《人工智能基础 A》实验报告二

AI 集成开发环境搭建及使用

一、实验目的

通过实验，学会搭建人工智能开发环境。具体目标要求如下：

1） 复习或学习掌握 AI 语言 Python 工具安装。

2） 学习掌握 AI 的开发环境 Anaconda 平台的安装部署，并熟悉其用于 Python 虚拟环境管理的功能。

3） 学习掌握 PyCharm/Vscode 等编程 IDE 的安装部署使用

4） 学习掌握基于 Jupyter lab 的 notebook 环境使用运行 Python 代码的方法。

5） 学习 Mo 平台的使用，包括如何创建项目、管理数据集及进行模型训练。

6） 学习掌握常用 python 库如 pandas、numpy、matplotlib 的使用。

二、实验内容及要求

本次实验需要撰写实验报告，请同学们在实验报告中记录每一步成功操作的截图，需要截图的步骤均已加粗标注。

实验步骤（仔细阅读，按照步骤完成实验）

1. Python 安装

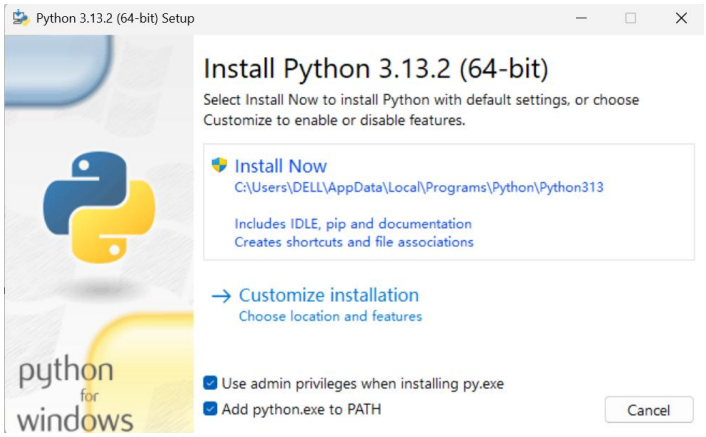
首先进入 Python 的官方网站（https://www.python.org/downloads/ ），在下图中选择对应系统进入 python 版本选择页面，下载适合的 Python 版本的安装程序。

python 官网下载页面

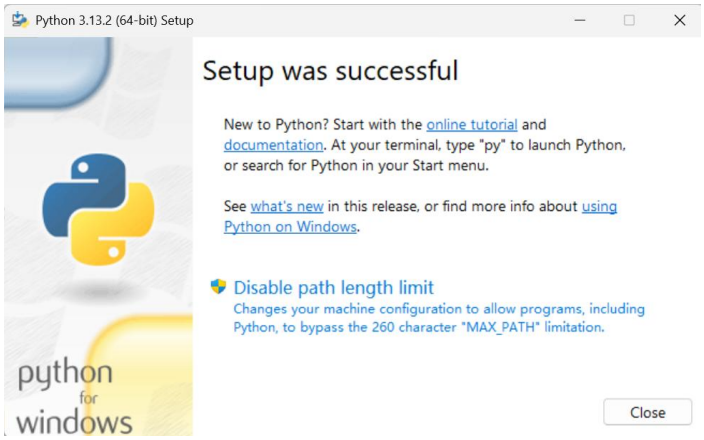
（1） 下载完成后打开 Python 安装程序，按照安装向导的指示进行安装并截图。

windows 系统下只需点击黄色的 Download 按键，就能下载 exe 文件，打开后进入安装程序。

勾选下面的两个选项：



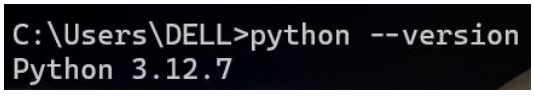
选择自动安装：



安装成功。

（2） 在 cmd（Windows 状态栏搜索 cmd 打开）中输入“Python --version”验证是否安装成功，若安装成功，将显示版本信息。若未显示版本信息，请检查环境变量是否设置无误。

打开命令行窗口输入命令：



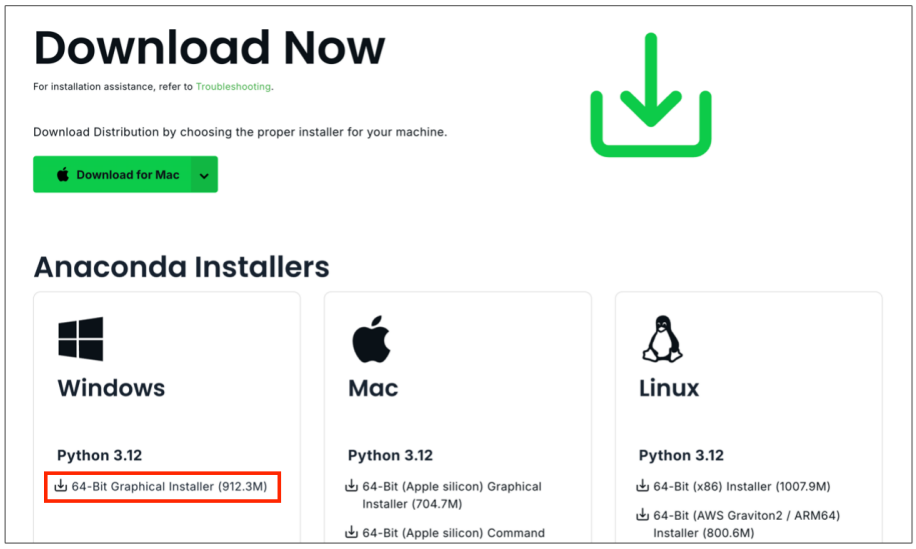
我的电脑已经预装过 python，所以我使用 py 启动器管理不同版本。
指定版本的命令来查看版本号：

```
C:\Users\DELL>py -3.13 --version
Python 3.13.2

C:\Users\DELL>py -3.12 --version
Python 3.12.0
```

2. Anaconda 安装部署

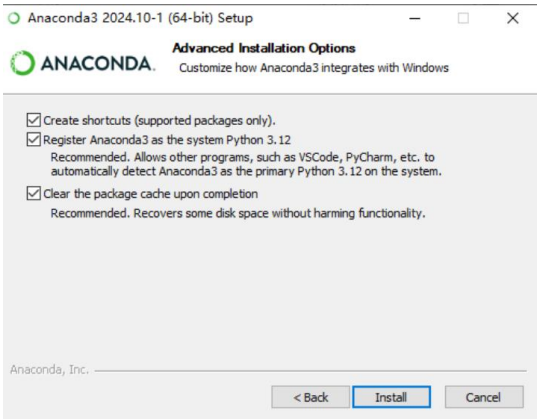
安装 Anaconda 时，具体步骤可能因 Anaconda 的版本和操作系统的不同还是会有所差异的，但总体流程是相似的，下面以 windows 为例。

（1）首先要从 Anaconda 的官方网站（https://www.anaconda.com/download/）下载 windows 版本，选择对应的 Python 版本，一般建议选最新的 Python 3.x，然后下载相应的安装程序。

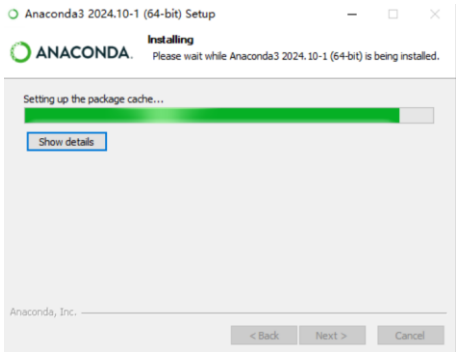


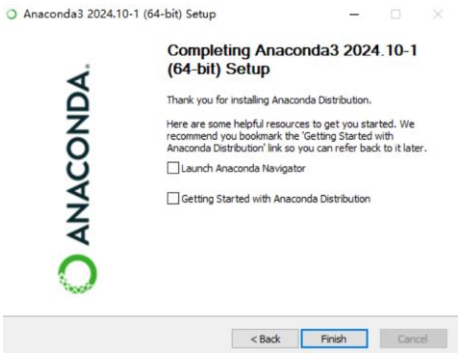
（2） 下载完成后打开安装包，按照安装向导的指示进行安装。

我下载的是官网首页的 windows 默认安装包，打开后进入安装向导：

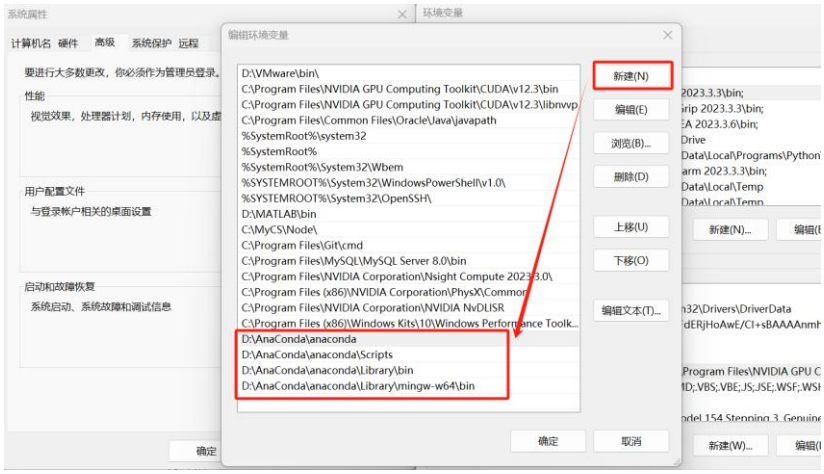


- 等待安装完成





- 此时 cmd 还不能识别 conda 命令，需要继续配置系统环境变量；
- 在系统变量的 Path 中新建环境变量：

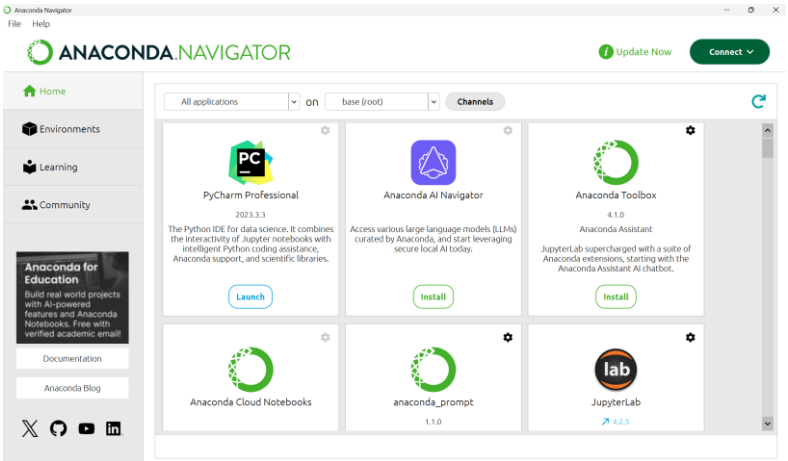


此时 conda 安装完成。

（3） 打开 cmd，输入 conda --versin 或 conda -V 命令来验证是否安装成功。若安装成功，将显示版本信息。

```
C:\Users\DELL>conda --version
conda 24.9.2
```

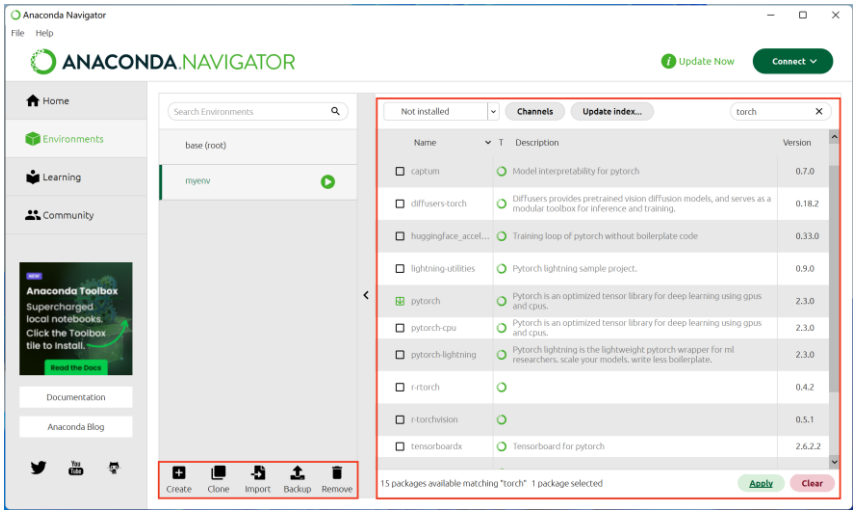
（4） 在应用中找到 Anaconda Navigator 并打开。



3. Anaconda 的使用（以 windows 为例）

3.1 虚拟环境的管理

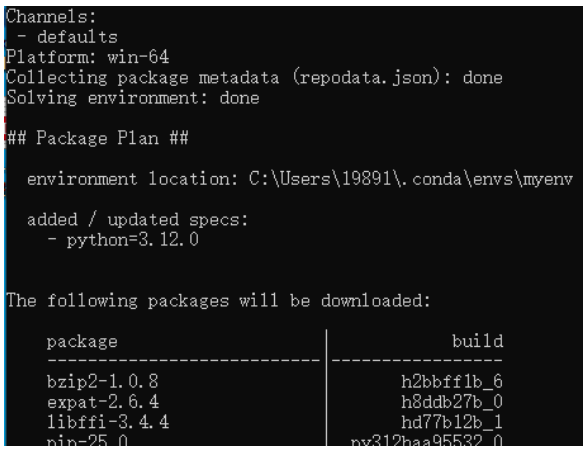
在 Anaconda Navigator 左边栏选择 Environments 进入虚拟环境管理页面，如下图所示。在这里可以对虚拟环境进行创建、删除等，还可以对指定环境中的包进行管理。

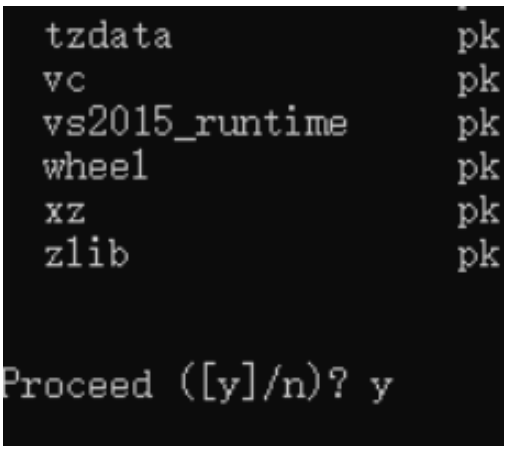


虚拟环境的管理还可以使用 conda 命令在 cmd 中完成（mac 系统在终端中完成）

（1） 使用 conda 命令创建虚拟环境，命令如下，其中 myenv 为自定义的虚拟环境名称。

```
conda create -n myenv python=3.12.0
```





conda 列出了即将下载的包，输入一个 y 才能继续下载；

(2) 查看虚拟环境列表，验证虚拟环境是否创建成功，命令如下：

```
conda env list
```

```
C:\Users\19891>conda env list
# conda environments:
#
base                  C:\Anaconda
myenv                 C:\Users\19891\.conda\envs\myenv
```

包括 base 环境和刚刚创建的新环境

(3) 切换到指定虚拟环境，并在此状态下查看虚拟环境列表

```
conda activate myenv
```

此时出现报错 CondaError: Run 'conda init' before 'conda activate'，先输入一个 conda init 命令对 conda 进行初始化（需要管理员权限），然后再次运行 conda activate myenv 激活了虚拟环境；

```
C:\Windows\system32>conda activate myenv

(myenv) C:\Windows\system32>conda env list
# conda environments:
#
base                     C:\Anaconda
myenv                 *  C:\Users\19891\.conda\envs\myenv
```

激活虚拟环境后，命令行开头括号内会显示当前所在环境名。

(4) 用 conda 安装 python 包，以 numpy 为例：

```
conda install numpy
```

```
(myenv) C:\Windows\system32>conda install numpy
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

（安装过程同样需要输入一个 y 来确认）

(5) 退出虚拟环境

```
conda deactivate
```

```
(myenv) C:\Windows\system32>conda deactivate

C:\Windows\system32>
```

（6） 移除指定虚拟环境

```
conda remove -n myenv --all
```

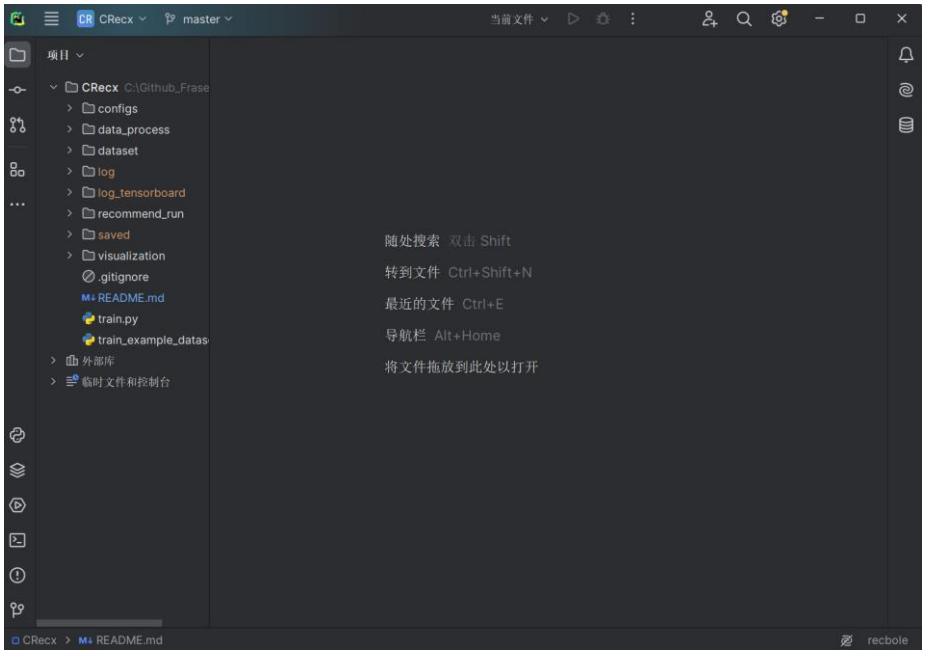
```
Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Everything found within the environment (C:\Users\19891\.conda\envs\myenv),
ns and any non-conda files, will be deleted. Do you wish to continue?
 (y/[n])? y


C:\Windows\system32>
```

3.2 IDE 安装

PyCharm 和 Visual Studio Code 均为编程 IDE，同学们二选一（两个软件的具体配置还需要同学们自行网上学习，大家要善于利用大语言模型！搜索引擎最好用 Google 或者 Bing），**对应截图也只需一个软件的图片。**

（1） Pycharm 安装。去到 PyCharm 官方下载页面（MacOS 注意更改对应平台），需要将页面往下滑看到 PyCharm Community Edition，点击下载（PyCharm Professional 需要付费，学校正版软件平台也有对应授权方法，同学们视自己情况选择不同版本安装）。参考配置

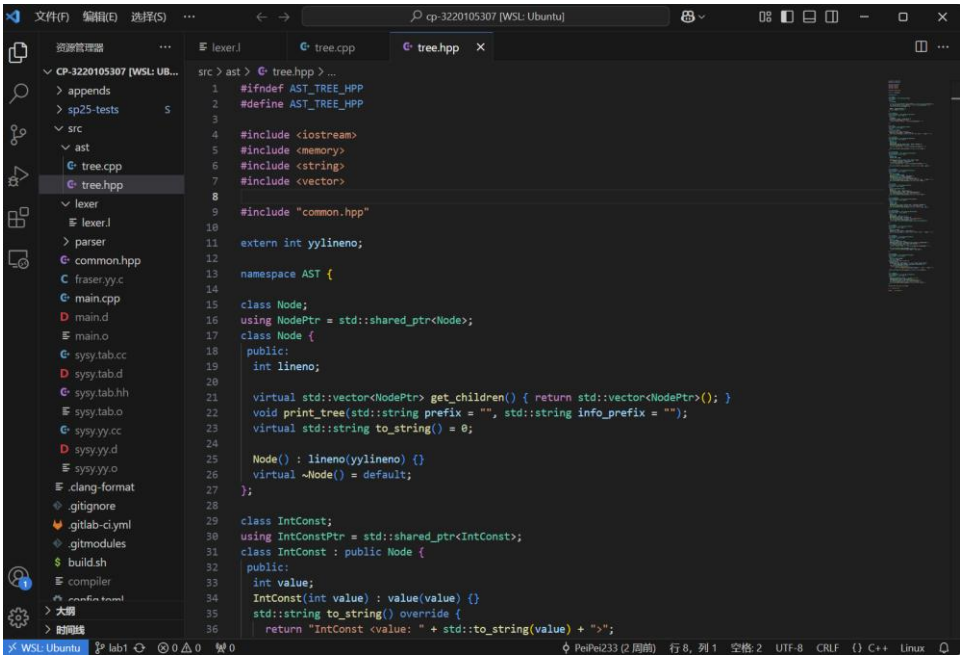
安装后启动软件并截图：



我安装的是 Professional 版本，通过 JetBarain 学生认证获得免费使用许可。

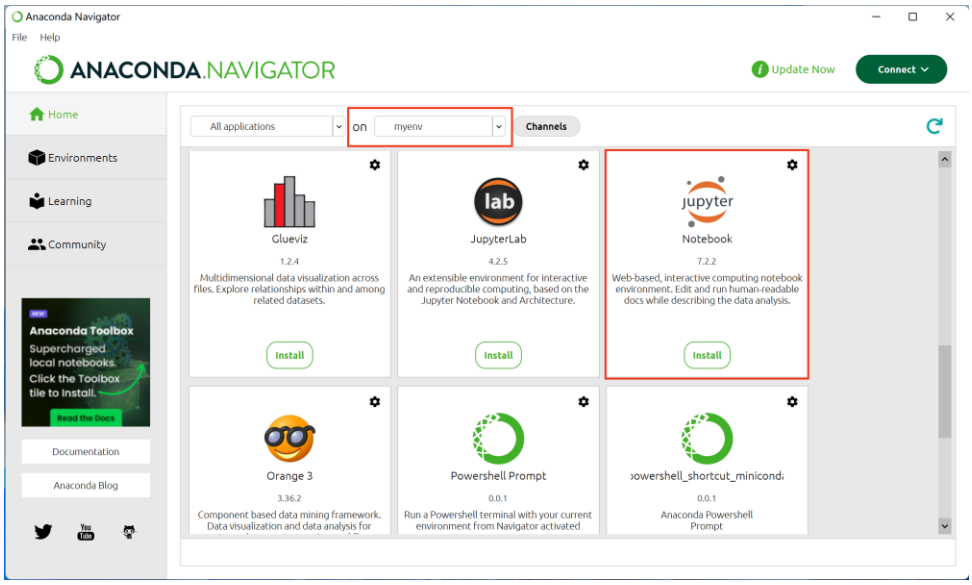
（2） Visual Studio Code 安装。去到官方下载页面（Windows 平台选择 User Installer，MacOS 平台选择.zip，注意处理器类别，Windows 通常选择 x64，MacOS 通常选择 Apple Silicon。参考配置链接：quick start、getting started、Python in VScode、Python Envrionments

安装后启动软件并截图：



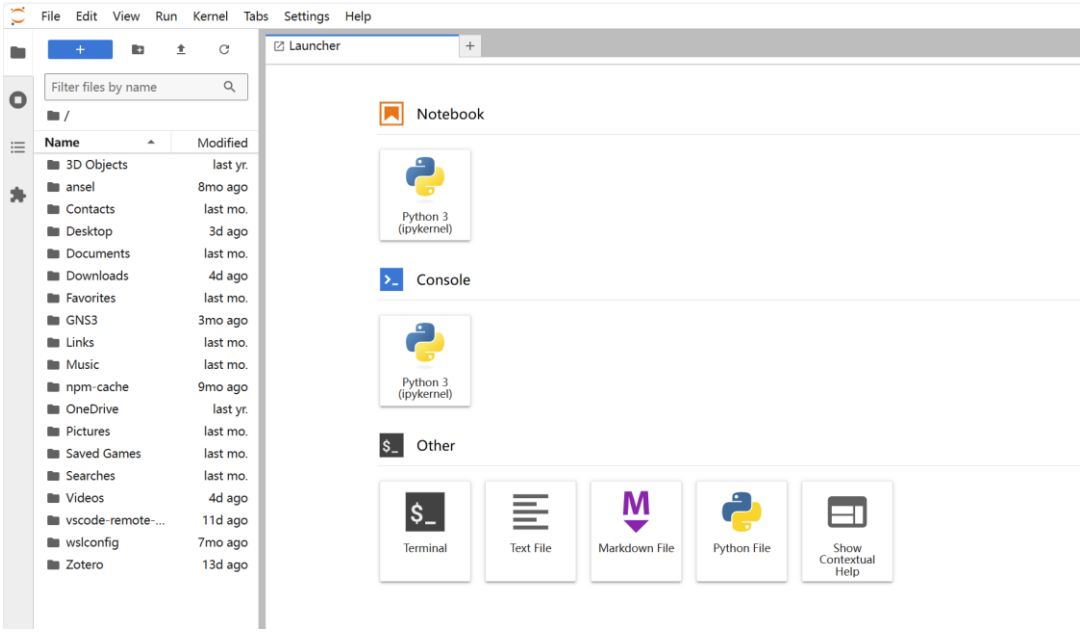
3.3 Jupyter Notebook

Jupyter notebook 是一个基于网页的交互式计算环境，其优点是交互式强，易于可视化，尤其适用于需要频繁修改、实验的场景，比如数据分析、测试机器学习模型等。



（1） 进入 Anaconda Navigator，选择需要运行 Jupyter notebook 的虚拟环境，找到 Jupyter notebook 点击下方的“Install”按钮进行安装。

（2） 安装完成后点击“Launch”按钮就可以打开 Jupyter notebook。打开后进行截图。

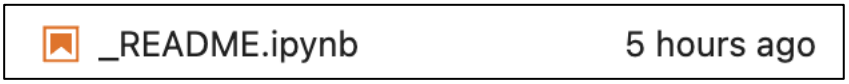


4. Mo 平台的使用

首先进入官网（https://mo.zju.edu.cn/），用浙大通行证登陆，点击右上角“我的学习”，点击我的课程。

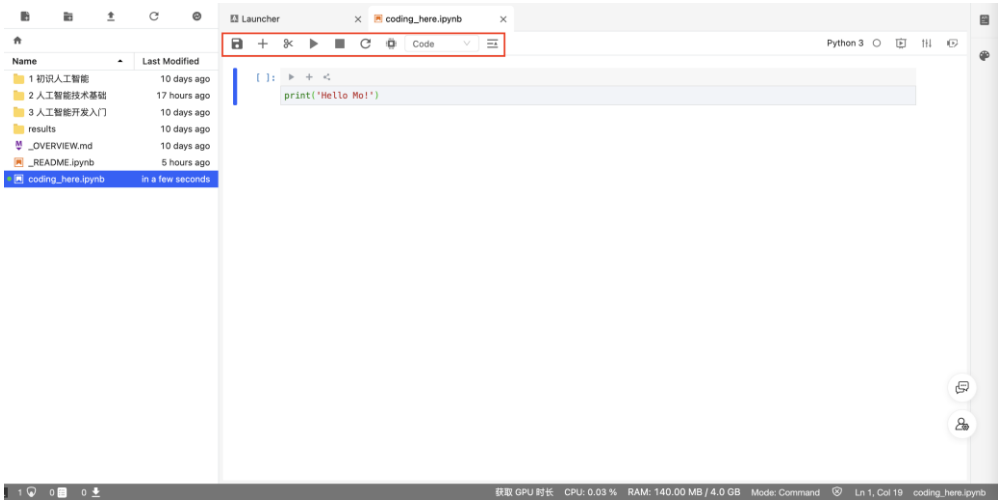
找到本课程并点击进入项目。

在文件一栏中找到下图所示文件，并双击点开。



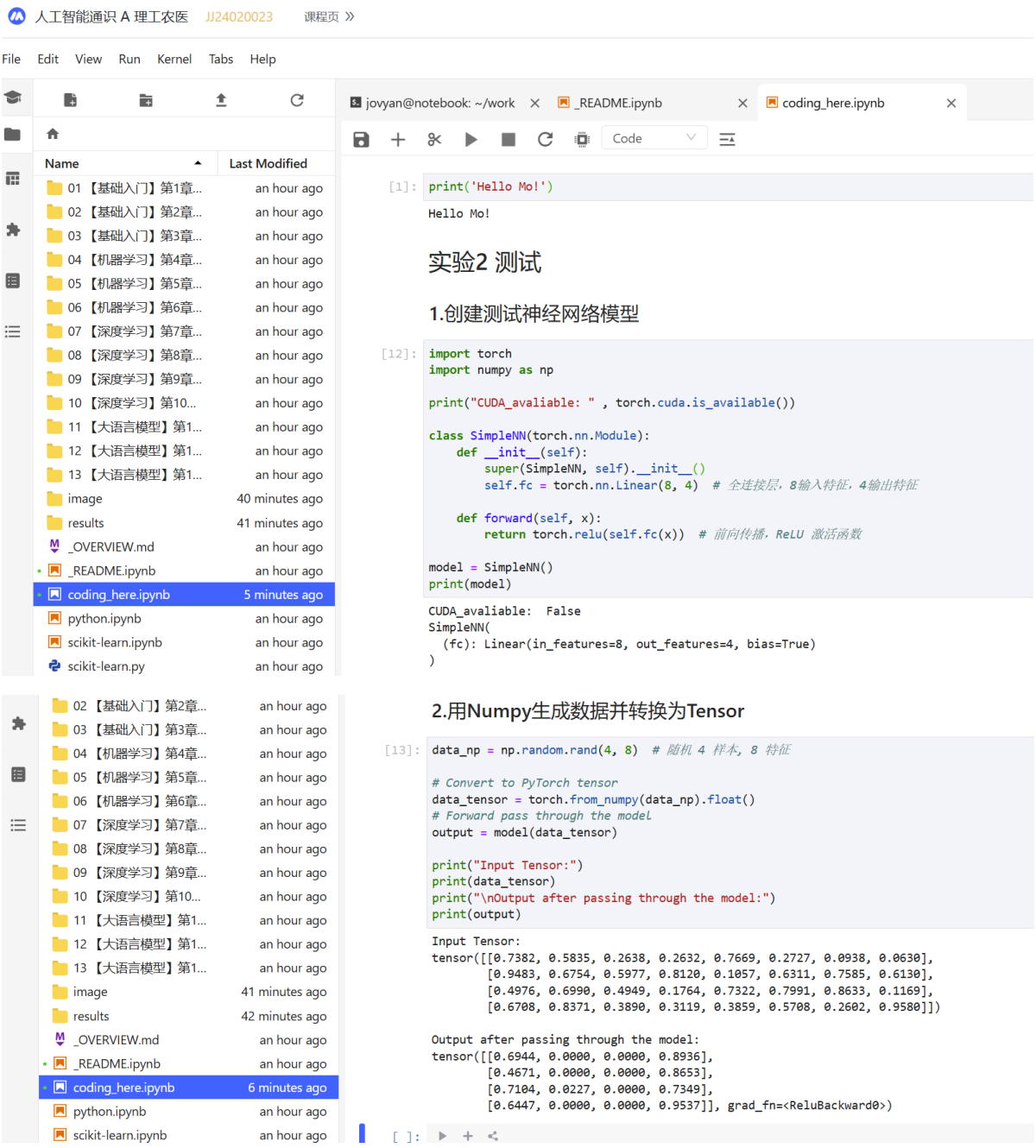
仔细阅读，学习如何使用 Mo 平台开发项目。

可以在 coding_here.ipynb 中学习如何使用 Jupyter notebook，试着使用红框中的功能键，写入一些代码后运行并截图。



等待环境创建：

在 coding_here.ipynb 中写入一些代码，然后运行：



程序测试了 pytorch，该环境已经预装 pytorch，tensorflow 等机器学习库，直接在网页端运行代码不支持 GPU，需要用专门的方式创建任务调用 GPU。

5. Python 科学计算常用包的使用

以下用到的 Python 包均可通过 Anaconda Navigator 下载或者在命令行中用 pip 命令下载。

下面演示在命令行中创建一个虚拟环境并进入到该虚拟环境，最后下载需要的 Python 包。

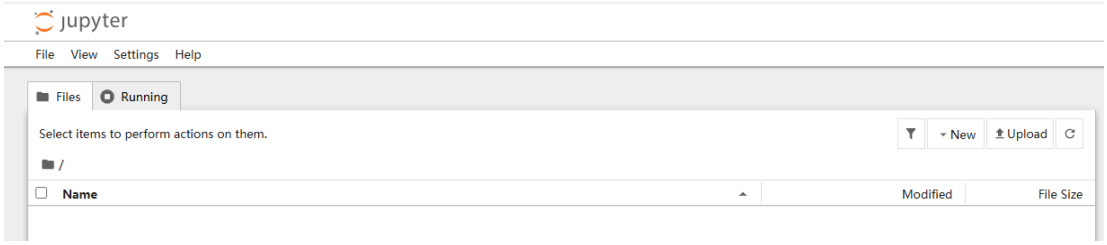
1. 在命令行中输入 conda create -n xxx 创建虚拟环境。xxx 为虚拟环境名，同学们可自行决定。
2. 在命令行中输入 conda activate xxx 进入虚拟环境（这一步很重要）。
3. 在命令行中输入 conda install pip 来安装 pip 包管理器。
4. 在命令行中输入 pip install pandas numpy matplotlib notebook 来安装 Python 包。若下载速度较慢，可转用国内镜像源。

（这里的 some-packages 为你想下载的包名，设为默认即代表以后下载 Python 包无须指定-i https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple 也可从清华源下载）。

5. 在命令行中输入 jupyter notebook，此时浏览器应自动跳转到 jupyter 界面，若跳转失败或需要输入 token，可在命令输出中找到 url 以及 token。进入 jupyter 界面后选择自己想存放代码的文件夹并新建 notebook 文件。
6. 若要退出虚拟环境，输入 conda deactivate。或者直接关闭命令行。
7. 对于每个 Python 包的使用，同学们可以参考官方文档或者求助于大语言模型。

完成以上步骤，我创建了虚拟环境名为 AIAlab2，安装完所有包以后查看已安装 python 包环境（pip list）：

可以看到所需软件包均已安装，接下来输入 jupyter notebook：



以下作业不需要提供代码！！！提供对应代码截图以及运行结果即可。

5.1 Python 数据分析处理工具库 Pandas 的使用操作。quick-start

```
import pandas as pd
data = {
    "products" : ["Phone", "Laptop","Pad", "Earphone", "Smart Watch",
                  "Camera", "Television", "Speaker", "Printer", "Router"],
    "Sales" : [150, 80, 90,200,120, 60,50,130,70,40],
    "Prices($)" : [3000, 6000, 2000, 800, 1500, 5000, 4000, 1500, 1000, 600],
    "Sold Date" : pd.date_range(start="2025-01-01", periods=10, freq='D')
}
df = pd.DataFrame(data)
print(df)
```

将上述代码复制粘贴到 notebook 文件中，并在 Jupyter notebook 中实现：

(1)筛选出销售量大于 100 的产品

```
[2]: filtered_df = df[df['Sales'] > 100]
     print(filtered_df)
```

	products	Sales	Prices($)	Sold Date
0	Phone	150	3000	2025-01-01
3	Earphone	200	800	2025-01-04
4	Smart Watch	120	1500	2025-01-05
7	Speaker	130	1500	2025-01-08

(2)按销售量从大到小排序并输出

```
[3]: sorted_df = df.sort_values(by='Sales', ascending=False)
     print(sorted_df)
```

	products	Sales	Prices($)	Sold Date
3	Earphone	200	800	2025-01-04
0	Phone	150	3000	2025-01-01
7	Speaker	130	1500	2025-01-08
4	Smart Watch	120	1500	2025-01-05
2	Pad	90	2000	2025-01-03
1	Laptop	80	6000	2025-01-02
8	Printer	70	1000	2025-01-09
5	Camera	60	5000	2025-01-06
6	Television	50	4000	2025-01-07
9	Router	40	600	2025-01-10

(3)计算所有产品的平均销售量

```
[4]: average_sales = df['Sales'].mean()
     print(average_sales)
```

```
99.0
```

(4)计算每种产品的总销售额，并添加总销售额为新列

```
[5]: df['Revenue($)'] = df['Sales'] * df['Prices($)']
     print(df)
```

	products	Sales	Prices($)	Sold Date	Revenue($)
0	Phone	150	3000	2025-01-01	450000
1	Laptop	80	6000	2025-01-02	480000
2	Pad	90	2000	2025-01-03	180000
3	Earphone	200	800	2025-01-04	160000
4	Smart Watch	120	1500	2025-01-05	180000
5	Camera	60	5000	2025-01-06	300000
6	Television	50	4000	2025-01-07	200000
7	Speaker	130	1500	2025-01-08	195000
8	Printer	70	1000	2025-01-09	70000
9	Router	40	600	2025-01-10	24000

5.2 Python 数值计算科学计算库 Numpy 使用操作。quick start、

在 Jupyter notebook 中实现。

示例：创建一个 2x12 大小的符合标准正态分布的随机矩阵，对其所有元素乘 10，再进行下取整操作。

```
import numpy as np
a = np.random.randn(2,12)
print(a)
print("-------------------------------------------------")
a *= 10
print(a)
print("-------------------------------------------------")
a = np.floor(a)
print(a)
```

```
[[ 0.35809334 -0.77518876 -0.49921386 -0.18700687  2.15278214  0.97630448
   0.04445876 -0.35529367  1.04052255 -1.10333605  0.79156596 -1.38997722]
 [-0.50471712 -0.52152354  0.15042424  1.55641821 -1.03450385 -1.82443713
  -0.11770865 -0.94571319 -0.1591186  -0.11108928  0.65144027  1.10281316]]
-------------------------------------------------
[[  3.58093336  -7.75188759  -4.99213856  -1.87006873  21.52782137
    9.76304479   0.44458761  -3.5529367   10.40522551 -11.03336054
    7.91565962 -13.89977218]
 [ -5.04717117  -5.21523539   1.50424242  15.56418211 -10.34503852
  -18.24437126  -1.17708649  -9.4571319   -1.59118601  -1.11089278
    6.51440266  11.02813162]]
-------------------------------------------------
[[  3.  -8.  -5.  -2.  21.   9.   0.  -4.  10. -12.   7. -14.]
 [ -6.  -6.   1.  15. -11. -19.  -2. -10.  -2.  -2.   6.  11.]]
```

（1） 创建一个从 1 到 20 的 NumPy 一维数组，并筛选出偶数

[2]:
```
arr = np.arange(1, 21)
even_arr = arr[arr % 2 == 0]
print("偶数:", even_arr)
```

```
偶数: [ 2  4  6  8 10 12 14 16 18 20]
```

（2） 计算并输出一维数组的均值、标准差、最大值、最小值

[3]:
```
mean_val = np.mean(arr)
std_dev = np.std(arr)
max_val = np.max(arr)
min_val = np.min(arr)
print("均值: {}, 标准差: {}, 最大值: {}, 最小值: {}".format(mean_val, std_dev, max_val, min_val))
```

```
均值: 10.5, 标准差: 5.766281297335398, 最大值: 20, 最小值: 1
```

（3） 任意创建一个 NumPy 二维数组，然后进行形状变换

[4]:
```
array_2d = np.array([[1, 2, 3], [4, 5, 6]])
reshaped_array = array_2d.reshape(3, 2)
print(reshaped_array)
```

```
[[1 2]
 [3 4]
 [5 6]]
```

（4） 再创建一个二维数组（注意如何设定形状才能进行矩阵乘法），与上面的二维数组进行矩阵乘法运算

[5]:
```
another_array_2d = np.array([[7, 8, 9], [10, 11, 12]])
matrix_multiplication_result = np.dot(reshaped_array, another_array_2d)
print(matrix_multiplication_result)
```

```
[[ 27  30  33]
 [ 61  68  75]
 [ 95 106 117]]
```

（5） 生成 4 行 3 列的随机数组

[6]:
```
random_array = np.random.rand(4, 3)
print(random_array)
```

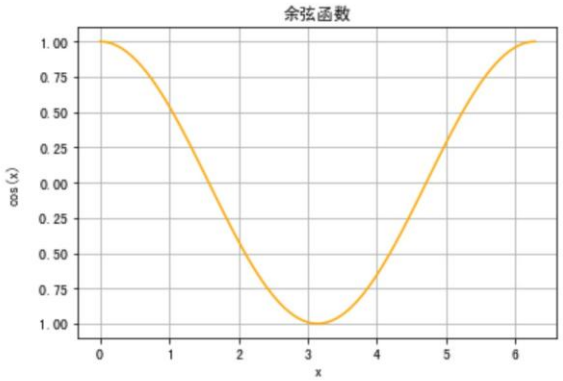
```
[[0.17782212 0.29667763 0.49201043]
 [0.73877087 0.81752669 0.35993492]
 [0.72341908 0.00964166 0.82260943]
 [0.83469766 0.28885752 0.32826695]]
```

5.3 Python 可视化工具 Matplotlib 的简单使用。quick start

（1）示例 1 绘制余弦函数 cos(x)

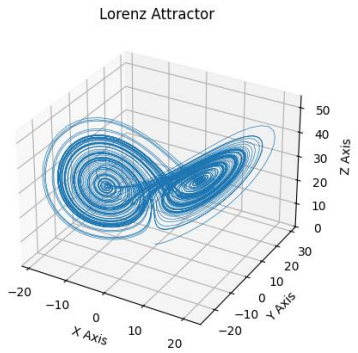
[2]:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 2 * np.pi, 100)
y = np.cos(x)

plt.plot(x, y, color='orange')
plt.title('余弦函数')
plt.xlabel('x')
plt.ylabel('cos(x)')
plt.grid()
plt.show()
```



在 Jupyter Notebook 中实现。

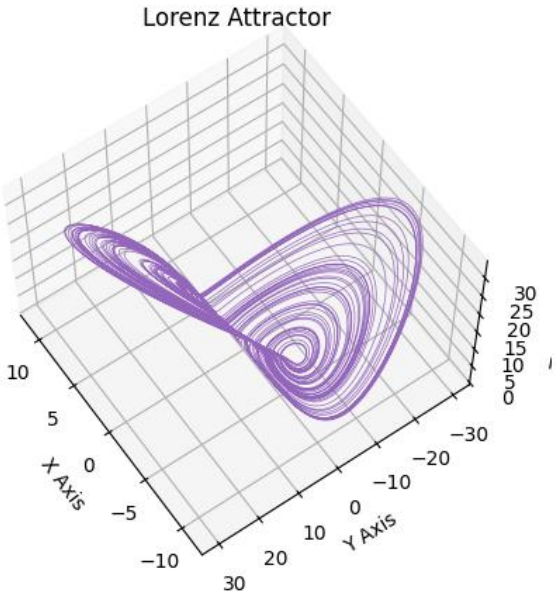
（1）画出一个 Lorenz Attrator（同学们可以进行一些自由探索：选择其他的 attrators，更改 figure 的朝向，线条的粗细、颜色等，甚至画出动画。加分可选项）。



Lorenz Attrator

复制链接中页面的代码，简单修改颜色和朝向绘制图像：

```
ax = plt.figure().add_subplot(projection='3d')

ax.plot(*xyzs.T, lw=0.5, color='#9467bd')
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("Lorenz Attractor")
ax.view_init(elev=60, azim=145)

plt.show()
```



其他 Attractor

我选择了网页中提供的 ACT Attractor 绘制动画

- 首先将网页提供的微分方程及其参数修改进代码中

```
def act_attractor(xyz, *, a=1.8, b=-0.07, d=1.5, m=0.02):
    x, y, z = xyz
    x_dot = a * (x - y)
    y_dot = -4 * a * y + x * z + m * (x**3)
    z_dot = -d * a * z + x * y + b * z*z
    return np.array([x_dot, y_dot, z_dot])
```

- 根据网页给出的 ACT Attractor 基本参数，设置初始条件

```
dt = 0.01
num_steps = 20000

xyzs = np.empty((num_steps + 1, 3))
xyzs[0] = (0.1, 0.1, 0.1)  # initial values

for i in range(num_steps):
    xyzs[i + 1] = xyzs[i] + act_attractor(xyzs[i]) * dt
```

- 设置动画的线条，由于我的动画展现图像中曲线的生成过程，因此初始数据线为空,在动画绘制过程中不断生成

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')

# 初始化时为空的数据线
line, = ax.plot([], [], [], lw=0.5)
ax.set_xlim(np.min(xyzs[:, 0]), np.max(xyzs[:, 0]))
ax.set_ylim(np.min(xyzs[:, 1]), np.max(xyzs[:, 1]))
ax.set_zlim(np.min(xyzs[:, 2]), np.max(xyzs[:, 2]))
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("ACT Attractor Growth")
```

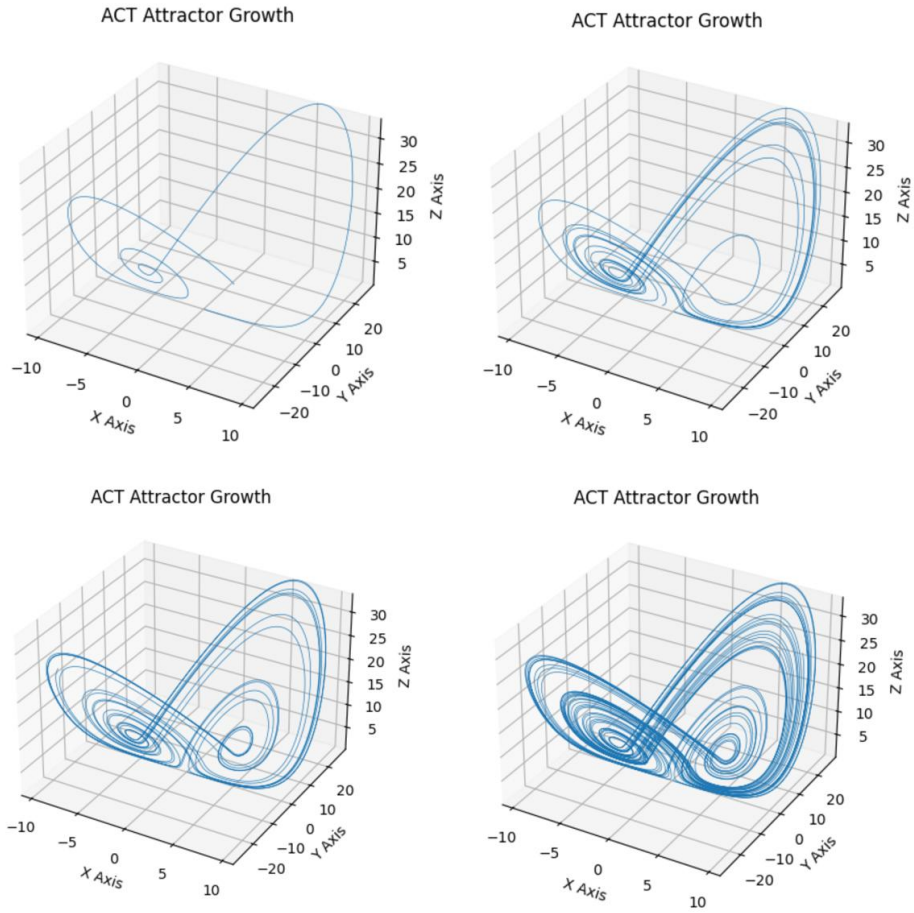
- 定义动画更新函数，随着帧数 n 的增加，越来越多的数据点被添加到图中，从而产生线条逐渐增长的效果

```
def update(n):
    line.set_data_3d(xyzs[:n, 0], xyzs[:n, 1], xyzs[:n, 2])
    return line,
```

- 绘制动画并保存为 GIF

```
ani = FuncAnimation(fig, update, frames=np.arange(0, num_steps+1, 10), interval=20, blit=False)

ani.save('act_growth.gif', writer=PillowWriter(fps=24))
plt.show()
```

- 动画展示 ACT Attractor 的曲线生成过程，下面展示其中 4 个时刻截图



三、实验感受与记录

3.1 实验感受（总结实验过程中的收获或疑问）

（1）在实际工程项目中，大多使用 conda 虚拟环境来管理 python 版本，因此本次实验第一步虽然在系统中直接安装了一个 python 环境，但在实际开发用基本不会用到这个非虚拟环境下的 python。

（2）Anaconda 等应用的安装，最重要的是保证系统环境变量正确，一般在 cmd 中提示找不到命令都是由于环境变量问题引起，需要多检查。

（3）由于 Anaconda 默认的虚拟环境建立在 C 盘，我修改了配置文件，将其移动到 D 盘目录下；同时，我为 conda 配置了镜像源来加快下载速度。

（4）Pycharm 虽然提供了免费的 Community 版本，但使用过程存在诸多限制，因此还是需要下载专业版。虽然学校正版软件平台提供了下载渠道，但是该版本需要每隔 10 分钟连接一次校园网认证（最长可断连 48 小时），在校外、家里或网络条件不佳的情况下将直接无法使用。JetBrain 为中国用户提供学信网认证，我使用这种方法申请了免费许可证。

（5）Pycharm 在使用最新版 matplotlib 绘制图像时，可能存在图形后端适配问题，如果出现绘制失败的报错，需要回退 matplotlib 版本到 3.7.1，可能还需要回退与之相关的部分包版本。

（6）在绘制其他 Attrator 时，点进链接的网页中只有几个微分方程和参数，一开始不知道怎么修改；后来发现可以手动把微分方程使用 python 写出来，替换掉 Lorenz attractor 中的三个微分方程即可。

（7）Mo 平台当前仅为每个账号提供 4GB 内存，且 GPU 使用存在限制，在实际机器学习任务中的性能表现相比个人电脑不一定有优势。关于具体使用，我找到了一个 CC98 帖子有做介绍：https://www.cc98.org/topic/6089731 。

3.2 实验记录

（以下填写实验操作过程产生的内容，包括文字和适当截取的中间操作图片）

实验记录已集成到实验步骤中，每个步骤后均有记录，此处不再重复列出。