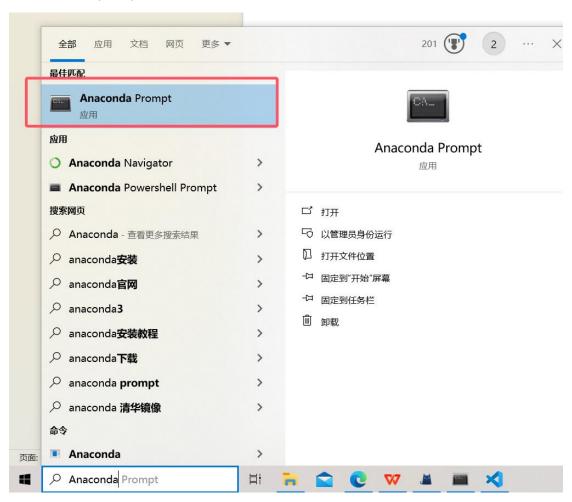
# 1.安装 Anconda

参考博客: https://blog.csdn.net/qq\_45281589/article/details/134597810

# 2.配环境(python 版本和相关的库)

打开 Anconda prompt



依次输入以下命令:

## 创建一个名为 study 的虚拟环境 指定 python 版本为 3.5

conda create --name study python=3.5

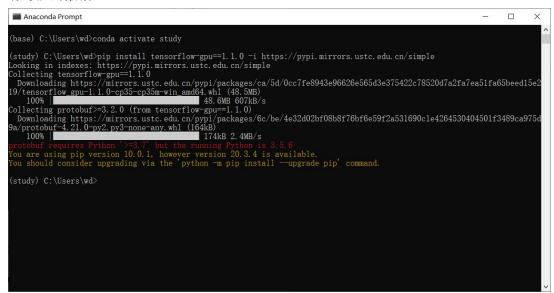
## 启用 study 这个虚拟环境

conda activate study

因为 tensorflow 需要 1.1.0 版本

#### pip install tensorflow==1.1.0 -i <a href="https://pypi.mirrors.ustc.edu.cn/simple">https://pypi.mirrors.ustc.edu.cn/simple</a>

#### 有可能会报错



### 此时需要升级 pip

python -m pip install --upgrade pip -i https://pypi.mirrors.ustc.edu.cn/simple 再次运行

pip install tensorflow==1.1.0 -i <a href="https://pypi.mirrors.ustc.edu.cn/simple">https://pypi.mirrors.ustc.edu.cn/simple</a>

```
(study) C:\Users\wd\pip install tensorflow-gpu==1.1.0 -i https://pypi.mirrors.ustc.edu.cn/simple

DEPRECATION: Python 3.5 reached the end of its life on September 13th, 2020. Please upgrade your Python as Python 3.5 is
no longer maintained. pip 21.0 will drop support for Python 3.5 in January 2021. pip 21.0 will remove support for this
functionality.

Looking in indexes: https://pypi.mirrors.ustc.edu.cn/simple

Collecting tensorflow-gpu==1.1.0

Using cached https://mirrors.ustc.edu.cn/pypi/packages/ca/5d/0cc7fe8943e96626e565d3e375422c78520d7a2fa7ea51fa65beed15e
219/tensorflow.gpu=1.1.0-cp35-cp35m-win_amd64.whl (48.5 MB)

Collecting six>=1.10.0

Downloading https://mirrors.ustc.edu.cn/pypi/packages/b7/ce/149a00dd41f10bc29e5921b496af8b574d8413afcd5e30dfa0ed46c2cc
5e/six-1.17.0-py2.py3-none-any.whl (11 kB)

Collecting werkzeug>=0.11.10

Downloading https://mirrors.ustc.edu.cn/pypi/packages/cc/94/5f7079a0e00bd6863ef8f1da638721e9da21e5bacee597595b318f71d6

2e/Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)

298 kB 2.2 MB/s

Requirement already satisfied: wheel>=0.26 in d:\anaconda3\envs\study\lib\site-packages (from tensorflow-gpu==1.1.0) (0.37.1)

Collecting numpy>=1.11.0

Downloading https://mirrors.ustc.edu.cn/pypi/packages/ed/09/ff8f529a5548ff788765f66a81ef751130f26f8c7d517e94d3dbf3bale

d5/numpy-1.18.5-cp35-cp35m-win_amd64.whl (12.7 MB)

12.7 MB 3.2 MB/s

Collecting protobuf>=3.2.0

Downloading https://mirrors.ustc.edu.cn/pypi/packages/32/27/1141a8232723dcb10a595cc0ce4321dcbbd5215300bf4acfc142343205

bf/protobuf-3.19.6-py2.py3-none-any.whl (162 kB)

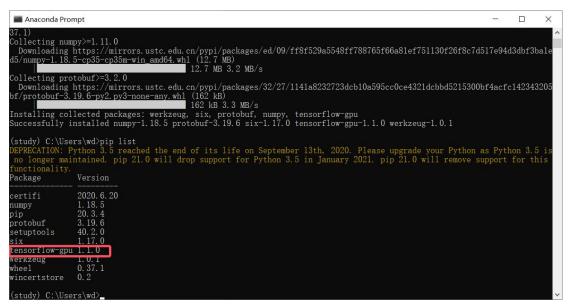
162 kB 3.3 MB/s

Installing collected packages: werkzeug, six, protobuf, numpy, tensorflow-gpu-1.1.0 werkzeug-1.0.1

(study) C:\Users\wd>_
```

应该是安装成功啦!!

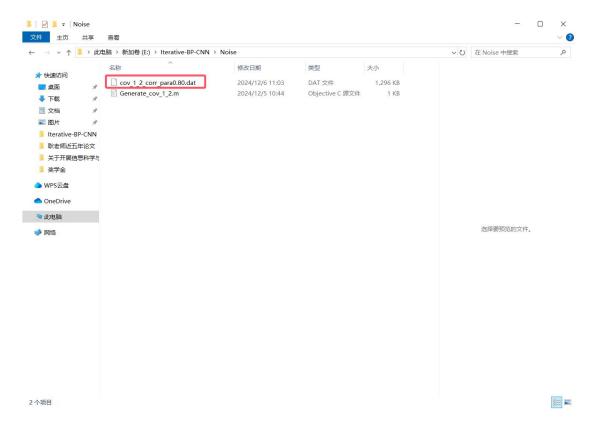
## 输入 pip list 看是否成功安装 tensorflow



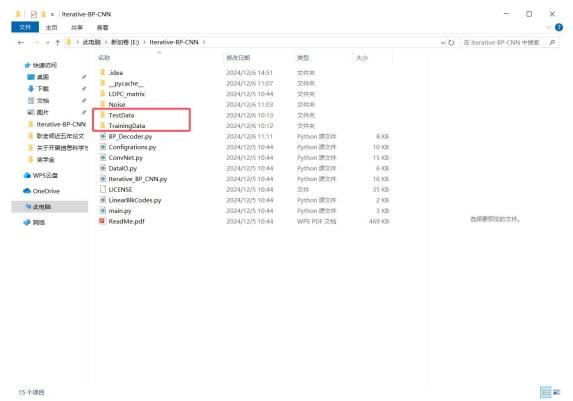
确实成功啦!!!!!!! 开心~

# 3.运行项目

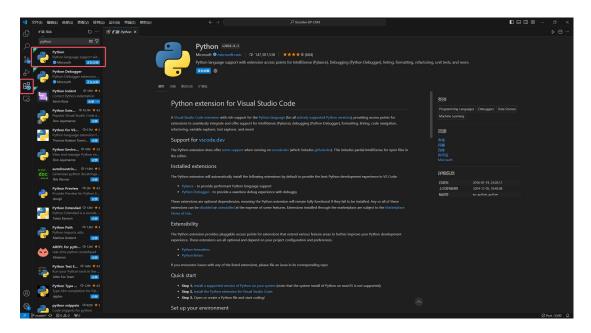
首先运行 matlab 文件 在 Noise 目录下会生成数据



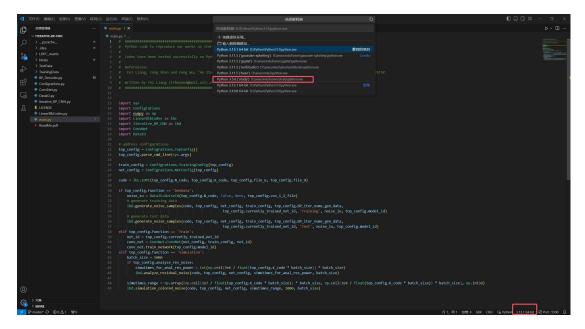
## 创建两个空的文件夹 TestData 和 TrainingData



用 vscode 打开项目(我个人喜换 vscode~~ 其实是 pycharm 出现问题啦 我不会!!) 安装 python 的扩展



接下来配置好 python 版本 就用刚才的 3.5 版本 找到它! study 那个环境



查看配置文件 Configrations.py 大概在第七行有一行 self.function = 'Train'的 代码 这里的注释分别就代表着 Readme.pdf 里面的三步,GenData Train 以及

## Simulation

首先需要改个东西(这个作者把配置写死啦 实际生成的文件后缀是 80 它是 50 修改一下 改为 0.8)

```
文件(E) 编辑(E) 选择(S) 查看(M) 转到(G) 运行(B) 终端(I) 帮助(H)

♠ Configrations.py ×

                                                                           Ch Ch O O Configrations.py > STopConfig > Ch _ init_ import numpy as np
                                                                                                                                   ## TopConfig defines some top configurations. Other configurations are set based on TopConfig.

class TopConfig:

def_init_(self):

# select_functions to be
                   > __pycache__
> .idea
> LDPC_matrix
                 ✓ Noise

E cov_1_2_corr_pa a0.80.dat U
                                                                                                                                                                C Generate_cov_1_2.m
> TestData
> TrainingData
                   BP_Decoder.py
                   ConvNet.py
                  DatalO.py
Iterative_BP_CNN.py
                                                                                                                                                               # noise information
self.blk [an - self.hl code
self.corr_para = 0.5  # correlation parameters of the colored noise
self.corr_para simu = self.corr_para # correlation parameters for simulation. t
self.cov_1_2 file = format(','MoiseCov_1_2.corr_para%.2f.dat'% self.corr_para)
self.cov_1_2_file_simu = self.cov_1_2_file

    ↑ LICENSE
    ◆ LinearBlkCodes.py
    ◆ Linea
                  main.py
ReadMe.pdf
                                                                                                                                                                 self.BP_iter_nums_gen_data = np.array([5])  # the number of BP iterations
self.BP_iter_nums_simu = np.array([5,5])
                                                                                                                                                               # con config

self.currently_trained_net_id = 0 # denote the cnn denoiser which is in training currently

self.cun_net_number = 1 # the number of cnn denoisers in final simulation

self.layer_num = 4 # the number of cnn layers

self.filter_sizes = np.array([9,3,3,15]) # the convolutional filter size. The length of this list should be equal to the self.fistore_network_from_file = False # whether to restore previous saved network for training

self.model_id = np.array([0]) # differentiate models trained with the same configurations. Its length should be equal # the ith network in the BP-CNN-DP-CNN-... structure.
Configrations.py X
       🍨 Configrations.py > ધ TopConfig > 🛇 _init_
                                                def __init__(self):
    __# select functions to be executed, including generating data(GenData), training(Train), and sim
                                                            self.function = 'Train'
                                                                self.file_6 = format('./LDPC_matrix/LDPC_gen_mat_%d_%d.txt' % (self.N_code, self.K_code))
self.file_H = format('./LDPC_matrix/LDPC_chk_mat_%d_%d.txt' % (self.N_code, self.K_code))
                                                                 self.blk_len = self.N_code
                                                                 self.corr_para = 0.5 # correlation parameters of the colored noise
self.corr_para_simu = self.corr_para # correlation parameters for simulation. this should be
                                                                 self.cov 1 2 file = format('./Noise/cov 1 2_corr_para%.2f.dat'% self.corr_para)
self.cov_1_2_file_simu = self.cov_1_2_file
```

## 首先将 Train 改为 GenData 用于生成数据 然后运行 main.py

```
| Second Content of the Content of t
```

## 后续同理!!!

再改回 Train 运行 main.py 用于训练

最后改为 Simulation 用于仿真