

ubuntu16.04+七彩虹 GTX1060 的 NVIDIA 驱动 +Cuda8.0+cuda5.1+tensorflow+keras 搭建深度学习环境

平台信息:

PC: ubuntu16.04、i5、七彩虹 GTX1060 显卡

作者: 庄泽彬(欢迎转载, 请注明作者)

说明:参考了网上的一堆的资料搭建了深度学习的开发环境, 下班在宿舍折腾了好几个晚上才搞定, 写篇文章记录一下。

一、安装 Nvidia GTX1060 显卡驱动

1.1 图 1 是本人使用的显卡, 暂时没有那么多钱, 买个便宜的玩玩, 把显卡插入电脑主板,然后开启 ubuntu 系统.

1.2 在终端输入 `lspci |grep VGA` 查看显卡型号, ubuntu 系统自带 nouveau 驱动, 因此我们启动系统之后可以识别到显卡, 但这毕竟不是 GTX1060 专用的显卡驱动, 我们不能使用这个驱动要屏蔽这个显卡驱动, 安装 NVIDIA GTX1060 显卡驱动, 我们在看看图 3 现在系统使用的是 CPU 集成显卡, 没有使用我们的独立显卡, 我们要切换到我们的 GTX1060 显卡.

```
Last login: Mon Sep 17 18:09:35 2018 from 192.168.1.102
zhuang@zhuang:~$ lspci |grep VGA
00:02.0 VGA compatible controller: Intel Corporation Device 5912 (rev 04)
01:00.0 VGA compatible controller: NVIDIA Corporation Device 1c02 (rev a1)
zhuang@zhuang:~$ █
```

图 2



图 3

1.3 在终端输入 `sudo ubuntu-drivers devices` 查看自己的系统适合安装什么版本的显卡驱动,当然也可以下载最新的, 这里自己选择, 本人选择系统推荐的版本 **nvidia-384**

```
zhuang@zhuang:~$ sudo ubuntu-drivers devices
[sudo] zhuang 的密码:
== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==
modalias : pci:v000010DEd00001C02sv00007377sd00000000bc03sc00i00
vendor    : NVIDIA Corporation
driver    : nvidia-384 ← distro non-free recommended
driver    : xserver-xorg-video-nouveau - distro free builtin

zhuang@zhuang:~$ █
```

图 4

到这里链接下载对应型号的显卡动:<https://www.geforce.cn/drivers>,设置见图 5, 之后点击开始搜索, 在搜索结果中选择自己下载的显卡驱动版本



注意: 如果是笔记本的话, 选择类型是 (notebook)



搜索结果

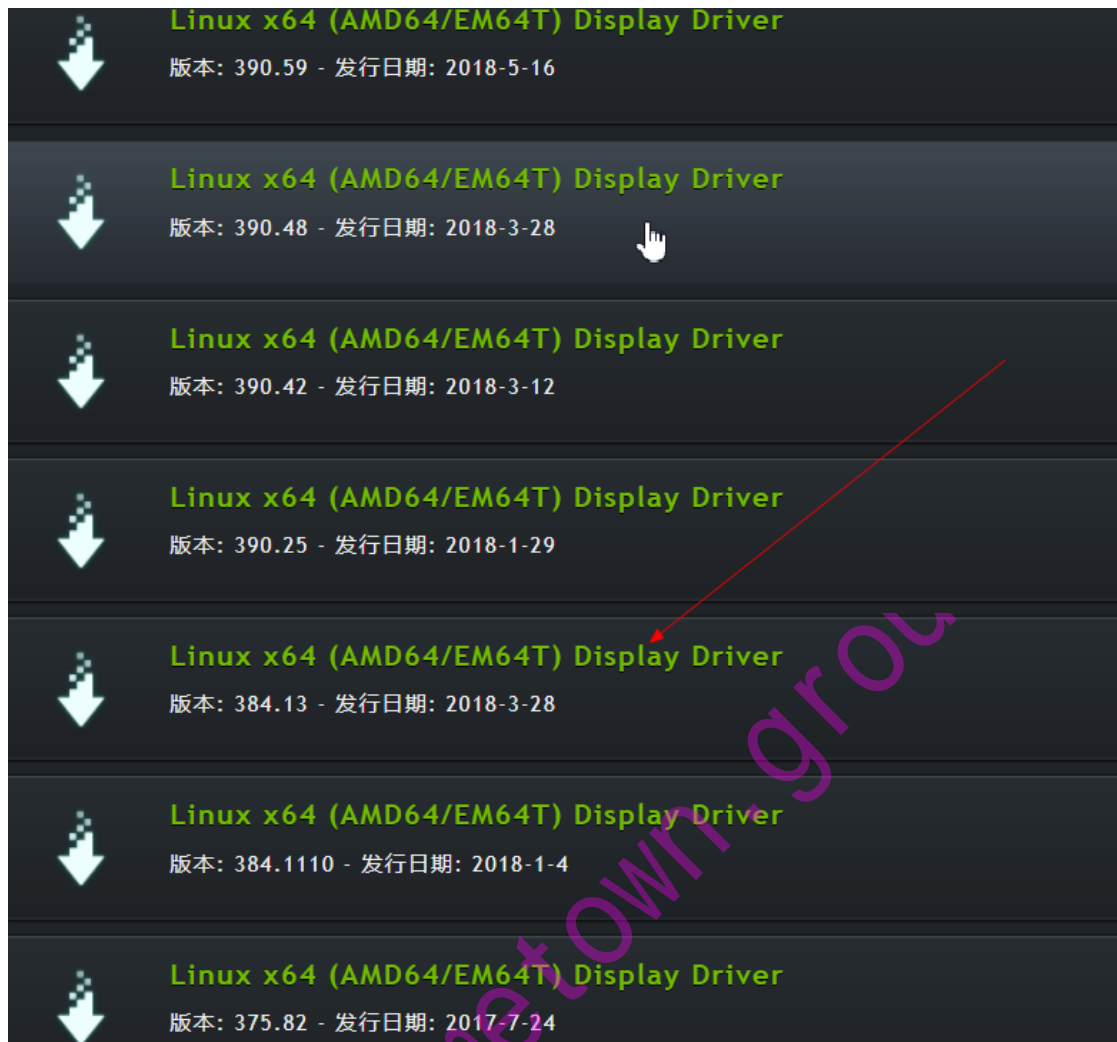


图 5

1.4 在正式安装之前执行先卸载原有的显卡驱动:

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

禁止 nouveau:

```
sudo vim /etc/modprobe.d/blacklist.conf
```

在这个文件最后添加:

```
blacklist nouveau
```

之后执行 `sudo update-initramfs -u` 重启系统

在终端输入 `lsmod | grep nouveau` 如果没有输出信息证明系统没有加载 nouveau 驱动, 可以进行下一步的操作。

1.5 安装驱动:

添加可执行权限:

```
sudo chmod a+x NVIDIA-Linux-x86_64-384.130.run
```

关闭图形化环境:

进入控制台模式: `ctrl+alt+f1`,

```
sudo /etc/init.d/lightdm stop
```

正式安装:

```
sudo ./NVIDIA-Linux-x86_64-384.130.run --no-x-check --no-nouveau-check --no-opengl-files
```

执行完之后跳出图 6 这个界面，后找了一下资料，其实出现这个界面是正常的，只是 NVIDIA 的人提醒你是否要进行安装，可以看看 /usr/lib/nvidia/pre-install 这个文件的内容图 7，其实没有做什么就退出，所以出现图 6 的界面不用担心，继续执行就好了啊。

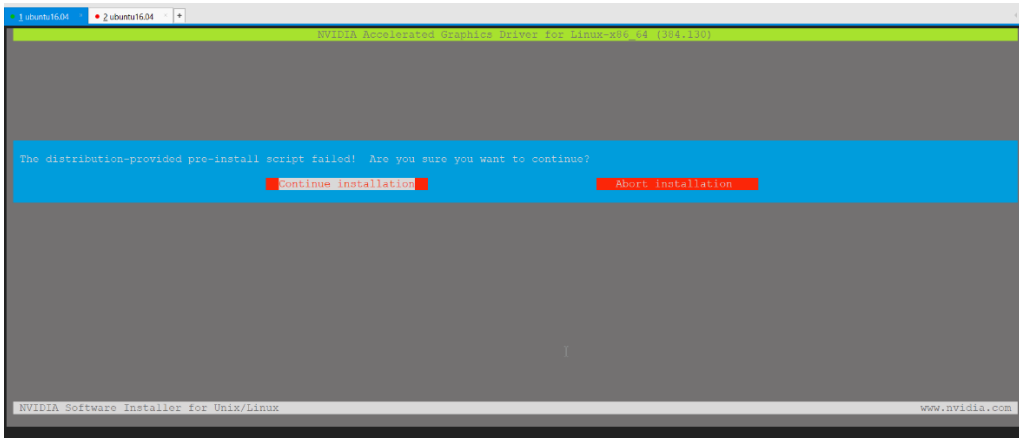


图 6

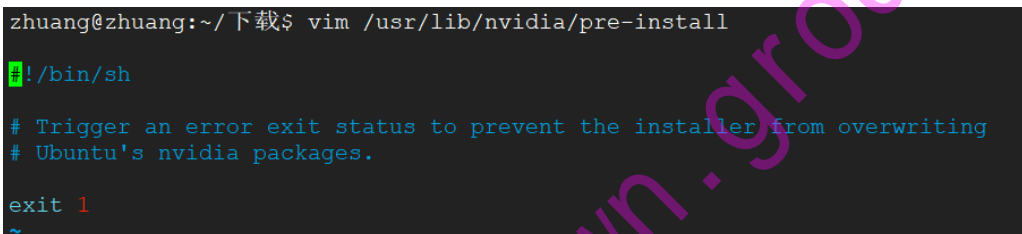


图 7

在安装的过程会出现的选项：

Install NVIDIA's 32-bit compatibility libraries?本人选择 **yes**

Would you like to run the nvidia-xconfig utility to automatically update your X configuration file so that the NVIDIA X driver will be used when you restart X? Any pre-existing X configuration file will be backed up.本人选 **NO**

等待安装完毕之后执行

sudo /etc/init.d/lightdm start 恢复图形界面

查看显卡驱动版本:从图 8 可以看出我们安装 GTX1060 显卡驱动已经搞定了，但是我们系统现在使用的还是图 3 所示的集成显卡，我们安装完 cuda8.0 之后就能切换过去，因此革命尚未成功仍需 nvli

cat /proc/driver/nvidia/version

nvidia-smi

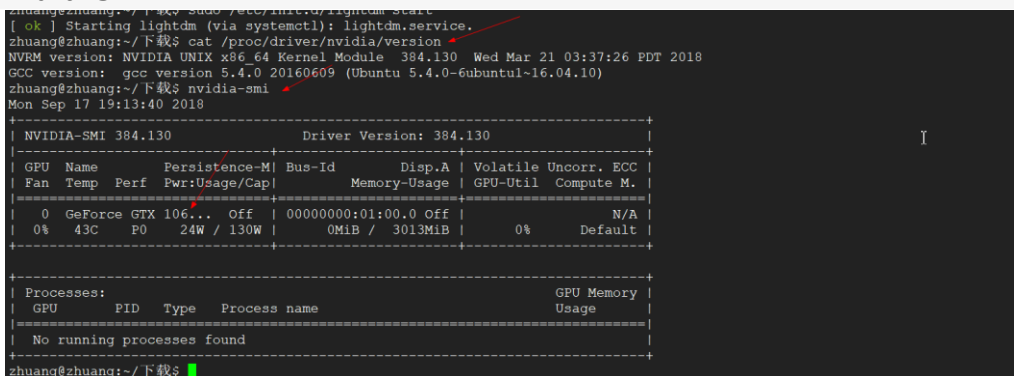


图 8

如果需要升级版本卸载: `sudo ./NVIDIA-Linux-x86_64-384.130.run --uninstall`

二、安装 cuda8.0

2.1 下载 cuda8.0 的下载链接:<https://developer.nvidia.com/cuda-80-ga2-download-archive>, 本人的下载设置见图 9,之后点击下载, 需要注意的是 CUDA 包括了 CPU 计算库和显卡驱动, 因此在安装的时候不要重复安装显卡驱动, 不然前面装了那么久 GTX1060 显卡驱动就白装了啊。

注意: 最好安装 cuda9.0 及以上, cuda8 不支持新的 tensorflow 版本, 新版本安装方法一样。注意下面配置中要把 8 改成 9

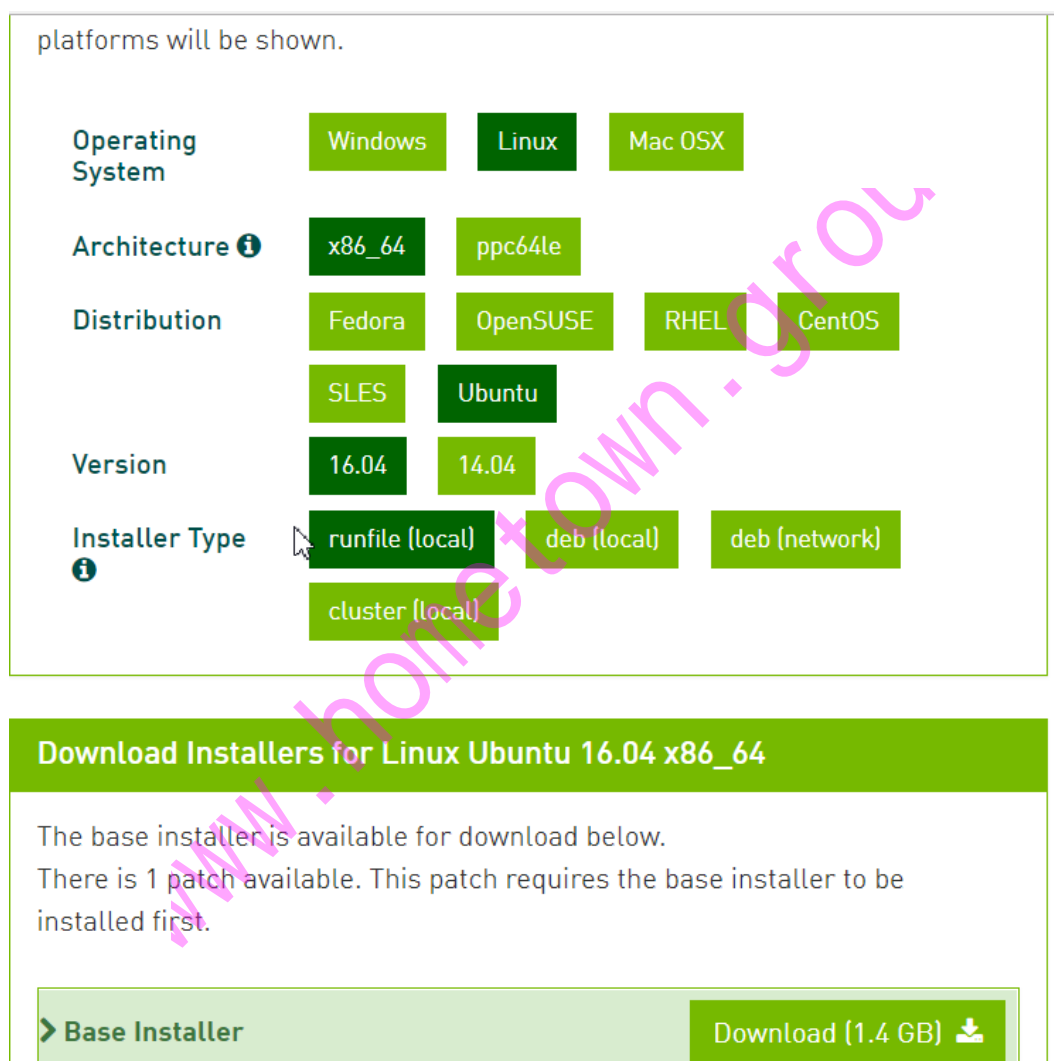


图 9

2.2、安装 cuda8.0

设置可执行权限

```
sudo chmod a+x cuda_8.0.61_375.26_linux.run
```

执行安装:

在刚开始安装的时候会出现一个声明, 本人直接回车到 100%看完声明, 进行安装。

Do you accept the previously read EULA?

accept/decline/quit: **accept**

cd 0_Simple/clock 执行简单的测试 demo

```
zhuang@zhuang:~/NVIDIA_CUDA-8.0_Samples/0_Simple$ cd clock
zhuang@zhuang:~/NVIDIA_CUDA-8.0_Samples/0_Simple/clock$ ls
clock  clock.cu  clock.o  Makefile  NsightEclipse.xml  readme.txt
zhuang@zhuang:~/NVIDIA_CUDA-8.0_Samples/0_Simple/clock$ ./clock
CUDA Clock sample
GPU Device 0: "GeForce GTX 1060 3GB" with compute capability 6.1

Average clocks/block = 3469.937500
zhuang@zhuang:~/NVIDIA_CUDA-8.0_Samples/0_Simple/clock$ █
```

如果需要升级卸载:

```
cd /usr/local/cuda/bin
sudo ./uninstall_cuda_8.0.pl
```

若安装的是 deb 文件使用这种方式卸载:

```
sudo apt-get autoremove --purge cuda
```

三、安装 cudnn

3.1 下载链接:<https://developer.nvidia.com/rdp/cudnn-archive>, 下载 cudnn5.1 需要注册一个 NVIDIA 账号. 下载设置见图 10, 会得到 cudnn-8.0-linux-x64-v5.1.solitairetheme8 这个文件.

注意: cudnn 版本不限, 但是要和前面的 cuda 版本配套



图 10

solitairetheme8 这个格式的文件太奇怪了处理一下:

```
sudo cp cudnn-8.0-linux-x64-v5.1.solitairetheme8 cudnn-8.0-linux-x64-v5.1.tgz
```

解压:

```
sudo tar -zxvf ./cudnn-8.0-linux-x64-v5.1.tgz
```

复制头文件:

```
cd cuda/include
sudo cp cudnn.h /usr/local/cuda/include
```

复制库文件:

```
cd ../lib64/
sudo cp lib* /usr/local/cuda/lib64/
```

删除系统原有的动态库文件:

```
cd /usr/local/cuda/lib64/  
sudo rm -rf libcudnn.so libcudnn.so.5  
sudo ln -s libcudnn.so.5.1.10 libcudnn.so.5  
sudo ln -s libcudnn.so.5 libcudnn.so  
sudo ldconfig
```

四、安装 tensorflow

下载对应版本的 tensorflow : <https://pypi.org/project/tensorflow/#files>

查看 TensorFlow 版本:

```
sudo pip show tensorflow
```

安装和卸载命令:

```
pip install tensorflow_gpu==1.1
```

```
sudo pip uninstall tensorflow-gpu
```

但是运行却包了这个错误:

```
tensorflow: name tensorflow not defined  
>>> import tensorflow as tf  
>>> hello = tf.constant("Hello, TensorFlow")  
>>> sess = tf.Session()  
2018-09-18 15:16:09.811689: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.  
2018-09-18 15:16:09.811754: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.  
2018-09-18 15:16:09.811778: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.  
2018-09-18 15:16:09.811803: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.  
2018-09-18 15:16:09.811828: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.  
2018-09-18 15:16:10.596510: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:901] successful NUMA node read from SysFS had negative value (-1), but the file must be at least one NUMA node, so returning NUMA node zero  
2018-09-18 15:16:10.596732: I tensorflow/core/common_runtime/gpu/gpu_device.cc:887] Found device 0 with properties:  
name: GeForce GTX 1060 3GB  
major: 6 minor: 1 memoryClockRate (GHz) 1.7335  
pciBusID 0000:01:00.0  
Total memory: 2.94GiB  
Free memory: 2.88GiB  
2018-09-18 15:16:10.596747: I tensorflow/core/common_runtime/gpu/gpu_device.cc:908] DMA: 0  
2018-09-18 15:16:10.596752: I tensorflow/core/common_runtime/gpu/gpu_device.cc:918] 0: Y  
2018-09-18 15:16:10.596761: I tensorflow/core/common_runtime/gpu/gpu_device.cc:977] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1060 3GB, pci bus id: 0000:01:00.0)  
>>> print(sess.run(hello))  
Hello, TensorFlow  
>>>
```

有强迫症患者的看到这堆警告肯定很不爽，后面上网又找了一遍，我们还是用源码编译的方式安装 TensorFlow 吧。哎，路还很长，搭建深度学习的开发环境容易啊。

五、安装 keras

```
sudo pip install -U --pre pip setuptools wheel
```

```
sudo pip install -U --pre numpy scipy matplotlib scikit-learn scikit-image
```

```
pip install keras
```

原文链接 : <https://www.cnblogs.com/zzb-Dream-90Time/p/9663542.html>