ubuntu 18.04-pyTorch\_gpu

ubuntu 18.04

RTX 2070S

cuda\_10.0.130\_410.48\_linux.run

cuDNN 7.6.5 for cuda 10.0

Anaconda3-2020.02-Linux-x86\_64.sh (python 3.7)

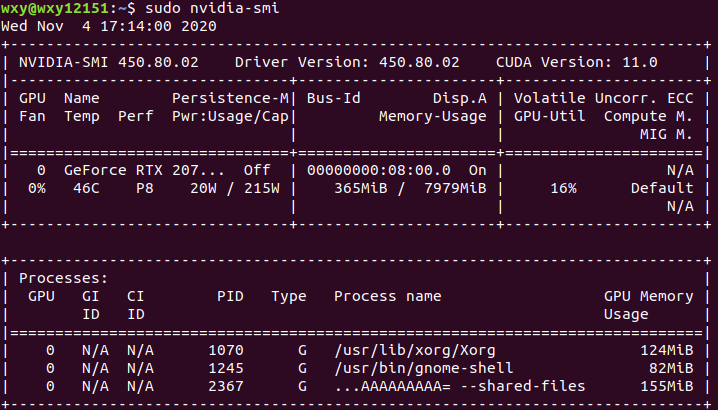
pytorch 1.2.0

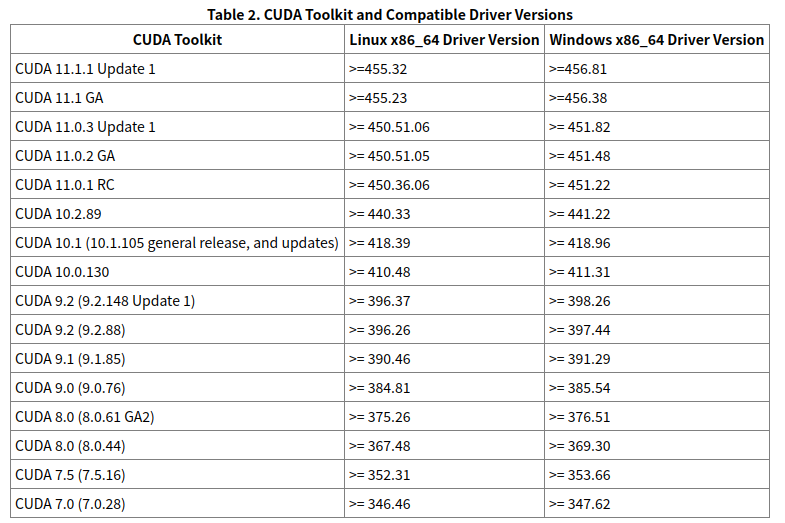
1. 准备工作
2. 检查版本和类型

在终端输入

sudo nvidia-smi

则Driver Version 450.80.2





1. 禁用nouveau驱动

在终端输入：

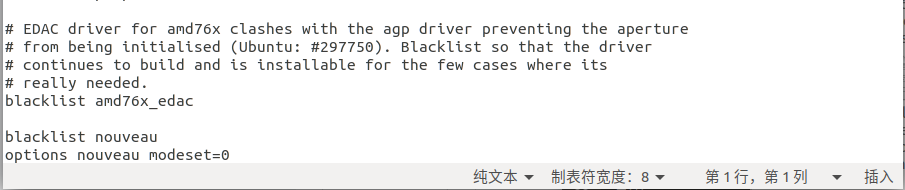
sudo vim /etc/modprobe.d/blacklist.conf

在blacklist.conf文本最后添加：

blacklist nouveau

options nouveau modeset=0

如图：



然后在终端更新并重启：

sudo update-initramfs -u

sudo reboot

重启后检查nouveau是否禁用成功：

lsmod | grep nouveau

若屏幕没哟输出，则表明禁用成功

1. 安装驱动

安装所有推荐的驱动

sudo ubuntu-drivers autoinstall

也可以只安装一个

sudo apt install nvidia-390

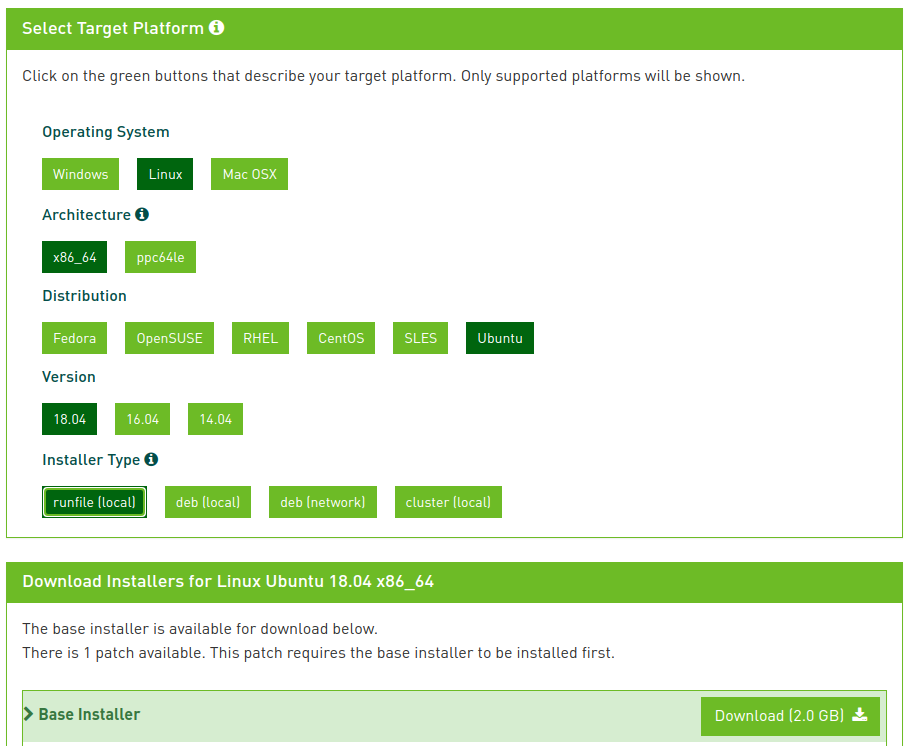
安装完成后重启，然后检查nvidia是否启用成功

lsmod | grep nvidia

若有显示nvidia信息，则驱动启用成功

1. 安装cuda
2. [https://developer.nvidia.com/cuda-downloads](https://developer.nvidia.com/cuda-downloads官网上下载合适版本，我选择的CUDA10.0，选择runfile下载)

官网上下载合适版本，我选择的CUDA10.0，选择runfile下载



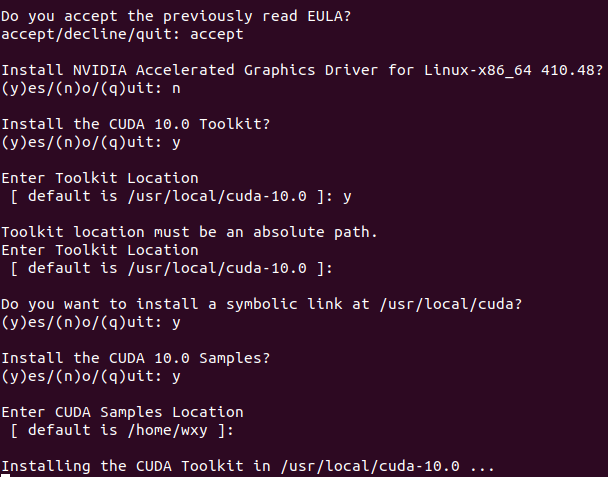
1. 下载后的文件为cuda\_10.0.130\_410.48\_linux.run文件赋予执行权限

chmod a+x cuda\_10.0.130\_410.48\_linux.run

1. 进行安装

sudo sh cuda\_10.0.130\_410.48\_linux.run

1. 开始安装，按Ctrl+C可以直接完成说明阅读。安装时，除了驱动其他都选y，cuda sample非必须可以不装，cuda安装路径默认。



1. 安装完成后，修改环境变量(i编辑；esc退出到命令行；输入冒号：wq保存退出)

sudo vim /etc/bash.bashrc

在/etc/bash.bashrc文件的最后加入以下配置

export CUDA\_HOME=/usr/local/cuda-10.0

export LD\_LIBRARY\_PATH=${CUDA\_HOME}/lib64

export PATH=${CUDA\_HOME}/bin:${PATH}

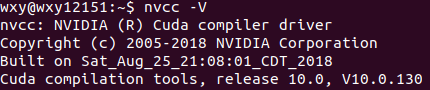
1. 更新环境变量

source /etc/bash.bashrc

1. 查看安装的版本信息

ncvv -V

结果如下：



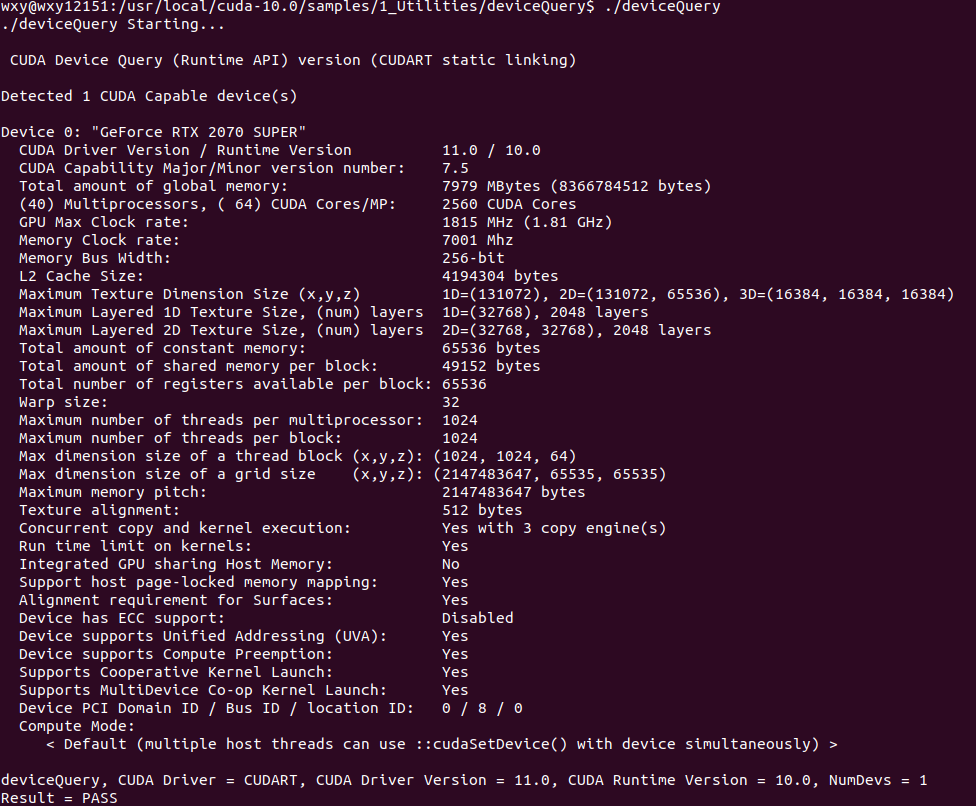
1. 测试安装是否成功

cd /usr/local/cuda-10.0/samples/1\_Utilities/deviceQuery

make(出现致命从错误权限不够前面＋sudo)

./deviceQuery

若输出如下，则安装成功



Tip：cuda driver和cuda runtime不一致：

1） 两者不是必须一致。

2）CUDA Driver Version应该是跟着GPU驱动走的，Runtime Version取决于当前设置。Driver Version一般 >= Runtime Version, 否则insufficient。

3） 软件运行时调用的应该是Runtime Version。

1. 安装cudnn
2. 下载对应版本的cudnn

登陆<https://developer.nvidia.com/rdp/form/cudnn-download-survey>并选择合适的版本进行下载，由于我安装的CUDA 10.0，这里选择对应的版本cuDNN 7.6.5 for cuda 10.0进行下载，如下图：



1. 解压下载后的cudnn

下载得到的是一个压缩文件，需要解压，命令中的download是cudnn被下载的位置

tar -zxvf 文件拖进来

1. 复制解压后的cudnn到cuda的目录中

进入解压后的文件目录，然后进行复制

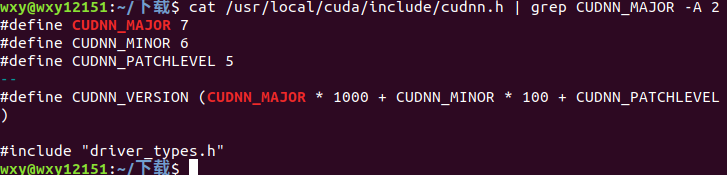
cd /download/

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

sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64

sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcdnn\*

结果如下：



1. 安装Anaconda
2. 下载安装文件

官网：<https://www.anaconda.com/products/individual>

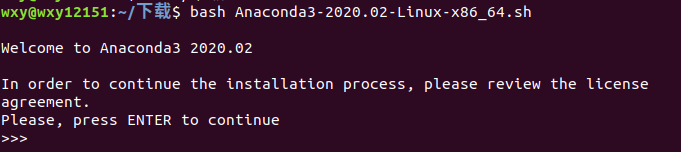
本次安装Anaconda3-2020.02-Linux-x86\_64.sh

1. 启动安装程序

在终端输入，bash后面是下载安装包的包名

sudo bash Anaconda3-2020.02-Linux-x86\_64.sh

进入安装界面

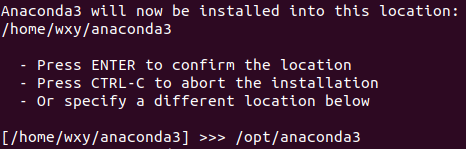


回车，查看许可协议，ctrl+c可以直接完成阅读



确认安装，选择yes

设置安装路径，直接回车则安装至为默认路径，即用户个人文件夹，这里我将安装目录修改为/opt/anaconda3，因为/opt目录下一般存放程序



进行安装

过程中会询问是否修改个人账户的环境变量配置，这里选择no，因为后面会修改整个系统的环境变量。

还会询问是否安装VSCode，个人选择。

1. 配置环境变量

source /etc/bash.bashrc

在bash.bashrc文件中加入，其中"/opt/anaconda3/"是安装路径

export PATH="/opt/anaconda3/bin:$PATH"

更新环境变量

source /etc/bash.bashrc

1. 安装pytorch
2. 加入清华镜像源（目的提速）

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

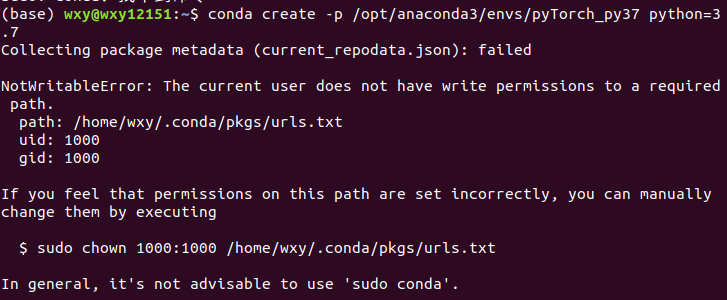
conda config --set show\_channel\_urls yes

1. 创建环境

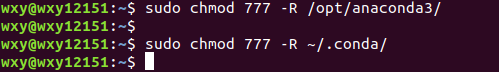
在/opt/anaconda/envs/中创建一个新的环境，我这里命名为pyTorch\_py37

conda create -p /opt/anaconda3/envs/pyTorch\_py37 python=3.7

出现问题：



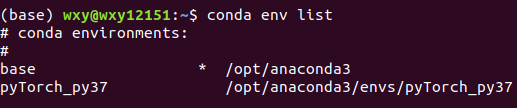
退出base环境授予文件夹权限解决：



也可以用下面的语句进行创建，但这样环境会添加到用户个人路径下，其他人没办法用这个环境。

conda create -n pyTorch\_py37 python=3.7

完成环境创建后，看一下环境列表，pyTorch\_py37环境已成功配置

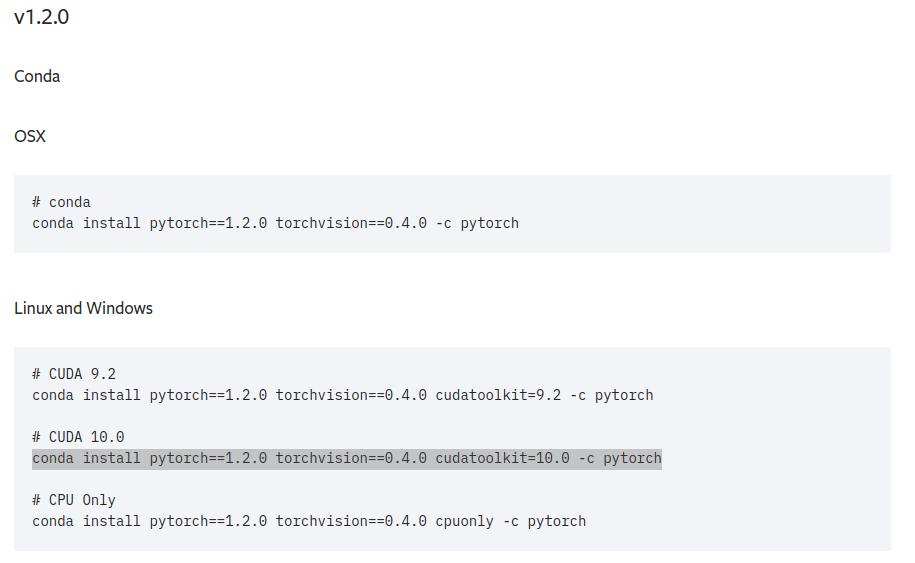


1. 激活环境

conda activate pyTorch\_py37

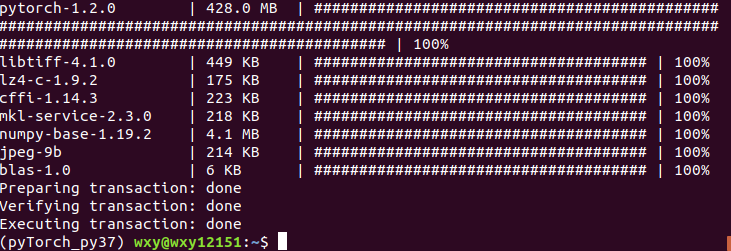
1. 配置环境，安装

从<https://pytorch.org/get-started/previous-versions/#wheel-1>中找到匹配自己版本的安装指令，这里选择1.2.0版本的torch





下载过程比较慢



完结撒花！

1. 验证（来源于pytorch官网）

为了确保正确安装了PyTorch，我们可以通过运行示例PyTorch代码来验证安装。在这里，我们将构造一个随机初始化的张量。

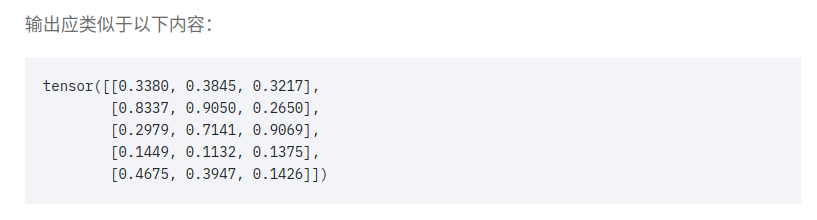
from \_\_future\_\_ import print\_function

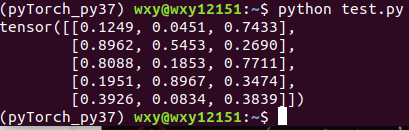
import torch

x = torch.rand(5, 3)

print(x)

输出结果应该是：





验证通过;

此外，要检查PyTorch是否启用了GPU驱动程序和CUDA并可以通过其访问，请运行以下命令以返回是否启用了CUDA驱动程序：

import torch

# 若正常则静默

a = torch.tensor(1.)

# 若正常则静默

a.cuda()

# 若正常则返回 tensor(1., device='cuda:0')

from torch.backends import cudnn

# 若正常则静默

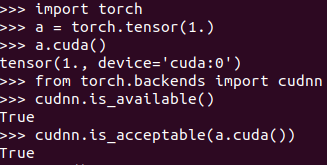
cudnn.is\_available()

# 若正常则返回 True

cudnn.is\_acceptable(a.cuda())

# 若正常则返回 True

我的结果：



没有问题，安心完结！