Pytorch环境配置

环境安装

（1）PyTorch环境配置

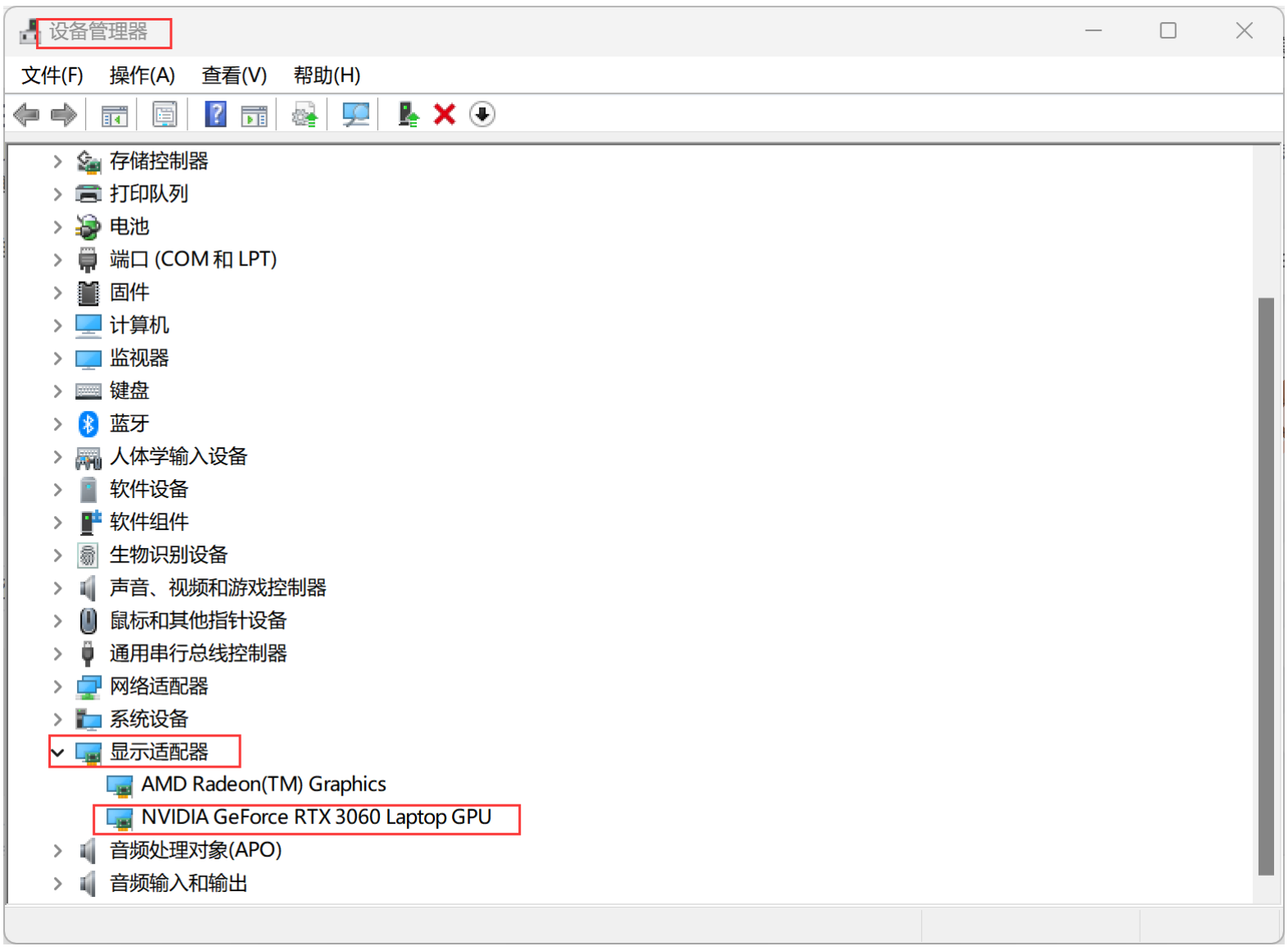
创建一个虚拟python环境：

```
conda create -n DL_Pytorch
conda activate DL_Pytorch
```

- 查看虚拟环境列表：
  - conda info --envs/-e
  - conda env list
- 激活虚拟环境
  - source activate .... （conda旧版本：4.3及更早的版本）
  - conda activate .... （conda新版本：4.4及以后）

（2）Pytorch-CUDA版本环境配置

电脑如果是Windows平台下的Nvidia GPU的用户，需配置Pytorch的CUDA版本，分为三步：



1. 安装或更新NVIDA显卡驱动

官方驱动下载地址：https://www.nvidia.cn/Download/index.aspx?lang=cn



2. 安装`CUDA Toolkit + cudnn`:

1）CUDA安装

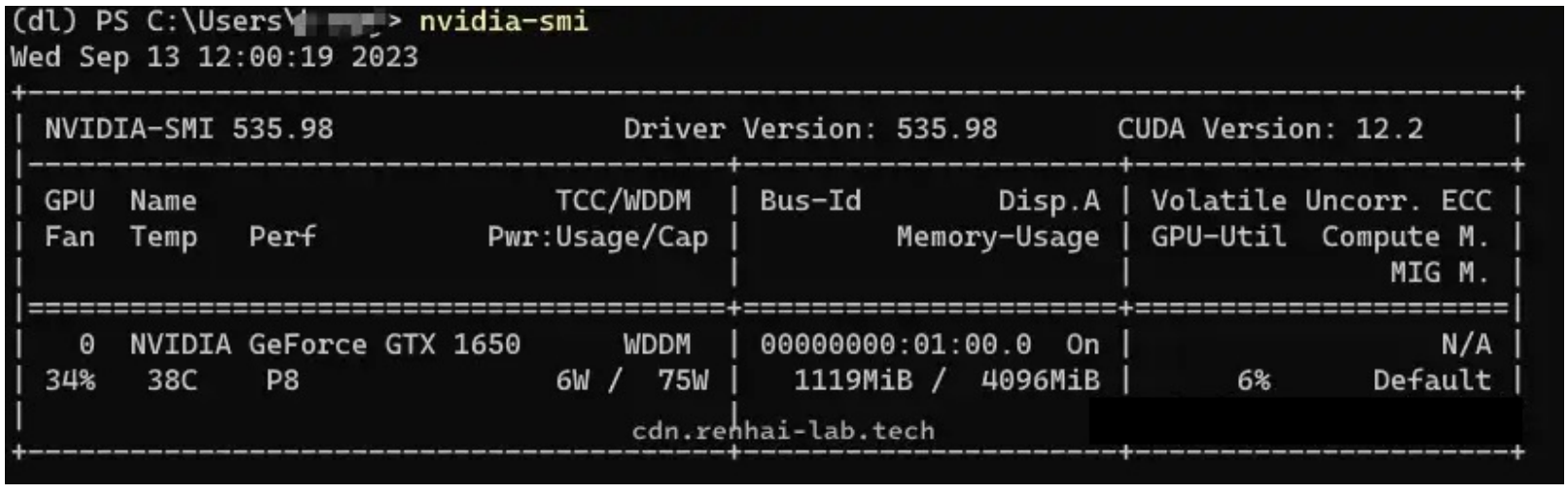
在`CUDA Toolkit`安装前用以下命令查询机器上显卡最高支持的CUDA 版本:

终端输入:

```
nvidia-smi
```

下图中CUDA Version是12.2。

> 如果你没有安装`cuda toolkit`或者需要升级，可以去官网下载:
> https://developer.nvidia.com/cuda-toolkit-archive



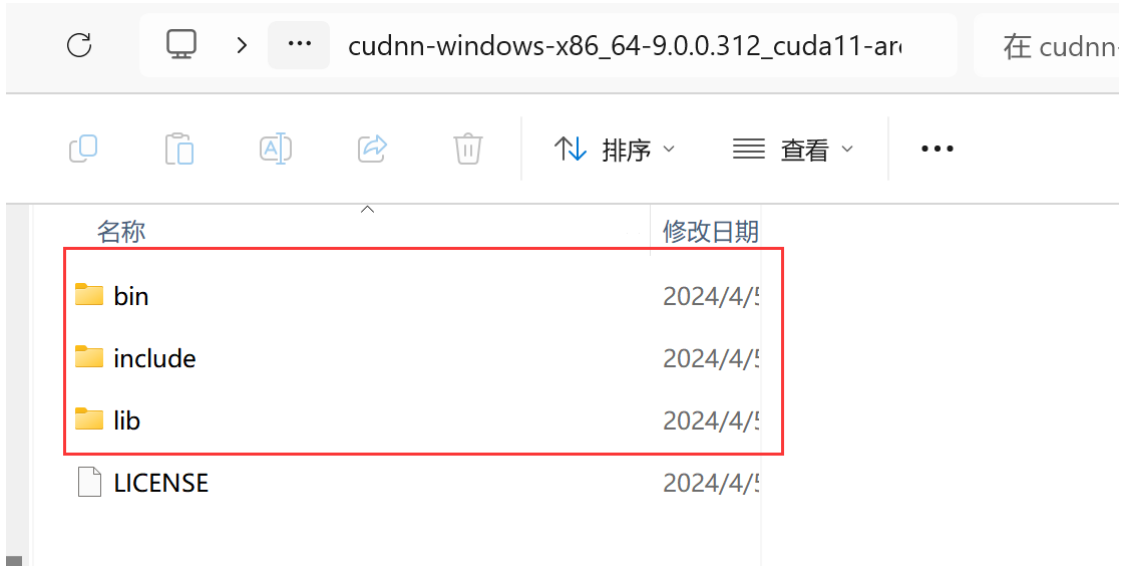
2）cuDNN安装

NVIDIA CUDA深度神经网络库 (cuDNN) 是一个 **GPU 加速**的**深度神经网络基元库**，能够以高度优化的方式实现标准例程（如前向和反向卷积、池化层、归一化和激活层）。

全球的深度学习研究人员和框架开发者都依赖 cuDNN 来实现高性能 GPU 加速。借助 cuDNN，研究人员和开发者可以专注于训练神经网络及开发软件应用，而不必花时间进行低层级的 GPU 性能调整。cuDNN 可加速广泛应用的深度学习框架，包括 Caffe2、Keras、MATLAB、MxNet、PaddlePaddle、PyTorch和 TensorFlow。

下载地址：cuDNN Archive | NVIDIA Developer

（1）下载并解压文件



（2）复制内容到CUDA安装路径

CUDA安装默认路径：

- Windows：`C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA`
- Linux：`/usr/local/cuda`



3. 安装Pytorch

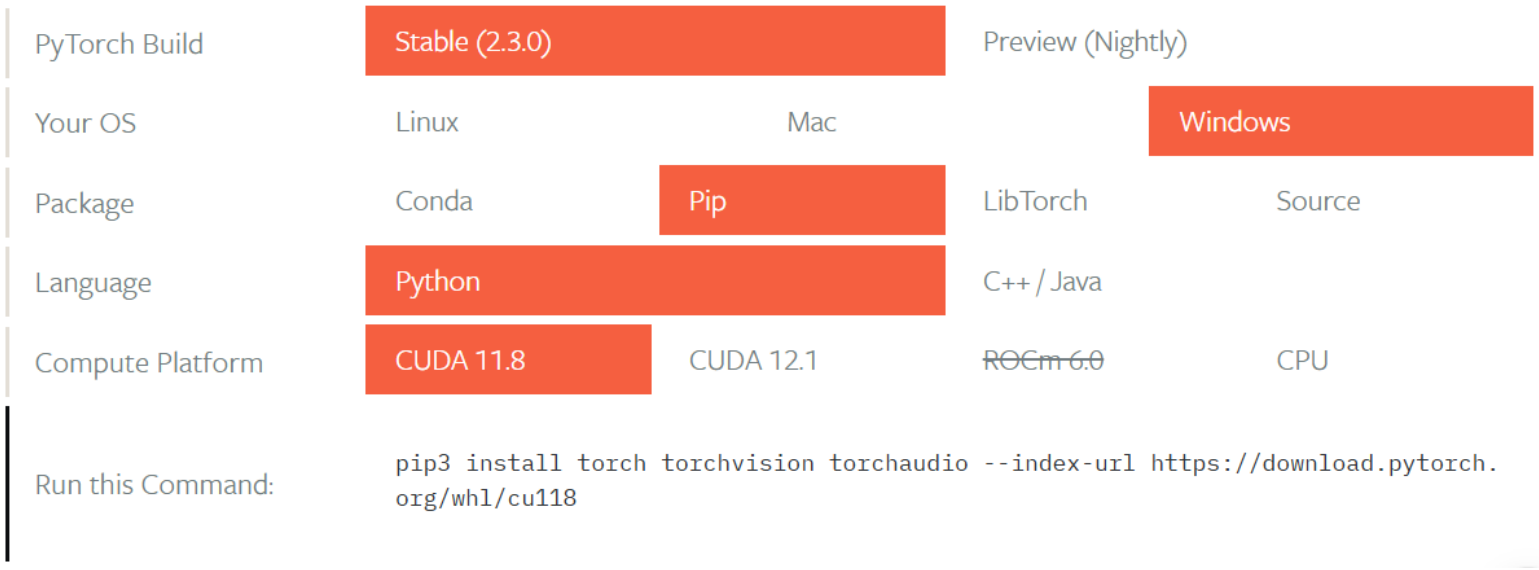
（1）在线安装

打开[pytorch安装指导网站](), 选择合适的系统平台，关键是在`compute platform`选择一个不高于你电脑上的`CUDA Version`，复制命令安装。

- pip install torch==版本号
- conda install torch==版本号

```
# 使用conda安装
conda install python pytorch torchvision torchaudio pytorch-cuda=11.7 -c
pytorch -c nvidia
# 使用pip安装
pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu117
或者
pip install torch==2.0.0+cu118 torchvision==0.15.0+cu118
torchaudio==2.0.1+cu118 -f
https://download.pytorch.org/whl/torch_stable.html
```

NOTE: Latest PyTorch requires Python 3.8 or later.



(2) 离线安装

- 离线包下载地址：download.pytorch.org/whl/torch_stable.html
- 安装方式

```
pip install torch-2.0.1+cu118-cp310-cp310-win_amd64.whl
```

注意：

1）Pytorch与torchvision版本对应问题

[Pytorch与torchvision版本配套]

The following is the corresponding `torchvision` versions and supported Python versions.

`torch`	`torchvision`	Python
`main` / `nightly`	`main` / `nightly`	`>=3.8`, `<=3.11`
`2.2`	`0.17`	`>=3.8`, `<=3.11`
`2.1`	`0.16`	`>=3.8`, `<=3.11`
`2.0`	`0.15`	`>=3.8`, `<=3.11`

▼ older versions

`torch`	`torchvision`	Python
`1.13`	`0.14`	`>=3.7.2`, `<=3.10`
`1.12`	`0.13`	`>=3.7`, `<=3.10`
`1.11`	`0.12`	`>=3.7`, `<=3.10`
`1.10`	`0.11`	`>=3.6`, `<=3.9`
`1.9`	`0.10`	`>=3.6`, `<=3.9`
`1.8`	`0.9`	`>=3.6`, `<=3.9`
`1.7`	`0.8`	`>=3.6`, `<=3.9`
`1.6`	`0.7`	`>=3.6`, `<=3.8`
`1.5`	`0.6`	`>=3.5`, `<=3.8`
`1.4`	`0.5`	`==2.7`, `>=3.5`, `<=3.8`
`1.3`	`0.4.2` / `0.4.3`	`==2.7`, `>=3.5`, `<=3.7`
`1.2`	`0.4.1`	`==2.7`, `>=3.5`, `<=3.7`
`1.1`	`0.3`	`==2.7`, `>=3.5`, `<=3.7`
`<=1.0`	`0.2`	`==2.7`, `>=3.5`, `<=3.7`

> 如果你的conda解决环境很慢，可以试一试pip安装。

2）使用镜像源

- 使用镜像源：
  - pip install torch -i [镜像源]
  - conda install torch -c [镜像源]
- 常用镜像源
  - 清华源：https://pypi.tuna.tsinghua.edu.cn/simple
  - 豆瓣源：https://pypi.doubanio.com/simple/

3）安装验证

```
import torch
# 打印出正在使用的PyTorch和CUDA版本。
print(torch.__version__)
print(torch.version.cuda)

# 测试GPU是否生效
print(torch.cuda.is_available())
```

（3）导入PyToch

导入 PyTorch 并检查正在使用的版本。

```
import torch
torch.__version__
>>> '2.0.1'
```