

BasicDeepLearningEnvironment

by fcl 2020.10.27

前言

根据自己要跑的算法，列以下表格

0. 删除原装驱动，安装英伟达驱动
1. 首先看你跑的程序需要装CUDA几
2. 查询CUDA对应的GCC版本
3. 安装CUDA
4. 安装CUDNN(与cuda对应)
5. 安装Anaconda

本文案例中的硬件与版本选择

硬件类别	以长蝰哥电脑为例，目前在410
显卡：	GeForce GTX 1070 Ti/PCIe/SSE2
CPU：	i7-4790
内存：	7.7 GiB
系统：	Ubuntu18.04
gcc：	7.3
CUDA：	CUDA10.1
cuDNN：	cuDNN v8.0.3
Anaconda：	Python 3.8 Anaconda 4.8.3

配置查询命令

`uname -a` # 查看ubuntu系统内核版本

```
fcl@fcl-All-Series:~$ uname -a
Linux fcl-All-Series 5.4.0-48-generic #52~18.04.1-Ubuntu SMP Thu Sep 10 12:50:22
UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

`lspci | grep VGA` # 查看集成显卡

`lspci | grep NVIDIA` # 查看NVIDIA显卡

`ls /usr/bin/gcc*` #查看已安装的gcc

一些查询网站

显卡驱动与CUDA对应关系

Table 2. CUDA Toolkit and Compatible Driver Versions		
CUDA Toolkit	Linux x86_64 Driver Version	Windows x86_64 Driver Version
11.1	>=455.23	>=456.38
Table 2. CUDA Toolkit and Compatible Driver Versions		
CUDA Toolkit	Linux x86_64 Driver Version	Windows x86_64 Driver Version
11.1.1 Update 1	>=455.32	>=456.81
11.1 GA	>=455.23	>=456.38
11.0.3 Update 1	>= 450.51.06	>= 451.82
11.0.2 GA	>= 450.51.05	>= 451.48
11.0.1 RC	>= 450.36.06	>= 451.22
10.2.89	>= 440.33	>= 441.22
10.1 (10.1.105 general release, and updates)	>= 418.39	>= 418.96
10.0.130	>= 410.48	>= 411.31
9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26
9.2 (9.2.88)	>= 396.26	>= 397.44
9.1 (9.1.85)	>= 390.46	>= 391.29
9.0 (9.0.76)	>= 384.81	>= 385.54
8.0 (8.0.61 GA2)	>= 375.26	>= 376.51
8.0 (8.0.44)	>= 367.48	>= 369.30
7.5 (7.5.16)	>= 352.31	>= 353.66
7.0 (7.0.28)	>= 346.46	>= 347.62

CUDA与GCC对应关系，这里以CUDA10.1为例

CUDA 11.1

Distribution	GCC2,3
Ubuntu 18.04.z (z <= 5)	9.x

CUDA 10.1

Distribution	Kernel*	GCC
Ubuntu 18.04.3 (**)	5.0.0	7.4.0

CUDA 10.0

Distribution	Kernel*	GCC
Ubuntu 18.04.1 (**)	4.15.0	7.3.0

NVIDIA显卡驱动安装

禁用自带的驱动

输入一下指令查看有无输出

```
lsmod | grep nouveau
```

如果没有输出就不需要做下面的步骤

如果有输出，则进行以下步骤

```
sudo gedit /etc/modprobe.d/blacklist-nouveau.conf
```

添加以下两行

```
blacklist nouveau
options nouveau modeset=0
```

更新驱动信息

```
sudo update-initramfs -u
```

重启电脑之后，因为禁止了显卡的驱动，所以重启后显示的效果很不好，通过这个可以看出，是否完成这一步操作

再次输入`lsmod | grep nouveau`确认无输出内容即表示默认的自带显卡已经被干掉了

删除以前的nvidia驱动

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

添加Graphic Drivers PPA

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

```
sudo apt-get update
```

查看合适的驱动版本

```
ubuntu-drivers devices
```

```
fcl@fcl-All-Series:~$ ubuntu-drivers devices
```

```
== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==
```

```
modalias : pci:v000010DEd00001B82sv000019DAsd00002436bc03sc00i00
```

```
vendor   : NVIDIA Corporation
```

```
model    : GP104 [GeForce GTX 1070 Ti]
```

```
driver   : nvidia-driver-450-server - distro non-free
```

```
driver   : nvidia-driver-410 - third-party free
```

```
driver   : nvidia-driver-418-server - distro non-free
```

```
driver   : nvidia-driver-435 - distro non-free
```

```
driver   : nvidia-driver-455 - third-party free recommended
```

```
driver   : nvidia-driver-415 - third-party free
```

```
driver   : nvidia-driver-390 - distro non-free
```

```
driver   : nvidia-driver-440-server - distro non-free
```

```
driver   : nvidia-driver-450 - distro non-free
```

```
driver   : xserver-xorg-video-nouveau - distro free builtin
```

安装该驱动

```
sudo apt-get install nvidia-driver-455
```

重启机器验证

```
sudo reboot
```

检查显卡驱动生效未否

```
nvidia-smi
```

Install GCC

查看已经安装的gcc

```
ls /usr/bin/gcc*
```

```
fcl@fcl-All-Series:~$ ls /usr/bin/gcc*
/usr/bin/gcc      /usr/bin/gcc-ar    /usr/bin/gcc-nm    /usr/bin/gcc-ranlib
/usr/bin/gcc-6    /usr/bin/gcc-ar-6  /usr/bin/gcc-nm-6  /usr/bin/gcc-ranlib-6
**/usr/bin/gcc-7  /usr/bin/gcc-ar-7  /usr/bin/gcc-nm-7  /usr/bin/gcc-ranlib-7**
/usr/bin/gcc-9    /usr/bin/gcc-ar-9  /usr/bin/gcc-nm-9  /usr/bin/gcc-ranlib-9
```

可根据CUDA需求，安装好一个需要的版本（就不用换版本了），我这里预装了三种版本

查看当前系统使用的gcc

```
gcc -v
```

```
g++ --version
```

设置gcc优先级（gcc的版本切换）

指令讲解：

方式一：

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 70 --slave
```

```
/usr/bin/g++ g++ /usr/bin/g++-9
```

--slave后面加入g++是指，当切换gcc版本时也同时切换g++版本

方式二：

注：另一种方法在~/.bashrc中添加

```
alias gcc='/usr/bin/gcc-5'
alias g++='/usr/bin/g++-5'
```

这两种方法选一种即可，推荐第一种

因为本例中我需要使用gcc7，所以我把gcc7的优先级设为80（最大就行），gcc9的优先级设为70

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 80 --slave
```

```
/usr/bin/g++ g++ /usr/bin/g++-7
```

设置好后查看gcc优先级

```
sudo update-alternatives --config gcc
```

如果安装了多个gcc，且多个gcc分配了优先级则会出现下面的提示，星号是当前默认使用的版本

```
fc1@fc1-All-Series:~$ `sudo update-alternatives --config gcc`
有 2 个候选项可用于替换 gcc (提供 /usr/bin/gcc)
  选择                路径                优先级    状态
*  0                  /usr/bin/gcc-7      80        自动模式
  1                  /usr/bin/gcc-7      80        手动模式
  2                  /usr/bin/gcc-9      70        手动模式
```

如果只安装了一个版本的gcc则会出现下面的提示

```
fc1@fc1-All-Series:~$ sudo update-alternatives --config gcc
链接组 gcc (提供 /usr/bin/gcc)中只有一个候选项: /usr/bin/gcc-9
无需配置。
```

删除gcc的指令

```
sudo update-alternatives --remove gcc /usr/bin/gcc-5
```

安装CUDA

下载cuda

[官网](#)的cuda是最新版本的

右下角有一个Archive of Previous CUDA Releases

选择CUDA Toolkit10.1，点击Base Installer的[Download](#)

安装依赖库（根据提示）

```
sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev
libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev
```

正式安装cuda

```
sudo sh cuda_10.1.105_418.39_linux.run
```

在出现的图形界面中除了driver不用安装其他都要装

如果安装失败了，可以尝试再添加以下库

```
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 libopenblas-dev liblapack-dev libatlas-base-dev
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

配置环境变量

```
sudo gedit ~/.bashrc
```

添加路径

```
export PATH=/usr/local/cuda-10.1/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-
10.1/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

```
source ~/.bashrc
```

验证CUDA安装成功

```
cd NVIDIA_CUDA-10.1_Samples/5_Simulations/fluidsGL
make clean && make
./fluidsGL
```

出现浮萍即代表安装完成

```
cd /usr/local/cuda/samples/1_Utilities/deviceQuery
sudo make
./deviceQuery
```

此时可以查看CUDA版本 `nvcc -V`

安装cuDNN

下载地址<https://developer.nvidia.com/rdp/cudnn-archive>

选择Download cuDNN v8.0.3 (August 26th, 2020), for CUDA 10.1

下载以下三个

```
cuDNN Runtime Library for Ubuntu18.04 (Deb)
cuDNN Developer Library for Ubuntu18.04 (Deb)
cuDNN Code Samples and User Guide for Ubuntu18.04 (Deb)
```

下载完成后在其目录下打开终端

输入以下命令

```
sudo dpkg -i libcudnn7_7.0.3.11-1+cuda9.0_amd64.deb
sudo dpkg -i libcudnn7-dev_7.0.3.11-1+cuda9.0_amd64.deb
sudo dpkg -i libcudnn7-doc_7.0.3.11-1+cuda9.0_amd64.deb
```

验证安装完成

```
cp -r /usr/src/cudnn_samples_v8/ $HOME
cd cudnn_samples_v8/mnistCUDNN
make clean && make
./mnistCUDNN
```

最终如果有提示信息：“Test passed!”，则说明安装成功

安装Anaconda

安装anaconda过程

下载地址<https://www.anaconda.com/products/individual#linux>

下载后在其目录下打开终端输入 `bash Anaconda3-2020.07-Linux-x86_64.sh`

按回车或者空格确认，再输入yes和enter安装

安装好后，打开

```
sudo gedit ~/.bashrc
```

在最后一行输入

```
conda deactivate
```

创建自己的环境并指定python版本

```
conda create -n tensorflow python=3.8
```

进入环境

```
conda activate tensorflow
```

可以输入

``

安装国内镜像(非必要)

如果想换源可以进行以下操作

清华镜像

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgsg/free/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
```

设置搜索时显示通道地址

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

中科大镜像

```
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgsg/main/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkgsg/free/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/msys2/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/bioconda/
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/cloud/menpo/
conda config --set show_channel_urls yes
```