



昇腾全栈AI软硬件平台

--Mindspore实践

昇腾全栈 AI 软硬件平台，构筑智能世界的基石

能源、金融、交通、电信、制造、医疗等行业应用



ModelArts



HiAI Service



第三方平台

应用使能

Mind X



MindX DL



MindX Edge



ModelZoo



MindX SDK

全流程开发工具链



MindStudio

AI框架



昇思 MindSpore

TensorFlow/PyTorch 等第三方框架

异构计算架构

CANN

Atlas 系列硬件

实验介绍

前馈神经网络是一种**最简单的**神经网络，各**神经元**分层排列。每个神经元只与前一层的神经元相连。接收前一层的输出，并输出给下一层，各层间没有反馈。是目前应用最广泛发展最迅速的人工神经网络之一。

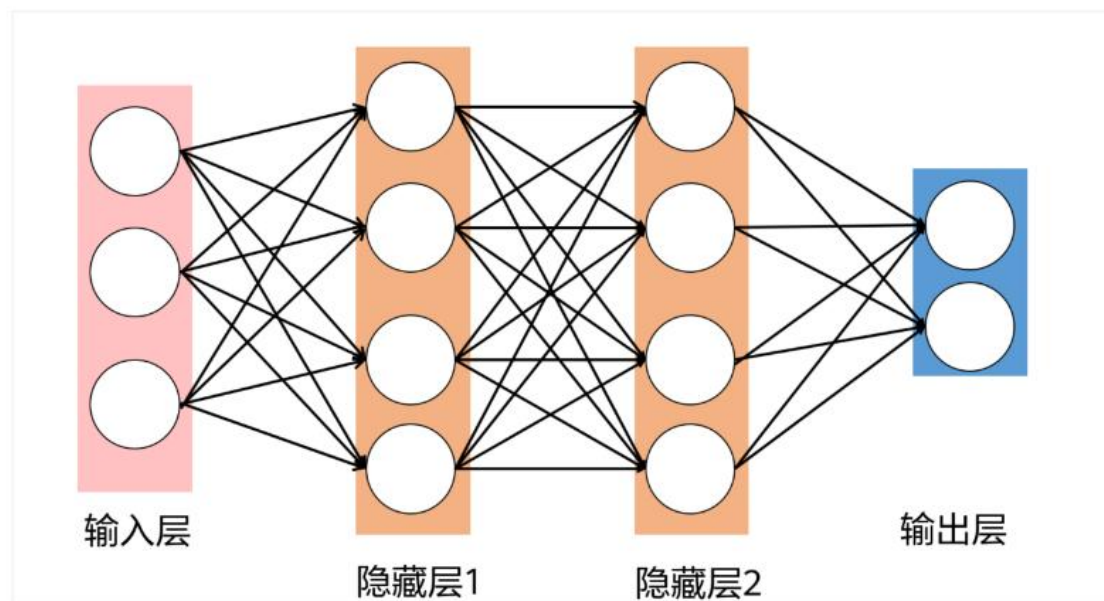
本次实验主要专注于理解前馈神经网络(Fully Connected Neural Network) 原理，基于**Mindspore**框架，利用网络实现Fashion-Mnist**图像分类**实验任务。

目的：

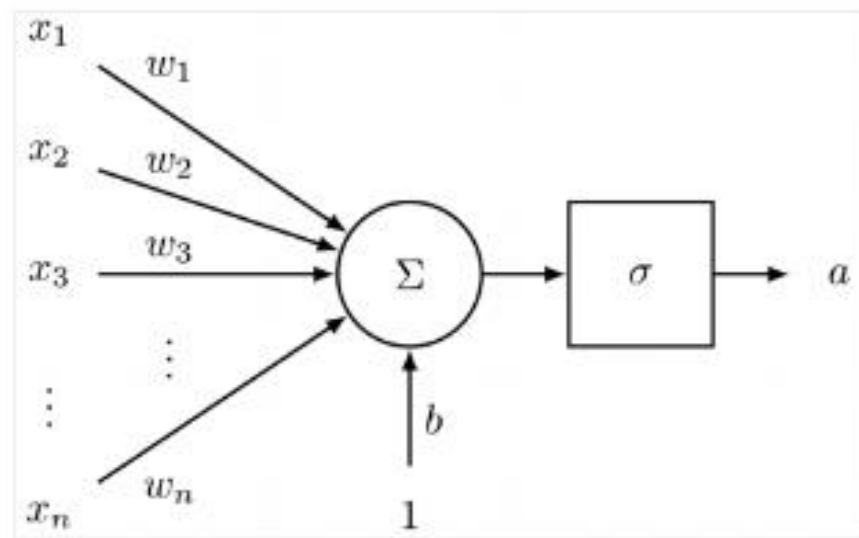
- 1.掌握如何使用MindSpore进行简单前馈神经网络的开发。
- 2.了解如何使用MindSpore进行简单图片分类任务的训练。
- 3.了解如何使用MindSpore进行简单图片分类任务的测试和预测。

预备知识

前馈神经网络(Feedforward neural network, FNN), 简称前馈网络,是人工神经网络的一种。前馈神经网络采用一种**单向多层结构**, 如下图所示。图中每一个圆圈代表一个**神经元**, 每一层包含若干个神经元。在此种神经网络中, 各神经元可以接收前一层神经元的信号, 产生输出到下一层。第0层叫**输入层**, 最后一层叫**输出层**, 其他中间层叫做**隐含层**(或隐藏层、隐层)。隐层可以是一层, 也可以是多层。



网络中的每一个神经元都完成了一次计算工作,如下图所示。接受高维的输入($x_0, x_1, x_2 \dots x_n$)与同维度的参数($w_0, w_1, w_2 \dots w_n$)对应相乘,累加后,经过激活函数得到该神经元的输出,传递至下一层神经元。

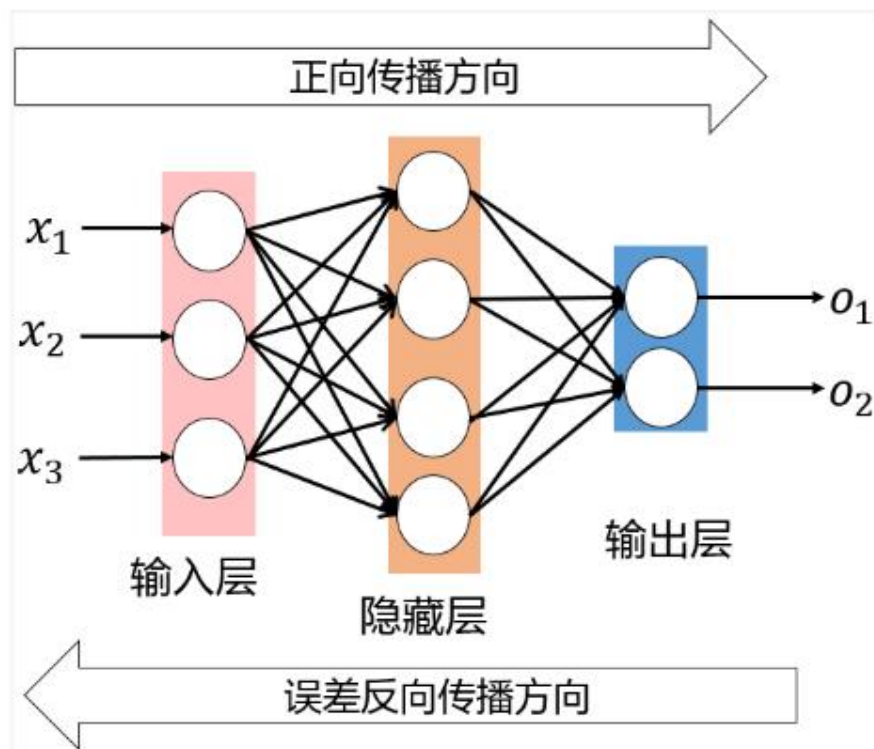


神经网络中输入数据的信息是沿着正向传播的,而预测结果和标签的误差则是通过反向传播的,在完成反向传播后就可以对所有参数进行更新。这个阶段也就是训练神经网络的过程。网络在建立时往往会随机生成所有参数。可想而知,此时的参数并不是理想参数,需要通过调整参数来达到我们想要的理想效果。这就需要训练,或者说反向传播更新参数来达到。

当第一个样本输入刚刚初始化的网络，会得到一个输出值 o_d ，而对于该样本，我们知道他本该得到输出值 t_d ，因此，通过下面的**损失函数**，我们可以得出这个样本的损失值。当然，这只是一种损失函数：二次代价函数。

$$E(W) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

然后，就可以利用**梯度下降法**对网络参数进行更新。梯度下降法的思想是让损失函数沿着负梯度的方向进行搜索，不断迭代更新参数，最终使得损失值最小化。



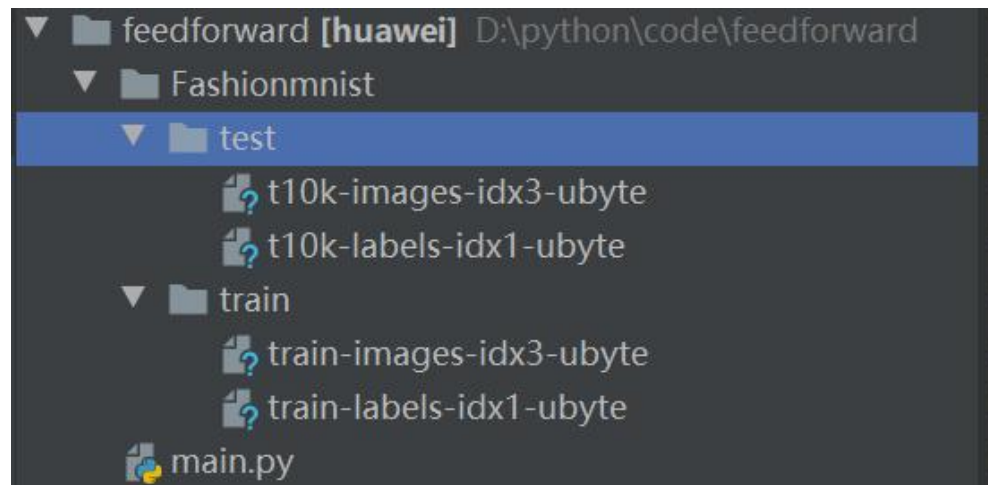
数据集介绍

Fashion-MNIST是一个替代MNIST手写数字集的图像数据集。其涵盖了来自10种类别的共7万个不同商品的正面图片。包括60000/10000的训练测试数据划分，28x28x1的灰度图片。

经典的MNIST数据集包含了大量的手写数字。MNIST数据集非常简单，很多深度学习算法在测试集上的准确率已经达到99.6%。

将脚本和数据集组织为如下形式：

```
feedforward
├── Fashion-MNIST
│   ├── test
│   │   ├── t10k-images-idx3-ubyte
│   │   └── t10k-labels-idx1-ubyte
│   └── train
│       ├── train-images-idx3-ubyte
│       └── train-labels-idx1-ubyte
└── main.py
```



实验环境



MindSpore

Miniconda/Anaconda

- Conda 是一个开源的软件包管理系统和环境管理系统，用于安装多个版本的软件包及其依赖关系，并在它们之间轻松切换。
- Conda 是为 Python 程序创建的，适用于 Linux, OSX 和 Windows, 也可以打包和分发其他软件。
- 最流行的 Python 环境管理工具。

总之，Anaconda是一个非常强大的Python发行版，它能够大大简化Python的安装和管理，提高开发者的工作效率。对于数据科学工作者来说，Anaconda更是必不可少的工具之一。

安装mindspore

- mindspore: <https://www.mindspore.cn/install>

一、获取安装命令

查看所有版本和接口变更 >

版本	2.0.0-rc1	1.10.1 ✓	Nightly		
硬件平台	Ascend 910	Ascend 310	GPU CUDA 10.1	GPU CUDA 11.1	CPU ✓
操作系统	Linux-aarch64	Linux-x86_64	Windows-x64 ✓	MacOS-aarch64	MacOS-x86_64
编程语言	Python 3.7	Python 3.8	Python 3.9 ✓		
安装方式	Pip ✓	Conda	Source	Docker	Binary
安装命令	<pre>pip install https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.10.1/MindSpore/cpu/x86_64/mindspore-1.10.1-cp39-cp39-win_amd64.whl --trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i https://pypi.tuna.tsinghua.edu.cn/simple</pre> <p># 注意参考下方安装指南，添加运行所需的环境变量配置</p>				

安装mindspore

- 安装命令: `pip install https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.10.1/MindSpore/cpu/x86_64/mindspore-1.10.1-cp39-cp39-win_amd64.whl --trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i https://pypi.tuna.tsinghua.edu.cn/simple`
- 验证是否安装成功: `python -c "import mindspore;mindspore.run_check()"`

```
(mindspore) C:\Users\16906>python -c "import mindspore;mindspore.run_check()"
MindSpore version: 1.10.1
The result of multiplication calculation is correct, MindSpore has been installed successfully!
```

安装其他所需要的包：

使用“conda install 包的名字”或者“pip install 包的名字”进行其他依赖包的安装。

- mindspore, 用于神经网络的搭建
- numpy, 用于处理一些数据
- matplotlib, 用于画图、图像展示
- struct, 用于处理二进制文件
- easydict, 用于方便操作字典

```
import os
import struct
from easydict import EasyDict as edict
import matplotlib.pyplot as plt
import numpy as np

import mindspore
import mindspore.dataset as ds
import mindspore.nn as nn
from mindspore import context
from mindspore.train import Model
from mindspore.train.callback import ModelCheckpoint, CheckpointConfig, LossMonitor, TimeMonitor
from mindspore import Tensor
import argparse
```

实验要求

1. **补全代码**，实现前馈神经网络。包括网络的初始化和前向过程。
2. 训练前馈神经网络，**调整网络参数或者修改网络结构**，输出分类准确率。
4. **6月12日晚上12:00**之前将代码（.py或者.ipynb文件）、实验报告（doc或pdf文件）一并打包上传至邮箱。邮箱：**mlspring2023@163.com**

压缩包和实验报告命名方式：

实验序号_学号_姓名，例如：实验1_111702xxxxxx_王xx

```
# 定义前馈神经网络
class Forward_fashion(nn.Cell):
    def __init__(self, num_class=10): # 一共分十类，图片通道数是1
        super(Forward_fashion, self).__init__()
        self.num_class = num_class
        self.flatten = nn.Flatten()
        # TODO: 初始化前馈神经网络

    def construct(self, x):
        # TODO: 定义前馈神经网络前向过程
```

实验结果举例

①

```
Forward_fashion<
  (flatten): Flatten<>
  (fc1): Dense<input_channels=784, output_channels=128, has_bias=True>
  (relu): ReLU<>
  (fc2): Dense<input_channels=128, output_channels=10, has_bias=True>
  >
```

```
epoch: 20 step: 16000, loss is 0.0019754625391215086
epoch: 20 step: 17000, loss is 0.0009831213392317295
epoch: 20 step: 18000, loss is 0.048107683658599854
epoch: 20 step: 19000, loss is 8.136026735883206e-05
epoch: 20 step: 20000, loss is 0.0035286860074847937
```

```
{'acc': 0.88375}
```

第0个 sample 预测结果:	Dress	真实结果:	Dress
第1个 sample 预测结果:	Trouser	真实结果:	Trouser
第2个 sample 预测结果:	Pullover	真实结果:	Shirt
第3个 sample 预测结果:	Coat	真实结果:	Coat
第4个 sample 预测结果:	Sandal	真实结果:	Sandal
第5个 sample 预测结果:	Shirt	真实结果:	Shirt
第6个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第7个 sample 预测结果:	Pullover	真实结果:	Coat
第8个 sample 预测结果:	T-shirt/top	真实结果:	T-shirt/top
第9个 sample 预测结果:	Trouser	真实结果:	Trouser
第10个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第11个 sample 预测结果:	Dress	真实结果:	Dress
第12个 sample 预测结果:	Pullover	真实结果:	Pullover
第13个 sample 预测结果:	Coat	真实结果:	Coat
第14个 sample 预测结果:	Dress	真实结果:	Dress

②

```
Forward_fashion<
  (flatten): Flatten<>
  (fc1): Dense<input_channels=784, output_channels=256, has_bias=True>
  (relu1): ReLU<>
  (fc2): Dense<input_channels=256, output_channels=128, has_bias=True>
  (relu2): ReLU<>
  (fc3): Dense<input_channels=128, output_channels=10, has_bias=True>
```

```
epoch: 20 step: 17000, loss is 0.0009553695563226938
epoch: 20 step: 18000, loss is 0.05153035745024681
epoch: 20 step: 19000, loss is 0.00030835019424557686
epoch: 20 step: 20000, loss is 0.0005264079081825912
```

```
{'acc': 0.8882178714859438}
```

第0个 sample 预测结果:	Coat	真实结果:	Coat
第1个 sample 预测结果:	Ankle boot	真实结果:	Ankle boot
第2个 sample 预测结果:	Shirt	真实结果:	Shirt
第3个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第4个 sample 预测结果:	Ankle boot	真实结果:	Ankle boot
第5个 sample 预测结果:	Trouser	真实结果:	Trouser
第6个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第7个 sample 预测结果:	Bag	真实结果:	Bag
第8个 sample 预测结果:	Coat	真实结果:	Coat
第9个 sample 预测结果:	Pullover	真实结果:	Pullover
第10个 sample 预测结果:	Dress	真实结果:	Dress
第11个 sample 预测结果:	Ankle boot	真实结果:	Ankle boot
第12个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第13个 sample 预测结果:	Sneaker	真实结果:	Sneaker
第14个 sample 预测结果:	Pullover	真实结果:	Pullover

附件1：pycharm安装和使用

下载pycharm，下载社区版，也可下载专业版，经过学生认证后也可免费使用。下载链接：

<https://www.jetbrains.com/pycharm/download/#section=windows>

下载 PyCharm

Windows

macOS

Linux

Professional

适用于科学和 Web Python 开发。支持 HTML、JS 和 SQL。

下载

免费试用

Community

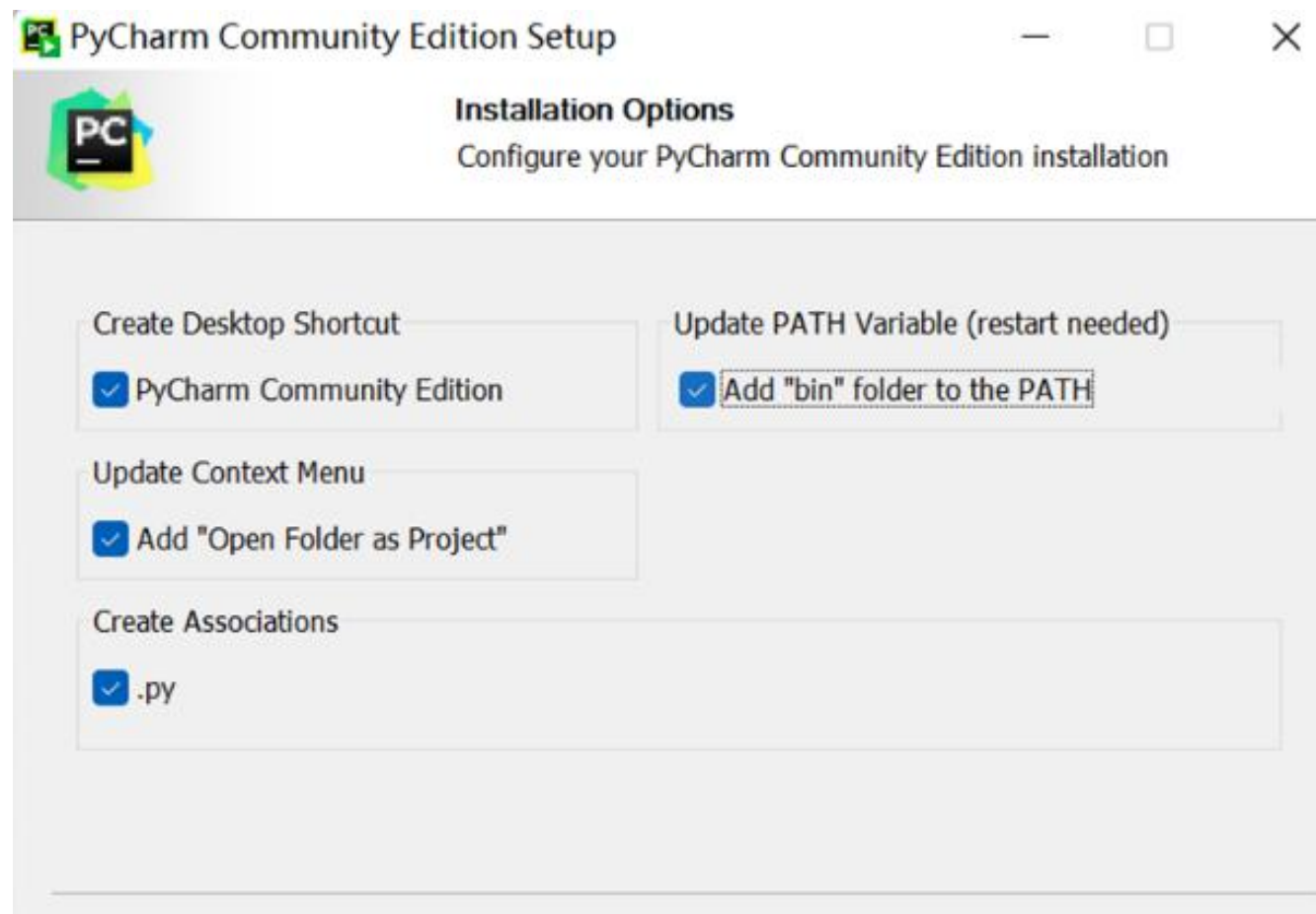
适用于纯 Python 开发

下载

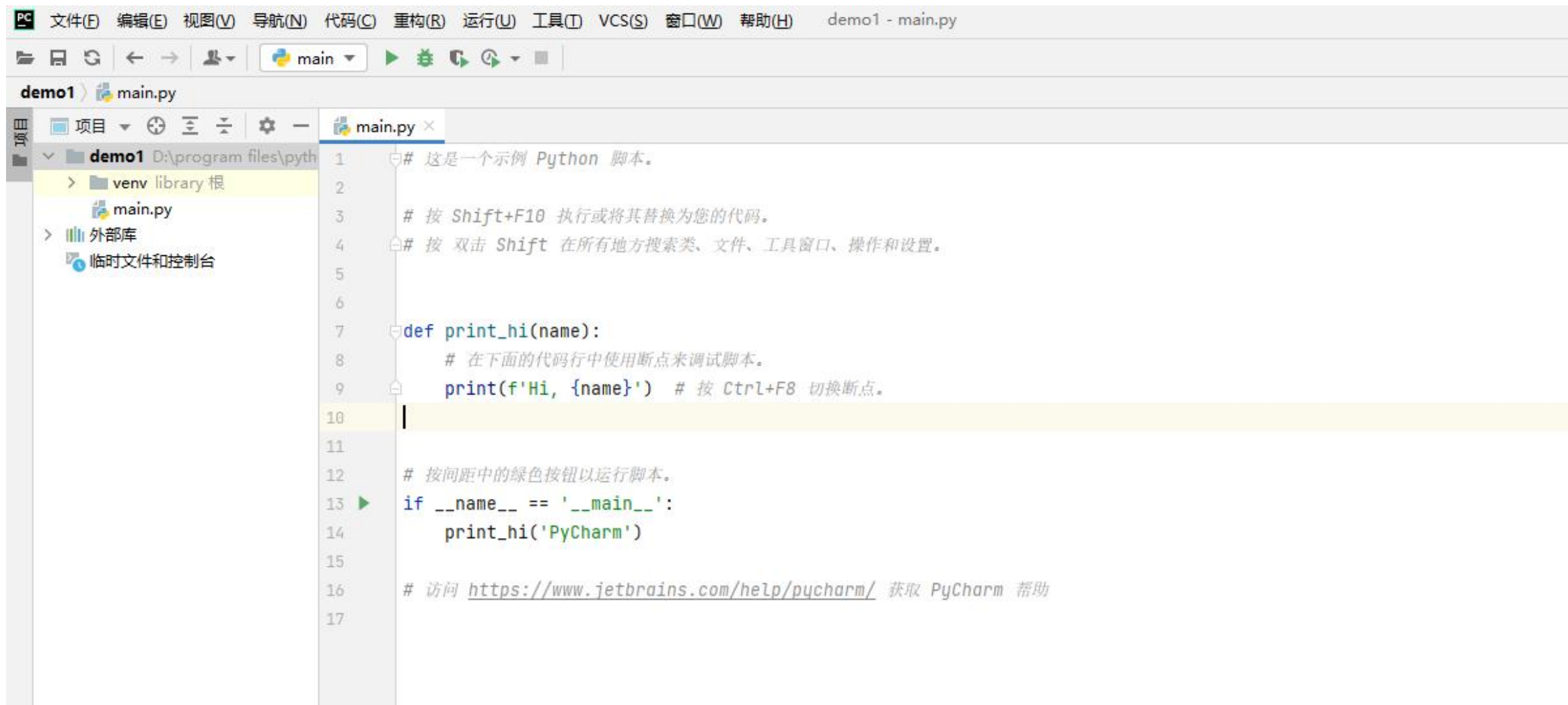
免费，开源构建



2.安装时选择自己安装路径，勾选如下选项，其他的全部默认即可。安装完后启动pycharm



3.新建项目后即可开始写python代码

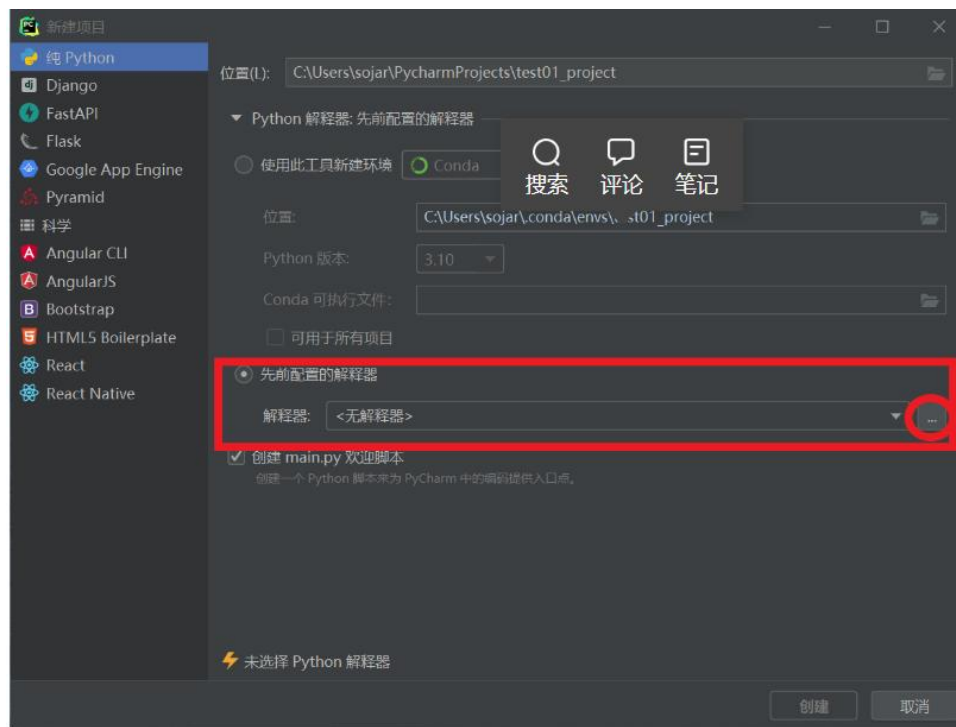


附件2：在Pycharm中配置conda的环境

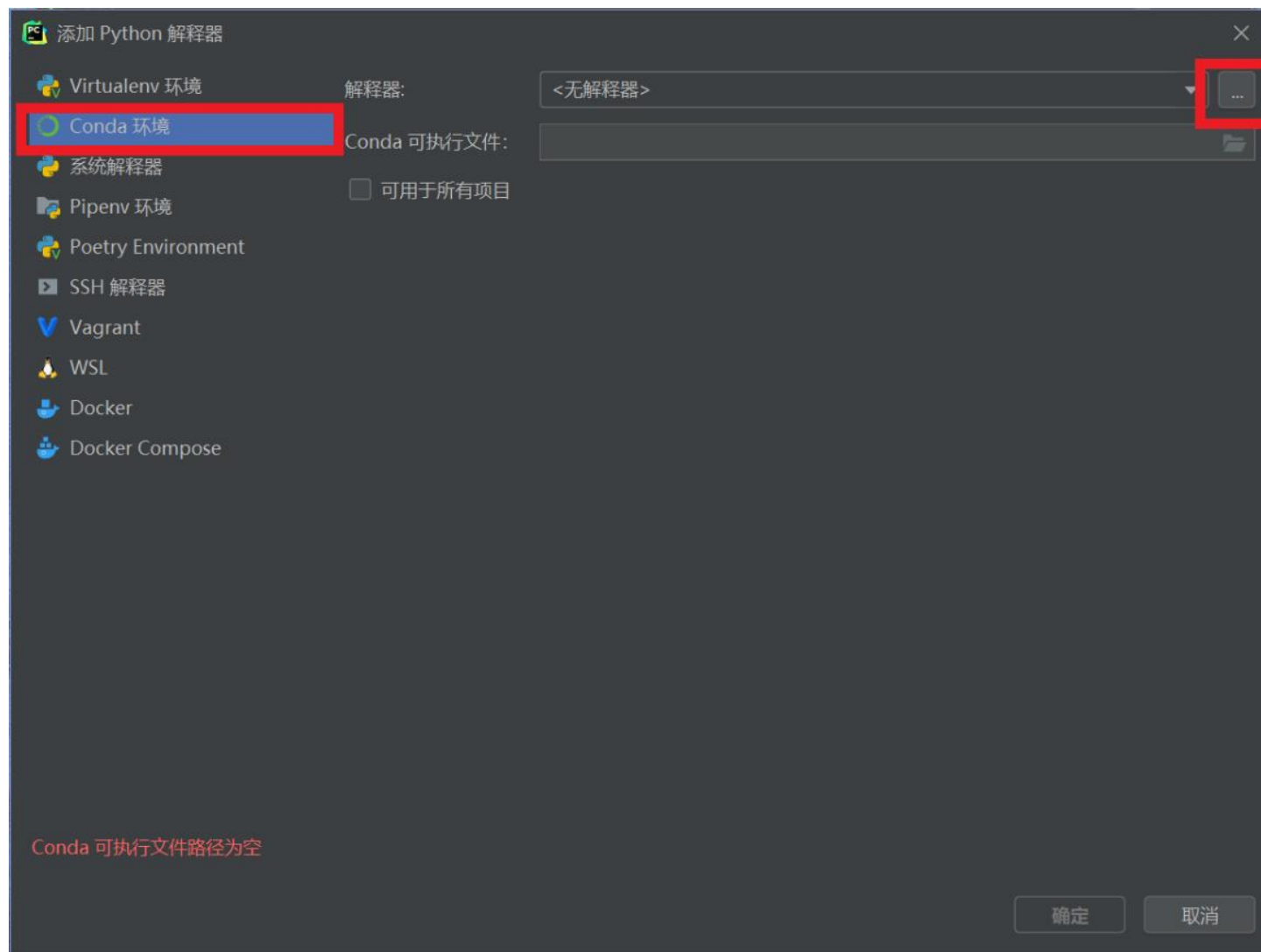
首先你要提前安装好Pycharm和Anaconda在电脑上

首先你要提前安装好Pycharm和Anaconda在电脑上，Anaconda在安装时会配置好一个python环境和带上很多python的第三方库，可以利用conda自带的python环境而不用另外给pycharm配置python环境

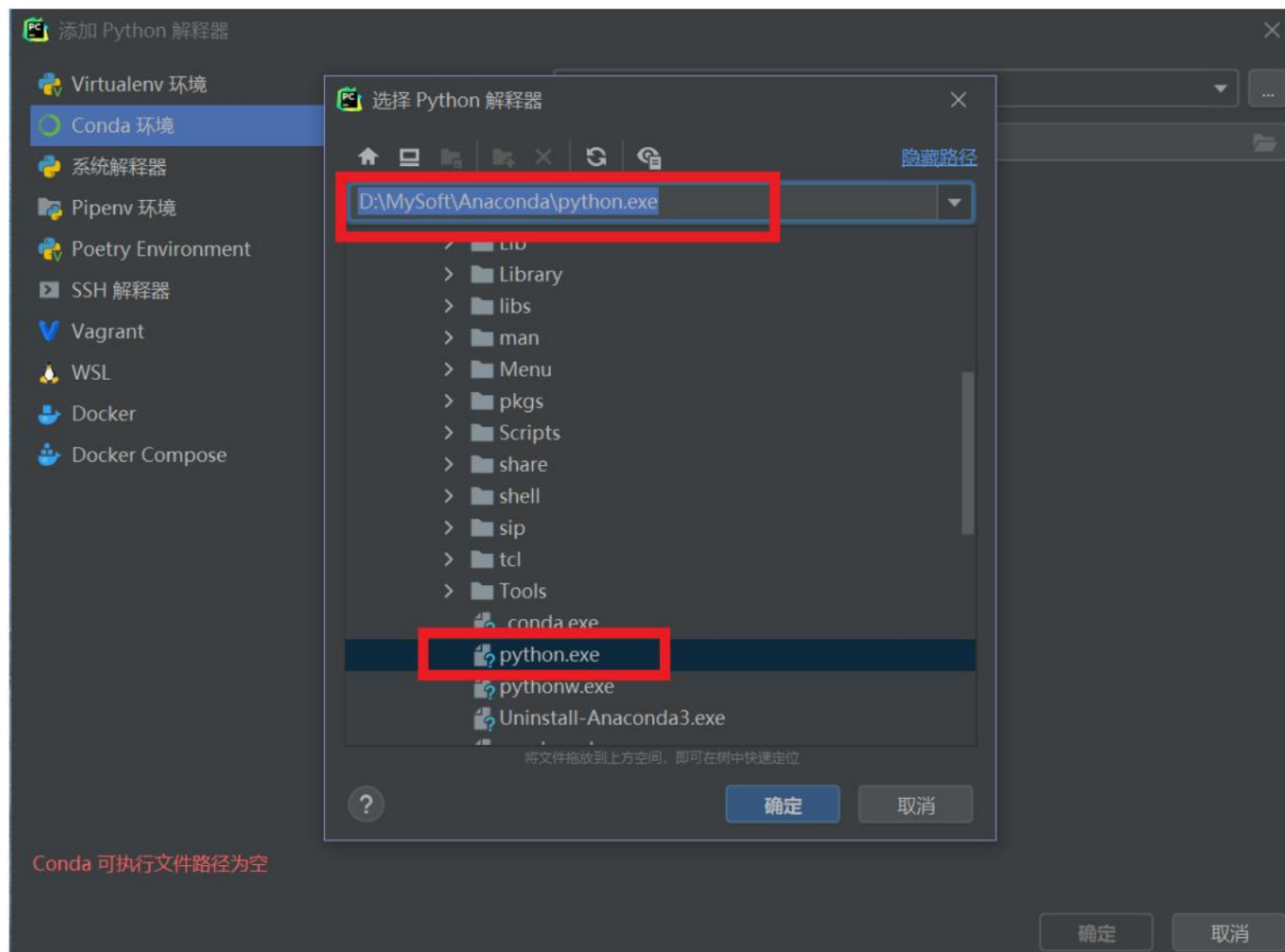
1、先在pycharm新建一个项目，选中如图所示的 先前配置的解器，再点击右边的那三个小点：



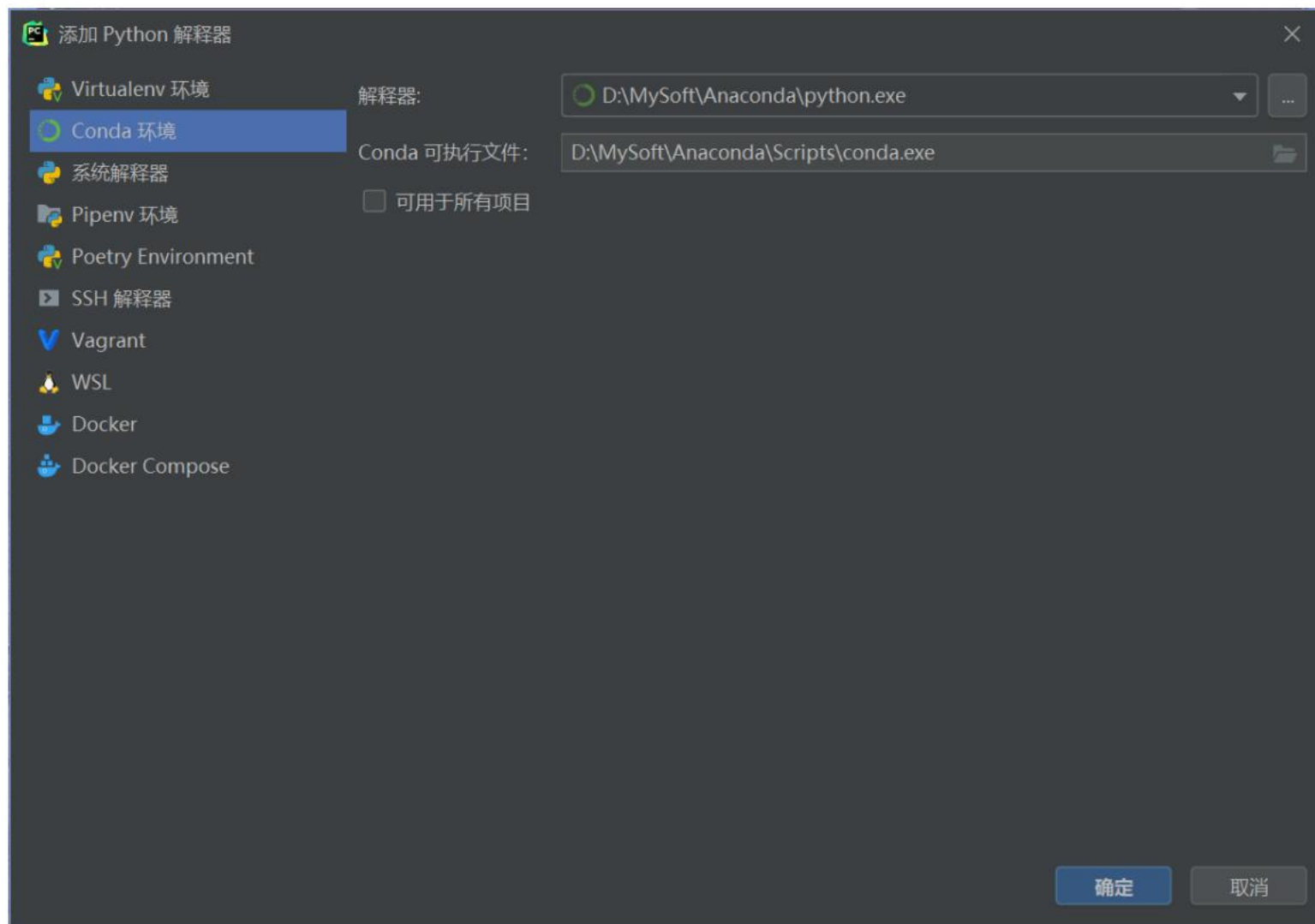
2、再点击左边的 Conda环境，点击右边的三个小点在Anaconda文件中寻找解释器：



3、找到你在电脑上的Anaconda的安装路径，并且在这个目录下找到python.exe(在Tools下面)，然后点击确认：



4、进入这个界面后继续点击确认：



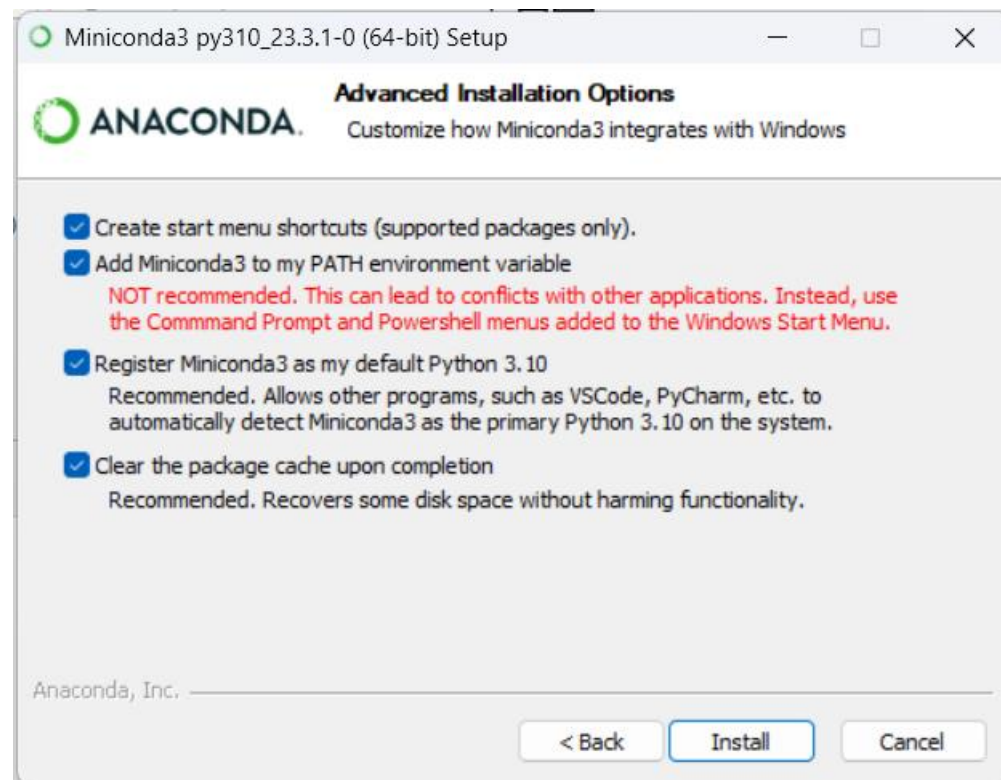
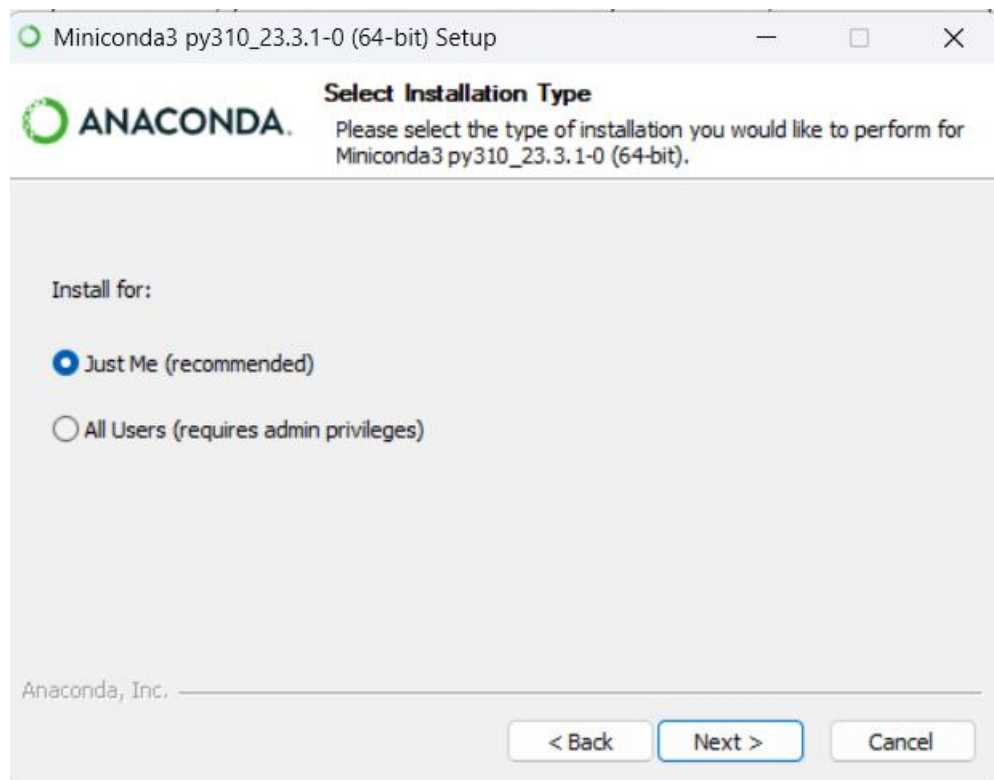
附件三：安装Anaconda/miniconda

- 下载地址：<https://docs.conda.io/en/latest/miniconda.html>
<https://www.anaconda.com/>
- 选择适合自己操作系统的版本，这里选择的是Windows64位版本

Latest Miniconda Installer Links

Latest - Conda 23.3.1 Python 3.10.10 released April 24, 2023

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	307194e1f12bbeb52b083634e89cc67db4f7980bd542254b43d3309eaf7cb358
	Miniconda3 Windows 32-bit	4fb64e6c9c28b88beab16994bfba4829110ea3145baa60bda5344174ab65d462
macOS	Miniconda3 macOS Intel x86 64-bit bash	5abc78b664b7da9d14ade330534cc98283bb838c6b10ad9cfd8b9cc4153f8104
	Miniconda3 macOS Intel x86 64-bit pkg	cca31a0f1e5394f2b739726dc22551c2a19afd689c13a25668887ba706cba58
	Miniconda3 macOS Apple M1 64-bit bash	9d1d12573339c49050b0d5a840af0ff6c32d33c3de1b3db478c01878eb003d64
	Miniconda3 macOS Apple M1 64-bit pkg	6997472c5ff90a772eb77e6397f4e3e227736c83a7f7b839da33d6cc7facb75d
Linux	Miniconda3 Linux 64-bit	aef279d6baea7f67940f16aad17ebe5f6aac97487c7c03466ff01f4819e5a651
	Miniconda3 Linux-aarch64 64-bit	6950c7b1f4f65ce9b87ee1a2d684837771ae7b2e6044e0da9e915d1dee6c924c
	Miniconda3 Linux-ppc64le 64-bit	b3de538cd542bc4f5a2f2d2a79386288d6e04f0e1459755f3cefe64763e51d16
	Miniconda3 Linux-s390x 64-bit	ed4f51afc967e921ff5721151f567a4c43c4288ac93ec2393c6238b8c4891de8



为本用户安装，并添加到环境变量

常用的conda命令

- 新建环境: `conda create -n 环境名 python==3.9`

例如: `conda create -n mindspore python==3.9`会创建一个名为mindspore的python环境。

- 激活环境: `conda activate 环境名`

默认的conda环境是base, 如图所示:

```
(base) C:\Users\16906>
```

使用`conda activate mindspore`可以切换到刚刚创建的python3.9环境:

```
(base) C:\Users\16906>conda activate mindspore
```

```
(mindspore) C:\Users\16906>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.
>>>
```


常用的conda命令

- 查看环境列表: conda env list

```
(mindspore) D:\python\code\feedforward>conda env list
# conda environments:
#
base                                D:\Anaconda
CSP_modified-master                D:\Anaconda\envs\CSP_modified-master
Facerecg_code                      D:\Anaconda\envs\Facerecg_code
                                   D:\Anaconda\envs\Facerecg_code\keshe
code                               D:\Anaconda\envs\code
huawei                              D:\Anaconda\envs\huawei
mindspore                          * D:\Anaconda\envs\mindspore
                                   D:\Anaconda\envs\papercode\adaptivw-inattention
python数据分析                     D:\Anaconda\envs\python数据分析
tensor                             D:\Anaconda\envs\tensor
tf                                  D:\Anaconda\envs\tf
untitled                           D:\Anaconda\envs\untitled
人工智能                           D:\Anaconda\envs\人工智能
                                   D:\Anaconda\envs\人工智能\env
```

- 删除环境: conda remove -n 环境名 --all