chengkai 149 views

Contents

[\[隐藏\]](#)

- 1 环境
- 2 安装过程概述
- 3 安装 NVIDIA 显卡驱动
 - 3.1 解决桌面循环登陆问题
- 4 准备后续安装文件
 - 4.1 下载 CUDA 8.0
 - 4.2 下载 cudnn 6.0
 - 4.3 下载 anaconda 2
- 5 安装 CUDA 8.0
 - 5.1 安装步骤
 - 5.2 设置环境变量
 - 5.3 测试 CUDA 是否安装成功
 - 5.4 安装缺失的库



- 6 安装 cuDNN 6.0
- 7 安装 anaconda 2
 - 7.1 安装步骤
 - 7.2 更换国内镜像
 - 7.3 遇到的问题
- 8 安装 tensorflow 1.4.0
 - 8.1 安装步骤
 - 8.2 tensorflow 清华镜像网站
- 9 在 pycharm 中使用 tensorflow

1. 环境

- 系统：Ubuntu 16.04 LTS
- 显卡：GTX 1070

2. 安装过程概述

1. 安装 NVIDIA 显卡驱动
2. 准备后续安装文件
3. 安装 CUDA8.0
4. 安装 cudnn6.0
5. 安装 anaconda2
6. 安装 tensorflow1.4.0
7. 在 pycharm 中使用 tensorflow

3. 安装 NVIDIA 显卡驱动

首先说一下我走过的坑：我安装完 Ubuntu 系统之后是没有安装 NVIDIA 显卡驱动的，后来在 **System Settings->Software & Updates->Additional Drivers** 里查到是一个 **X** 开头的驱动，只可惜没有记下来，因为安装上 NVIDIA 的驱动之后就不见了。由于是第一次搭建整个环境，上来就直接安装 **CUDA8.0** 的 **deb** 文件，而且没有注意到里面有 **NVIDIA** 显卡的驱动，就稀里糊涂的全部同意，安装完之后重启电脑就傻眼了，系统一直在登陆界面循环，就是进不了系统，而且 **CTRL+ALT+F1~F6** 也进不了字符界面，屏幕上只有一个光标，其他什么东西也没有，而且无法输入任何字符或命令。解决这个问题办法见本节的最后。后来我从官网单独下载了显卡 **.run** 驱动文件，安装完之后，也遇到了循环登陆的问题。后来尝试了好多办法，最后用下面的办法成功安装上了驱动。另外，网上有人说需要退出图形界面才能成功安装 **NVIDIA** 驱动，但是我没有退出也成功安装了，不知为何。说完我走过的坑，下面就介绍我成功安装 **NVIDIA** 显卡驱动的方法。

首先卸载驱动（不管有没有安装 **NVIDIA** 的驱动，先卸载一下再说）：

```
sudo nvidia-uninstall
```

在线安装:

```
sudo apt-add-repository ppa:graphics
```

```
sudo apt-get update
```

```
sudo apt-get install nvidia-384 #我用的是这个驱动
```

安装完之后, 在终端输入 `nvidia-smi`, 出现下面的信息, 说明显卡驱动安装成功

```
1 Tue Mar 20 17:23:58 2018
2 +-----+
3 | NVIDIA-SMI 384.111                Driver Version: 384.111                |
4 +-----+-----+-----+-----+-----+-----+
5 | GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
6 | Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
7 +-----+-----+-----+-----+-----+-----+
8 | 0   GeForce GTX 1070    Off      | 00000000:01:00.0 On  |          N/A         |
9 | 37%   32C    P8      15W / 250W | 304MiB / 8112MiB |      18%    Default  |
10 +-----+-----+-----+-----+-----+-----+
11
12
13 +-----+-----+-----+-----+-----+-----+
14 | Processes:                                GPU Memory |
15 |   GPU       PID    Type    Process name                     Usage      |
16 +-----+-----+-----+-----+-----+-----+
17 |    0         1318     G   /usr/lib/xorg/Xorg                     219MiB |
18 |    0         2015     G   compiz                             43MiB |
19 |    0         2667     G   ...-token=ABCA5882048C240CCA072B2C7142D502 35MiB |
20 |    0         2951     G   unity-control-center                   3MiB |
21 +-----+-----+-----+-----+-----+-----+
```

如果找不到 `nvidia-smi` 命令, 重启电脑就可以, 我就是重启电脑才能成功运行 `nvidia-smi` 命令。

3.1. 解决桌面循环登陆问题

重启电脑, 在启动的时候不选择 **Ubuntu**, 而选择第二项: `Advanced Options for Ubuntu`, 然后在接下来的界面中使用方向键选择 `root`, 会进入黑色的命令行, 且以 `root` 身份登陆。这时如果进入 `/home`, 使用 `ll` 查看文件夹信息, 会发现没有任何用户文件夹, 这需要等一会儿, 可以多运行几个 `cd` 和 `ll` 等不会改变文件的命令。过一会儿会发生变化, 一直按 **Enter** 键, 等正常之后, 再次进入 `/home`, 并使用 `ll` 查看, 这时就可以看到用户文件夹了。此刻才可以正常执行卸载显卡的命令。如果你们的电脑没有上面这个问题, 就直接执行卸载显卡驱动的命令就行, 可能是我的电脑有点儿奇葩。

执行以下命令卸载 **NVIDIA** 显卡驱动

```
sudo apt-get remove --purge nvidia-*
```

中间有一个确认提示, 输入 `y`, 并 **Enter** 就行了

最后 `reboot`, 重启之后就可以正常登陆了。

4. 准备后续安装文件

4.1. 下载 CUDA 8.0

CUDA Toolkit 8.0 GA2 (Feb 2017)

在这里选择 `Linux->x86_64->Ubuntu->16.04->runfile (local)` 下载。注意，一定要选择 `runfile`，不要选择 `deb`，因为 `deb` 文件自带显卡驱动，会把之前安装好的显卡驱动覆盖掉。下载后的文件名是 `cuda_8.0.61_375.26_linux.run`。

4.2. 下载 cudnn 6.0

下载 `cudnn` 之前需要注册并填写调查问卷，请认真填写这个调查问卷。

在 [这里](#) 下载，依次选择 “Download cuDNN v6.0 (April 27, 2017), for CUDA 8.0”->cuDNN v6.0 Library for Linux。下载下来是一个 `.tgz` 的压缩包，文件名是 `cudnn-8.0-linux-x64-v6.0.tgz`。

4.3. 下载 anaconda 2

官网下载比较慢，可以在 `byrbit` 上下载（针对在校生），也可以在清华大学开源软件镜像站里下载，选择 `Anaconda2-4.4.0-Linux-x86_64.sh` 下载。等安装完 `CUDA 8.0`、`cudnn 6.0` 和 `anaconda 2` 之后再下载 `tensorflow 1.4.0`，所以在此先不下载。

把下载好的 3 个文件拷贝的一个比较容易访问的目录，我把文件放到了 `~/Downloads` 下面。

5. 安装 CUDA 8.0

5.1. 安装步骤

首先进入 `~/Downloads` 目录，在终端执行命令

```
sudo sh ./cuda_8.0.44_linux.run
```

首先是一大堆条款，可以直接回车到底
输入 `accept` 接受条款，即下面

Do you accept the previously read EULA?

accept/decline/quit: **accept**

接下来是询问是否安装驱动，此处一定要选择 **no**，此处一定要选择 **no**，此处一定要选择 **no**，不然之前安装的驱动就被覆盖了，还可能导致循环登陆的问题。输入 `n`，不安装 `NVIDIA` 驱动。

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?

(y)es/(n)o/(q)uit: **n**

输入 `y`，安装 `CUDA Toolkit`

Install the CUDA 8.0 Toolkit?

(y)es/(n)o/(q)uit: **y**

回车，使用默认安装路径 `/usr/local/cuda-8.0`

Enter Toolkit Location

[default is /usr/local/cuda-8.0]:

输入 `y`。安装指向 `/usr/local/cuda` 的符号链接

Do you want to install a symbolic link at /usr/local/cuda?

(y)es/(n)o/(q)uit: `y`

输入 `y`。安装 Samples。

Install the CUDA 8.0 Samples?

(y)es/(n)o/(q)uit: `y`

回车。使用 Samples 默认安装路径 `/home/xxxx` 此处的 `xxxx` 指的是你的用户名

Enter CUDA Samples Location

[default is /home/xxxx]:

接下来就是安装，并显示一些信息。安装完之后会显示如下信息：

```

1 Installing the CUDA Toolkit in /usr/local/cuda-8.0 ...
2 Missing recommended library: libGLU.so
3 Missing recommended library: libX11.so
4 Missing recommended library: libXi.so
5 Missing recommended library: libXmu.so
6
7 Installing the CUDA Samples in /home/chengkai ...
8 Copying samples to /home/chengkai/NVIDIA_CUDA-8.0_Samples now...
9 Finished copying samples.
10
11 =====
12 = Summary =
13 =====
14
15 Driver:    Not Selected
16 Toolkit:   Installed in /usr/local/cuda-8.0
17 Samples:   Installed in /home/chengkai, but missing recommended libraries
18
19 Please make sure that
20 - PATH includes /usr/local/cuda-8.0/bin
21 - LD_LIBRARY_PATH includes /usr/local/cuda-8.0/lib64, or, add
22   /usr/local/cuda-8.0/lib64 to /etc/ld.so.conf and run ldconfig as root
23
24 To uninstall the CUDA Toolkit, run the uninstall script in
25 /usr/local/cuda-8.0/bin
26
27 Please see CUDA_Installation_Guide_Linux.pdf in
28 /usr/local/cuda-8.0/doc/pdf for detailed information on setting up CUDA.
29
30 ***WARNING: Incomplete installation! This installation did not install
31 the CUDA Driver. A driver of version at least 361.00 is required for CUDA
32 8.0 functionality to work.
33 To install the driver using this installer, run the following command,
34 replacing <CudaInstaller> with the name of this run file:
35     sudo <CudaInstaller>.run -silent -driver
36
37     Logfile is /tmp/cuda_install_3662.log

```

5.2. 设置环境变量

在终端中执行

```
sudo vim ~/.bashrc
```

在末尾添加

```
export CUDA_HOME=/usr/local/cuda-8.0
export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64/${LD_LIBRARY_PATH:+:
${LD_LIBRARY_PATH}}
export LPATH=/usr/lib/nvidia-384:$LPATH
export LIBRARY_PATH=/usr/lib/nvidia-384:$LIBRARY_PATH
```

注意：`LPATH`和`LIBRARY_PATH`要输入自己的显卡驱动路径，我的是 384。

保存退出，然后使用下面代码刷新使环境变量生效

```
source ~/.bashrc
```

注意到上面信息中提示缺少一些库文件，比如 `libGLU.so`，`libX11.so`，`libXi.so`，`libXmu.so`，使用下面命令添加缺少的库文件：

```
sudo apt-get install libglu1-mesa libxi-dev libxmu-dev libglu1-mesa-dev
```

如果现在输入 `nvcc -V` 的话，可能会提示命令 `nvcc` 找不到，需要重启电脑。在终端里执行

```
sudo reboot
```

重启完之后，输入

```
nvcc -V
```

可以看到 CUDA 的版本信息，如下所示

```
1 nvcc: NVIDIA (R) Cuda compiler driver
2 Copyright (c) 2005-2016 NVIDIA Corporation
3 Built on Tue_Jan_10_13:22:03_CST_2017
4 Cuda compilation tools, release 8.0, V8.0.61
```

5.3. 测试 CUDA 是否安装成功

然后进入 **CUDA Samples** 目录（如果没有更改 **Samples** 的安装目录的话，应该是在 `/home/username` 目录下，有一个 `NVIDIA_CUDA-x.x_Samples` 的文件，其中 `x.x` 是版本号），在目录里执行 `make` 命令，编译 NVIDIA 给的例子。（有可能会因为 `gcc` 的版本太高而编译出错，可以到网上搜索降级 `gcc` 的办法，把 `gcc` 降到 4.8 版本，我用的是 4.8.5 版本。查看 `gcc` 版本的方法是 `gcc --version`）。

5.4. 安装缺失的库

在编译的过程中，我遇到了下面的问题，找不到 `lglut` 库，错误提示如下：

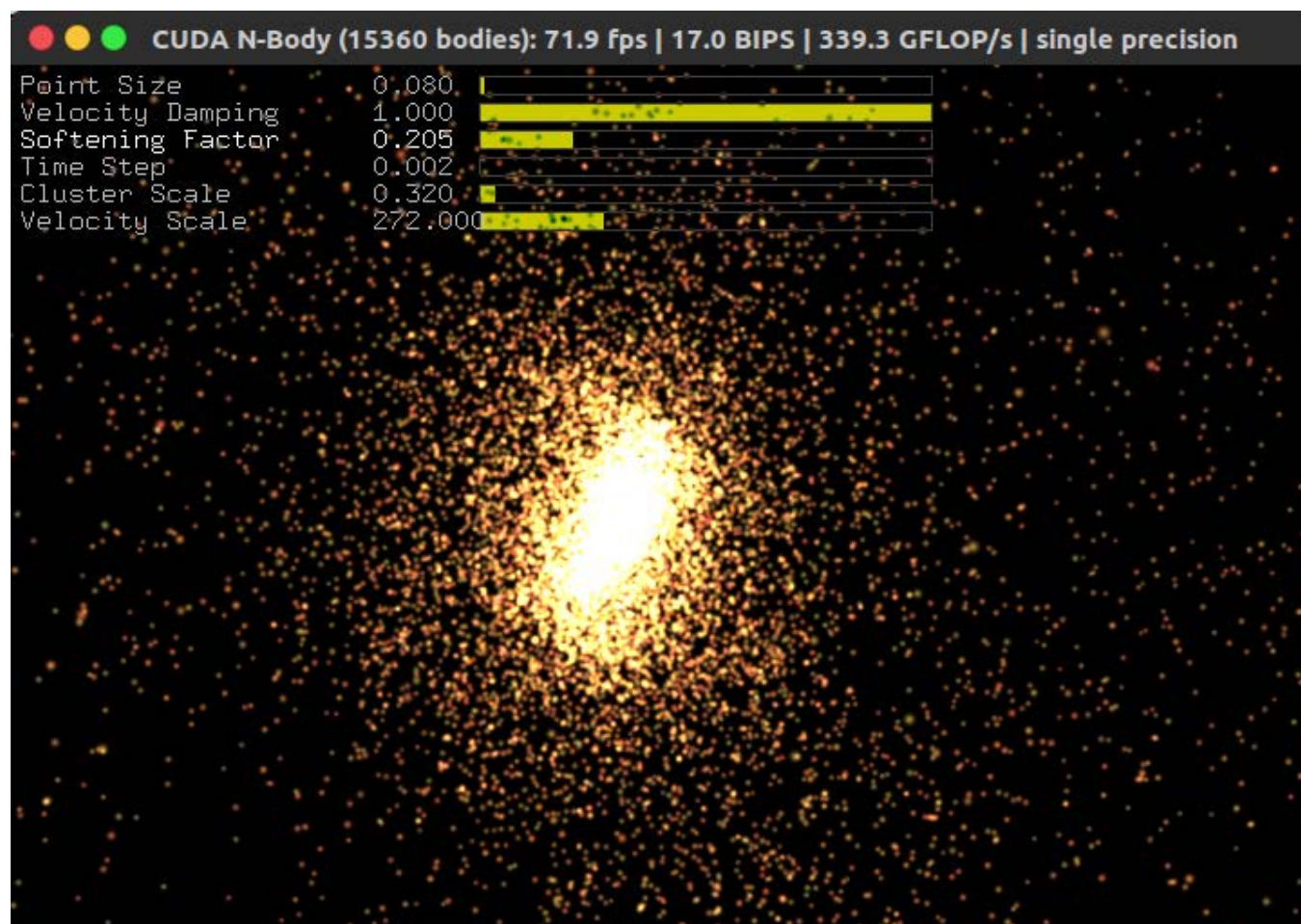
```
1 /usr/bin/ld: cannot find -lglut
2 collect2: error: ld returned 1 exit status
3 Makefile:270: recipe for target 'marchingCubes' failed
4 make[1]: *** [marchingCubes] Error 1
5 make[1]: Leaving directory '/home/chengkai/NVIDIA_CUDA-8.0_Samples/2_Graphics/marchingCu
6 bes'
7 Makefile:52: recipe for target '2_Graphics/marchingCubes/Makefile.ph_build' failed
make: *** [2_Graphics/marchingCubes/Makefile.ph_build] Error 2
```


然后用关键词 `lglt` 搜索，找到如下解决方案：

```
sudo apt-get install freeglut3 freeglut3-dev
```

通过上面命令就可以把缺失的库安装上。

然后执行 `make clean`，把编译错误的文件删掉，重新执行 `make` 命令编译。编译时间比较长，耐心等待。编译完之后，会在对应的文件夹中生成编译好的可执行文件。比如进入 `5_Simulation/nbody` 文件夹，在终端中执行 `./nbody`，会看到如下效果。



到此，CUDA 8.0 安装成功。

6. 安装 cuDNN 6.0

相比 CUDA 来讲，cuDNN 就简单多了。cuDNN 不是安装文件，而是一个头文件和一些库文件，只需要把这些文件复制到对应的目录上，并修改权限就可以了。安装过程如下：

进入到 `~/Downloads` 目录，使用 `tar` 进行解压

```
tar -xvf cudnn-8.0-linux-x64-v6.0.tgz
```

解压后把相应的文件拷贝到对应的 CUDA 目录下即可

```
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
```

```
sudo chmod a+r /usr/local/cuda/lib64/libcudnn*
```

这样 cuDNN 6.0 就安装好了。

升级的时候可能会用到，暂时不用管

```
1 tar zxvf cudnn-8.0-linux-x64-v5.1.tgz #解压
2 cd cuda/include #进入 include 目录
3 sudo cp cudnn.h /usr/local/cuda/include/ #复制头文件
4 cd ../lib64 #进入 lib64 目录
5 sudo cp lib* /usr/local/cuda/lib64/ #复制动态链接库
6 cd /usr/local/cuda/lib64/
7 sudo rm -rf libcudnn.so libcudnn.so.5 #删除原有动态文件
8 # 以下的两步设置软连接时，
9 一定要注意自己电脑的 /usr/local/cuda/lib64/ 下的 libcudnn.so.5.1.5 名字，
10 有可能是 libcudnn.so.5.0.5 等，要依据自己的电脑上的文件来定
11 sudo ln -s libcudnn.so.5.1.5 libcudnn.so.5 #生成软链接
12 sudo ln -s libcudnn.so.5 libcudnn.so #生成软链接
13 sudo ln -s libcudnn.so.5.1.10 libcudnn.so.5 #生成软链接
14 sudo ln -s libcudnn.so.5 libcudnn.so #生成软链接
```

7. 安装 anaconda 2

5.1. 安装步骤

进入终端，执行如下命令安装

```
sudo bash Anaconda2-4.4.0-Linux-x86_64.sh
```

读完 **license**，输入 **yes**

确认安装地址（可保留默认）

确认安装路径（可保留默认）

安装完之后，重新载入 **.bashrc** 文件（安装文件会在 **.bashrc** 文件末尾添加一行文字），载入方法如下：

```
source ~/.bashrc
```

7.2. 更换国内镜像

参考自：<https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/>

```
conda config --add channels
```

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/>

```
conda config --add channels
```

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/>

```
conda config --set show_channel_urls yes
```

即可添加 **Anaconda Python** 免费仓库。

7.3. 遇到的问题

1. 如果开了代理等科学上网工具，使用 **conda** 可能会出现错误，只需要把代理关掉就行。
2. 默认情况下，**conda** 会使用国外的镜像，导致速度比较慢，可以通过上一小节更换成国内镜像。

8. 安装 tensorflow 1.4.0

终于看到了 **tensorflow**，一步步下来不容易啊。

5.1. 安装步骤

参考自官方教程：https://www.tensorflow.org/install/install_linux#installing_with_anaconda

打开终端，创建 `conda` 环境，此处将环境命名为 `tensorflow`

```
conda create -n tensorflow pip python=2.7 # or python=3.3, etc.
```

激活 `conda` 环境

```
source activate tensorflow
```

```
(tensorflow)$ # Your prompt should change
```

使用如下代码安装 `tensorflow`，由于使用谷歌的镜像速度太慢，我换成了清华镜像，并选择 `tensorflow 1.4` 的 GPU 加速版本。

```
pip install \
```

```
-i https://pypi.tuna.tsinghua.edu.cn/simple/ \
```

```
https://mirrors.tuna.tsinghua.edu.cn/tensorflow/linux/gpu/tensorflow_gpu-1.4.0-cp27-none-linux_x86_64.whl
```

可以在 `conda` 环境中使用如下命令测试 `tensorflow` 是否安装好

```
1 import tensorflow as tf
2 hello = tf.constant('Hello, TensorFlow!')
3 sess = tf.Session()
4 print(sess.run(hello))
```

如果输出了

```
Hello, TensorFlow!
```

就代表 `tensorflow` 安装好了。

8.2. tensorflow 清华镜像网站

<https://mirrors.tuna.tsinghua.edu.cn/help/tensorflow/>

GPU 加速 1.4 版本 python2.7 的镜像链接如下

```
pip install \
```

```
-i https://pypi.tuna.tsinghua.edu.cn/simple/ \
```

```
https://mirrors.tuna.tsinghua.edu.cn/tensorflow/linux/gpu/
```

9. 在 pycharm 中使用 tensorflow

如何安装 `pycharm` 就不在这里说了，可自行百度。

安装完之后启动 `pycharm` 并创建一个空的 `Python` 工程。接下来有两步需要做：

第一步：修改 `Project Interpreter`。依次选择 `File->Settings...->Project: projectName->Project Interpreter`。在右边有个小齿轮，单击这个小齿轮，选择 `Add Local`，然后选择在第 5 步中创建的名为 `tensorflow` 的 `conda` 环境所在的目录中的 `python`，比如我的路径是 `~/.conda/envs/tensorflow/bin/python2.7`，选完路径之后，会发现在下面的列表中会出现 `tensorflow-gpu` 字样，就说明路径选对了，否则请再确认路径。最后点击 `Apply` 或 `OK` 应用。

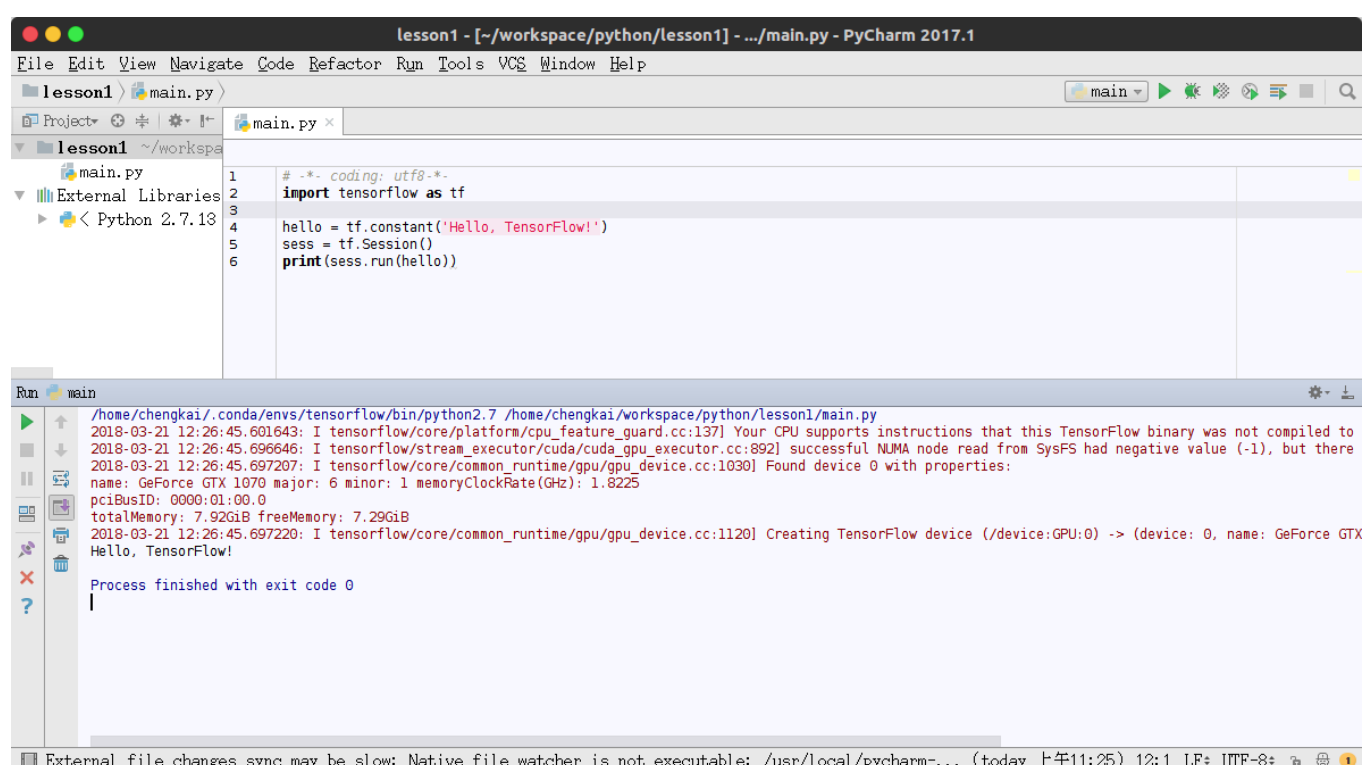
完成第一步，在 **windows** 环境下应该可以使用了，但是在 **Ubuntu** 环境下还不行，否则会出现类似于下面的错误

```
_mod = imp.load_module('_pywrap_tensorflow_internal', fp, pathname, description)
```

```
ImportError: libcusolver.so.8.0: cannot open shared object file: No such file or directory
```

第二步：出现上面错误的原因是由于通过桌面快捷方式启动 **pycharm** 或通过在终端中执行 **pycharm安装路径/bin/pycharm.sh** 的方式启动 **pycharm**，使得 **pycharm** 没有获取到环境变量，具体来说就是没有获取到 **~/.bashrc** 中的 **LD_LIBRARY_PATH** 变量的值（该变量是在安装 **CUDA 8.0** 的时候设置的）。这个问题的方法很简单（尽管很简单，但依旧花费了我一上午的时间才解决），就是在终端中执行 **charm** 命令就可以启动 **pycharm** 并获取环境变量。参考链接 <https://github.com/tensorflow/tensorflow/issues/9530>

完成以上两步之后就可以成功在 **pycharm** 中 **import tensorflow** 了。效果如下：



分享到： 0

☐ Algorithm ☐ Leave a comment

← **Ubuntu 安装 cuda9.1 之后循环登陆现象**

tensorboard 的使用方法 →

发表评论

要发表评论，您必须先[登录](#)。

登录 / 注册

[注册](#)

[登录](#)

[文章](#) [RSS](#)

[评论](#) [RSS](#)

[WordPress.org](#)

搜索

近期文章

[Solution2 for GSH](#)

[Solution for GSH](#)

[clion 安装与激活](#)

[Linux for Matlab 中文注释乱码（亲测有效）](#)

[正则表达式匹配](#)

近期评论

[chengkai](#) 发表在《[Establish Private VPN on VPS](#)》

writeessay 发表在《Establish Private VPN on VPS》

writeessay 发表在《Establish Private VPN on VPS》

张, 瑞瑞发表在《Compute Inversions of an Array in $\Theta(n\lg n)$ 》

张, 瑞瑞发表在《Exchange two numbers without the third number》

标签

addClass **Ajax** AJAX 同步 AJAX 异步 **BootStrap** c00ce56e Grid System hasClass html IE IT++ IT++ 分析与实例
IT++ 安装 **JavaScript** js **LaTeX** _{LTE} **MySQL** php php 连接 MySQL 数据库 removeClass root
toggleClass **VS 2010** 传输信道 偏移列 图片轮播 大图片自适应 密码 引用 **CSS** 文件 引用样式表文件 文档引用 无线通信系统仿真 物理信道
空白页 网格系统 自适应 获取元素位置 设置元素位置 路径 逻辑信道 重设 **root** 密码 页眉 页眉页脚 页脚