

ubuntu16.04 + opencv3 + Nvidia 显卡驱动 + cuda8.0 + cudnn5.1 + caffe 安装 + python2

安装opencv3

直接使用pip install opencv-python 即可安装成功，如果需要一些拓展功能，则需要使用pip安装contrib版本。

安装Nvidia 显卡驱动

1. 去Nvidia官网选择自己电脑显卡对应的型号的驱动，下载.run文件

2. 卸载原先所有的驱动：

```
sudo apt-get remove --purge nvidia*  
sudo chmod a+x *.run
```

3.禁用集成的nouveau驱动。

将驱动添加到黑名单blacklist.conf中，但是由于该文件的属性不允许修改。所以需要先修改文件属性。

```
查看属性  
sudo ls -lh /etc/modprobe.d/blacklist.conf  
修改属性  
sudo chmod 666 /etc/modprobe.d/blacklist.conf  
  
sudo vim /etc/modprobe.d/blacklist.conf
```

然后在文件后添加以下几行：

```
blacklist vga16fb  
blacklist nouveau  
blacklist riva  
blacklist rivatv  
blacklist nvidiafb
```

4.开始安装

先按Ctrl+Alt+F1到控制台，关闭当前图形环境

```
sudo service lightdm stop
```

安装驱动

```
sudo chmod a+x NVIDIA-LINUX...run
sudo ./NVIDIA....run -no-x-check -no-nouveau-check -no-opengl-files
```

最后重启图形环境

```
sudo service lightdm start
```

5. 查看显卡驱动版本

可以通过以下命令确定驱动是否正确安装

```
cat /proc/driver/nvidia/version
```

或者

```
nvidia-smi
```

如果有一个表个显示自己的显卡型号，说明安装成功

```
Thu Jun  7 16:56:13 2018
```

NVIDIA-SMI 390.67 Driver Version: 390.67									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute	M.		
0	GeForce GTX 1050	Off	00000000:01:00.0	Off			N/A		
N/A	46C	P0	N/A / N/A	0MiB / 4040MiB	0%		Default		

Processes:					GPU Memory
GPU	PID	Type	Process name		Usage
No running processes found					

ps: 如果使用nvidia-setting 时 本人出现X-server错误的情况，但是按照网上很多教程没有能够弄好。但是对于后面用GPU进行训练没有什么影响。

安装cuda8.0+cudnn5.1

cuda是nvidia的编程语言平台，想使用GPU就必须使用cuda。

1. 下载cuda8.0源文件

[cuda官网](#)

选择runfile(local)

2. 安装

```
sudo sh cuda_8.0.....run
```

在安装的过程中，由于之前已经安装了nvidia的驱动，所以在是否安装驱动这一选项选择n其他直接默认护在哦和y就行。

3.修改配置文件

修改启动bash的配置文件，将一些路经直接预加载。

```
sudo vim /etc/profile
```

在文件末尾添加以下语句

```
export PATH=/usr/local/cuda-8.0/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64:$LD_LIBRARY_PATH
```

4. 重启电脑

```
sudo shutdown -r now
```

5.测试cuda的Samples

```
cd /usr/local/cuda-8.0/samples/1_Uutilities/deviceQuery
sudo make
./deviceQuery
```

```
Device 0: "GeForce GTX 1050"
  CUDA Driver Version / Runtime Version      9.1 / 8.0
  CUDA Capability Major/Minor version number: 6.1
  Total amount of global memory:              4040 MBytes (4236312576 bytes)
  ( 5) Multiprocessors, (128) CUDA Cores/MP: 640 CUDA Cores
  GPU Max Clock rate:                        1493 MHz (1.49 GHz)
  Memory Clock rate:                          3504 Mhz
  Memory Bus Width:                           128-bit
  L2 Cache Size:                             524288 bytes
  Maximum Texture Dimension Size (x,y,z)     1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers 1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
  Total amount of constant memory:            65536 bytes
  Total amount of shared memory per block:    49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                  32
  Maximum number of threads per multiprocessor: 2048
  Maximum number of threads per block:        1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size (x,y,z):  (2147483647, 65535, 65535)
  Maximum memory pitch:                       2147483647 bytes
  Texture alignment:                           512 bytes
  Concurrent copy and kernel execution:       Yes with 2 copy engine(s)
  Run time limit on kernels:                   No
  Integrated GPU sharing Host Memory:          No
  Support host page-locked memory mapping:     Yes
  Alignment requirement for Surfaces:          Yes
  Device has ECC support:                      Disabled
  Device supports Unified Addressing (UVA):    Yes
  Device PCI Domain ID / Bus ID / location ID: 0 / 1 / 0
  Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 9.1, CUDA Runtime Version = 8.0, NumDevs = 1, Device 0 = GeForce GTX 1050
Result = PASS
wangdh15@wangdh15-computer:/usr/local/cuda-8.0/samples/1_Uutilities/deviceQuery$ |
```

6.安装cudnn

首先去官网下载cudnn，下载时需要注册帐号。然后选择对应的cuda版本进行下载即可。

下载完cudnn之后，进行解压，然后将内部文件复制到cuda安装路径即可。

```
sudo cp cudnn.h /usr/local/cuda/include/    #复制头文件
```

然后在cd到lib64目录下进行复制链接：

```
sudo cp lib* /usr/local/cuda/lib64/    #复制动态链接库
cd /usr/local/cuda/lib64/
sudo rm -rf libcudnn.so libcudnn.so.5    #删除原有动态文件
sudo ln -s libcudnn.so.5.0.5 libcudnn.so.5    #生成软链接
sudo ln -s libcudnn.so.5 libcudnn.so    #生成软链接
```

这样就完成了cudnn的安装

安装caffe

1. 首先安装各种依赖包

```
sudo apt-get update
sudo apt-get install -y build-essential cmake git pkg-config
sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnappy-dev libhdf5-serial-dev
sudo apt-get install -y libatlas-base-dev
sudo apt-get install -y --no-install-recommends libboost-all-dev
sudo apt-get install -y libgflags-dev libgoogle-glog-dev liblmdb-dev
sudo apt-get install -y python-pip
sudo apt-get install -y python-dev
sudo apt-get install -y python-numpy python-scipy
```

2. 然后从github上下载caffe源码

cd 到要安装的位置，然后：

```
git clone https://github.com/BVLC/caffe.git //从github上git caffe
cd caffe //打开到刚刚git下来的caffe
sudo cp Makefile.config.example Makefile.config //将Makefile.config.example的内容复制到Makefile.config
//因为make指令只能make Makefile.config文件，而Makefile.config.example是caffe给出的makefile
sudo gedit Makefile.config //打开Makefile.config文件
```

3. 对Makefile.config进行修改。

- 如果使用cudnn 则去掉cudnn前面的注释
- 如果使用的是opencv3 则去掉opencv3前面的注释

- //重要的一项 将# Whatever else you find you need goes here.下面的 INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include LIBRARY_DIRS := \$(PYTHON_LIB) /usr/local/lib /usr/lib
修改为： INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
LIBRARY_DIRS := \$(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial //这是因为ubuntu16.04的文件包含位置发生了变化，尤其是需要用到的hdf5的位置，所以需要更改这一路径
- 可能需要修改numpy的安装路径，因为给的文档好像没有加local。
- 去掉CUDA_ARCH的前两行，不然在后面编译的时候会有警告。
- 其余默认即可

4. 执行下面的语句来进行编译

```
make all -j8
make test -j8
make runtest -j8
make pycaffe -j8
```

5.可能遇到的问题

error while loading shared libraries: libcudart.so.8.0: cannot open shared object file: No such file or directory

解决方法：

```
sudo cp /usr/local/cuda-8.0/lib64/libcudart.so.8.0 /usr/local/lib/libcudart.so.8.0
sudo cp /usr/local/cuda-8.0/lib64/libcublas.so.8.0 /usr/local/lib/libcublas.so.8.0
sudo cp /usr/local/cuda-8.0/lib64/libcurand.so.8.0 /usr/local/lib/libcurand.so.8.0
ps. ldconfig命令是一个动态链接库管理命令，是为了让动态链接库为系统共享
```

6. 将caffe加到Python导入时的路径中

```
sudo vim ~/.bashrc
在文档末尾添加
export PYTHONPATH=/home/wangdh15/caffe_cuda_cudnn/python:$PYTHONPATH

重新加载配置文件
source ~/.bashrc
```

7. 测试是否成功

```
python
import caffe
```

```
In [1]: import caffe

In [2]: caffe
caffe      caffe.cloc

In [2]: caffe
Out[2]: <module 'caffe' from '/home/wangdh15/caffe_cuda_cudnn/python/caffe/__init__.pyc'>

In [3]: |
```

8. 安装完成

参考文献

[Ubuntu16.04+cuda8.0+caffe安装教程](#)

[ubuntu16.04安装cuda8.0](#)