

## 一、深度学习基础环境配置

本实验基于 `python` 编程语言以及 `pytorch` 深度学习框架进行。实验首先需要完成本地基础环境配置,包括 `Anaconda` 环境配置、`python` 环境配置、`pytorch` 深度学习框架配置以及实验所需 `python` 依赖包配置。

### 1) `Anaconda` 环境配置

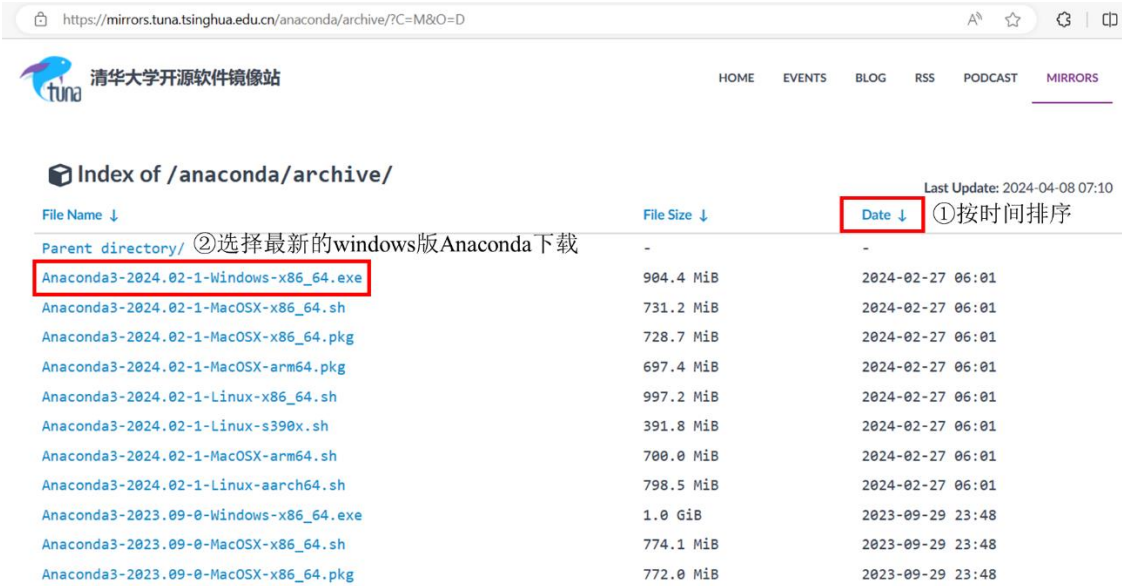
首先完成 `Anaconda` 环境配置。`Anaconda` 是一个安装、管理 `python` 相关包的软件,还自带 `Jupyter Notebook`、`Spyder` 以及管理包的 `conda` 工具。`Anaconda` 包含了 `conda`、`Python` 在内的超过 180 个科学包及其依赖项。

之所以需要在安装 `python` 前首先配置 `Anaconda` 环境,是因为 `Anaconda` 中的 `conda` 工具可以便捷的安装 `python` 依赖包,并高效管理不同的 `python` 开发环境,在后续其他实验/项目中,只需要使用不同的 `anaconda` 虚拟环境即可使用不同版本的 `python` 及依赖包。`Windows` 系统和 `Ubuntu` 系统下 `Anaconda` 的环境配置有所不同,下面将分别列出两种操作系统下的 `Anaconda` 环境配置方法,请按照自己的电脑情况选择对应配置方法。

#### a) `Windows Anaconda` 环境配置

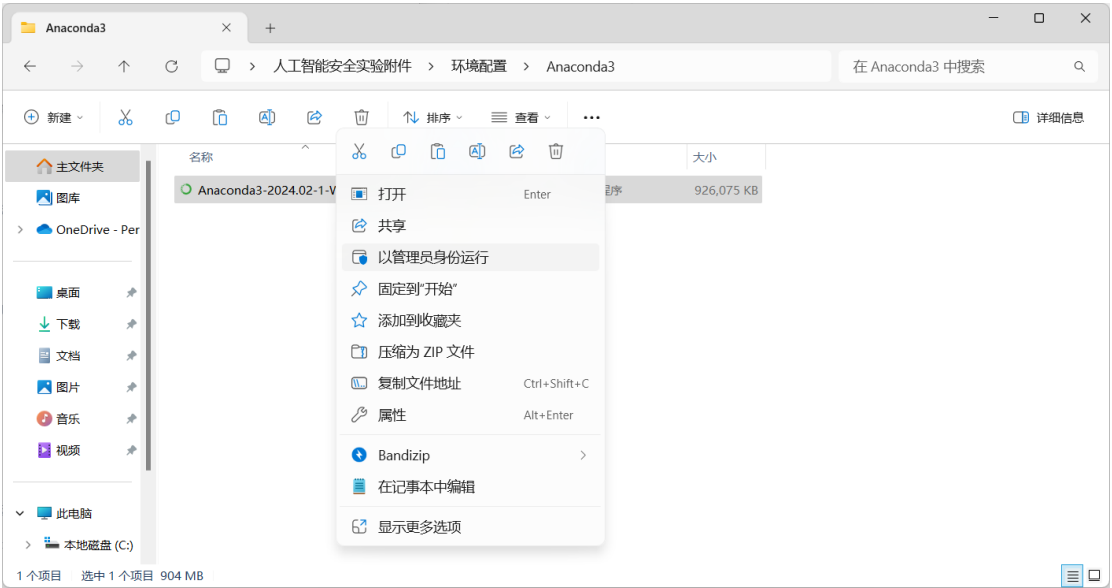
首先在清华镜像源网站

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/> 下载 Anaconda, 按照时间排序, 选择最新的 windows 版 Anaconda .exe 执行文件进行下载:

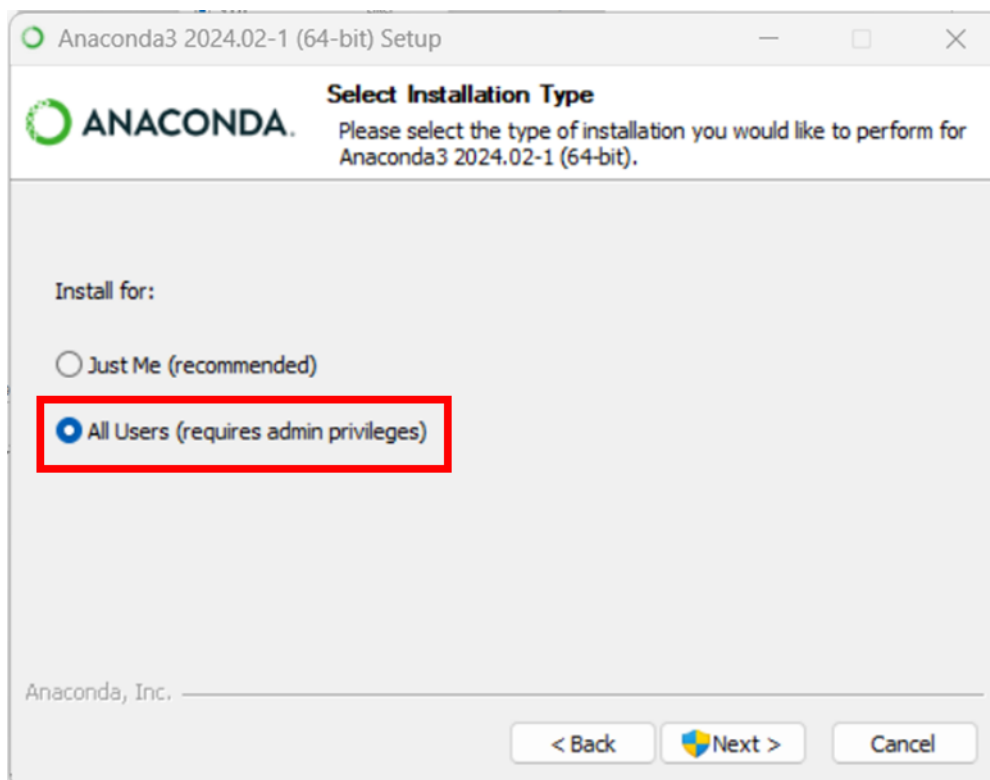


File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
Anaconda3-2024.02-1-Windows-x86_64.exe	904.4 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-MacOSX-x86_64.sh	731.2 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-MacOSX-x86_64.pkg	728.7 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-MacOSX-arm64.pkg	697.4 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-Linux-x86_64.sh	997.2 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-Linux-s390x.sh	391.8 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-MacOSX-arm64.sh	700.0 MiB	2024-02-27 06:01
Anaconda3-2024.02-1-Linux-aarch64.sh	798.5 MiB	2024-02-27 06:01
Anaconda3-2023.09-0-Windows-x86_64.exe	1.0 GiB	2023-09-29 23:48
Anaconda3-2023.09-0-MacOSX-x86_64.sh	774.1 MiB	2023-09-29 23:48
Anaconda3-2023.09-0-MacOSX-x86_64.pkg	772.0 MiB	2023-09-29 23:48

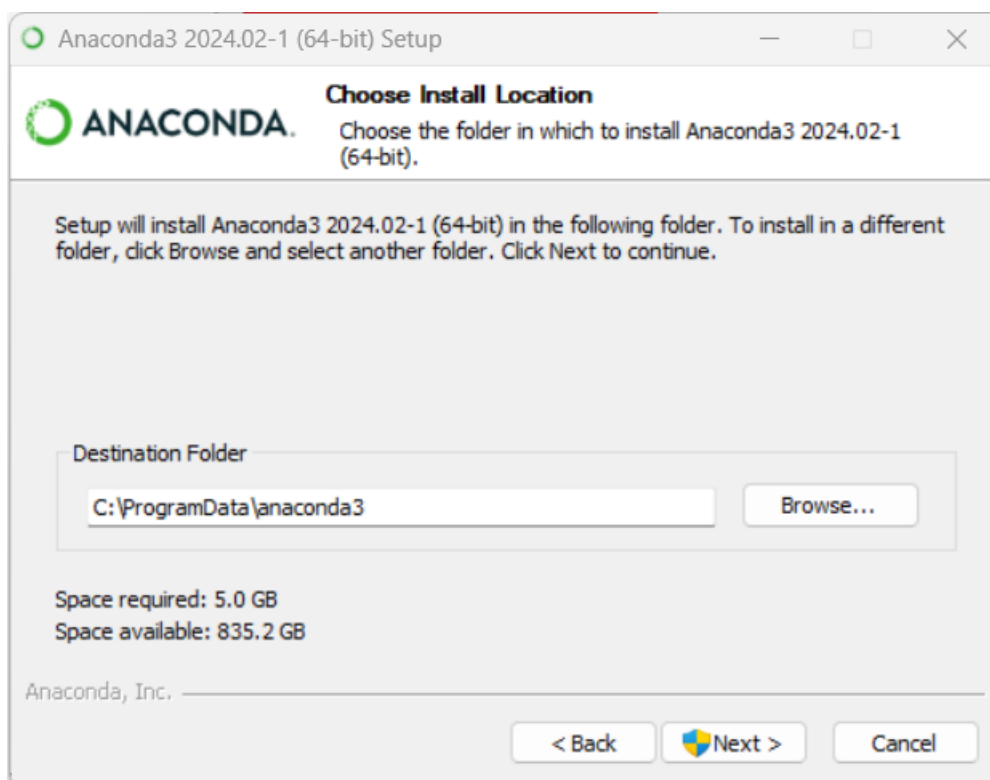
右键单击下载后的可执行文件, 选择以管理员身份运行:



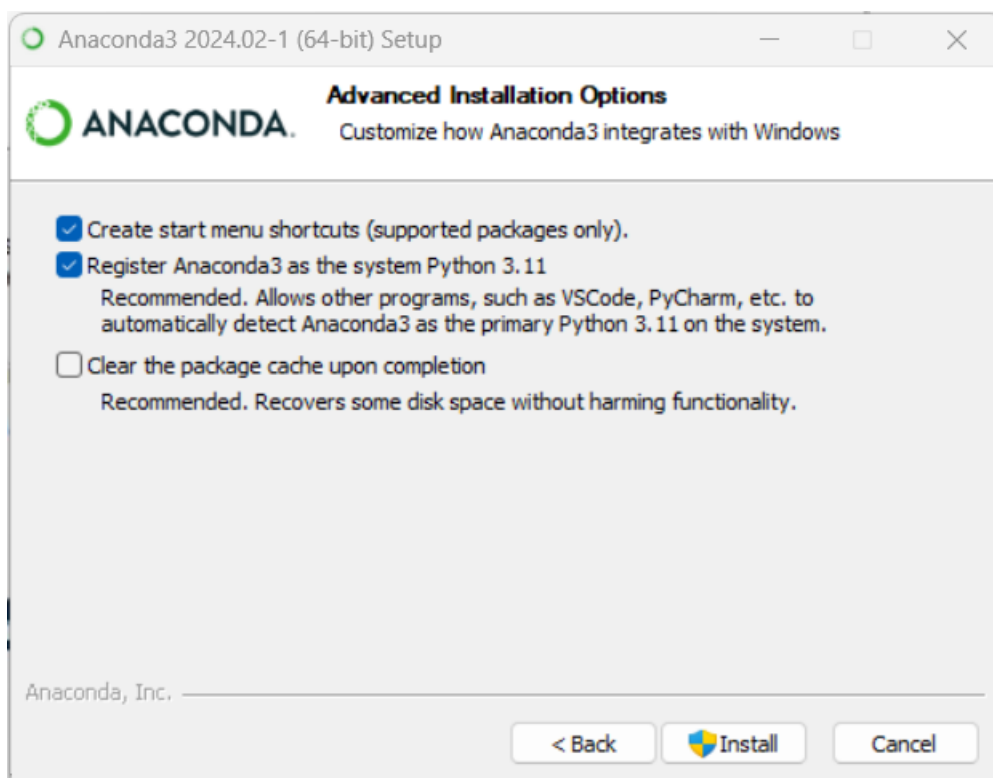
顺序点击 next/I agree, 为所有用户安装 Anaconda:



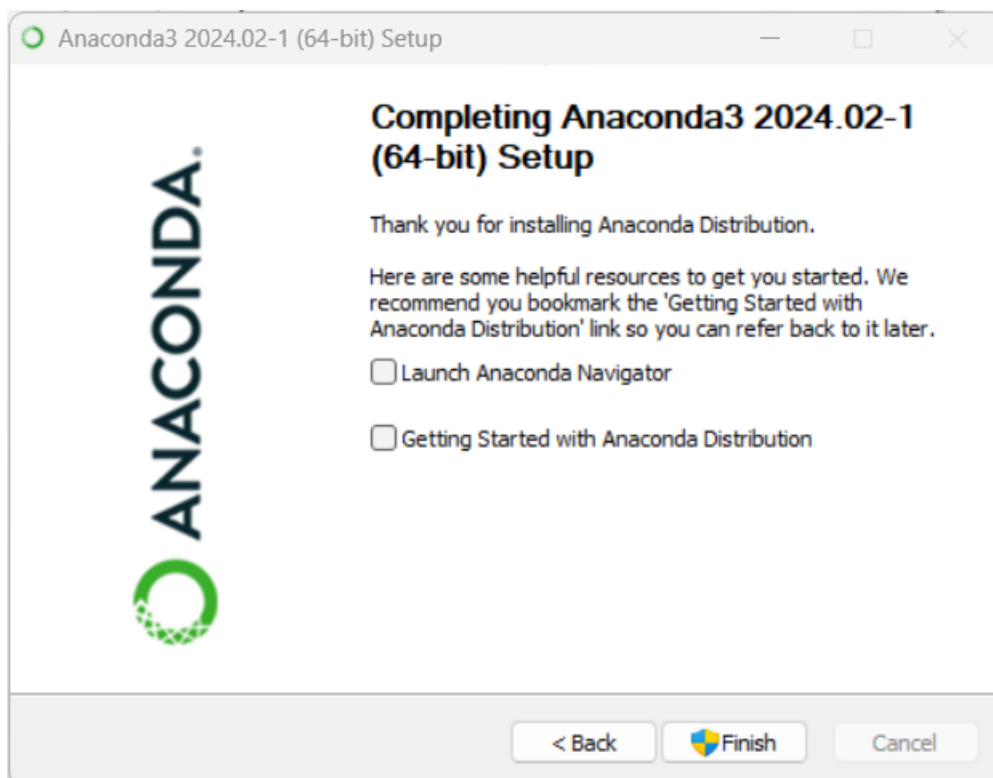
选择安装路径，如果 c 盘空间不够，可以选择在其他盘安装，注意安装路径不要出现中文、空格、特殊字符：



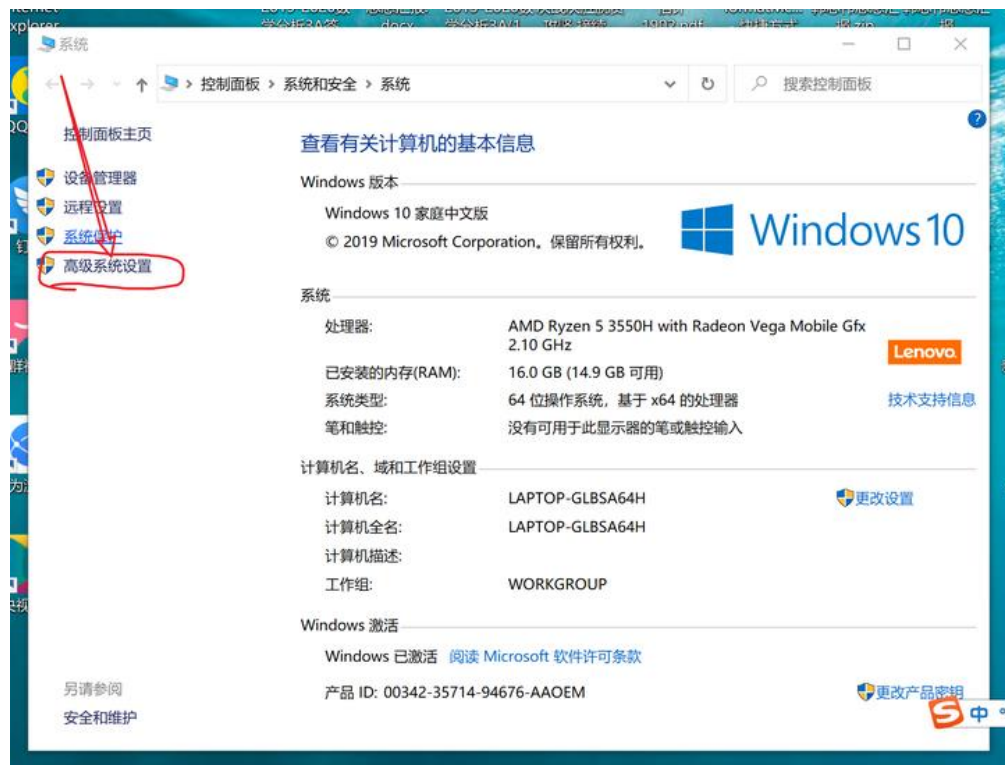
保持默认选项，开始安装 Anaconda3：



取消勾选两个选项，结束 Anaconda3 安装：



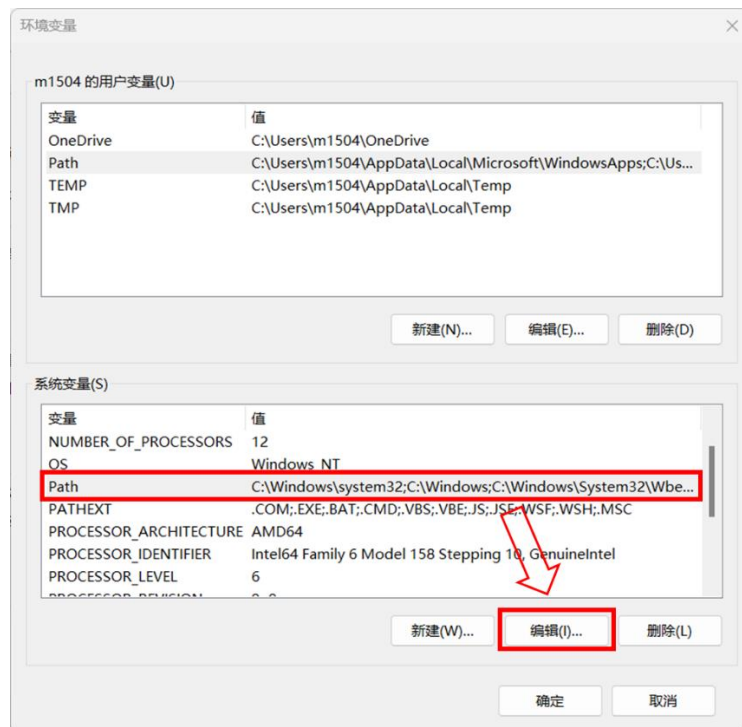
接下来为系统添加 **anaconda3** 环境变量，打开电脑控制面板→系统和安全→系统→高级系统设置（此处 win10 和 win11 界面显示不同，按照自己系统界面显示为准）：



选择高级选项卡下的环境变量选项：

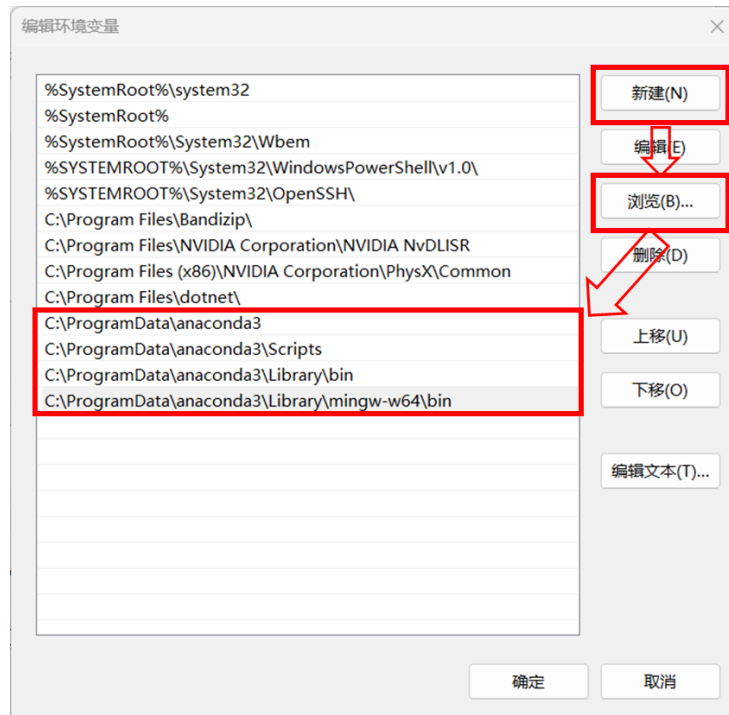


选中下方系统变量选项中的‘Path’一行，点击下方编辑：

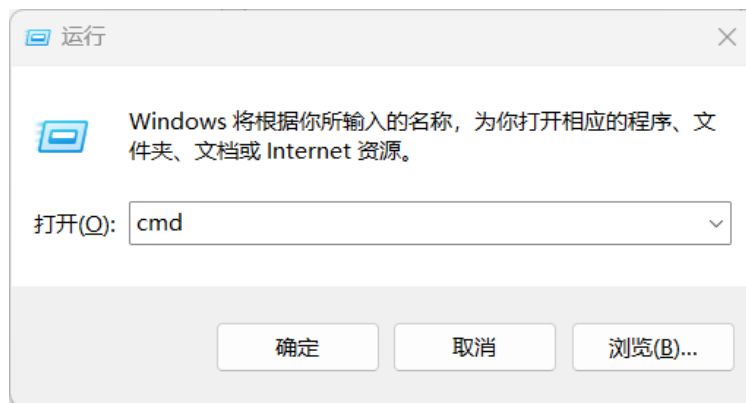


选择新建→浏览，将图中四个文件夹加入环境变量（路径前缀与

前面 Anaconda3 安装路径保持一致), 单击确定, 完成环境变量配置:



接下来验证 Anaconda3 安装成功, 同时按下 win+R, 输入 cmd 打开命令行:



在打开的命令行中输入 conda, 若命令行返回 conda 可用指令列表, 则 Anaconda3 安装成功:

```
C:\Windows\system32\cmd.e: X + v
Microsoft Windows [版本 10.0.22631.3296]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\m1504>conda
usage: conda-script.py [-h] [-v] [--no-plugins] [-V] COMMAND ...

conda is a tool for managing and deploying applications, environments and packages.

options:
  -h, --help            Show this help message and exit.
  -v, --verbose          Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG
                        logging, four times for TRACE logging.
  --no-plugins          Disable all plugins that are not built into conda.
  -V, --version          Show the conda version number and exit.
```

## b) Ubuntu 系统 Anaconda3 环境配置

参考博客 <https://blog.csdn.net/thy0000/article/details/122878599>

### 2) 实验所需虚拟环境配置

完成 Anaconda3 基础环境配置后,以 Anaconda 虚拟环境为基础,搭建本实验所需环境。实验所需 python、pytorch、以及相关依赖包版本如下:

Requirements	Version
python	>=3.5
torch	==1.0
torchvision	==0.2.1
numpy	>=1.15.2
matplotlib	>=3.0.0

在命令行中输入以下命令新建 conda 虚拟环境,其中, -n 后面的参数为新建虚拟环境的名称,python==后面的参数为指定的虚拟环境 python 版本:

```
C:\Users\m1504>conda create -n AI_security_exp1 python==3.5.6
```

安装过程中会询问是否继续安装,输入 y 回车执行,anaconda 自



动开始虚拟环境的配置：

```
C:\Windows\system32\cmd.e: x + v
added / updated specs:
- python==3.5.6

The following packages will be downloaded:

package | build | size
-----|-----|-----
certifi-2020.6.20 | pyhd3eb1b0_3 | 155 KB
pip-10.0.1 | py35_0 | 1.6 MB
python-3.5.6 | he025d50_0 | 14.5 MB
setuptools-40.2.0 | py35_0 | 497 KB
wheel-0.37.1 | pyhd3eb1b0_0 | 33 KB
wincertstore-0.2 | py35hfebbdb8_0 | 14 KB
-----|-----|-----
Total: | 16.7 MB

The following NEW packages will be INSTALLED:

certifi | pkgs/main/noarch::certifi-2020.6.20-pyhd3eb1b0_3
pip | pkgs/main/win-64::pip-10.0.1-py35_0
python | pkgs/main/win-64::python-3.5.6-he025d50_0
setuptools | pkgs/main/win-64::setuptools-40.2.0-py35_0
vc | pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime | pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel | pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore | pkgs/main/win-64::wincertstore-0.2-py35hfebbdb8_0

Proceed ([y]/n)? y
```

等待 `anaconda3` 将所需依赖包下载完成，使用 `conda activate` 指令激活刚才新建的虚拟（首次使用 `anaconda3` 此处会出现报错，此时输入 `conda init` 指令，等待 `anaconda` 初始化配置完成后重新启动命令行，即可正常进入刚才新建的虚拟环境）：

```
C:\Users\m1504>conda activate AI_security_exp1
```

命令行的前面出现虚拟环境名，即代表成功进入虚拟环境：

```
(AI_security_exp1) C:\Users\m1504>
```

使用 `conda install` 指令安装对应版本的 `pytorch`：

```
(AI_security_exp1) C:\Users\m1504>conda install pytorch-cpu==1.0.0 torchvision-cpu==0.2.1 cpuonly -c pytorch
```

继续使用 `conda install` 指令安装 `numpy`、`matplotlib` 依赖包：

```
(AI_security_exp1) C:\Users\m1504>conda install numpy
```

```
(AI_security_exp1) C:\Users\m1504>conda install matplotlib
```

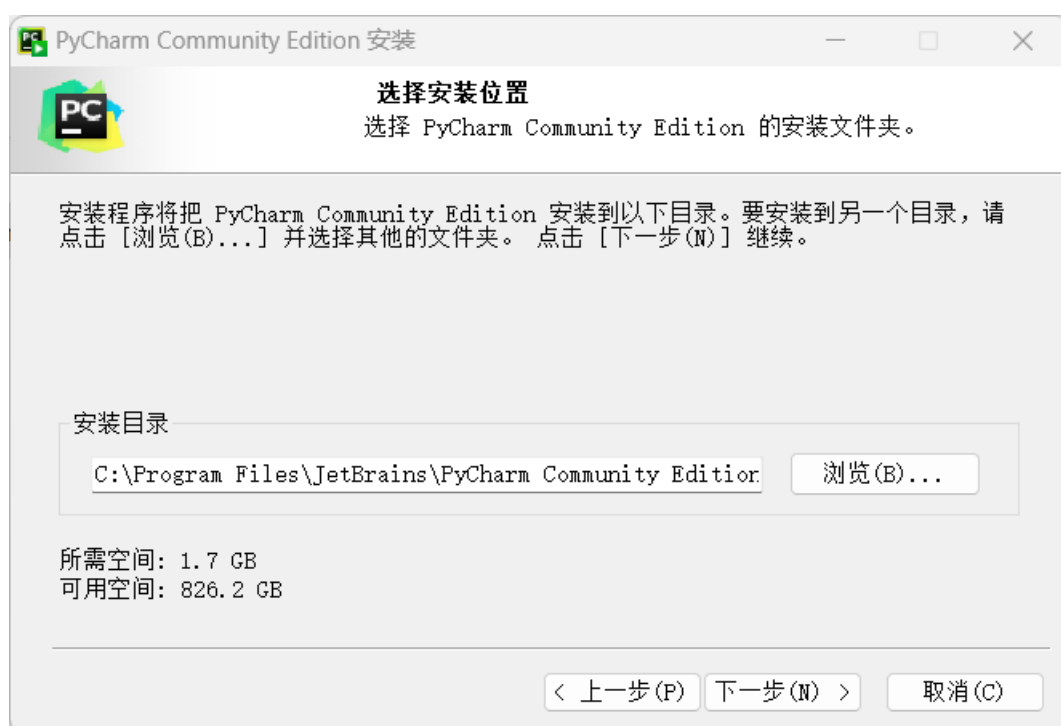
至此，完成深度学习基础环境配置。

## 二、Pycharm 开发环境配置

完成深度学习基础环境配置后，下载 Pycharm 作为开发 IDE，打开

<https://www.jetbrains.com/pycharm/download/?section=windows#section=windows> 链接，下载 Pycharm 教育版。

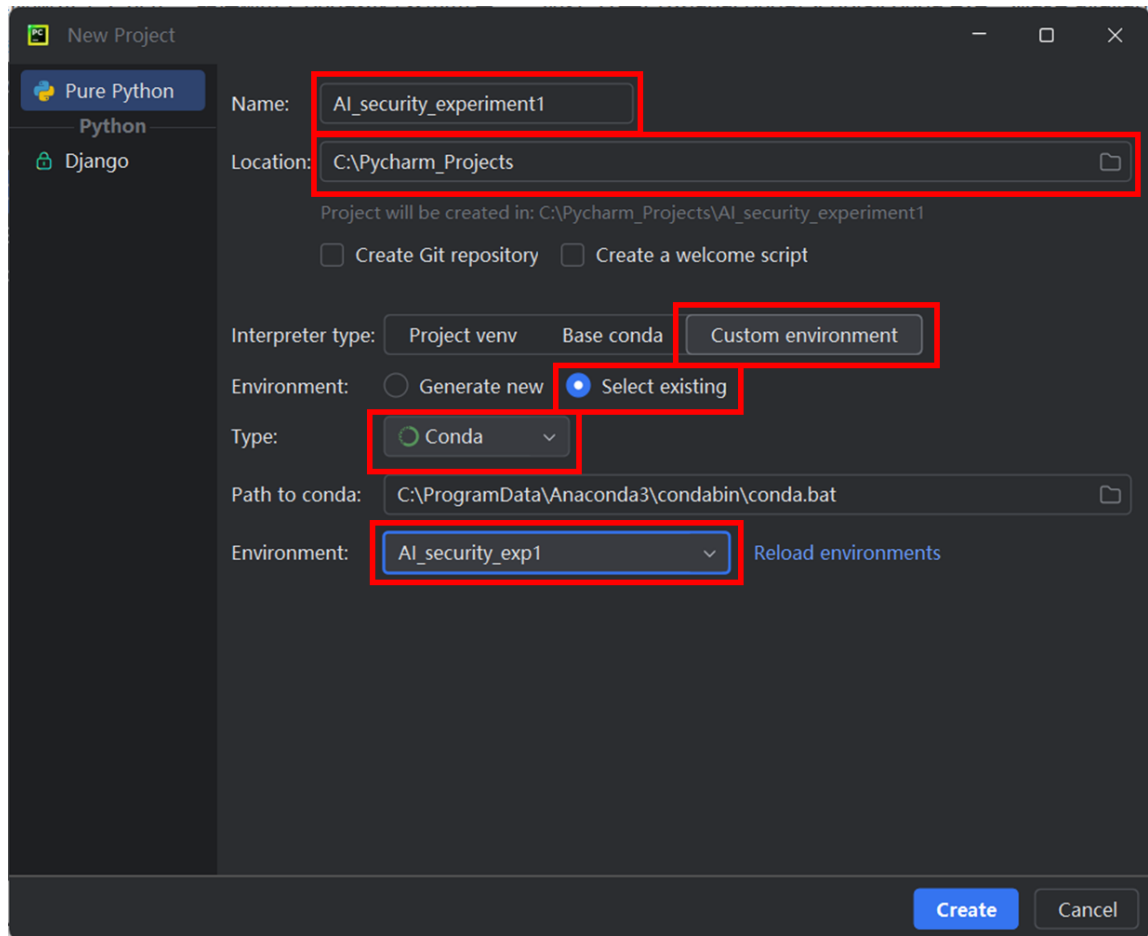
打开下载好的安装程序，选择安装路径，同样注意路径不要出现中文、空格、特殊字符：



点击下一步，在配置选项页中勾选所有的选项，然后点击下一步，完成安装：



打开安装好的 Pycharm，选择 New Project，Name 栏填写实验项目文件夹名称，Location 栏填写存放 python 项目文件夹路径，解释器选择部分选择 custom environment，将创建好的 anaconda 虚拟环境“AI\_security\_exp1”链接至本实验。



至此，完成本实验的全部环境配置工作。