



## 目 录

|                            |    |
|----------------------------|----|
| 引入.....                    | 1  |
| 一、安装 Anaconda .....        | 2  |
| 1.1 卸载 Anaconda（可选） .....  | 2  |
| 1.2 安装 Anaconda .....      | 3  |
| 1.3 配置环境变量.....            | 4  |
| 1.4 桌面快捷方式.....            | 6  |
| 二、安装 PyTorch（GPU 版）库 ..... | 7  |
| 2.1 创建虚拟环境.....            | 7  |
| 2.2 GPU 基础.....            | 8  |
| 2.3 安装 CUDA（可选） .....      | 9  |
| 2.4 安装 PyTorch.....        | 12 |
| 2.5 检验 cuda 是否可用 .....     | 15 |
| 三、Jupyter 代码编辑器.....       | 16 |
| 3.1 修改工作路径（可选） .....       | 16 |
| 3.2 修改默认字体（可选） .....       | 16 |
| 3.3 虚拟环境连接 Jupyter.....    | 17 |
| 四、PyCharm 代码编辑器.....       | 18 |
| 4.1 卸载 PyCharm（可选） .....   | 18 |
| 4.2 安装 PyCharm.....        | 19 |
| 4.3 初次设置 PyCharm.....      | 20 |
| 4.4 虚拟环境连接 PyCharm.....    | 21 |



# 引入

## 0.1 课程内容

- 第一章：卸载 Anaconda、安装 Anaconda、配置环境变量、桌面快捷方式；
- 第二章：创建虚拟环境、安装 CUDA、安装 PyTorch、检验 cuda 可用；
- 第三章：修改工作路径、修改默认字体、虚拟环境连接 Jupyter；
- 第四章：卸载 PyCharm、安装 PyCharm、虚拟环境连接 PyCharm。

## 0.2 适宜人群

- Windows 系统的用户，且不区分 win7、win9、win10 与 win11 等；
- 计算机内没装过任何 Python 软件，没有任何 Python 常识的小白；
- 计算机内此时环境十分混乱，Python 解释器下载过几十个不同的版本的高手（因为会新建虚拟环境，此新环境与计算机的基础环境平行且独立）。

## 0.3 视频特点

- 清晰度：本视频分辨率为 1080P，请调高分辨率；
- 讲义：Bilibili 视频的置顶评论中附有网盘链接，内含讲义与问题集锦；
- 交流群：微信搜索“Boolart003”的小火柴，暗号“杰哥”，即可拉入群聊。

## 0.4 安装常识

### （1）Python 解释器

Python 解释器的名称由 Python 与版本号组成，如 Python 3.9.0。

解释器是向计算机解释 Python 语言的工具，只有下载了解释器，才能进行 Python 编程。因此，下载完 Python 后就可以在命令提示符（cmd）中运行 Python 代码，当然，B 站上有人在记事本里写代码，再改后缀为.py 的也是这个原理。

### （2）代码编辑器

代码编辑器是编辑代码的工具，使用代码编辑器前必须下载 Python 解释器。你可以将代码编辑器理解为“可运行代码的记事本”，没有解释器它就是个壳子。

常用的 Python 代码编辑器有 Jupyter Notebook、PyCharm、Visual Studio Code。类似的，C 的代码编辑器包括 Visual C++、Dev C++、Visual Studio Code。代码编辑器任选其一即可，没有高低贵贱之分，相比之下，写代码的人更重要。

### （3）Anaconda 环境集成平台

Anaconda 是一款巨大的 Python 环境集成平台，这是我们必须下载的工具。其内含 Python 解释器、Jupyter Notebook 代码编辑器以及很多三方库。



## 一、安装 Anaconda

据悉，Anaconda 的 2023.3 版本并不稳定，因此本教程使用 2022.10 版本。如果你的 Anaconda 不是 2022.10 版本，可以跟 UP 一起卸载掉。

经实践，低版本的 Anaconda 仍然可以安装较新版本的三方库（这是因为在虚拟环境里安装第三方库时会连接网络），因此 Anaconda 2022.10 永不过时。

### 1.1 卸载 Anaconda（可选）

首先，请先进入 B 站视频置顶评论中的百度网盘链接，下载的 Everything 小工具（一个 2M 左右无需安装的 exe 文件），Everything 可以根据文件名，秒搜计算机中任何位置的任何文件，如图 1-1 所示，启动后进入状态需要等待约 20 秒。

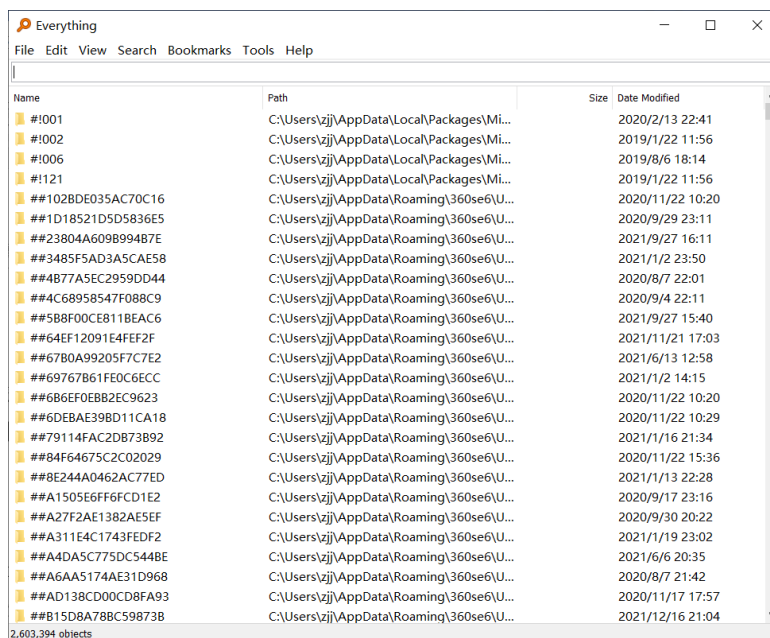


图 1-1 启动 Everything，并进入状态

在 Everything 中搜索“Uninstall-Anaconda”，如图 1-2 所示。

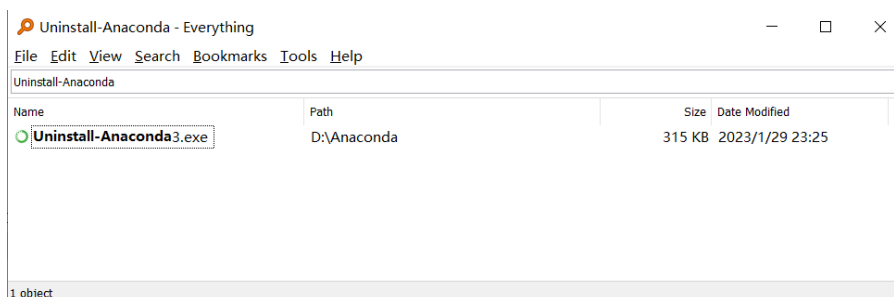


图 1-2 用 Everything 找到 Anaconda 的卸载程序

可直接双击图 1-2 中的卸载程序，对 Anaconda 进行卸载。

卸载 Anaconda 后，用 Everything.exe 查找并删除残存在 C 盘里关于 .condarc、jupyter、ipython、continuum、matplotlib、anaconda 以及 conda 的文件。

## 1.2 安装 Anaconda

### (1) 下载 Anaconda

由于官方服务器在国外，我们用的话很慢，因此去中国大学的镜像源下载。镜像源地址为 <https://mirrors.bfsu.edu.cn/anaconda/archive/>，下载 2022.10-Win 版本，如图 1-3 所示，其基础环境（base 环境）下的 Python 为 3.9 版本。

| Index of /                           |        |                     |
|--------------------------------------|--------|---------------------|
| Filename                             | Size   | Last Modified       |
| .winzip/                             | -      |                     |
| Anaconda3-2022.10-Linux-aarch64.sh   | 534.5M | 2022-10-17 16:15:40 |
| Anaconda3-2022.10-Windows-x86_64.exe | 621.2M | 2022-10-17 16:15:39 |
| Anaconda3-2022.10-Linux-x86_64.sh    | 737.6M | 2022-10-17 16:15:39 |
| Anaconda3-2022.10-MacOSX-x86_64.pkg  | 688.6M | 2022-10-17 16:15:38 |
| Anaconda3-2022.10-MacOSX-arm64.sh    | 472.5M | 2022-10-17 16:15:38 |
| Anaconda3-2022.10-MacOSX-x86_64.sh   | 681.6M | 2022-10-17 16:15:37 |
| Anaconda3-2022.10-Linux-s390x.sh     | 282.4M | 2022-10-17 16:15:37 |
| Anaconda3-2022.10-Linux-ppc64le.sh   | 360.0M | 2022-10-17 16:15:37 |
| Anaconda3-2022.10-MacOSX-arm64.pkg   | 484.1M | 2022-10-17 16:15:36 |
| Anaconda3-2022.05-MacOSX-arm64.sh    | 304.8M | 2022-06-07 12:40:25 |
| Anaconda3-2022.05-MacOSX-arm64.pkg   | 316.4M | 2022-06-07 12:40:24 |

图 1-3 下载 Anaconda 安装包

考虑到后面会用虚拟环境，创建虚拟环境时可以设置此环境中的 Python 解释器版本，所以这里下载哪一版 Anaconda 并不重要。

### (2) 安装 Anaconda

双击刚刚下载的 exe 文件，会有三个分岔口，分别按下列规则选择。

- ① Just me 和 All Users，选择 Just me；
- ② 安装路径选择最大的盘（一般是 D 盘），放在新建的【D:\Anaconda】里；

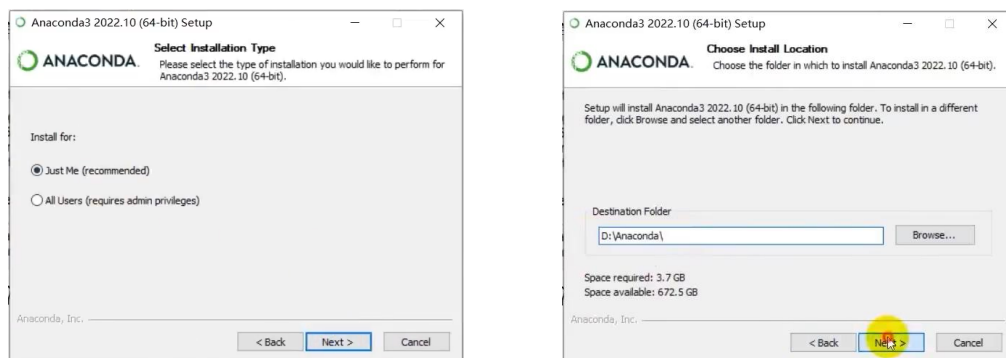


图 1-4 前两个分岔口

- ③ 最后一个分岔口，不勾选第一个方框，按照图 1-5 所示选择。

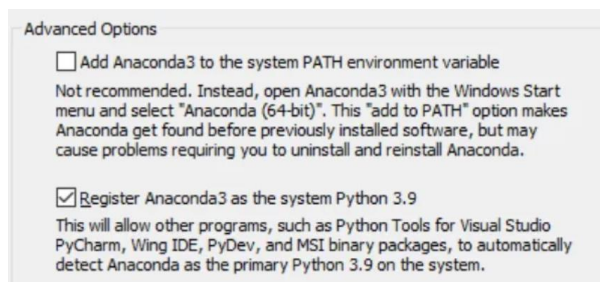


图 1-5 第三个分岔口

安装过程十分漫长，进度条会停在约 90%的位置共 20 分钟，请耐心等待。

### 1.3 配置环境变量

桌面按下鼠标右键，点击“显示设置”，如图 1-6 所示。



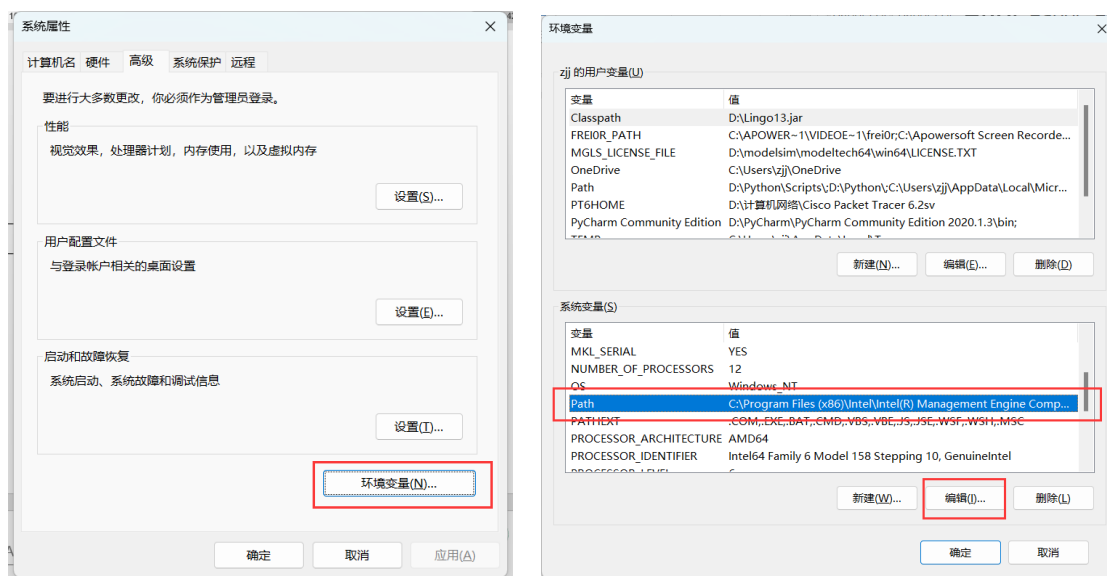
图 1-6 显示设置

左上角“查找设置”中输入“环境变量”，点击“编辑系统环境变量”。



图 1-7 点击“编辑系统环境变量”

在弹出的“系统属性”窗口中点击“环境变量”，再在弹出的“环境变量”窗口中选中 Path 路径，并点击编辑。



(a) “系统属性”窗口

(b) “环境变量”窗口

图 1-8 打开“编辑系统环境变量”



通过右侧的“新建”按钮，可新建环境变量的路径，将【D:\Anaconda】、【D:\Anaconda\Scripts】与【D:\Anaconda\Library\bin】添加到环境变量。

若您的 Anaconda 安装路径不是 D:\Anaconda，而是 E:\Anaconda，以上三个环境变量需要对应地进行更改，即改为【E:\Anaconda】、【E:\Anaconda\Scripts】与【E:\Anaconda\Library\bin】。

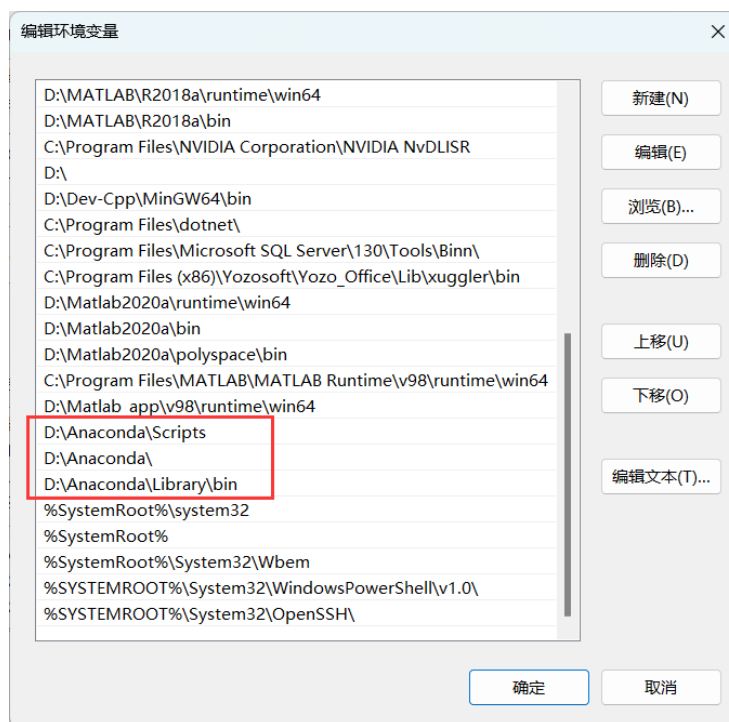


图 1-9 添加环境变量

请注意，据悉，若此前您为其它单独的 Python 解释器添加过环境变量，请在删除它的环境变量，否则 Anaconda 的环境变量会被挤出来（即点击确定后，再点进来会自动消失）。



## 1.4 桌面快捷方式

先找到 Jupyter 的安装路径，win10 和 win11 的方法如图 1-10 所示。



图 1-10 找到 Jupyter 的位置

找到 Jupyter 的位置后，把 Jupyter 和 Prompt 复制到桌面，如图 1-11 所示。

| 名称                                    | 修改日期           | 类型   | 大小   |
|---------------------------------------|----------------|------|------|
| Anaconda Navigator (Anaconda)         | 2023/2/1 10:36 | 快捷方式 | 1 KB |
| Anaconda Powershell Prompt (Anaconda) | 2023/2/1 10:36 | 快捷方式 | 3 KB |
| Anaconda Prompt (Anaconda)            | 2023/2/1 10:36 | 快捷方式 | 2 KB |
| Jupyter Notebook (Anaconda)           | 2023/2/1 10:36 | 快捷方式 | 1 KB |
| Reset Spyder Settings (Anaconda)      | 2023/2/1 10:36 | 快捷方式 | 1 KB |

图 1-11 直接把快捷方式复制到桌面

对于 UP 的 2022.10 版本来说，其地址为：

C:\Users\zjj\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Anaconda3 (64-bit)



## 二、安装 PyTorch（GPU 版）库

很多时候我们需要复刻其它演示代码中的环境，因此，虚拟环境必须掌握。初始的虚拟环境基本没什么库，演示代码里说需要安装什么版本的库，我们就手动安装什么版本的库。虚拟环境还有一个更大的优点，即想创建多少个，就创建多少。这样一来，就可以在同一台计算机中复刻多个不同的环境。

### 2.1 创建虚拟环境

点击 Prompt 进入 Anaconda 的环境中，接下来的命令均在 Prompt 中执行。

#### (1) 清屏

```
# 清屏  
cls
```

#### (2) base 环境下的操作

```
# 列出所有的环境  
conda env list
```

```
# 创建名为“环境名”的虚拟环境，并指定 Python 的版本  
conda create -n 环境名 python=3.9
```

```
# 创建名为“环境名”的虚拟环境，并指定 Python 的版本与安装路径  
conda create --prefix=安装路径\环境名 python=3.9
```

```
# 删除名为“环境名”的虚拟环境  
conda remove -n 环境名 --all
```

```
# 进入名为“环境名”的虚拟环境  
conda activate 环境名
```

#### (3) 虚拟环境内的操作

```
# 列出当前环境下的所有库  
conda list
```

```
# 安装 NumPy 库，并指定版本 1.21.5  
pip install numpy==1.21.5 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# 安装 Pandas 库，并指定版本 1.2.4  
pip install Pandas==1.2.4 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# 安装 Matplotlib 库，并指定版本 3.5.1  
pip install Matplotlib==3.5.1 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# 查看当前环境下某个库的版本（以 numpy 为例）  
pip show numpy
```

```
# 退出虚拟环境  
conda deactivate
```

以上演示了三个库的安装，您可以根据规律去安装其它库（PyTorch 除外）。



## 2.2 GPU 基础

对计算机而言，中央处理器 CPU 是主板上的芯片，图形处理器 GPU 是显卡上的芯片。每台计算机必有主板，但少数计算机可能没有显卡。显卡可以用来加速深度学习的运算速度（GPU 比 CPU 快 10-100 倍）。

目前显卡有三家厂商：Intel、Nvidia、AMD。深度学习所需要的显卡是 NVIDIA，没有此显卡的同学无法进行 GPU 加速。查看自己的计算机的显卡的方法是：任务管理器——性能——左侧栏划到最下面。



图 5-1 查看显卡

NVIDIA 显卡中的运算平台是 CUDA，不过，即使您的计算机有 NVIDIA 显卡，但您的显卡中也不一定含有 CUDA，没有的话就要下载 CUDA。

而 PyTorch 的下载组件里也会包含一个内置的 cuda。

为了区分，显卡内的 CUDA 用大写，PyTorch 内置的 cuda 用小写。

一般来讲，要满足：CUDA 版本  $\geq$  cuda 版本。

查看 CUDA 版本的方法是：Win+R 后输入 cmd，进入命令提示符，我们需要输入 nvcc -V，如图 5-2。

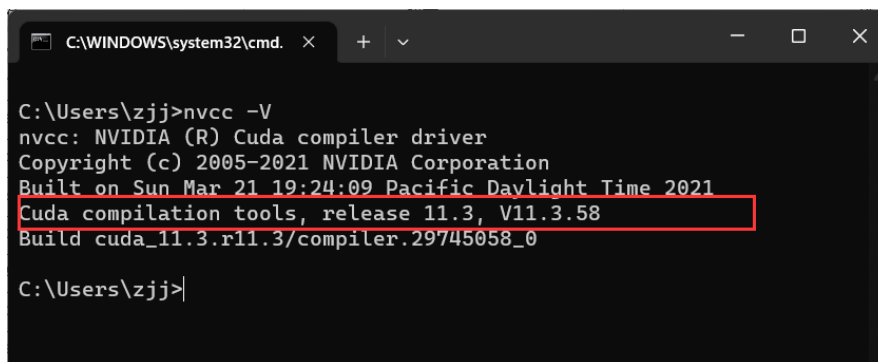


图 5-2 显卡里的 CUDA 版本

**如果显示“nvcc -V 不是内部或外部命令”，则说明需要安装 CUDA。**

后面我们将安装 torch 1.12.0 版本，其可选的内置 cuda 版本是 11.3。因此，如果您显卡里的 CUDA 低于了 11.3，需要进行升级。

**注：第 12 页有各 torch 内置 cuda 的版本，据此选择 CUDA 版本进行安装。**



## 2.3 安装 CUDA（可选）

CUDA 的下载链接：<https://developer.nvidia.com/cuda-toolkit-archive>，以其中的 CUDA 11.3 为例，点击进入，如图 5-3 所示。

[CUDA Toolkit 11.4.4 \(February 2022\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.3 \(November 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.2 \(September 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.1 \(August 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.0 \(June 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.3.1 \(May 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.3.0 \(April 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.2.2 \(March 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.2.1 \(February 2021\), Versioned Online Documentation](#)  
[CUDA Toolkit 11.2.0 \(December 2020\), Versioned Online Documentation](#)

图 5-3 进入 CUDA 11.3 界面

接下来要选择平台，点击 Windows，之后自动弹出更多内容，按图 5-4 选择，最后点击右下角的 Download（2.7GB），建议将其放置新建的 D:\CUDA 中。

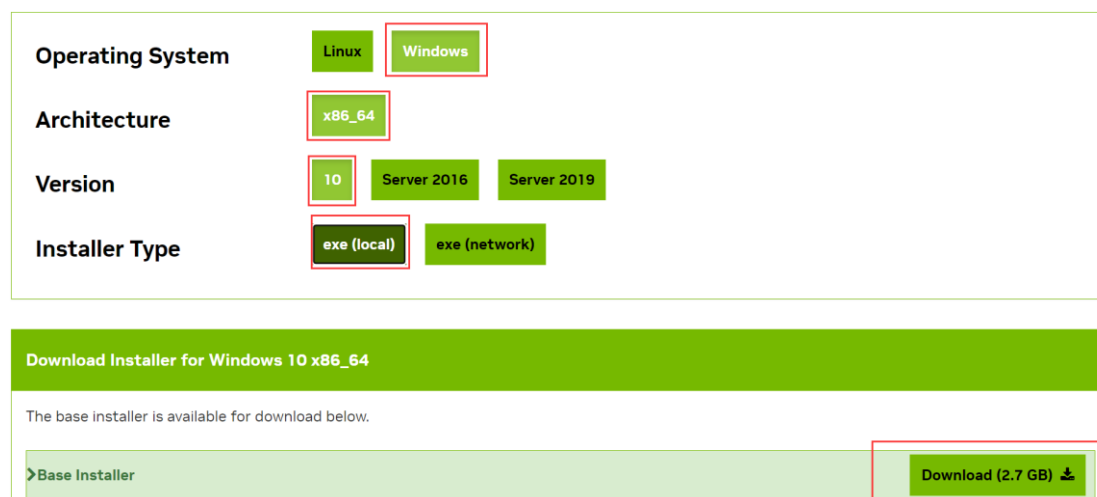


图 5-4 下载 CUDA

下载好之后，将 exe 文件放置在新建的 D:\CUDA 内，点击 exe 文件，大约要等两分钟，会弹出如图 5-5 的提示框，这里要选择临时的解压文件夹，考虑到解压后需要占用大约 7G 的内存，因此建议放在 D:\CUDA\Tem 内，安装结束后，该临时解压文件夹会自动删除。

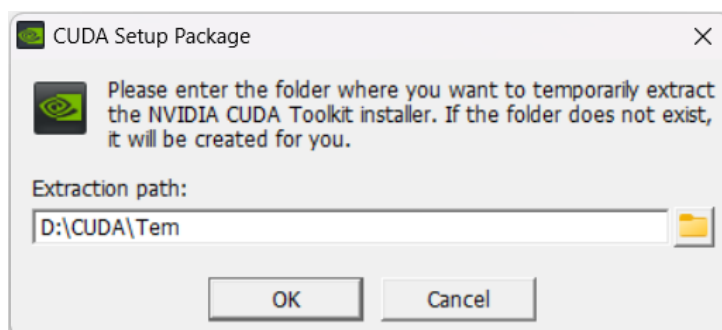


图 5-5 临时解压

解压好后，进入如图 5-6 的安装界面，同意并继续后，点击“自定义”。



(a) 同意并继续



(b) “自定义”安装选项

图 5-6 临时解压

接下来，仅仅选择 4 大项中的 CUDA，并取消 CUDA 中关于 VS 的选项。



(a) 仅仅选择 4 大项中的 CUDA



(b) 取消 CUDA 中关于 VS 的选项

图 5-7 临时解压

完成后，按照默认的 C 盘路径进行安装（大约 7G）即可，结束后如图 5-8。



图 5-8 安装结束



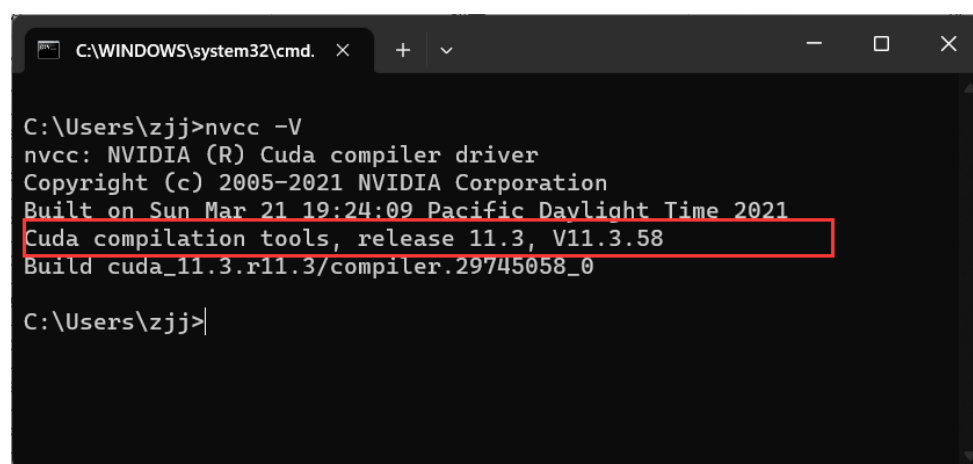
现在回头查看临时解压文件夹 D:\CUDA\Tem，会发现已经消失，顺便可以删除 D:\CUDA 了。

接下来配置环境变量（参考 1.3 小节的打开方式），如果你是按照默认路径的话，其路径应该是：

- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3\lib\x64
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3\bin
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3\libnvvp

如果你忘了你的路径，用 everything 搜索出来即可。

最后，回头检查一下 CUDA 版本，Win+R 后输入 cmd，进入命令提示符，输入 nvcc -V，如图 5-9 所示。



```
C:\WINDOWS\system32\cmd. x + v
C:\Users\zjj>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Sun Mar 21 19:24:09 Pacific Daylight Time 2021
Cuda compilation tools, release 11.3, V11.3.58
Build cuda_11.3.r11.3/compiler.29745058_0

C:\Users\zjj>|
```

图 5-9 检查 CUDA 版本

注意，尽管您输入 nvidia-smi 可能仍然显示老版本，但经查阅资料，CUDA 的实际使用版本以 nvcc -V 中的 CUDA 版本为准。

## 2.4 安装 PyTorch

PyTorch 一分为三：torch、torchvision 与 torchaudio。这三个库中，torch 有 2G 左右，而 torchvision 和 torchaudio 只有 2M 左右，因此一般在代码里只会 import torch。当 torch 的版本给定后，另外两个附件的版本也唯一确定了。

安装 torch 前，先给出一张安装表，其中 cu113 即 cuda 11.3，cp39 即 Python 解释器的版本是 Python3.9。注：NVIDIA 显卡 30 系列（如 NVIDIA GeForce RTX 3050）只能安装 cu110 及其以后的版本。

表 5-1 目前所有 torch 版本的安装表

| torch 版本     | 可选的 cuda 版本       | 支持的 Python 版本         |
|--------------|-------------------|-----------------------|
| torch 2.0.1  | cu117、cu118       | cp38、cp39、cp310、cp311 |
| torch 2.0.0  | cu117、cu118       | cp38、cp39、cp310、cp311 |
| torch 1.13.1 | cu116、cu117       | cp37、cp38、cp39、cp310  |
| torch 1.13.0 | cu116、cu117       | cp37、cp38、cp39、cp310  |
| torch 1.12.1 | cu113、cu116       | cp37、cp38、cp39、cp310  |
| torch 1.12.0 | cu113、cu116       | cp37、cp38、cp39、cp310  |
| torch 1.11.0 | cu113、cu115       | cp37、cp38、cp39、cp310  |
| torch 1.10.2 | cu102、cu111、cu113 | cp36、cp37、cp38、cp39   |
| torch 1.10.1 | cu102、cu111、cu113 | cp36、cp37、cp38、cp39   |
| torch 1.10.0 | cu102、cu111、cu113 | cp36、cp37、cp38、cp39   |
| torch 1.9.1  | cu102、cu111       | cp36、cp37、cp38、cp39   |
| torch 1.9.0  | cu102、cu111       | cp36、cp37、cp38、cp39   |
| torch 1.8.1  | cu101、cu102、cu111 | cp36、cp37、cp38、cp39   |
| torch 1.8.0  | cu101、cu111       | cp36、cp37、cp38、cp39   |
| torch 1.7.1  | cu101、cu110       | cp36、cp37、cp38、cp39   |
| torch 1.7.0  | cu101、cu110       | cp36、cp37、cp38        |
| torch 1.6.0  | cu101             | cp36、cp37、cp38        |
| torch 1.5.1  | cu92、cu101        | cp35、cp36、cp37、cp38   |
| torch 1.5.0  | cu92、cu101        | cp35、cp36、cp37、cp38   |
| torch 1.4.0  | cu92              | cp35、cp36、cp37、cp38   |
| torch 1.3.1  | cu92              | cp35、cp36、cp37        |
| torch 1.3.0  | cu92              | cp35、cp36、cp37        |
| torch 1.2.0  | cu92              | cp35、cp36、cp37        |

现在，按照李沐老师《动手学深度学习(PyTorch 版)》教程，安装 torch 1.12.0。

根据表 5-1，torch 1.12.0 支持的 cuda 是 11.3 或 11.6，任意选一个即可；其支持的 Python 是 3.7 - 3.10，刚刚新建的虚拟环境的 Python 是 3.9，满足条件。

进入 PyTorch 官网：<https://pytorch.org/get-started/previous-versions/>，在其中 Ctrl + F 搜索【 pip install torch==1.12.0 】，如图 5-10，请注意，这里使用 pip 安装，而不是 conda 安装（如果用 conda 安装，最后检验 cuda 时是不可用的）。



Linux and Windows

```
# ROCm 5.1.1 (Linux only)
pip install torch==1.12.0+rocm5.1.1 torchvision==0.13.0+rocm5.1.1 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/rocm5.1.1
# CUDA 11.6
pip install torch==1.12.0+cu116 torchvision==0.13.0+cu116 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cu116
# CUDA 11.3
pip install torch==1.12.0+cu113 torchvision==0.13.0+cu113 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cu113
# CUDA 10.2
pip install torch==1.12.0+cu102 torchvision==0.13.0+cu102 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cu102
# CPU only
pip install torch==1.12.0+cpu torchvision==0.13.0+cpu torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cpu
```

图 5-10 搜索结果

cu113 和 cu116 都可以，我们就选 cu113 吧，接下来安装有两种办法。

### (1) 方法一：直接安装（不建议，网差的话会死机）

复制网页里的那段代码，也即

```
pip install torch==1.12.0+cu113 torchvision==0.13.0+cu113 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cu113
```

双击 Prompt，进入 DL 环境下运行（不要在 base 环境下运行），如图 5-11。

```
(base) C:\Users\zjj>conda activate DL
(DL) C:\Users\zjj>pip install torch==1.12.0+cu113 torchvision==0.13.0+cu113 torchaudio==0.12.0 --extra-index-url https://download.pytorch.org/whl/cu113
Looking in indexes: https://pypi.org/simple, https://download.pytorch.org/whl/cu113
Collecting torch==1.12.0+cu113
  Downloading https://download.pytorch.org/whl/cu113/torch-1.12.0%2Bcu113-cp39-cp39-win_amd64.whl (2143.4 MB)
    2.1/2.1 GB ? eta 0:00:00
Collecting torchvision==0.13.0+cu113
  Downloading https://download.pytorch.org/whl/cu113/torchvision-0.13.0%2Bcu113-cp39-cp39-win_amd64.whl (4.7 MB)
    4.7/4.7 MB 8.0 MB/s eta 0:00:00
Collecting torchaudio==0.12.0
  Downloading https://download.pytorch.org/whl/cu113/torchaudio-0.12.0%2Bcu113-cp39-cp39-win_amd64.whl (1.2 MB)
    1.2/1.2 MB 3.2 MB/s eta 0:00:00
Collecting typing-extensions
  Using cached typing_extensions-4.5.0-py3-none-any.whl (27 kB)
Collecting pillow!=8.3.*,>=5.3.0
  Using cached Pillow-9.4.0-cp39-cp39-win_amd64.whl (2.5 MB)
Collecting numpy
  Using cached numpy-1.24.2-cp39-cp39-win_amd64.whl (14.9 MB)
Collecting requests
  Using cached requests-2.28.2-py3-none-any.whl (62 kB)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda\envs\dl\lib\site-packages (from requests->torchvision==0.13.0+cu113) (2022.12.7)
Collecting idna<4,>=2.5
  Using cached https://download.pytorch.org/whl/idna-3.4-py3-none-any.whl (61 kB)
Collecting urllib3<1.27,>=1.21.1
  Using cached urllib3-1.26.15-py2.py3-none-any.whl (140 kB)
Collecting charset-normalizer<4,>=2
  Using cached charset-normalizer-3.1.0-cp39-cp39-win_amd64.whl (97 kB)
Installing collected packages: urllib3, typing-extensions, pillow, numpy, idna, charset-normalizer, torch, requests, torchvision, torchaudio
Successfully installed charset-normalizer-3.1.0 idna-3.4 numpy-1.24.2 pillow-9.4.0 requests-2.28.2 torch-1.12.0+cu113 torchaudio-0.12.0+cu113 torchvision-0.13.0+cu113 typing-extensions-4.5.0 urllib3-1.26.15
(DL) C:\Users\zjj>
```

图 5-11 虚拟环境下安装 torch

看到最后几行代码里有 Successfully installed 就算成功。

安装命令的意思是，使用 pip 安装三个库，第一库是 torch==1.12.0+cu113，第二个库是 torchvision==0.13.0+cu113，第三个库是 torchaudio==0.12.0，库的下载地址是 https://download.pytorch.org/whl/cu113。

由于中国大学镜像源里没有 GPU 版本的 PyTorch，因此无法像之前安装 NumPy、Pandas 和 Matplotlib 一样方便，只能进入上述那个国外的网站下载。



## (2) 方法二：先下轮子再安装

首先，我们进入方法一提及的网站 <https://download.pytorch.org/whl/cu113>，如图 5-12 所示，进入 torch、torchvision、torchaudio 三大组件各自的网站。

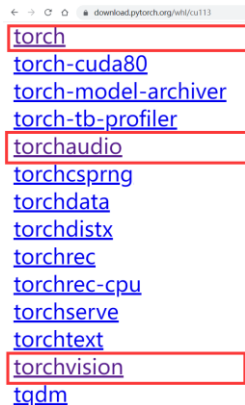


图 5-12 下载 PyTorch 组件的网站，进入三个组件的网站

回忆方法一的安装命令，第一个库是 `torch==1.12.0+cu113`，第二个库是 `torchvision==0.13.0+cu113`，第三个库是 `torchaudio==0.12.0`。

我们在这三个组件各自的网站里 Ctrl + F 搜索，如图 5-13 所示。

|   |   |
|---|---|
| <p>torch-1.11.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torch-1.11.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torch-1.12.0+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torch-1.12.0+cu113-cp310-cp310-win_amd64.whl<br/>                 torch-1.12.0+cu113-cp37-cp37m-linux_x86_64.whl<br/>                 torch-1.12.0+cu113-cp37-cp37m-win_amd64.whl<br/>                 torch-1.12.0+cu113-cp38-cp38-linux_x86_64.whl<br/>                 torch-1.12.0+cu113-cp38-cp38-win_amd64.whl<br/>                 torch-1.12.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torch-1.12.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torch-1.12.1+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torch-1.12.1+cu113-cp310-cp310-win_amd64.whl</p>   |   |
| (a) torch 网站里搜索   |   |
| <p>torchvision-0.12.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torchvision-0.12.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torchvision-0.13.0+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torchvision-0.13.0+cu113-cp310-cp310-win_amd64.whl<br/>                 torchvision-0.13.0+cu113-cp37-cp37m-linux_x86_64.whl<br/>                 torchvision-0.13.0+cu113-cp37-cp37m-win_amd64.whl<br/>                 torchvision-0.13.0+cu113-cp38-cp38-linux_x86_64.whl<br/>                 torchvision-0.13.0+cu113-cp38-cp38-win_amd64.whl<br/>                 torchvision-0.13.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torchvision-0.13.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torchvision-0.13.1+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torchvision-0.13.1+cu113-cp310-cp310-win_amd64.whl</p> | <p>torchaudio-0.11.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torchaudio-0.11.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torchaudio-0.12.0+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torchaudio-0.12.0+cu113-cp310-cp310-win_amd64.whl<br/>                 torchaudio-0.12.0+cu113-cp37-cp37m-linux_x86_64.whl<br/>                 torchaudio-0.12.0+cu113-cp37-cp37m-win_amd64.whl<br/>                 torchaudio-0.12.0+cu113-cp38-cp38-linux_x86_64.whl<br/>                 torchaudio-0.12.0+cu113-cp38-cp38-win_amd64.whl<br/>                 torchaudio-0.12.0+cu113-cp39-cp39-linux_x86_64.whl<br/>                 torchaudio-0.12.0+cu113-cp39-cp39-win_amd64.whl<br/>                 torchaudio-0.12.1+cu113-cp310-cp310-linux_x86_64.whl<br/>                 torchaudio-0.12.1+cu113-cp310-cp310-win_amd64.whl</p> |
| (b) torchvision 网站里搜索   | (c) torchaudio 网站里搜索  |

图 5-13 分别在三个组件的网站中搜索并下载

下载好后，将三个 whl 文件放在新建的 D:\whl 文件夹中。

安装命令为 `pip install 路径\轮子名.whl`，即

```
pip install D:\whl\torch-1.12.0+cu113-cp39-cp39-win_amd64.whl
pip install D:\whl\torchvision-0.13.0+cu113-cp39-cp39-win_amd64.whl
pip install D:\whl\torchaudio-0.12.0+cu113-cp39-cp39-win_amd64.whl
```

将上述代码放在虚拟环境 DL 下执行（而不是 base 环境，否则就会把这个库安装进 base 环境下）。

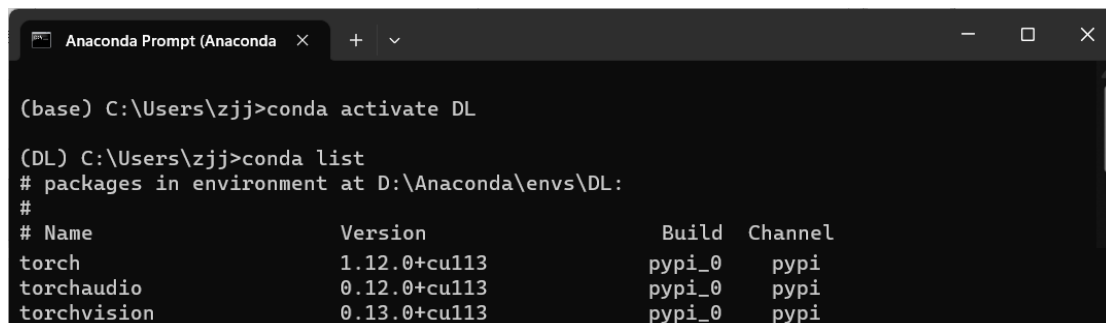
安装完毕后，即可删除 D:\whl 文件夹（但建议留着，之后可能还要安装）。



## 2.5 检验 cuda 是否可用

### (1) 方法一：查看当前环境的所有库

进入 DL 环境后，使用 `conda list` 命令列出当前的所有库，如图 5-14。



```
(base) C:\Users\zjj>conda activate DL
(DL) C:\Users\zjj>conda list
# packages in environment at D:\Anaconda\envs\DL:
#
# Name                      Version                Build      Channel
torch                       1.12.0+cu113           pypi_0    pypi
torchaudio                  0.12.0+cu113           pypi_0    pypi
torchvision                  0.13.0+cu113           pypi_0    pypi
```

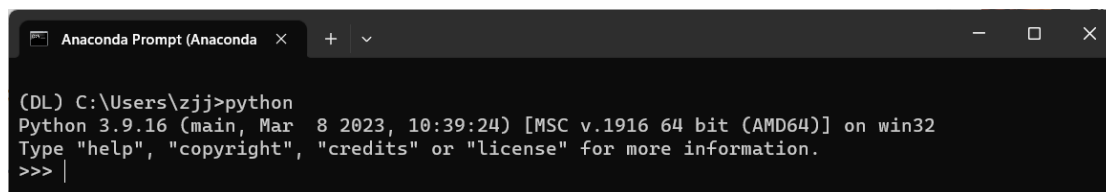
图 5-14 列出 DL 环境内的的所有库

图 5-14 仅仅展示了 torch 三大组件的版本，其版本后应该有“+cu113”。

### (2) 方法二：进入 Python 解释器检验

由于目前没有代码编辑器，因此直接进 Python 解释器，检验 cuda 是否可用。

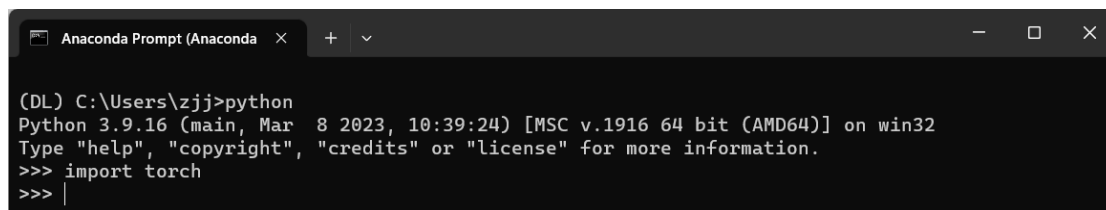
首先，进入虚拟环境 DL 后，输入 `python` 以进入解释器，如图 5-14。



```
(DL) C:\Users\zjj>python
Python 3.9.16 (main, Mar 8 2023, 10:39:24) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

图 5-14 在 DL 虚拟环境中进入 python 解释器

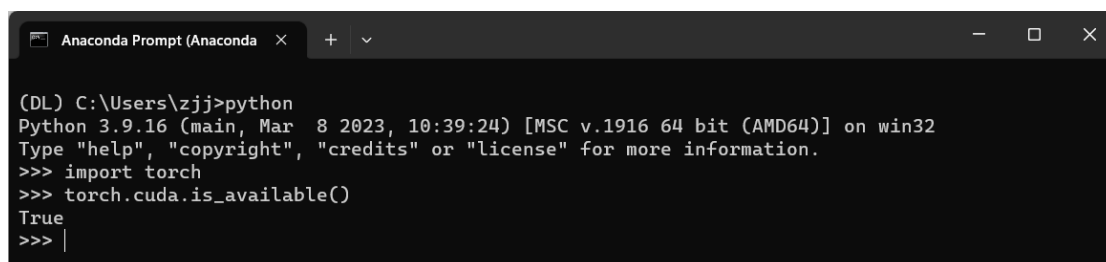
输入 `import torch` 导入 torch 库，如图 5-15 所示。若 torch 安装失败，则会返回 `No module named 'torch'`。若安装成功，不会返回任何语句，同时在下一行出现“>>>”，提示我们可以继续敲代码。



```
(DL) C:\Users\zjj>python
Python 3.9.16 (main, Mar 8 2023, 10:39:24) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> |
```

图 5-15 导入 torch

最后一步，输入 `torch.cuda.is_available()`，如图 5-16 所示。



```
(DL) C:\Users\zjj>python
Python 3.9.16 (main, Mar 8 2023, 10:39:24) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> |
```

图 5-16 测试 cuda 是否可用

根据 UP 测试，使用 `conda` 安装必是 `False`，使用 `pip` 安装方能为 `True`。

除了上述两个方法，还有两种方法，具体见 3.3 小节和 4.4 小节。

## 三、Jupyter 代码编辑器

计算机用户名（即 C:\Users\用户名）为中文，无法兼容 Jupyter。大家可以打开 Prompt 检查自己的用户名，如图 3-1。

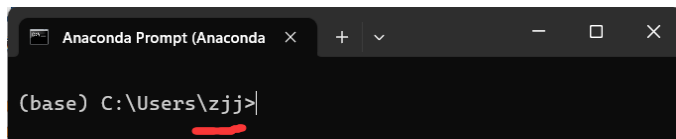


图 3-1 查看自己的用户名

如果发现自己的用户名携带有中文，除了更改自己的用户名为英文数字组合外，也可以直接跳过本章，使用第四章的 PyCharm 代码编辑器即可。

### 3.1 修改工作路径（可选）

Jupyter 初始的工作路径为【C:\Users\用户名】，需要进行修正，将其转移到新建的【D:\Jupyter】位置。

- ① 新建 D:\Jupyter;
- ② 打开桌面快捷方式中的 Prompt;
- ③ 输入 `jupyter notebook --generate-config` 命令并执行;
- ④ 打开上一步生成的配置文件地址，即

C:\Users\用户名\.jupyter

⑤ 在 `jupyter_notebook_config.py`（以记事本方式打开）中使用 `Ctrl+F` 查找并且修改如下配置项：

修改前：`# c.NotebookApp.notebook_dir = "`

修改后：`c.NotebookApp.notebook_dir = 'D:\Jupyter'`

也即删除前面的#号注释，在后面的单引号里输入要设置的目录路径，注意，'D:\Jupyter' 中不能有空格，否则 Jupyter 打开就闪退。保存后关闭。

⑥ 找到桌面的 `jupyter notebook` 快捷图标，鼠标右键>>属性>>快捷方式>>目标，删除最后的`"%USERPROFILE%/"`。

### 3.2 修改默认字体（可选）

初始字体可以进行修改，修改流程如下。

- ① 打开如下地址

D:\Anaconda\Lib\site-packages\notebook\static\components\codemirror\lib

- ② 打开 `codemirror.css` 文件;

- ③ `Ctrl+F`，搜索“`font-family: monospace;`”的文字，并将其改为

`font-family: 'Fira Code Light','Consolas';`

### 3.3 虚拟环境连接 Jupyter

我们已经在 Anaconda 里创建了一个叫 DL 的虚拟环境，但是现在这个叫 DL 的虚拟环境没有连接 Jupyter，换句话说，Jupyter 现在仅仅能与 base 环境相连。

为让虚拟环境与 Jupyter 相连，请在 Prompt 的虚拟环境下操作下列命令。

```
# 安装 ipykernel
```

```
pip install ipykernel -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# 将虚拟环境导入 Jupyter 的 kernel 中
```

```
python -m ipykernel install --user --name=环境名
```

```
# 删除虚拟环境的 kernel 内核
```

```
jupyter kernelspec remove 环境名
```

在 Jupyter 里，切换到 DL 内核后，点击 New，新建一个 DL 内核的脚本。



图 3-2 新建一个 DL 内核的脚本

在 DL 内核的脚本下，输入两段代码后运行，如图 3-3 所示。

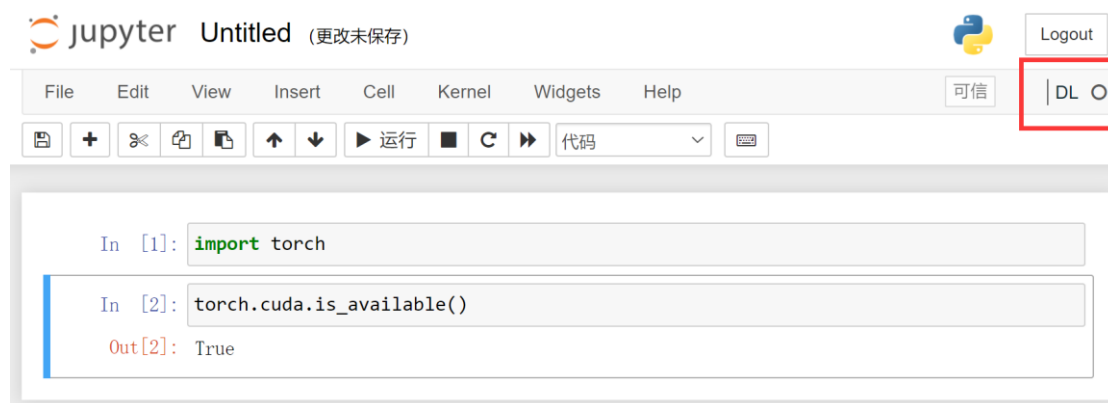


图 3-3 Jupyter 替代 Prompt



## 四、PyCharm 代码编辑器

本视频下载 PyCharm 2020.1.3-win 的社区版，如果您的版本不稳定，可卸载之（如果您的版本是专业版就别卸载了）。

### 4.1 卸载 PyCharm（可选）

找到 PyCharm 安装地址（UP 的安装地址为 D:\PyCharm），如果找不到，则在 PyCharm 快捷方式的属性中找，如图 4-1 所示。

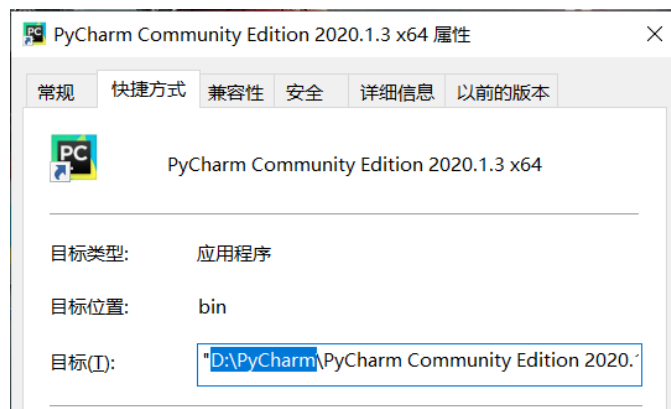


图 4-1 找到 PyCharm 的安装位置

在“PyCharm 安装地址\PyCharm Community Edition 2020.1.3\bin”中，找到 Uninstall.exe，如图 4-2 所示。

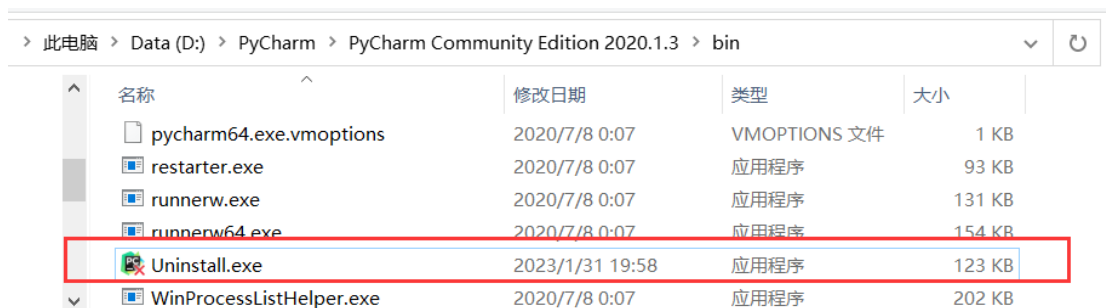


图 4-2 找到 PyCharm 的卸载程序

双击运行，勾选两个，点击确定，如图 4-3 所示。

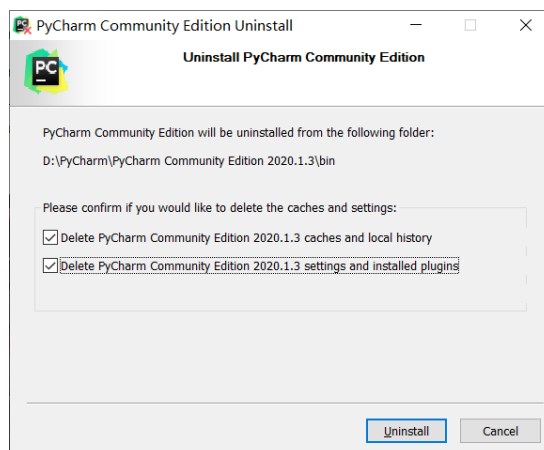


图 4-3 卸载 PyCharm

## 4.2 安装 PyCharm

首先，去 jetbrains 公司的官网下载 PyCharm，地址为  
<https://www.jetbrains.com/pycharm/download/other.html>

推荐下载社区版（足够个人使用）的 2020.1.3-win 版本，如图 4-4 所示。

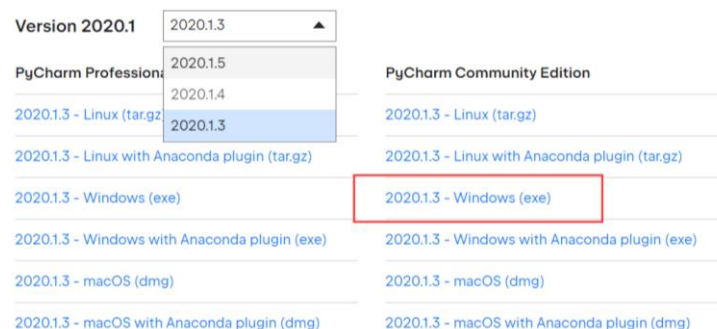


图 4-4 下载 PyCharm

安装时，请放在 D 盘的新建文件夹：D:\PyCharm 里。选好安装地址后，请勾选如图 4-5 所示。

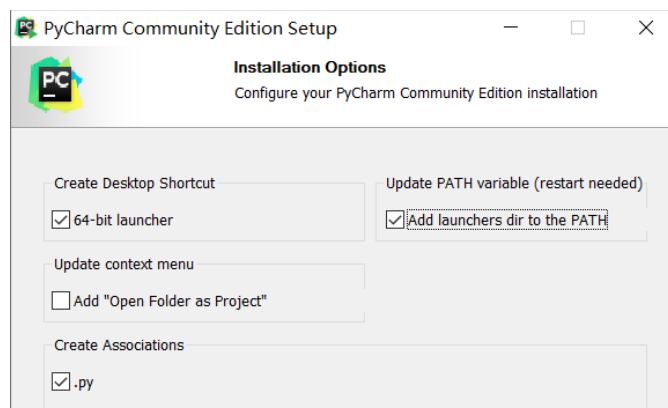


图 4-5 安装过程中唯一的分岔口

接着下一个窗口选择默认的 jetbrains 即可；最后一个窗口问你要不要重启，不重启好像也没啥事，如图 4-6 所示。

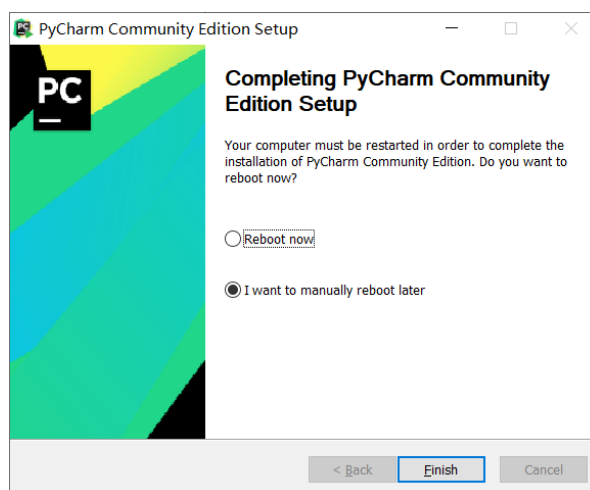


图 4-6 最后一个窗口

### 4.3 初次设置 PyCharm

安装好后，它会出现在桌面，双击进入。

第一个岔路，选择第二个，如图 4-7 所示。

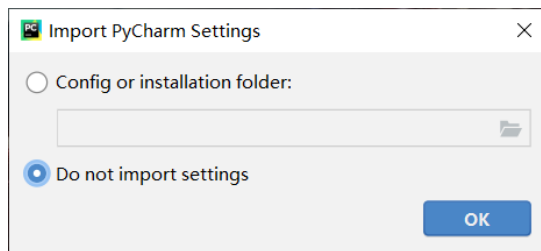


图 4-7 选择“不导入设置”

第二个岔路，选择主题，默认的黑色伤眼睛，UP 推荐亮色，如图 4-8 所示。

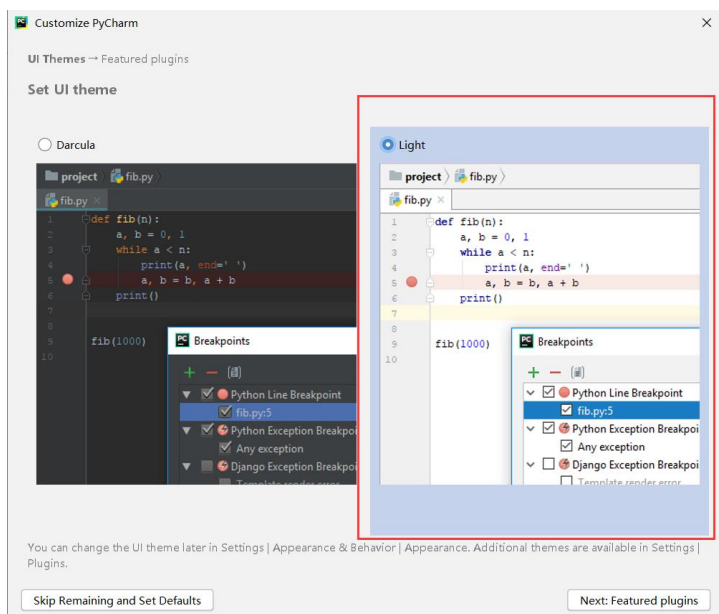


图 4-8 选择你喜欢的主题

最后一个配置，点左下角的按钮跳过即可，如图 4-9 所示。

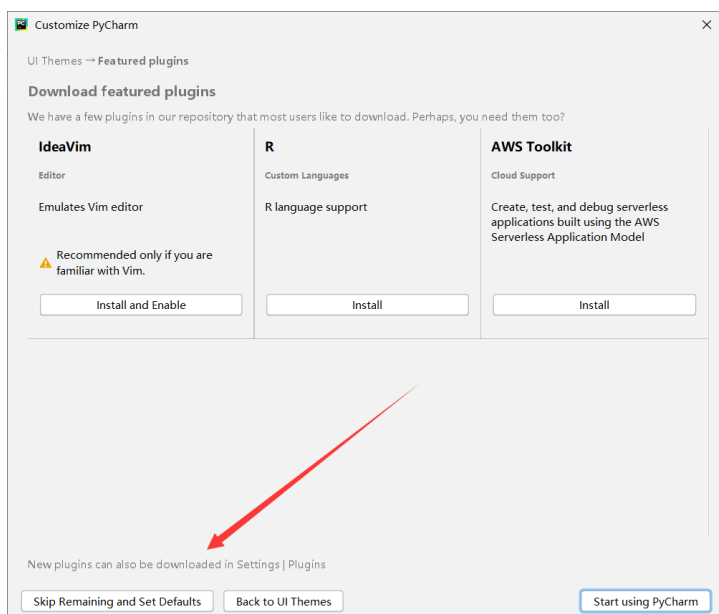


图 4-9 跳过最后一个配置

## 4.4 虚拟环境连接 PyCharm

完成第一次配置后，首先在 D:\PyCharm 中创建文件夹 Py\_Projects 存放工程。

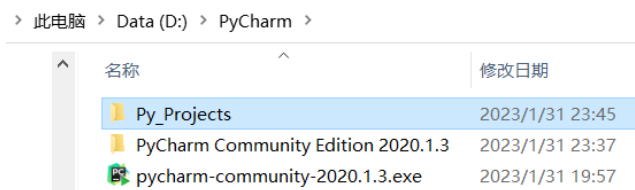


图 4-10 创建存放工程的文件夹

接着，在 PyCharm 里创建新工程，如图 4-11 所示。

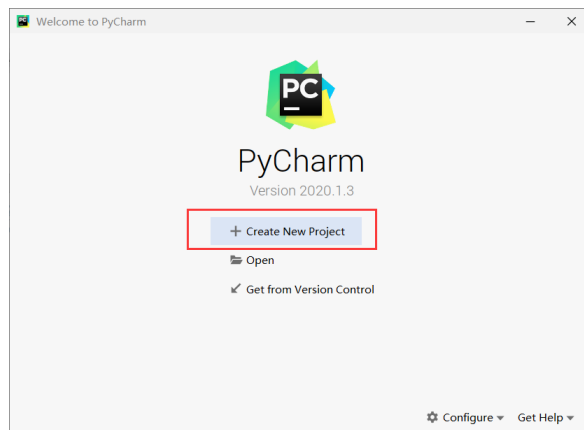


图 4-11 创建新工程

第一步，切换新工程的位置为刚刚创建的文件夹；第二步，点击“Existing interpreter”；第三步，由于是第一次进入 PyCharm，只能点击“...”来找解释器。

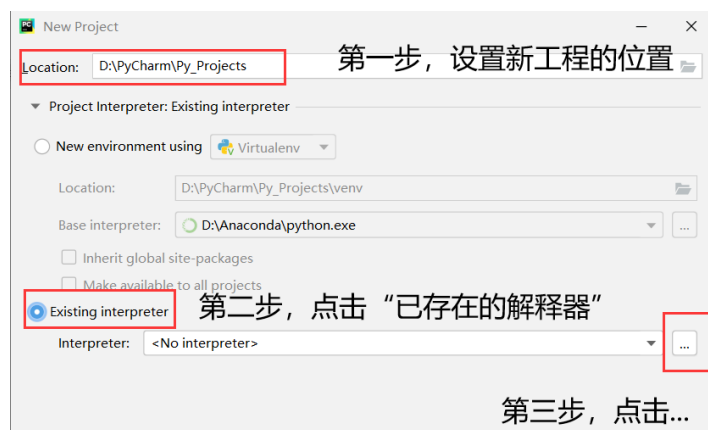


图 4-12 连接 Anaconda 前的准备

在弹出的界面中，首先点击左侧的 conda 环境，再将解释器设置为 Anaconda 虚拟环境解释器的地址，如图 4-13 所示。

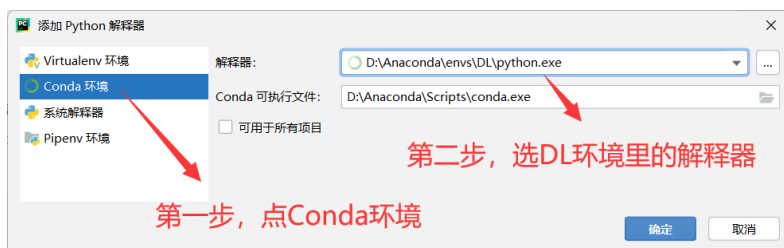


图 4-13 此工程将连接 Anaconda 的虚拟环境 DL



第一次进入工程页面，连接 Anaconda 需要很长一段时间（15 分钟是有的），点击右下角蓝色小圈圈可查看进度，如图 4-14 所示。

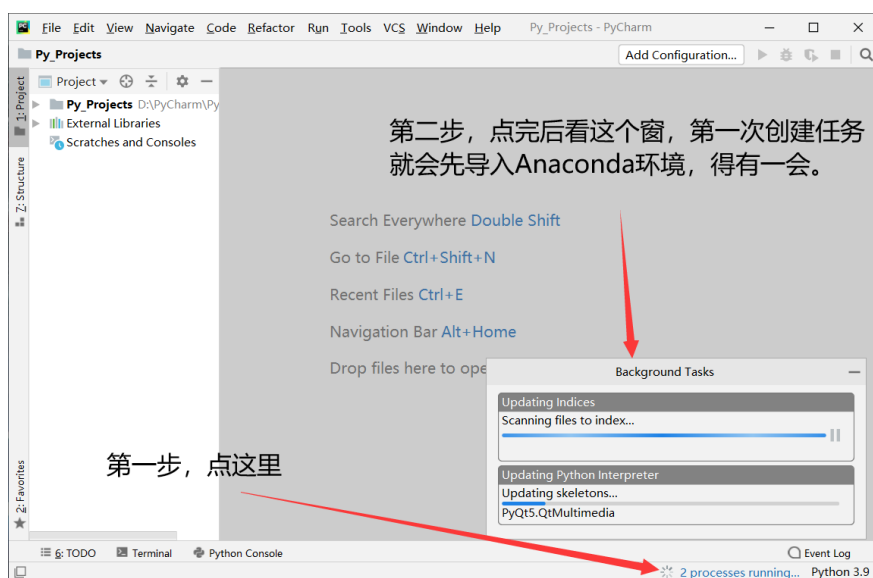


图 4-14 更新配置

更新好之后，我们可以把 PyCharm 设置为中文。File——Settings——Plugins——在搜索框输入 Chinese——安装第二个。

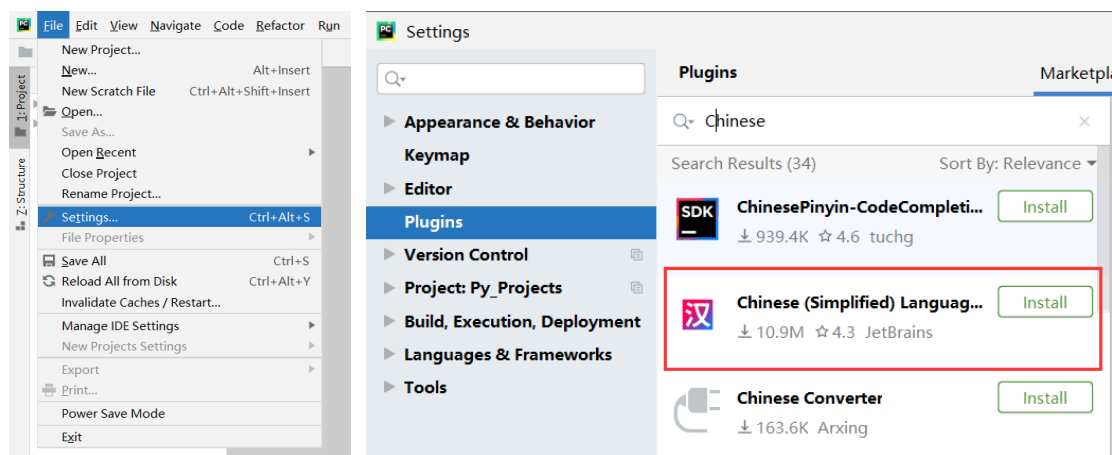


图 4-15 安装简体中文插件

下载好后，点击 Restart IDE，即重启 PyCharm，重启后，会发现软件已经变成了简体中文，但要第二次更新配置，但这一次更新速度会极快。

现在，在此工程里创建一个名为“test”的.py 文件，如图 4-16 所示。



图 4-16 创建.py 文件

先随便写行代码，运行一下 (Ctrl+Shift+F10)，再点击编辑配置。

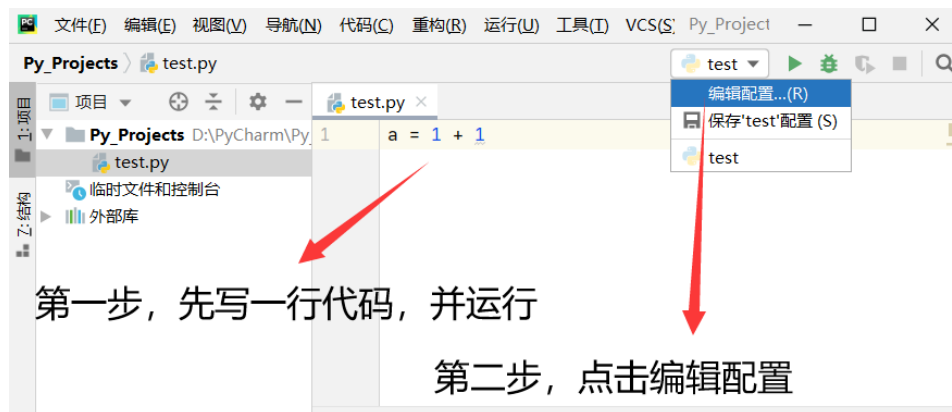


图 4-17 编辑配置

勾选“使用 Python 控制台运行”，并应用，如图 4-18 所示。

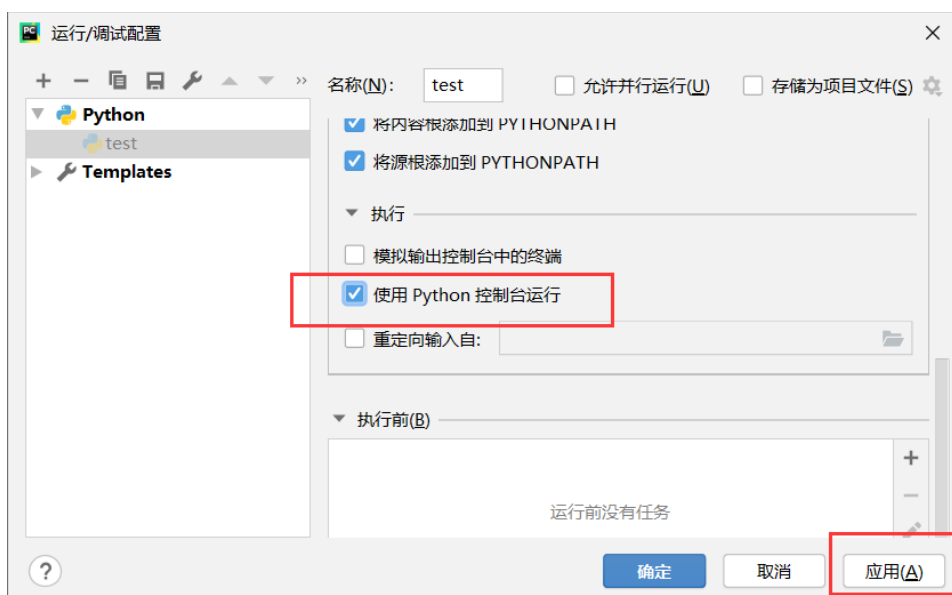


图 4-18 编辑配置

打开控制台，再运行.py 文件，控制台右侧即可显示每个变量的数值。



图 4-19 控制台

最后，检测 PyCharm 有无连接 GPU 版本的 PyTorch，如图 4-20 所示。

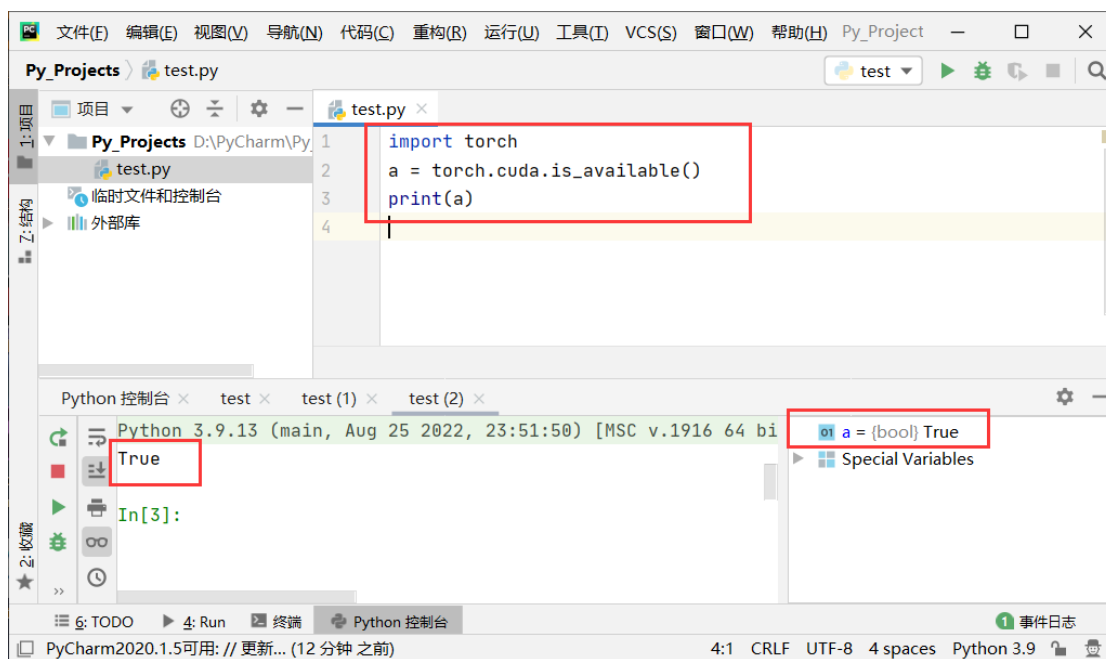


图 4-20 检测能否连接 CUDA