Ubuntu16.04 +cuda8.0+cudnn+caffe+theano+tensorflow配置明细

本文为原创作品，未经本人同意，禁止转载，禁止用于商业用途！本人对博客使用拥有最终解释权

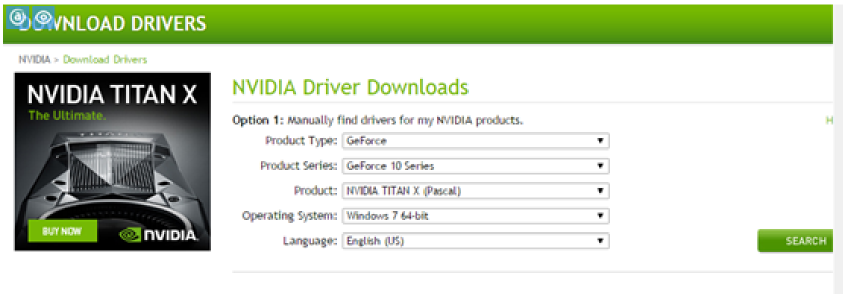
欢迎关注我的博客：http://blog.csdn.net/hit2015spring和http://www.cnblogs.com/xujianqing

本文主要是介绍在ubuntu16.04下，怎么配置当下流行的深度学习框架，cuda8.0+cudnn+caffe+theano+tensorflow

安装英伟达显卡驱动

首先去官网上查看适合你GPU的驱动

（http://www.nvidia.com/Download/index.aspx?lang=en-us）



NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

Product Type: GeForce
Product Series: GeForce 10 Series
Product: NVIDIA TITAN X (Pascal)
Operating System: Windows 7 64-bit
Language: English (US)

sudo add-apt-repository ppa:graphics-drivers/ppa

sudo apt-get update

sudo apt-get install nvidia-375（375是你查到的版本号）

sudo apt-get install mesa-common-dev

sudo apt-get install freeglut3-dev

执行完上述后，重启（reboot）。

重启后输入

nvidia-smi

如果出现了你的GPU列表，则说明驱动安装成功了。另外也可以通过，或者输入

nvidia-settings

出现

昵称：晨兑追风
园龄：2年8个月
粉丝：14
关注：1
+加关注

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

学习笔记(1)

随笔分类

C++
Jetson TX1
MATLAB
opencv(2)
机器学习(1)
图像处理(5)
学习笔记(10)
硬件 DSP6748
支持向量机

随笔档案

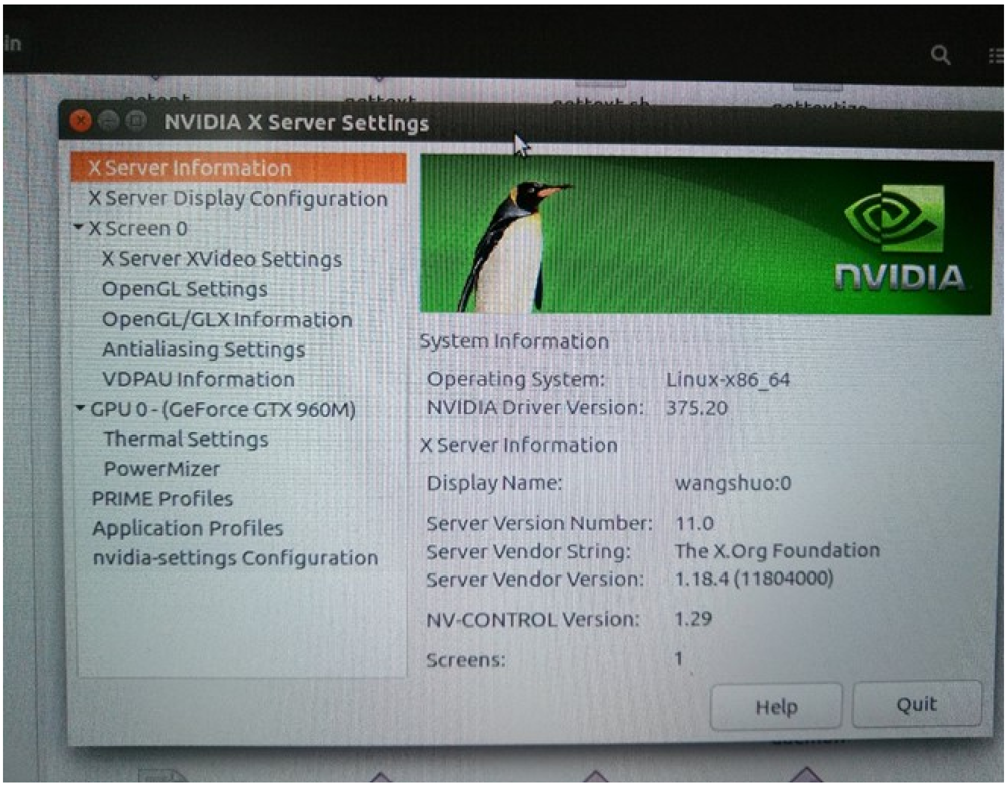
2017年6月 (8)
2017年5月 (2)
2016年12月 (6)
2016年10月 (1)
2016年9月 (2)
2016年8月 (1)
2016年5月 (2)
2016年4月 (2)
2016年1月 (4)
2015年11月 (1)

文章分类

opencv
图像处理
学习笔记

最新评论

1. Re:SIFT四部曲之——高斯滤波
受益匪浅，谢谢
--EllieToT
2. Re:SIFT四部曲之——高斯滤波
@tuji_sjp参照我的csdn博客：...



System Information

Operating System: Linux-x86_64
NVIDIA Driver Version: 375.20

X Server Information

Display Name: wangshuo:0

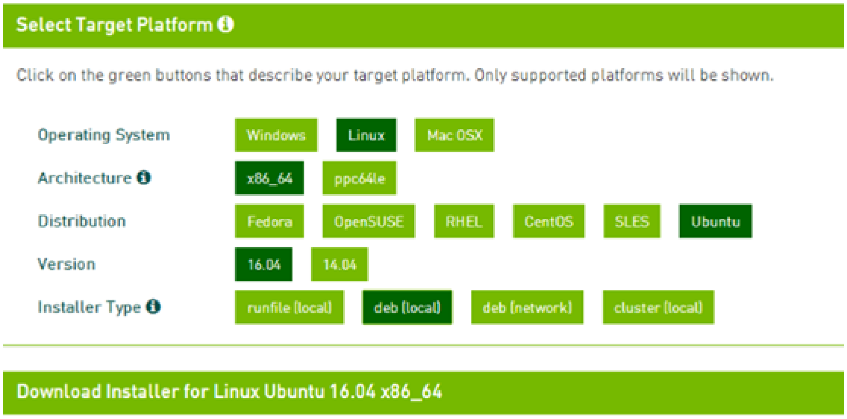
Server Version Number: 11.0
Server Vendor String: The X.Org Foundation
Server Vendor Version: 1.18.4 (11804000)

NV-CONTROL Version: 1.29

Screens: 1

1. 配置cuda

https://developer.nvidia.com/cuda-downloads



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows, Linux, Mac OSX
Architecture	x86_64, ppc64le
Distribution	Fedora, OpenSUSE, RHEL, CentOS, SLES, Ubuntu
Version	16.04, 14.04
Installer Type	runfile (local), deb (local), deb (network), cluster (local)

Download Installer for Linux Ubuntu 16.04 x86_64

在cuda所在目录打开terminal依次输入以下指令：

```
sudo dpkg -i cuda-repo-ubuntu1604-8-0-rc_8.0.27-1_amd64.deb
sudo apt-get update
sudo apt-get install cuda
```

ubuntu的gcc编译器是5.4.0，然而cuda8.0不支持5.0以上的编译器，因此需要降级，把编译器版本降到4.9：

在terminal中执行：

```
sudo apt-get install gcc -4.9 gcc-5 g++-4.9 g++-5
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 20
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-5 10
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.9 20
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-5 10
sudo update-alternatives --install /usr/bin/cc cc /usr/bin/gcc 30
sudo update-alternatives --set cc /usr/bin/gcc
sudo update-alternatives --install /usr/bin/c++ c++ /usr/bin/g++ 30
sudo update-alternatives --set c++ /usr/bin/g++
```

配置cuda8.0之后主要加上的一个环境变量声明，在文件~/.bashrc之后加上

--晨凫追风

3. Re:SIFT四部曲之——高斯滤波

你好，我刚刚入门学习图像识别。今天看了你的这篇文章，想问一下：您解决了SIFT算法四步中的哪一步呢？其他剩下的几步呢？应该怎样继续？
烦请您回答一下，不胜感激！

--tuji_sjp

4. Re:Ubuntu16.04 +cuda8.0+cudnn+caffe+theano+tensorflow配置明细

楼主您好，我在安装好tensorflow的时候，尝试导入 import tensorflowImport Error: libcudnn.so.5: cannot open shared object ......

--给点阳光yh

5. Re:打怪升级必备书单

做笔记，同感

--繁华0美丽

阅读排行榜

1. Ubuntu16.04 +cuda8.0+cudnn+caffe+theano+tensorflow配置明细(14628)
2. Opencv学习笔记——release和debug两个模式的运行问题(6674)
3. KKT条件和拉格朗日乘子法详解(4201)
4. 利用opencv自带源码，调试摄像头做人脸检测(4074)
5. Caffe学习笔记4图像特征进行可视化(3590)

评论排行榜

1. 打怪升级必备书单(10)
2. Ubuntu16.04 +cuda8.0+cudnn+caffe+theano+tensorflow配置明细(5)
3. SAR图像处理 MSTAR数据库利用问题(4)
4. SIFT四部曲之——高斯滤波(3)
5. Opencv学习笔记——release和debug两个模式的运行问题(2)

推荐排行榜

1. 打怪升级必备书单(3)
2. Opencv学习笔记——release和debug两个模式的运行问题(1)
3. SIFT四部曲之——高斯滤波(1)
4. Opencv模块功能介绍(1)
5. KKT条件和拉格朗日乘子法详解(1)

gedit ~/.bashrc

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}}

然后设置环境变量和动态链接库，在命令行输入

sudo gedit /etc/profile

在打开的文件里面加上（注意等号两边不能有空格）

export PATH=/usr/local/cuda/bin:$PATH

保存之后，创建链接文件

sudo gedit /etc/ld.so.conf.d/cuda.conf

在打开的文件中添加如下语句：

/usr/local/cuda/lib64

保存退出执行命令行：

sudo ldconfig

使链接立即生效。

2、测试cuda的Samples

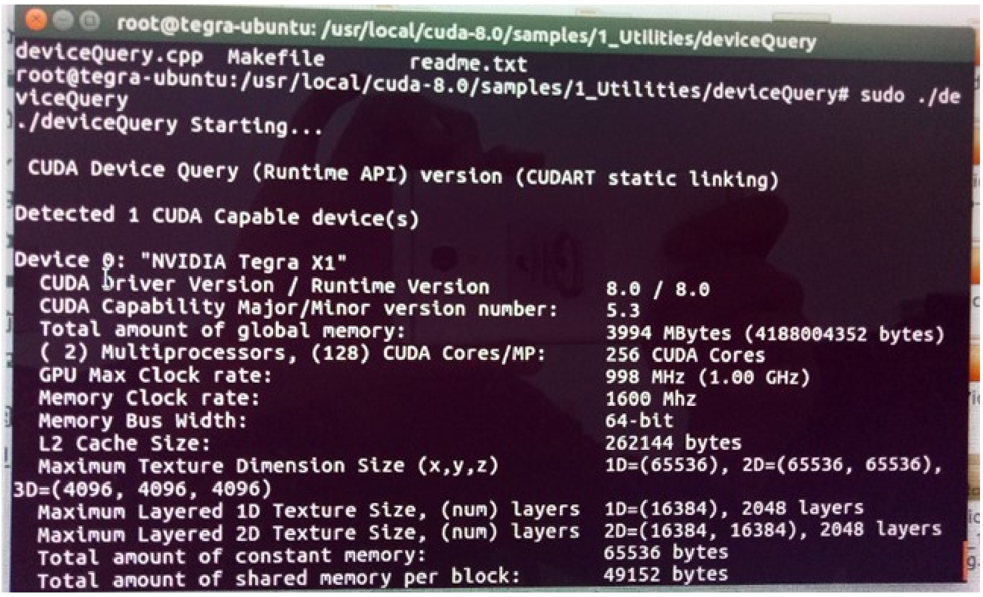
命令行输入（注意cuda-8.0是要相对应自己的cuda版本）

cd /usr/local/cuda-8.0/samples/1_Utilities/deviceQuery

make

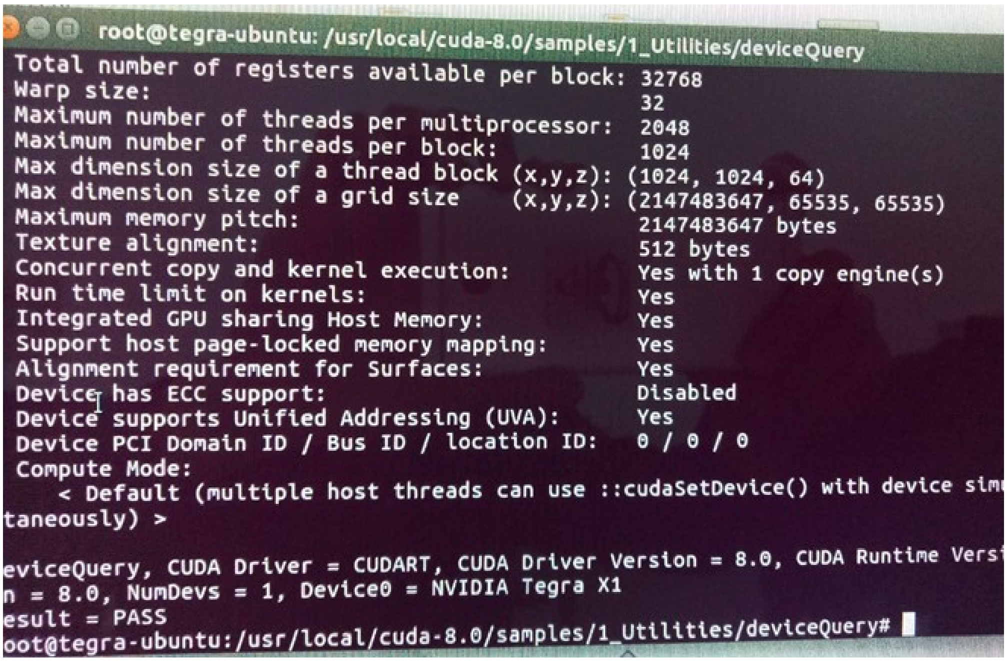
sudo ./deviceQuery

返回GPU的信息则表示配置成功



root@tegra-ubuntu: /usr/local/cuda-8.0/samples/1_Utilities/deviceQuery

```
deviceQuery.cpp  Makefile        readme.txt
root@tegra-ubuntu:/usr/local/cuda-8.0/samples/1_Utilities/deviceQuery# sudo ./de
viceQuery
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "NVIDIA Tegra X1"
  CUDA Driver Version / Runtime Version          8.0 / 8.0
  CUDA Capability Major/Minor version number:    5.3
  Total amount of global memory:                 3994 MBytes (4188004352 bytes)
  ( 2) Multiprocessors, (128) CUDA Cores/MP:     256 CUDA Cores
  GPU Max Clock rate:                            998 MHz (1.00 GHz)
  Memory Clock rate:                             1600 Mhz
  Memory Bus Width:                              64-bit
  L2 Cache Size:                                 262144 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65536),
3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
```

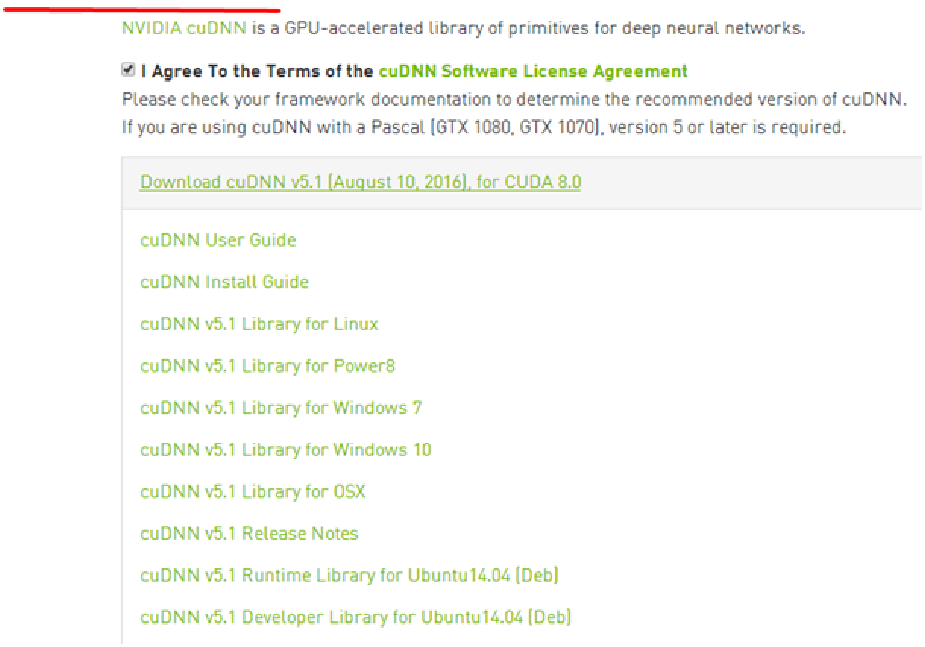

root@tegra-ubuntu: /usr/local/cuda-8.0/samples/1_Utilities/deviceQuery

```
Total number of registers available per block: 32768
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 1 copy engine(s)
Run time limit on kernels:                     Yes
Integrated GPU sharing Host Memory:            Yes
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 / 0 / 0
Compute Mode:
   < Default (multiple host threads can use ::cudaSetDevice() with device simu
taneously) >

eviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 8.0, CUDA Runtime Vers
n = 8.0, NumDevs = 1, Device0 = NVIDIA Tegra X1
esult = PASS
oot@tegra-ubuntu:/usr/local/cuda-8.0/samples/1_Utilities/deviceQuery#
```

3、使用cudnn

上官网下载对应的cudnn

https://developer.nvidia.com/cudnn



NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☑ I Agree To the Terms of the cuDNN Software License Agreement

Please check your framework documentation to determine the recommended version of cuDNN.
If you are using cuDNN with a Pascal (GTX 1080, GTX 1070), version 5 or later is required.

Download cuDNN v5.1 (August 10, 2016), for CUDA 8.0

cuDNN User Guide

cuDNN Install Guide

cuDNN v5.1 Library for Linux

cuDNN v5.1 Library for Power8

cuDNN v5.1 Library for Windows 7

cuDNN v5.1 Library for Windows 10

cuDNN v5.1 Library for OSX

cuDNN v5.1 Release Notes

cuDNN v5.1 Runtime Library for Ubuntu14.04 (Deb)

cuDNN v5.1 Developer Library for Ubuntu14.04 (Deb)

下载完cudnn后，命令行输入文件所在的文件夹 (ubuntu为本机用户名)

cd home/ubuntu/Downloads/

tar zxvf cudnn-8.0-linux-x64-v5.1.tgz #解压文件

cd进入cudnn5.1解压之后的include目录，在命令行进行如下操作：

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

再cd进入lib64目录下的动态文件进行复制和链接：（5.1.5为对应版本具体可修改）

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.5.1.5 libcudnn.so.5 #生成软衔接

sudo ln -s libcudnn.so.5 libcudnn.so #生成软链接

4、安装opencv3.1.0

从官网上下载opencv3.1.0

http://opencv.org/downloads.html

并将其解压到你要安装的位置，（下载的位置还是在home/ubuntu、Downloads文件夹下）

首先安装Ubuntu系统需要的依赖项，虽然我也不知道有些依赖项是干啥的，但是只管装就行，也不会占据很多空间的。

sudo apt-get install --assume-yes libopencv-dev build-essential cmake git libgtk2.0-dev pkg-config python-dev python-numpy libdc1394-22 libdc1394-22-dev libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libxine2-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-dev libtbb-dev libqt4-dev libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils unzip

然后安装OpenCV需要的一些依赖项，一些文件编码解码之类的东东。

sudo apt-get install build-essential cmake git

sudo apt-get install ffmpeg libopencv-dev libgtk-3-dev python-numpy python3-numpy libdc1394-22 libdc1394-22-dev libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libxine2-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libv4l-dev libtbb-dev qtbase5-dev libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils unzip

在终端中cd到opencv文件夹下（解压的那个文件夹），然后

mkdir build #新建一个build文件夹，编译的工程都在这个文件夹里

cd build/

cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=ON -D WITH_V4L=ON -D WITH_QT=ON -D WITH_OPENGL=ON -DCUDA_NVCC_FLAGS="-D_FORCE_INLINES" ..（后面两点不要忘记）

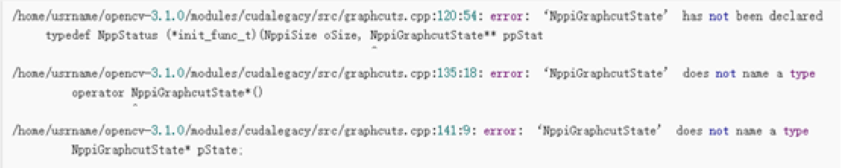
cmake成功后，会出现如下结果，提示配置和生成成功：

-- Configuring done

-- Generating done

-- Build files have been written to: /home/ise/software/opencv-3.1.0/build

由于CUDA 8.0不支持OpenCV的 GraphCut 算法，可能出现以下错误：



```
/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:120:54: error:  ‘NppiGraphcutState’  has not been declared
     typedef NppStatus (*init_func_t)(NppiSize oSize, NppiGraphcutState** ppStat
                                                      ^
/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:135:18: error:  ‘NppiGraphcutState’  does not name a type
         operator NppiGraphcutState*()
                  ^
/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:141:9: error:  ‘NppiGraphcutState’  does not name a type
         NppiGraphcutState* pState;
```

/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:120:54: error: 'NppiGraphcutState' has not been declared

typedef NppStatus (*init_func_t)(NppiSize oSize, NppiGraphcutState** ppStat

^

/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:135:18: error: 'NppiGraphcutState' does not name a type

operator NppiGraphcutState*()

^

/home/usrname/opencv-3.1.0/modules/cudalegacy/src/graphcuts.cpp:141:9: error: 'NppiGraphcutState' does not name a type

NppiGraphcutState* pState;

....

进入opencv-3.1.0/modules/cudalegacy/src/目录，修改graphcuts.cpp文件，将：

#include "precomp.hpp"

#if !defined (HAVE_CUDA) || defined (CUDA_DISABLER)

改为

#include "precomp.hpp"

#if !defined (HAVE_CUDA) || defined (CUDA_DISABLER) || (CUDART_VERSION >= 8000)

然后make编译就可以了

make -j8

上面是将opencv编译成功，但是并没有安装到我们的系统中，有很多的设置都没有写入到系统中，因此还要进行install。

sudo make install

sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'

sudo ldconfig

重启系统，重启系统后cd到build文件夹下：

sudo apt-get install checkinstall

sudo checkinstall

然后按照提示安装就可以了。

使用checkinstall的目的是为了更好的管理我安装的opencv，因为opencv的安装很麻烦，卸载更麻烦，其安装的时候修改了一大堆的文件，当我想使用别的版本的opencv时，将当前版本的opencv卸载就是一件头疼的事情，因此需要使用checkinstall来管理我的安装。

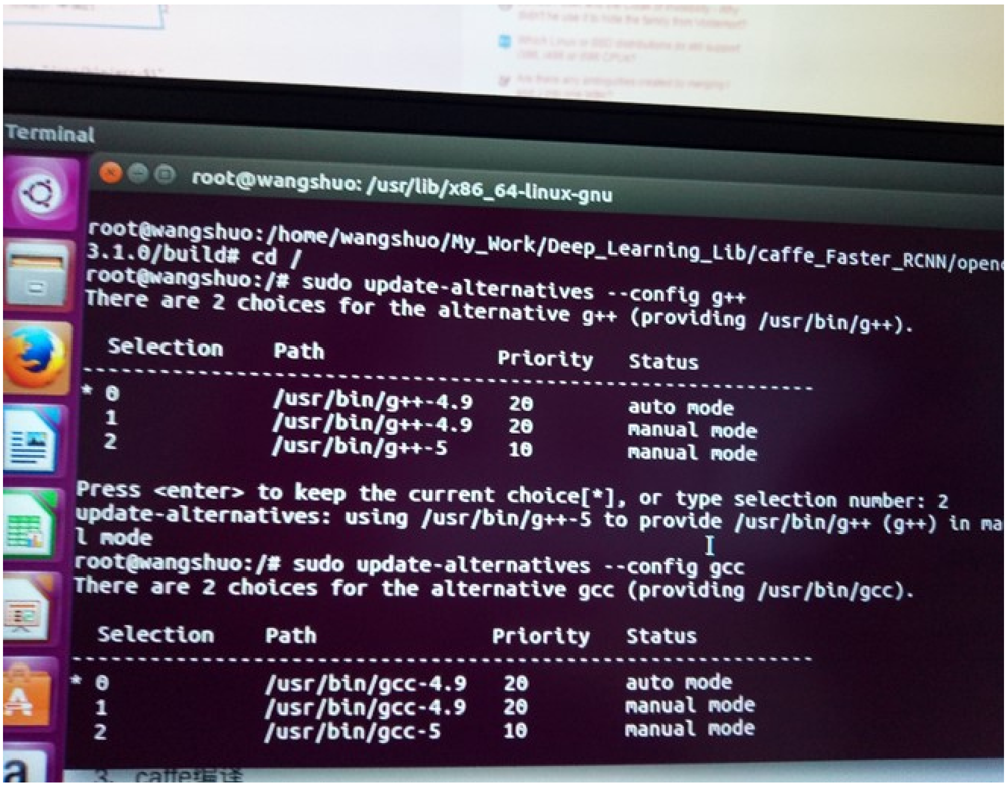
执行了checkinstall后，会在build文件下生成一个以backup开头的.tgz的备份文件和一个以build开头的.deb安装文件，当你想卸载当前的opencv时，直接执行dpkg -r build即可。

5、配置caffe环境

切换编译器

选择g++ 5.0以上的对应编号

sudo update-alternatives --config g++



```
root@wangshuo: /usr/lib/x86_64-linux-gnu
root@wangshuo:/home/wangshuo/My_Work/Deep_Learning_Lib/caffe_Faster_RCNN/openc
3.1.0/build# cd /
root@wangshuo:/# sudo update-alternatives --config g++
There are 2 choices for the alternative g++ (providing /usr/bin/g++).

  Selection    Path              Priority   Status
------------------------------------------------------------
* 0            /usr/bin/g++-4.9   20        auto mode
  1            /usr/bin/g++-4.9   20        manual mode
  2            /usr/bin/g++-5     10        manual mode

Press <enter> to keep the current choice[*], or type selection number: 2
update-alternatives: using /usr/bin/g++-5 to provide /usr/bin/g++ (g++) in ma
l mode
root@wangshuo:/# sudo update-alternatives --config gcc
There are 2 choices for the alternative gcc (providing /usr/bin/gcc).

  Selection    Path              Priority   Status
------------------------------------------------------------
* 0            /usr/bin/gcc-4.9   20        auto mode
  1            /usr/bin/gcc-4.9   20        manual mode
  2            /usr/bin/gcc-5     10        manual mode
```

sudo update-alternatives --config gcc

安装依赖库

sudo add-apt-repository universe

sudo apt-get update -y

sudo apt-get install cmake -y

# General Dependencies

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev \

libhdf5-serial-dev protobuf-compiler -y

sudo apt-get install --no-install-recommends libboost-all-dev -y

```
# BLAS
sudo apt-get install libatlas-base-dev -y
# Remaining Dependencies
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev -y
sudo apt-get install python-dev python-numpy –y
sudo apt-get install -y python-pip
sudo apt-get install -y python-dev
sudo apt-get install -y python-numpy python-scipy
```

编译 Caffe，cd到要安装caffe的位置

```
git clone https://github.com/BVLC/caffe.git
cd caffe
cp Makefile.config.example Makefile.config
```

修改Makefile.config：

```
gedit Makefile.config
```

对打开的文件编辑

```
# cuDNN acceleration switch (uncomment to build with cuDNN).
USE_CUDNN := 1

# Uncomment if you're using OpenCV 3 如果用的是opencv3版本
OPENCV_VERSION := 3

# Uncomment to support layers written in Python (will link against Python libs)
WITH_PYTHON_LAYER := 1
```

在问件里面添加文本由于hdf5库目录更改，所以需要单独添加：

```
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial/
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/aarch64-linux-gnu/hdf5/serial/
```

打开makefile文件

```
gedit Makefile
```

将

NVCCFLAGS +=-ccbin=$(CXX) - Xcompiler - fPIC$(COMMON_FLAGS)

替换

NVCCFLAGS += -D_FORCE_INLINES -ccbin=$(CXX) - Xcompiler - fPIC$(COMMON_FLAGS)

编辑/usr/local/cuda/include/host_config.h，将其中的第115行注释掉：

```
sudo gedit /usr/local/cuda/include/host_config.h
```

将

```
#error-- unsupported GNU version! gcc versions later than 4.9 are not supported!
```

改为

```
//#error-- unsupported GNU version! gcc versions later than 4.9 are not supported!
```

之后编辑即可

```
make -j4 all
make -j4 runtest
```

为了更好地使用pycaffe ，建议安装：

```
sudo apt-get install python-numpy python-setuptools python-pip cython python-skimage python-protobuf
make pycaffe
cd python
```

```
python
import caffe #测试安装成功
```

到这里Caffe开发环境就配置好了！

可以测试一下，输出AlexNet的时间测试结果：

```
cd ~/caffe
./build/tools/caffe time --gpu 0 --model ./models/bvlc_alexnet/deploy.prototxt
```

6、theano安装

1、直接输入命令：

```
sudo pip install theano
```

2、配置参数文件：.theanorc

```
sudo gedit ~/.theanorc
```

对打开的文件进行编辑

```
[global]
floatX=float32
device=gpu
base_compiledir=~/external/.theano/
allow_gc=False
warn_float64=warn
[mode]=FAST_RUN

[nvcc]
fastmath=True

[cuda]
root=/usr/local/cuda
```

3、运行测试例子：

```
sudo Vim test.py
```

```
from theano import function, config, shared, sandbox
import theano.tensor as T
import numpy
import time

vlen = 10 * 30 * 768 # 10 x #cores x # threads per core
iters = 1000

rng = numpy.random.RandomState(22)
x = shared(numpy.asarray(rng.rand(vlen), config.floatX))
f = function([], T.exp(x))
print(f.maker.fgraph.toposort())
t0 = time.time()
for i in range(iters):
r = f()
t1 = time.time()
print("Looping %d times took %f seconds" % (iters, t1 - t0))
print("Result is %s" % (r,))
if numpy.any([isinstance(x.op, T.Elemwise) for x in f.maker.fgraph.toposort()]):
```

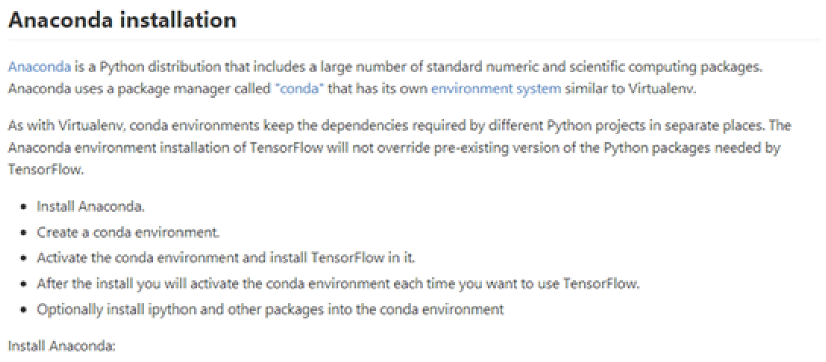
```
print('Used the cpu')

else:

print('Used the gpu')
```

可以看到结果：

```
/usr/bin/python2.7 /home/hjimce/PycharmProjects/untitled/.idea/temp.py

Using gpu device 0: GeForce GTX 960 (CNMeM is disabled, cuDNN not available)

[GpuElemwise{exp,no_inplace}(<CudaNdarrayType(float32, vector)>),
HostFromGpu(GpuElemwise{exp,no_inplace}.0)]

Looping 1000 times took 0.302778 seconds

Result is [ 1.23178029 1.61879349 1.52278066 ..., 2.20771813 2.29967761

1.62323296]

Used the gpu
```

说明安装成功

7、tensorflow 安装



Anaconda installation

Anaconda is a Python distribution that includes a large number of standard numeric and scientific computing packages. Anaconda uses a package manager called "conda" that has its own environment system similar to Virtualenv.

As with Virtualenv, conda environments keep the dependencies required by different Python projects in separate places. The Anaconda environment installation of TensorFlow will not override pre-existing version of the Python packages needed by TensorFlow.

- Install Anaconda.
- Create a conda environment.
- Activate the conda environment and install TensorFlow in it.
- After the install you will activate the conda environment each time you want to use TensorFlow.
- Optionally install ipython and other packages into the conda environment

Install Anaconda:

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g3doc/get_started/os_setup.md

先安装anaconda

https://repo.continuum.io/archive/Anaconda2-4.2.0-Windows-x86_64.exe

上面的地址下载 该包默认在downloads里面

```
cd /home/username/Downloads

sudo bash Anaconda2-4.2.0-Linux-x86_64.sh
```

配置环境变量

```
gedit /etc/profile
```

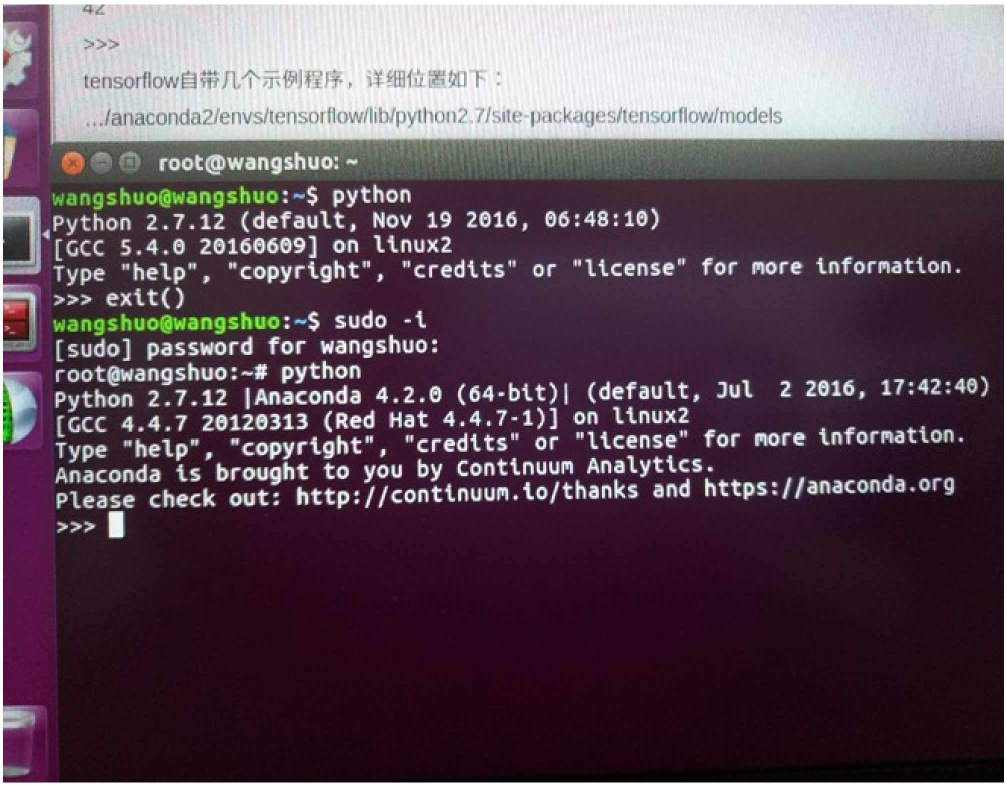
末尾添上,我是一路yes下来，所以安在了root下，你可以自己选路径，这时候的环境变量要改

```
export PATH=/root/anaconda2/bin:$PATH
```

重启

打开终端

```
python
```



```
42
>>>
```

tensorflow自带几个示例程序，详细位置如下：

.../anaconda2/envs/tensorflow/lib/python2.7/site-packages/tensorflow/models

```
root@wangshuo: ~
wangshuo@wangshuo:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
wangshuo@wangshuo:~$ sudo -i
[sudo] password for wangshuo:
root@wangshuo:~# python
Python 2.7.12 |Anaconda 4.2.0 (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> 
```

安装成功

2、创建conda环境 名字叫tensorflow

conda create -n tensorflow python=2.7

source activate tensorflow #使能该环境

#下面这句话只能下载给CPU用的tensorflow

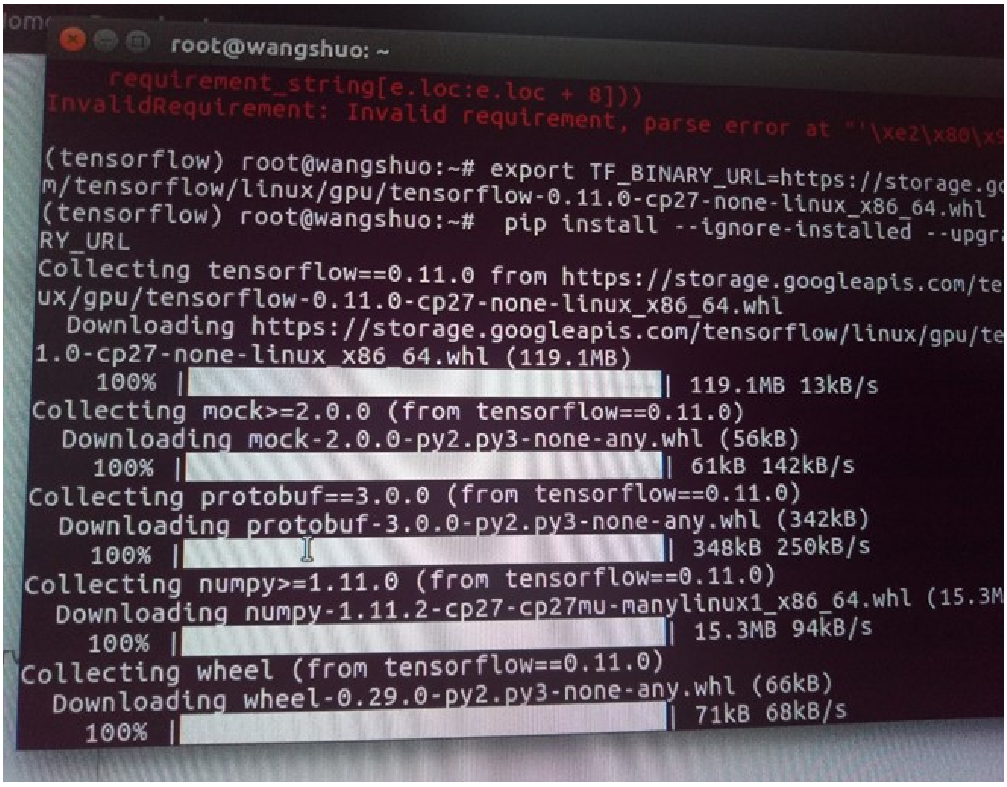
conda install -c conda-forge tensorflow

利用pip来下载给GPU用的tensorflow

export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.11.0-cp27-none-linux_x86_64.whl

下载安装

pip install --ignore-installed --upgrade $TF_BINARY_URL



```
root@wangshuo: ~
    requirement_string[e.loc:e.loc + 8]))
InvalidRequirement: Invalid requirement, parse error at "'\xe2\x80\x

(tensorflow) root@wangshuo:~# export TF_BINARY_URL=https://storage.go
m/tensorflow/linux/gpu/tensorflow-0.11.0-cp27-none-linux_x86_64.whl
(tensorflow) root@wangshuo:~#  pip install --ignore-installed --upgr
RY_URL
Collecting tensorflow==0.11.0 from https://storage.googleapis.com/te
ux/gpu/tensorflow-0.11.0-cp27-none-linux_x86_64.whl
  Downloading https://storage.googleapis.com/tensorflow/linux/gpu/te
1.0-cp27-none-linux_x86_64.whl (119.1MB)
    100% |████████████████████████████████| 119.1MB 13kB/s
Collecting mock>=2.0.0 (from tensorflow==0.11.0)
  Downloading mock-2.0.0-py2.py3-none-any.whl (56kB)
    100% |████████████████████████████████| 61kB 142kB/s
Collecting protobuf==3.0.0 (from tensorflow==0.11.0)
  Downloading protobuf-3.0.0-py2.py3-none-any.whl (342kB)
    100% |████████████████████████████████| 348kB 250kB/s
Collecting numpy>=1.11.0 (from tensorflow==0.11.0)
  Downloading numpy-1.11.2-cp27-cp27mu-manylinux1_x86_64.whl (15.3M
    100% |████████████████████████████████| 15.3MB 94kB/s
Collecting wheel (from tensorflow==0.11.0)
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
    100% |████████████████████████████████| 71kB 68kB/s
```

安装IPython

```
conda install ipython
```

关掉该环境

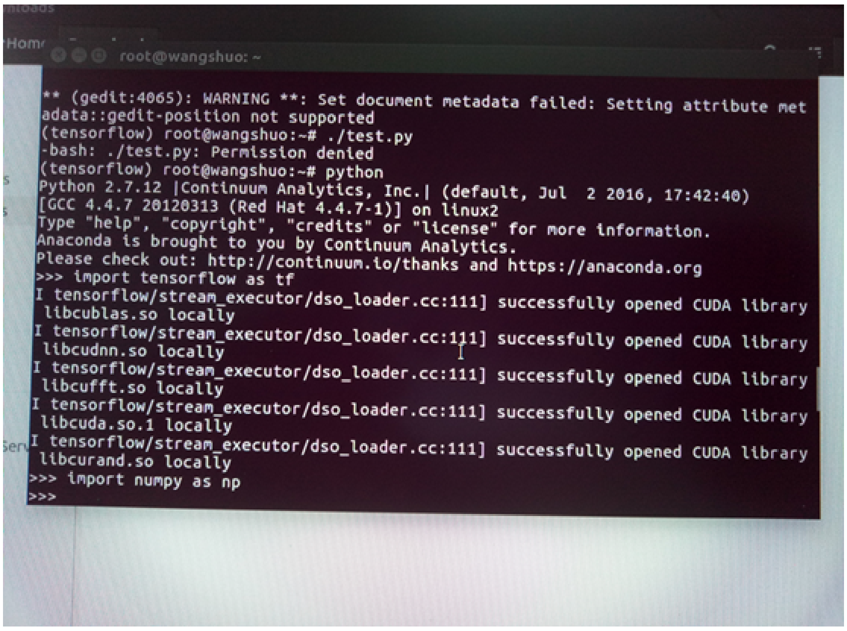
```
source deactivate
```

测试安装是否正确

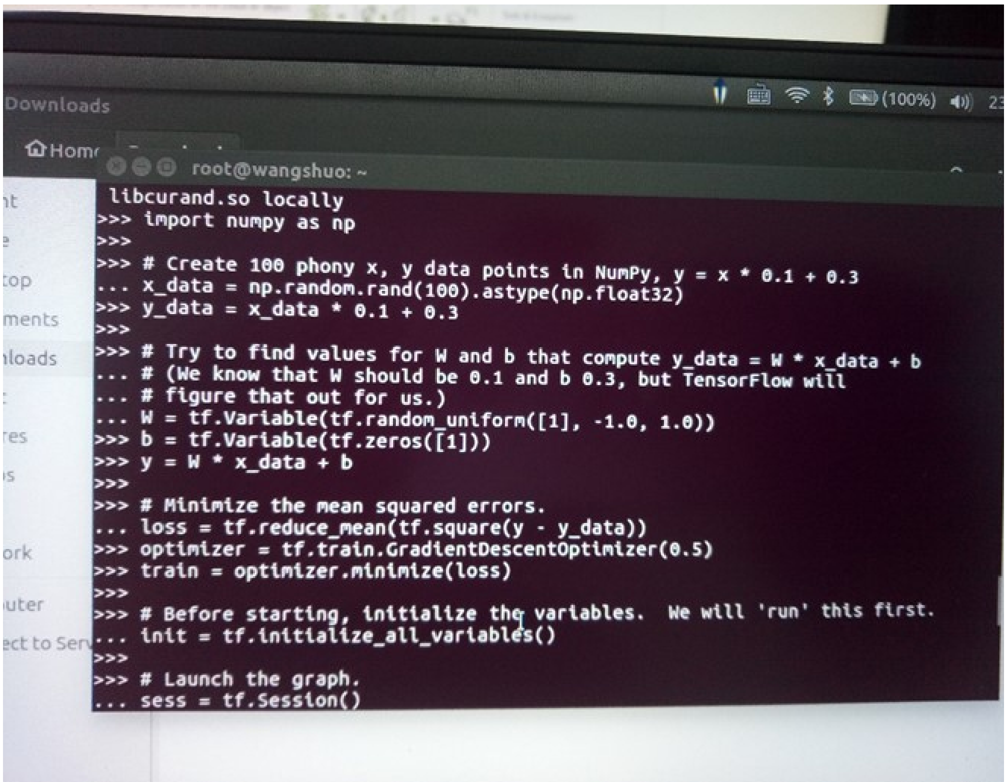
```
source activate tensorflow
python
```

输入

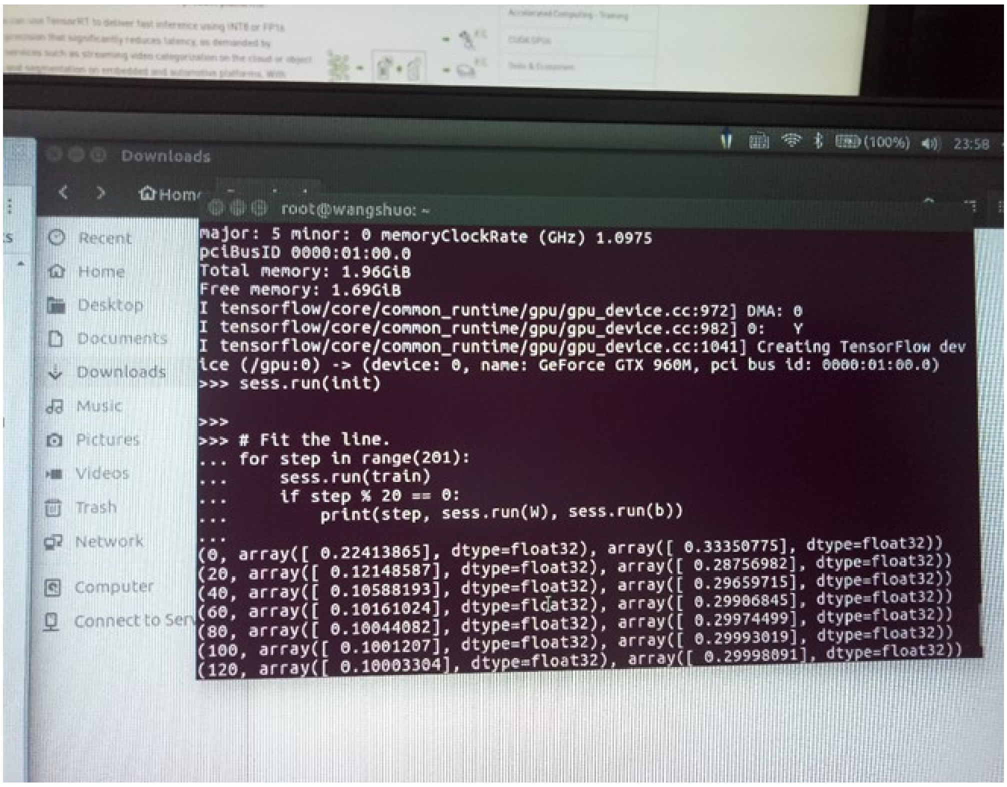
```
import tensorflow as tf
import numpy as np

# Create 100 phony x, y data points in NumPy, y = x * 0.1 + 0.3
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data * 0.1 + 0.3

# Try to find values for W and b that compute y_data = W * x_data + b
# (We know that W should be 0.1 and b 0.3, but TensorFlow will
# figure that out for us.)
W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b

# Minimize the mean squared errors.
loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

# Before starting, initialize the variables. We will 'run' this first.
init = tf.initialize_all_variables()

# Launch the graph.
sess = tf.Session()
sess.run(init)

# Fit the line.
for step in range(201):
sess.run(train)
if step % 20 == 0:
print(step, sess.run(W), sess.run(b))

# Learns best fit is W: [0.1], b: [0.3]
```



```
** (gedit:4065): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
(tensorflow) root@wangshuo:~# ./test.py
-bash: ./test.py: Permission denied
(tensorflow) root@wangshuo:~# python
Python 2.7.12 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> import tensorflow as tf
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library
 libcublas.so locally
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library
 libcudnn.so locally
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library
 libcufft.so locally
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library
 libcuda.so.1 locally
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library
 libcurand.so locally
>>> import numpy as np
>>>
```



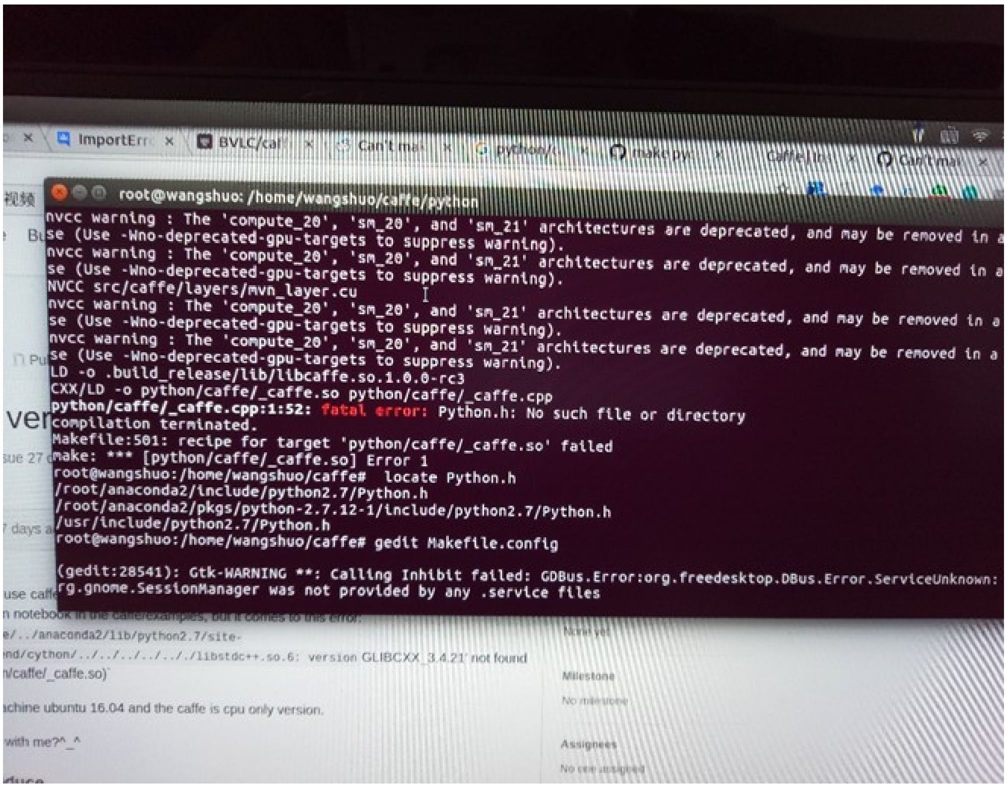
```
 libcurand.so locally
>>> import numpy as np
>>>
>>> # Create 100 phony x, y data points in NumPy, y = x * 0.1 + 0.3
... x_data = np.random.rand(100).astype(np.float32)
>>> y_data = x_data * 0.1 + 0.3
>>>
>>> # Try to find values for W and b that compute y_data = W * x_data + b
... # (We know that W should be 0.1 and b 0.3, but TensorFlow will
... # figure that out for us.)
... W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
>>> b = tf.Variable(tf.zeros([1]))
>>> y = W * x_data + b
>>>
>>> # Minimize the mean squared errors.
... loss = tf.reduce_mean(tf.square(y - y_data))
>>> optimizer = tf.train.GradientDescentOptimizer(0.5)
>>> train = optimizer.minimize(loss)
>>>
>>> # Before starting, initialize the variables.  We will 'run' this first.
... init = tf.initialize_all_variables()
>>>
>>> # Launch the graph.
... sess = tf.Session()
```



```
root@wangshuo: ~
major: 5 minor: 0 memoryClockRate (GHz) 1.0975
pciBusID 0000:01:00.0
Total memory: 1.96GiB
Free memory: 1.69GiB
I tensorflow/core/common_runtime/gpu/gpu_device.cc:972] DMA: 0
I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] 0:   Y
I tensorflow/core/common_runtime/gpu/gpu_device.cc:1041] Creating TensorFlow dev
ice (/gpu:0) -> (device: 0, name: GeForce GTX 960M, pci bus id: 0000:01:00.0)
>>> sess.run(init)

>>>
>>> # Fit the line.
... for step in range(201):
...     sess.run(train)
...     if step % 20 == 0:
...         print(step, sess.run(W), sess.run(b))
...
(0, array([ 0.22413865], dtype=float32), array([ 0.33350775], dtype=float32))
(20, array([ 0.12148587], dtype=float32), array([ 0.28756982], dtype=float32))
(40, array([ 0.10588193], dtype=float32), array([ 0.29659715], dtype=float32))
(60, array([ 0.10161024], dtype=float32), array([ 0.29906845], dtype=float32))
(80, array([ 0.10044082], dtype=float32), array([ 0.29974499], dtype=float32))
(100, array([ 0.1001207], dtype=float32), array([ 0.29993019], dtype=float32))
(120, array([ 0.10003304], dtype=float32), array([ 0.29998091], dtype=float32))
```

OK

8、Caffe配置错误



```
root@wangshuo: /home/wangshuo/caffe/python
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21' architectures are deprecated, and may be removed in a
se (Use -Wno-deprecated-gpu-targets to suppress warning).
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21' architectures are deprecated, and may be removed in a
se (Use -Wno-deprecated-gpu-targets to suppress warning).
NVCC src/caffe/layers/mvn_layer.cu
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21' architectures are deprecated, and may be removed in a
se (Use -Wno-deprecated-gpu-targets to suppress warning).
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21' architectures are deprecated, and may be removed in a
se (Use -Wno-deprecated-gpu-targets to suppress warning).
LD -o .build_release/lib/libcaffe.so.1.0.0-rc3
CXX/LD -o python/caffe/_caffe.so python/caffe/_caffe.cpp
python/caffe/_caffe.cpp:1:52: fatal error: Python.h: No such file or directory
compilation terminated.
Makefile:501: recipe for target 'python/caffe/_caffe.so' failed
make: *** [python/caffe/_caffe.so] Error 1
root@wangshuo:/home/wangshuo/caffe#  locate Python.h
/root/anaconda2/include/python2.7/Python.h
/root/anaconda2/pkgs/python-2.7.12-1/include/python2.7/Python.h
/usr/include/python2.7/Python.h
root@wangshuo:/home/wangshuo/caffe# gedit Makefile.config

(gedit:28541): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown:
rg.gnome.SessionManager was not provided by any .service files
```

ImportErr
BVLC/caf
Can't ma
python/
make py
Caffe.li
Can't mak
视频
use caffe
n notebook in the caffe/examples, but it comes to this error.
e/../anaconda2/lib/python2.7/site-
nd/cython/../../../../../../libstdc++.so.6: version GLIBCXX_3.4.21' not found
n/caffe/_caffe.so)
chine ubuntu 16.04 and the caffe is cpu only version.
with me?^_^
None yet
Milestone
No milestone
Assignees
No one assigned

问题：找不到Python.h

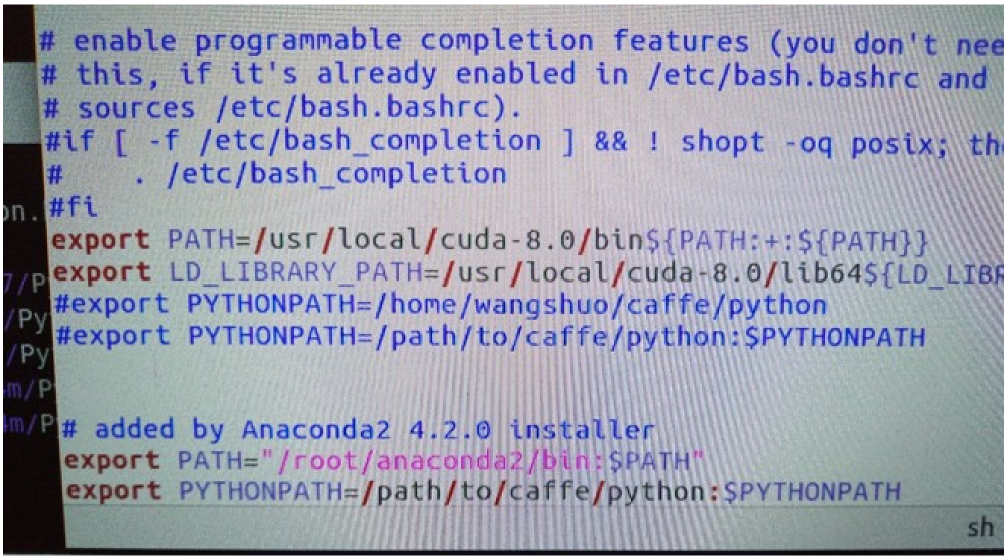
解决：给anaconda添加环境变量

gedit ~/.banshrc

添加

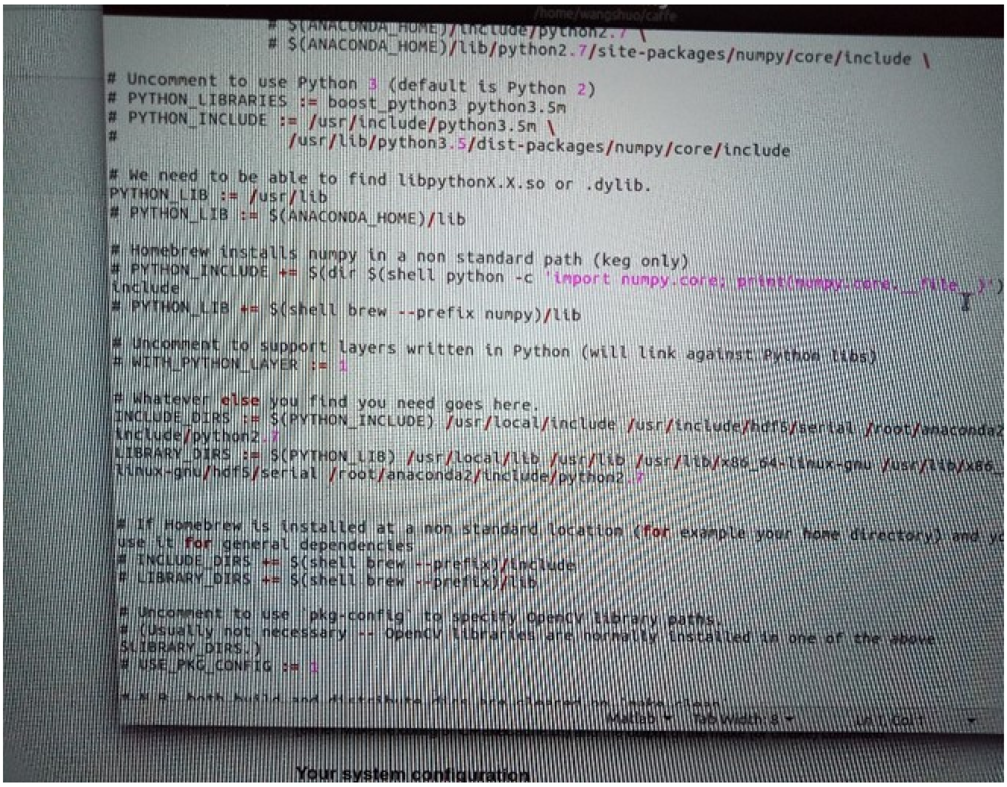
export PATH=/root/anaconda2/bin:$PATH

export PYTHONPATH=/path/to/caffe/python:$PATH



```
# enable programmable completion features (you don't nee
# this, if it's already enabled in /etc/bash.bashrc and
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; th
#    . /etc/bash_completion
#fi
export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64${LD_LIBR
#export PYTHONPATH=/home/wangshuo/caffe/python
#export PYTHONPATH=/path/to/caffe/python:$PYTHONPATH


# added by Anaconda2 4.2.0 installer
export PATH="/root/anaconda2/bin:$PATH"
export PYTHONPATH=/path/to/caffe/python:$PYTHONPATH
```

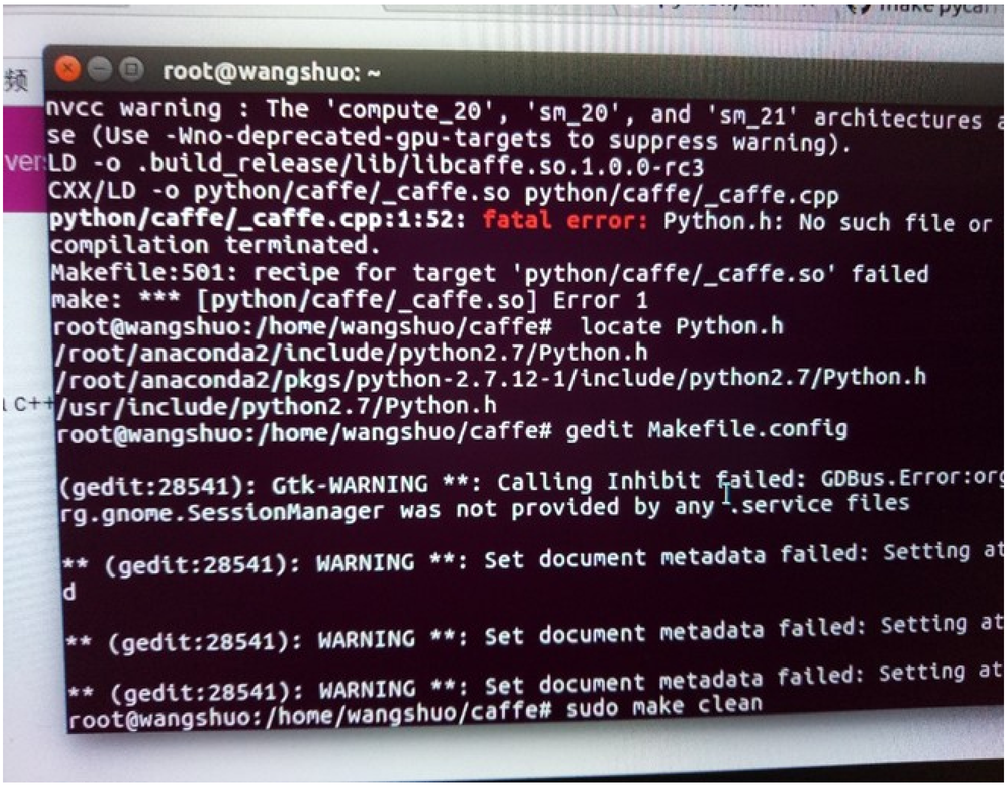
修改Makefile.config



```
        # $(ANACONDA_HOME)/include/python2.7 \
        # $(ANACONDA_HOME)/lib/python2.7/site-packages/numpy/core/include \

# Uncomment to use Python 3 (default is Python 2)
# PYTHON_LIBRARIES := boost_python3 python3.5m
# PYTHON_INCLUDE := /usr/include/python3.5m \
#                 /usr/lib/python3.5/dist-packages/numpy/core/include

# We need to be able to find libpythonX.X.so or .dylib.
PYTHON_LIB := /usr/lib
# PYTHON_LIB := $(ANACONDA_HOME)/lib

# Homebrew installs numpy in a non standard path (keg only)
# PYTHON_INCLUDE += $(dir $(shell python -c 'import numpy.core; print(numpy.core.__file__)'))/
include
# PYTHON_LIB += $(shell brew --prefix numpy)/lib

# Uncomment to support layers written in Python (will link against Python libs)
# WITH_PYTHON_LAYER := 1

# Whatever else you find you need goes here.
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial /root/anaconda2/
include/python2.7
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_
linux-gnu/hdf5/serial /root/anaconda2/include/python2.7


# If Homebrew is installed at a non standard location (for example your home directory) and yo
use it for general dependencies
# INCLUDE_DIRS += $(shell brew --prefix)/include
# LIBRARY_DIRS += $(shell brew --prefix)/lib

# Uncomment to use `pkg-config` to specify OpenCV library paths.
# (Usually not necessary -- OpenCV libraries are normally installed in one of the above
$LIBRARY_DIRS.)
# USE_PKG_CONFIG := 1

# N.B. both build and distribute dirs are cleared on `make clean`
```

在终端输入

locate Python.h

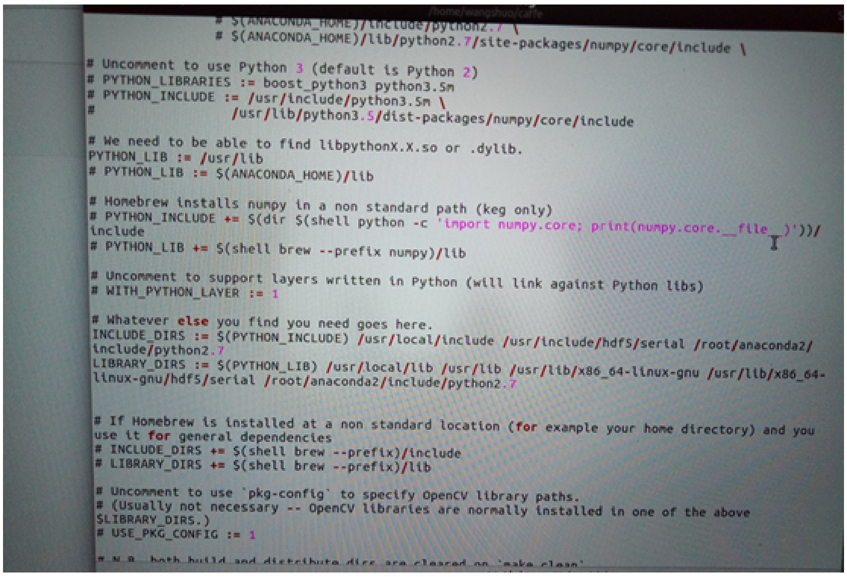


```
root@wangshuo: ~
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21' architectures a
se (Use -Wno-deprecated-gpu-targets to suppress warning).
LD -o .build_release/lib/libcaffe.so.1.0.0-rc3
CXX/LD -o python/caffe/_caffe.so python/caffe/_caffe.cpp
python/caffe/_caffe.cpp:1:52: fatal error: Python.h: No such file or
compilation terminated.
Makefile:501: recipe for target 'python/caffe/_caffe.so' failed
make: *** [python/caffe/_caffe.so] Error 1
root@wangshuo:/home/wangshuo/caffe#  locate Python.h
/root/anaconda2/include/python2.7/Python.h
/root/anaconda2/pkgs/python-2.7.12-1/include/python2.7/Python.h
/usr/include/python2.7/Python.h
root@wangshuo:/home/wangshuo/caffe# gedit Makefile.config

(gedit:28541): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org
rg.gnome.SessionManager was not provided by any .service files

** (gedit:28541): WARNING **: Set document metadata failed: Setting at
d

** (gedit:28541): WARNING **: Set document metadata failed: Setting at

** (gedit:28541): WARNING **: Set document metadata failed: Setting at
root@wangshuo:/home/wangshuo/caffe# sudo make clean
```

gedit Makefile.config

在INCLUDE_DIRS 和LIBRARY_DIRS后面添上

/root/anaconda2/include/python2.7



```
/home/wangshuo/caffe
                # $(ANACONDA_HOME)/include/python2.7 \
                # $(ANACONDA_HOME)/lib/python2.7/site-packages/numpy/core/include \

# Uncomment to use Python 3 (default is Python 2)
# PYTHON_LIBRARIES := boost_python3 python3.5m
# PYTHON_INCLUDE := /usr/include/python3.5m \
#                 /usr/lib/python3.5/dist-packages/numpy/core/include

# We need to be able to find libpythonX.X.so or .dylib.
PYTHON_LIB := /usr/lib
# PYTHON_LIB := $(ANACONDA_HOME)/lib

# Homebrew installs numpy in a non standard path (keg only)
# PYTHON_INCLUDE += $(dir $(shell python -c 'import numpy.core; print(numpy.core.__file__)'))/
include
# PYTHON_LIB += $(shell brew --prefix numpy)/lib

# Uncomment to support layers written in Python (will link against Python libs)
# WITH_PYTHON_LAYER := 1

# Whatever else you find you need goes here.
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial /root/anaconda2/
include/python2.7
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-
linux-gnu/hdf5/serial /root/anaconda2/include/python2.7


# If Homebrew is installed at a non standard location (for example your home directory) and you
use it for general dependencies
# INCLUDE_DIRS += $(shell brew --prefix)/include
# LIBRARY_DIRS += $(shell brew --prefix)/lib

# Uncomment to use `pkg-config` to specify OpenCV library paths.
# (Usually not necessary -- OpenCV libraries are normally installed in one of the above
$LIBRARY_DIRS.)
# USE_PKG_CONFIG := 1

# N.B. both build and distribute dirs are cleared on `make clean`
```

启用

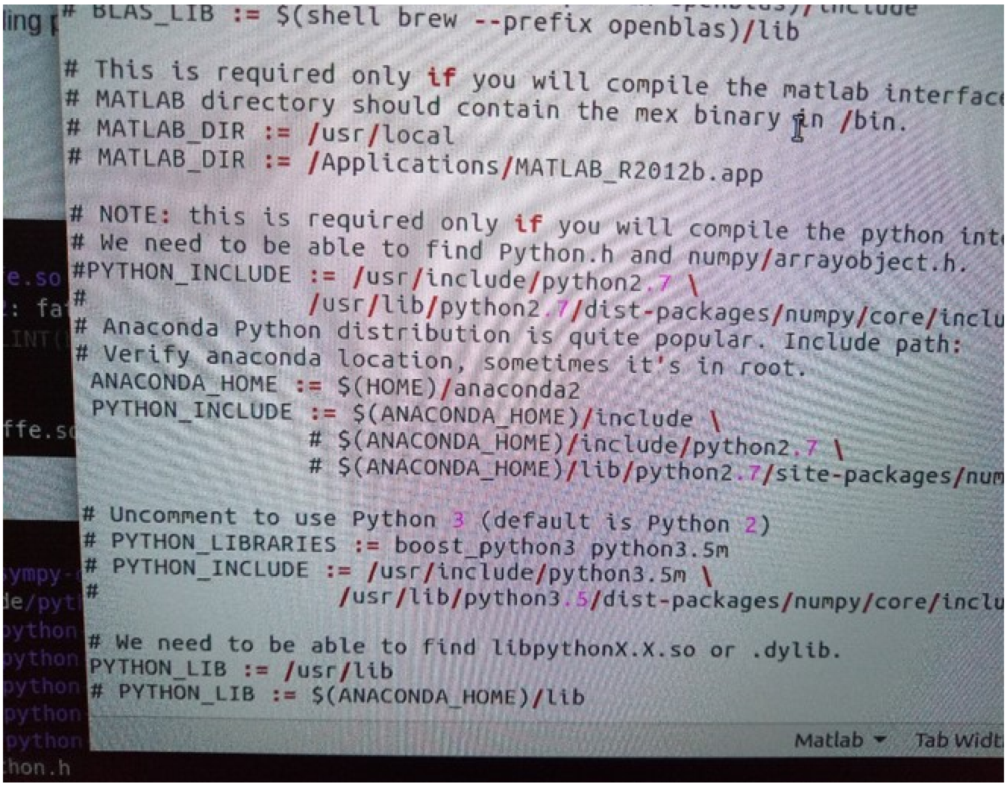
ANACONDA_HOME := $(HOME)/anaconda2

PYTHON_ INCLUDE =$(ANACONDA_HOME)/include\

,把前面的#去掉,那三行都去掉#，并在注释上面,

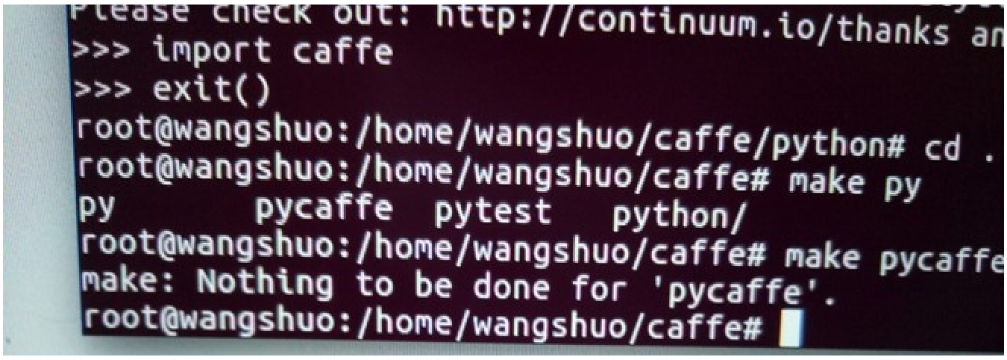
注释这两句PYTHON_INCLUDE := /usr/include/python2.7\

/usr/lib/python2.7…………..



```
# BLAS_LIB := $(shell brew --prefix openblas)/lib

# This is required only if you will compile the matlab interfac
# MATLAB directory should contain the mex binary in /bin.
# MATLAB_DIR := /usr/local
# MATLAB_DIR := /Applications/MATLAB_R2012b.app

# NOTE: this is required only if you will compile the python inte
# We need to be able to find Python.h and numpy/arrayobject.h.
#PYTHON_INCLUDE := /usr/include/python2.7 \
#               /usr/lib/python2.7/dist-packages/numpy/core/inclu
# Anaconda Python distribution is quite popular. Include path:
# Verify anaconda location, sometimes it's in root.
 ANACONDA_HOME := $(HOME)/anaconda2
 PYTHON_INCLUDE := $(ANACONDA_HOME)/include \
                # $(ANACONDA_HOME)/include/python2.7 \
                # $(ANACONDA_HOME)/lib/python2.7/site-packages/num

# Uncomment to use Python 3 (default is Python 2)
# PYTHON_LIBRARIES := boost_python3 python3.5m
# PYTHON_INCLUDE := /usr/include/python3.5m \
#                 /usr/lib/python3.5/dist-packages/numpy/core/inclu

# We need to be able to find libpythonX.X.so or .dylib.
PYTHON_LIB := /usr/lib
# PYTHON_LIB := $(ANACONDA_HOME)/lib
```

如果编译的时候发现有错，回来改完之后又得重新编译一遍pycaffe，于是出现如下错误



```
Please check out: http://continuum.io/thanks an
>>> import caffe
>>> exit()
root@wangshuo:/home/wangshuo/caffe/python# cd ..
root@wangshuo:/home/wangshuo/caffe# make py
py          pycaffe   pytest    python/
root@wangshuo:/home/wangshuo/caffe# make pycaffe
make: Nothing to be done for 'pycaffe'.
root@wangshuo:/home/wangshuo/caffe#
```

make: Nothing to be done for 'pycaffe'

则在终端输入：

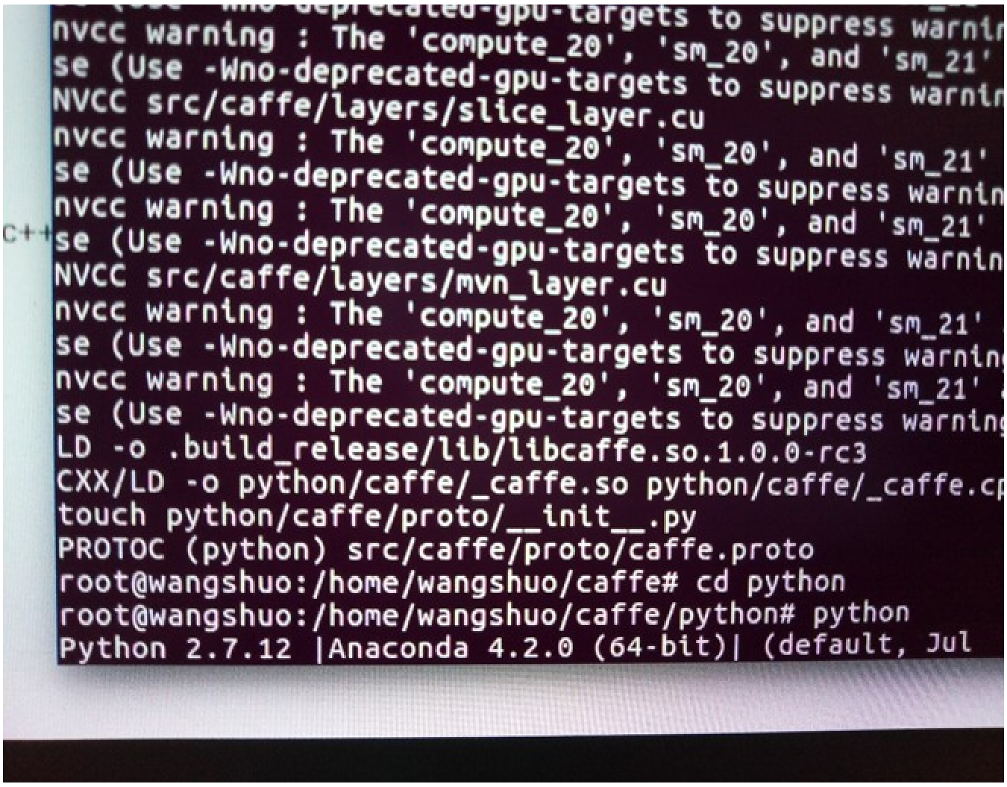
sudo make clean

修改完后再

sudo make pycaffe

这里要从make –j8 all那一步开始编译

编译完后，显示



```
se (Use -Wno-deprecated-gpu-targets to suppress warnin
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21'
se (Use -Wno-deprecated-gpu-targets to suppress warnin
NVCC src/caffe/layers/slice_layer.cu
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21'
se (Use -Wno-deprecated-gpu-targets to suppress warnin
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21'
se (Use -Wno-deprecated-gpu-targets to suppress warnin
NVCC src/caffe/layers/mvn_layer.cu
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21'
se (Use -Wno-deprecated-gpu-targets to suppress warnin
nvcc warning : The 'compute_20', 'sm_20', and 'sm_21'
se (Use -Wno-deprecated-gpu-targets to suppress warnin
LD -o .build_release/lib/libcaffe.so.1.0.0-rc3
CXX/LD -o python/caffe/_caffe.so python/caffe/_caffe.cp
touch python/caffe/proto/__init__.py
PROTOC (python) src/caffe/proto/caffe.proto
root@wangshuo:/home/wangshuo/caffe# cd python
root@wangshuo:/home/wangshuo/caffe/python# python
Python 2.7.12 |Anaconda 4.2.0 (64-bit)| (default, Jul
```

然后 cd python进入该目录

python

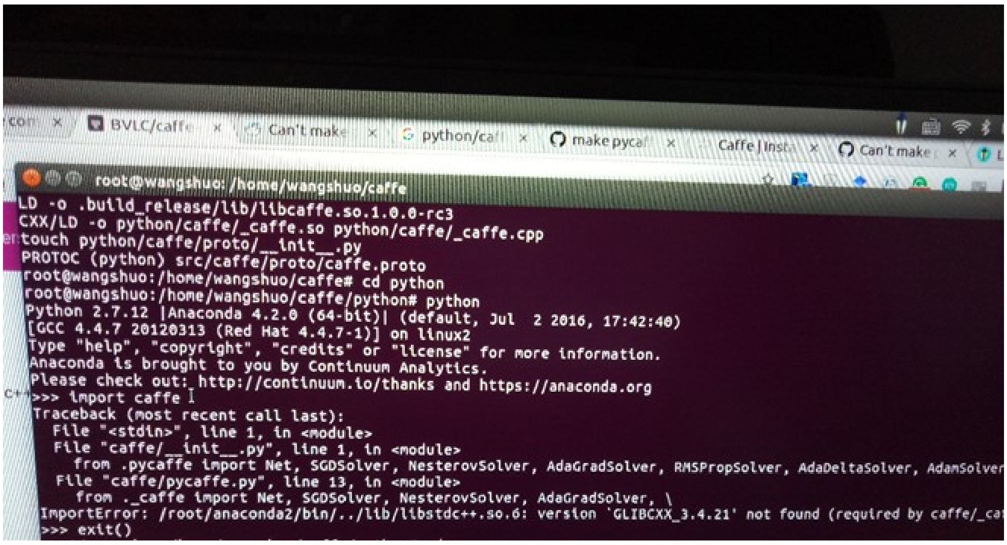
import caffe

若此时提示错误:

Traceback (most recent call last)

File

ImportError: /home/../anaconda2/lib/python2.7/site-packages/zmq/backend/cython/../../../../.././libstdc++.so.6: versionGLIBCXX_3.4.21' not found



```
LD -o .build_release/lib/libcaffe.so.1.0.0-rc3
CXX/LD -o python/caffe/_caffe.so python/caffe/_caffe.cpp
touch python/caffe/proto/__init__.py
PROTOC (python) src/caffe/proto/caffe.proto
root@wangshuo:/home/wangshuo/caffe# cd python
root@wangshuo:/home/wangshuo/caffe/python# python
Python 2.7.12 |Anaconda 4.2.0 (64-bit)| (default, Jul  2 2016, 17:42:40)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> import caffe
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "caffe/__init__.py", line 1, in <module>
    from .pycaffe import Net, SGDSolver, NesterovSolver, AdaGradSolver, RMSPropSolver, AdaDeltaSolver, AdamSolver
  File "caffe/pycaffe.py", line 13, in <module>
    from ._caffe import Net, SGDSolver, NesterovSolver, AdaGradSolver, \
ImportError: /root/anaconda2/bin/../lib/libstdc++.so.6: version `GLIBCXX_3.4.21' not found (required by caffe/_ca
>>> exit()
```

解决:

https://github.com/BVLC/caffe/issues/4953

https://gitter.im/BVLC/caffe/archives/2015/08/20

cd ..

pip install protobuf

sudo apt-get install python-protobuf

coda install libgcc

分类: 学习笔记

好文要顶 关注我 收藏该文

晨凫追风
关注 - 1
粉丝 - 14
+加关注

0 0

« 上一篇: Caffe学习笔记3
» 下一篇: Caffe学习笔记3

posted @ 2016-12-07 21:58 晨凫追风 阅读(14628) 评论(5) 编辑 收藏

评论

#1楼 2016-12-08 20:28 | 龙将

博主，你好。我按照你的博客安装tensorflow，在import tensorflow as tf
验证时出现
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named tensorflow
前面的都一样的结果，能告知一下是怎么回事，万分感谢

支持(0) 反对(0)

#2楼[楼主] 2016-12-08 20:35 | 晨凫追风

@ 龙将
那是一整段的Python代码，全部粘贴进去，试试

支持(0) 反对(0)

#3楼[楼主] 2016-12-08 20:37 | 晨凫追风

@ 龙将
source activate tensorflow
这句话用来激活该环境

支持(0) 反对(0)

#4楼 2017-01-12 11:30 | tzx0

十分感谢！外，opencv 3.2 不用修改源码就可以正确编译了。

支持(0) 反对(0)

#5楼 2017-12-08 00:29 | 给点阳光yh

楼主您好，我在安装好tensorflow的时候，尝试导入 import tensorflow

ImportError: libcudnn.so.5: cannot open shared object file: No such file or directory

报错，我发现/usr/local/cuda-8.0/lib64/下面都是8.0的，这怎么弄啊

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新IT新闻:
· 从年薪全美最高到1美元，他如何玩坏GoPro?
· 软银、春华资本正洽谈投资今日头条母公司字节跳动
· 滴滴：坚决拥护整改要求，持续加强安全运营保障工作
· 流利说王翌：我们随时可以盈利 今年已开始尝试国际化
· 研究显示美国医保公司数据成黑客首选"猎物"
» 更多新闻...

最新知识库文章:
· 为什么说 Java 程序员必须掌握 Spring Boot ?
· 在学习中，有一个比掌握知识更重要的能力
· 如何招到一个靠谱的程序员
· 一个故事看懂"区块链"
· 被踢出去的用户
» 更多知识库文章...