ubuntu16.04+cuda8.0+caffe 安装教程

第一步：安装 nvidia 显卡驱动

由于是台式机，显示器屏幕比较大，所以刚装好 Ubantu16.04 系统后分辨率比较低，看起来很不舒服，所以可以手动修改一下 grub 文件从而来提高分辨率，所以我们可以在终端输入

```
sudo vim /etc/default/grub
```

找到下面这几行：

```
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
# GRUB_GFXMODE=640×480
```

往最后添加一行设置自己想要的分辨率，推荐 1920x1080，只需要增加下面一行：

```
GRUB_GFXMODE=1920×1080 #这里分辨率自行设置
```

理论上设置完以后只需要在终端编辑

```
sudo update-grub
```

从而进行更新 grub,重新启动 Ubuntu 系统后就能生效了，但是，我们现在还没有安装显卡驱动，所以该次分辨率设置不能生效，所以现在我们应该做的工作是下载显卡驱动进行安装。

我们打算直接使用 Ubuntu 命令进行安装，要使用 Ubuntu 的命令进行安装，首先要做的就是更新 Ubuntu16.04 的源，终端输入

```
cd /etc/apt/     #进入 Ubuntu 高级软件包工具目录
sudo cp sources.list sources.list.bak    #先备份原来的源
sudo vi sources.list      #编辑 sources.list
```

把下面的源添加到 source.list 中:

```
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-security main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-updates main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-proposed main restricted universe multiverse
deb http://mirrors.ustc.edu.cn/ubuntu/ xenial-backports main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-security main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-updates main restricted universe multiverse
deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-proposed main restricted universe multiverse
```

deb-src http://mirrors.ustc.edu.cn/ubuntu/ xenial-backports main restricted universe multiverse

最后更新源和更新已安装的包，在终端中输入：

```
sudo apt-get update
sudo apt-get upgrade
```

我们继续来使用 add-apt-repository 脚本添加英伟达驱动 ppa 到当前库中并且自动导入公钥。

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

回车后继续

```
sudo apt-get update
sudo apt-get install nvidia-367
sudo apt-get install mesa-common-dev
sudo apt-get install freeglut3-dev
```

安装完成后就可以重启系统，然后 GTX1060 显卡驱动就会生效，由于先前我们设置好了分辨率，所以重新启动后我们的显示器显示的图像就会舒服多了。

重启完以后进行测试，终端输入：

```
nvidia-smi
```

出现下面这种效果就表示安装成功了。

```
chunhe@chunhe:~$ nvidia-smi
Sat Jul 29 15:59:32 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| 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 106...  Off  | 0000:01:00.0      On |                  N/A |
|  0%   55C    P2    32W / 120W |    377MiB /  6069MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0       907    G   /usr/lib/xorg/Xorg                           137MiB |
|    0      1728    G   compiz                                       163MiB |
|    0      1955    G   fcitx-qimpanel                                 7MiB |
|    0      2148    G   /usr/lib/firefox/firefox                       1MiB |
|    0      3398    C   /usr/lib/libreoffice/program/soffice.bin      63MiB |
+-----------------------------------------------------------------------------+
```

第二步：cuda 安装

所需文件：cuda_8.0.27_linux.run、cudnn-8.0-linux-x64-v5.1.tgz

百度云盘链接:http://pan.baidu.com/s/1boC8NAB 密码:sbzk

安装 cuda8.0

进入到 cuda_8.0.27_linux.run 所在目录，在终端执行命令如下：

```
sh cuda_8.0.27_linux.run --override
```

按下上面命令后，安装程序随即便启动了。接下来执行这几步操作：

1、一直按空格到最后，然后输入 accept 接受条款。

2、输入 n 不安装 nvidia 图像驱动，因为我们先前已经安装过了。

3、输入 y 安装 cuda8.0 工具。
4、回车确认 cuda 默认安装路径：/user/local/cuda-8.0。
5、输入 y 用 sudo 权限运行安装，输入密码。
6、可以输入 y 或者 n 安装或者不安装指向/usr/local/cuda 的符号链接。
7、输入 y 安装 CUDA 8.0Samples，以便后面进行测试。
8、回车确认 CUDA8.0 Samples 默认安装路径：/home/自己的用户名，该安装路径测试完就可以删除了。

出现“unsupported GNU version! gcc versions later than 5.3 are not supported!”的错误。这是由于 GCC 版本过高导致的。
解决办法：
终端执行如下命令

```
cd /usr/local/cuda-8.0/include
sudo cp host_config.h host_config.h.bak
sudo gedit host_config.h
```

接着可以通过 ctrl+f 寻找有“5.3”的地方，因为该文件只有一处 5.3，如下

```
# if __GNUC__ > 5 || (__GNUC__ == 5 && __GNUC_MINOR__ > 3)
#error -- unsupported GNU version! gcc versions later than 5.3 are not supported!
```

将两个 5 改成 6，即

```
#if __GNUC__ > 6 || (__GNUC__ == 6 && __GNUC_MINOR__ > 3)
```

安装 cudnn v5.1
先进入到 cudnn-8.0-linux-x64-v5.1.tgz 所在目录，然后解压该文件：

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

解压后会产生一个 cuda 目录，进入 cuda/include 目录下，执行如下操作：

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

再进入 cuda/lib64 目录下，执行如下操作：

```
cd ../lib64    #打开 lib64 目录
sudo cp lib* /usr/local/cuda/lib64/ #复制库文件
```

给所有用户增加这些文件的读权限

```
sudo chmod a+r /usr/local/cuda/include/cudnn.h
sudo chmod a+r /usr/local/cuda/lib64/libcudnn
```

建立软链接
首先在终端输入如下命令：

```
cd /usr/local/cuda/lib64/
sudo rm -rf libcudnn.so libcudnn.so.5
sudo ln -s libcudnn.so.5.1.5 libcudnn.so.5
sudo ln -s libcudnn.so.5 libcudnn.so
```

设置环境变量，终端输入

```
sudo gedit /etc/profile
```

在末尾加入

PATH=/usr/local/cuda/bin:$PATH
export PATH
保存后，创建链接文件
sudo vim /etc/ld.so.conf.d/cuda.conf
按 a 进入插入模式，增加下面一行
/usr/local/cuda/lib64
最后在终端输入 sudo ldconfig 使链接生效，ldconfig 原理如下链接：
http://blog.csdn.net/huangjin0507/article/details/50372721

注意：我在执行 ldconfig 命令时候遇到下面的错误：
/sbin/ldconfig.real: /usr/lib/nvidia-375/libEGL.so.1 不是符号连接
/sbin/ldconfig.real: /usr/lib32/nvidia-375/libEGL.so.1 不是符号连接
原因为：系统找的是一个符号连接，而不是一个文件。这应该是个 bug。
解决方法：
1、对这两个文件更名
2、重新建立符号连接
在终端执行下面的命令
sudo mv /usr/lib/nvidia-375/libEGL.so.1 /usr/lib/nvidia-375/libEGL.so.1.org
sudo mv /usr/lib32/nvidia-375/libEGL.so.1
/usr/lib32/nvidia-375/libEGL.so.1.org
sudo ln -s /usr/lib/nvidia-375/libEGL.so.375.39 /usr/lib/nvidia-375/libEGL.so.1
sudo ln -s /usr/lib32/nvidia-375/libEGL.so.375.39 /usr/lib32/nvidia-375/libEGL.so.1

cuda Samples 测试
打开 cuda 8.0 Samples 默认安装路径，终端输入
cd /home/username/NVIDIA_CUDA-8.0_Samples #username 是自己的用户名
sudo make all -j4 #4 核

小技巧:
查看 CPU 核心数：grep "cpu cores" /proc/cpuinfo|uniq
查看 CPU 个数：grep "physical id" /proc/cpuinfo|sort -u|wc -l
每个物理 CPU 上逻辑 CPU 个数：grep "siblings" /proc/cpuinfo|uniq
用上面的命令查看自己电脑的 CPU 核数。

执行完上面的 make all –j4 命令后会出现“unsupported GNU version! gcc versions later than 5.3 are not supported!”的错误。这是由于 GCC 版本过高导致的。
解决办法：
终端执行如下命令
cd /usr/local/cuda-8.0/include
sudo cp host_config.h host_config.h.bak
sudo gedit host_config.h
接着可以通过 ctrl+f 寻找有“5.3”的地方，因为该文件只有一处 5.3，如下
if __GNUC__ > 5 || (__GNUC__ == 5 && __GNUC_MINOR__ > 3)
#error -- unsupported GNU version! gcc versions later than 5.3 are not supported!

将两个 5 改成 6，即

#if __GNUC__ > 6 || (__GNUC__ == 6 && __GNUC_MINOR__ > 3)

保存退出，继续在终端输入

cd /home/username/NVIDIA_CUDA-8.0_Samples #username 是自己的用户名

sudo make all -j4 #4 核

等待编译完成

完成后继续向终端输入

cd bin/x86_64/linux/release

./deviceQuery

然后出现如下场景，表示已经成功安装了 cuda 了：

```
./deviceQuery Starting...

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

Detected 1 CUDA Capable device(s)

Device 0: "GeForce GTX 1060 6GB"
  CUDA Driver Version / Runtime Version          8.0 / 8.0
  CUDA Capability Major/Minor version number:    6.1
  Total amount of global memory:                 6070 MBytes (6364463104 bytes)
  (10) Multiprocessors, (128) CUDA Cores/MP:     1280 CUDA Cores
  GPU Max Clock rate:                            1772 MHz (1.77 GHz)
  Memory Clock rate:                             4004 Mhz
  Memory Bus Width:                              192-bit
  L2 Cache Size:                                 1572864 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:                     Yes
  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 = 8.0, CUDA Runtime Version = 8.0, NumDevs = 1, Device0 = GeForce GTX 1060 6GB
Result = PASS
```

三、依赖包安装

终端输入

sudo apt-get install build-essential #必要的编译工具依赖

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler

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

sudo apt-get install libatlas-base-dev

四、安装 python 的 pip 和 easy_install

终端输入

cd

wget --no-check-certificate https://bootstrap.pypa.io/ez_setup.py

sudo python ez_setup.py --insecure

wget https://bootstrap.pypa.io/get-pip.py

sudo python get-pip.py

五、安装科学计算和 python 所需的部分库

终端输入

```
sudo apt-get install libblas-dev liblapack-dev libatlas-base-dev gfortran python-numpy
```

六、安装 git，拉取源码

终端输入

```
sudo apt-get install git
git clone https://github.com/BVLC/caffe.git
```

七、安装 python 依赖

先执行命令

```
cd /home/username/caffe/python #username 是你自己的用户名
```

终端输入

```
sudo apt-get install python-pip 安装 pip
sudo su
for req in $(cat "requirements.txt"); do pip install -i https://pypi.tuna.tsinghua.edu.cn/simple $req; done
```

最后按 Ctrl+D 退出 sudo su 模式

八、编译 caffe

终端输入

```
cd /home/pawn/caffe
cp Makefile.config.example Makefile.config
gedit Makefile.config
```

然后将 USE_CUDNN := 1 取消注释，

然后将 INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include 后面打上一个空格 然后添加/usr/include/hdf5/serial 如果没有这一句可能会报一个找不到 hdf5.h 的错误

接着在终端输入

```
make all -j4
```

make 过程中出现找不到 lhdf5_hl 和 lhdf5 的错误，

解决方案：

在计算机中搜索 libhdf5_serial.so.10.1.0，在终端执行如下命令：

```
sudo find -name libhdf5_serial.so.10.1.0
```

找到后进入所在目录下

```
sudo ln libhdf5_serial.so.10.1.0 libhdf5.so
sudo ln libhdf5_serial_hl.so.10.0.2 libhdf5_hl.so
```

最后在终端输入 sudo ldconfig 使链接生效

原终端中输入 make clean 清除第一次编译结果

再次输入 make all -j4 重新编译

最后在终端输入

```
make test -j4
make runtest -j4
```

make pycaffe -j4
make distribute 生成发布安装包
这时候可以测试一下 python，终端输入
cd /home/pawn/caffe/python
python
import caffe
如果不报错就说明编译成功
如有其他问题请参考解决 caffe-1.0 编译问题.txt

九、设置 python 路径

在使用 make pycaffe -j4 命令完成 caffe 的 python 接口生成之后，还需要将 python 接口的路径进行设置。
终端执行
gedit ~/.bashrc
来对路径进行设置，在文件最后一行加入路径：
export
PYTHONPATH=/home/startag/caffe/python/:/home/startag/caffe/python/caffe/
注销或者重启，路径生效。
解释：PYTHONPATH 是 Python 搜索路径，默认我们 import 的模块都会从 PYTHONPATH 里面寻找。
敲下 import urllib 后，Python 解释器会逐个从上面的路径列表选出一个路径然后搜索 urllib 模块直到找到为止。这里最后在 D:\Python3\lib 下找到（ubuntu 自带 python3）

十、mnist 测试

下载 mnist 数据集，终端输入
cd /home/pawn/caffe/data/mnist/
./get_mnist.sh #获取 mnist 数据集
在/home/pawn/caffe/data/mnist/目录下会多出训练集图片、训练集标签、测试集图片和测试集标签等 4 个文件
mnist 数据格式转换，终端输入
cd /home/pawn/caffe/
./examples/mnist/create_mnist.sh
必须要在第一行之后运行第二行，即必须要在 caffe 根目录下运行 create_mnist.sh
此时在/caffe/examples/mnist/目录下生成 mnist_test_lmdb 和 mnist_train_lmdb 两个 LMDB 格式的训练集和测试集
LeNet-5 模型描述在/caffe/examples/mnist/lenet_train_test.prototxt
Solver 配置文件在/caffe/examples/mnist/lenet_solver.prototxt
训练 mnist，执行文件在/caffe/examples/mnist/train_lenet.sh
终端输入
cd /home/pawn/caffe/
./examples/mnist/train_lenet.sh