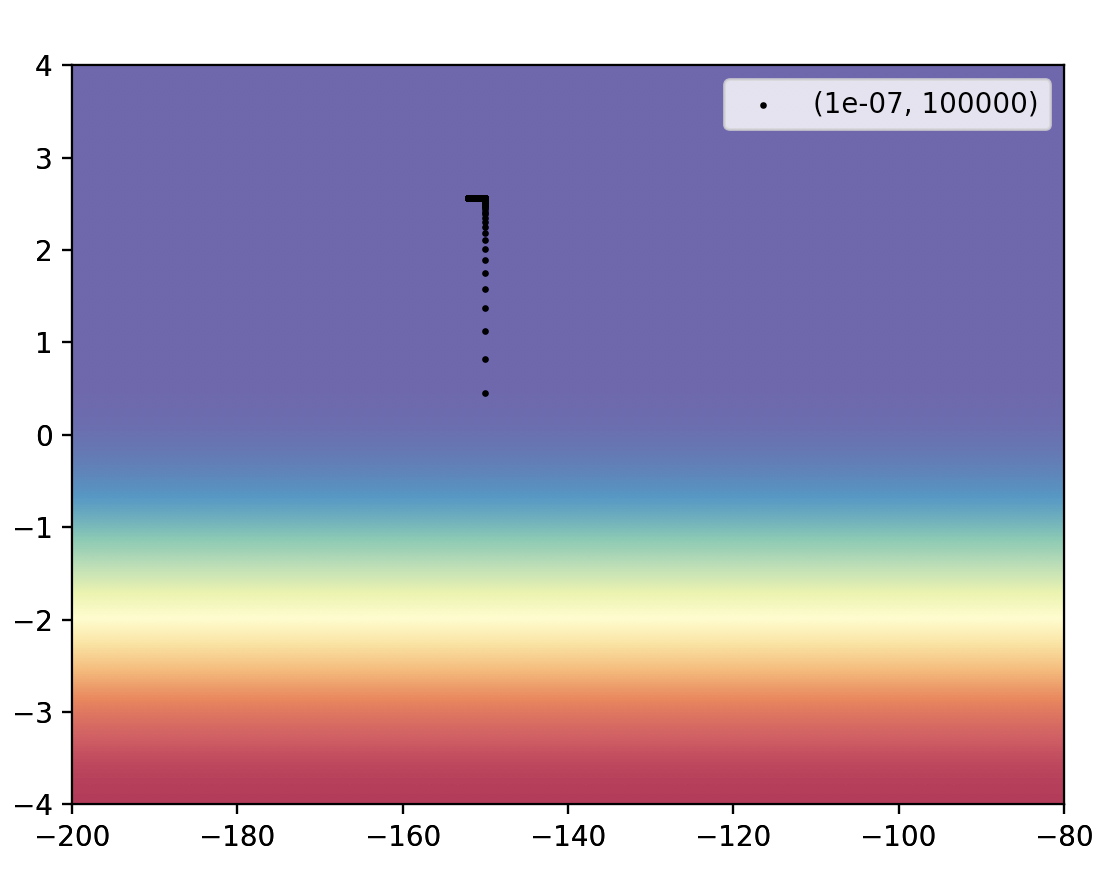
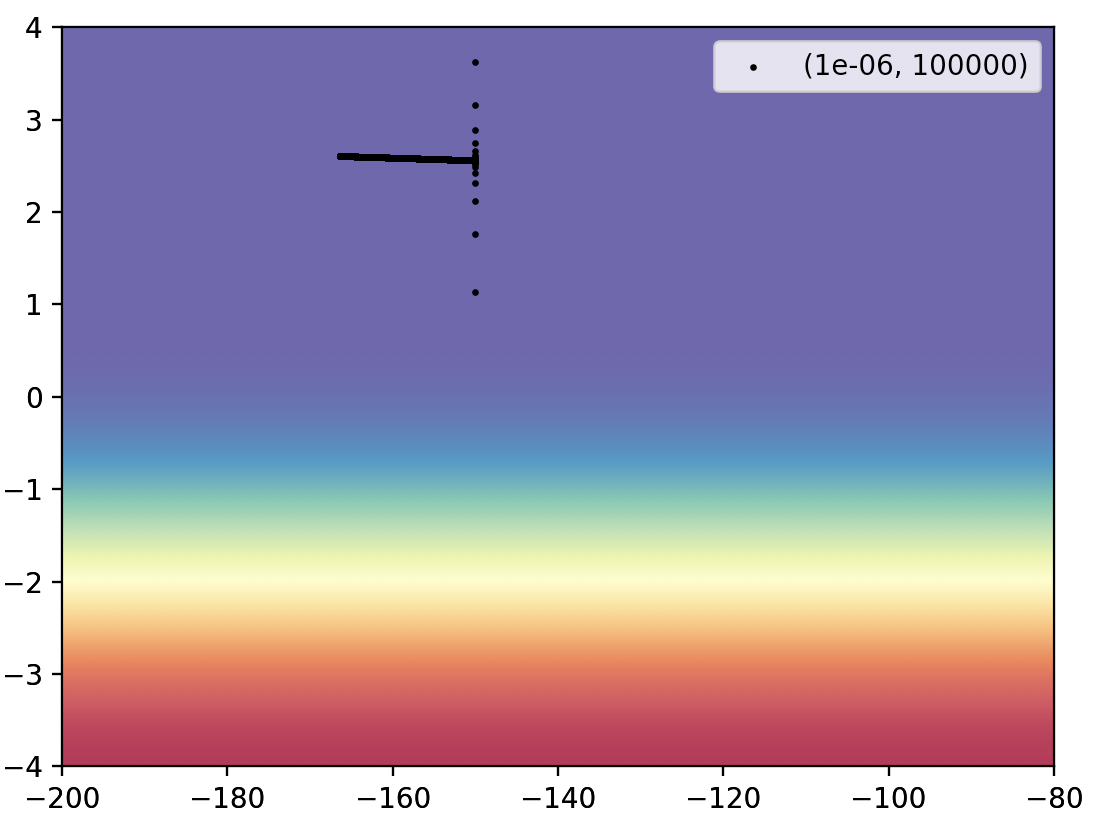
任务1

梯度下降算法

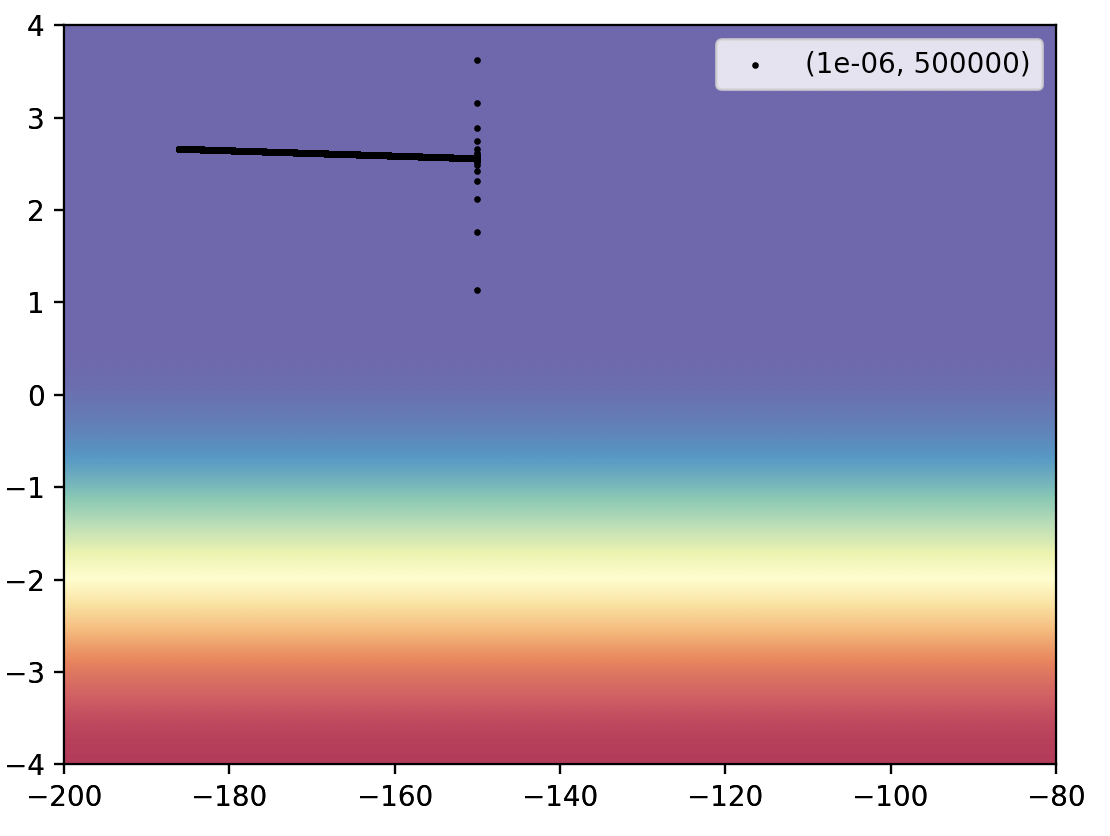
学习率：0.0000001，迭代次数：100000；



学习率：0.000001，迭代次数：100000；

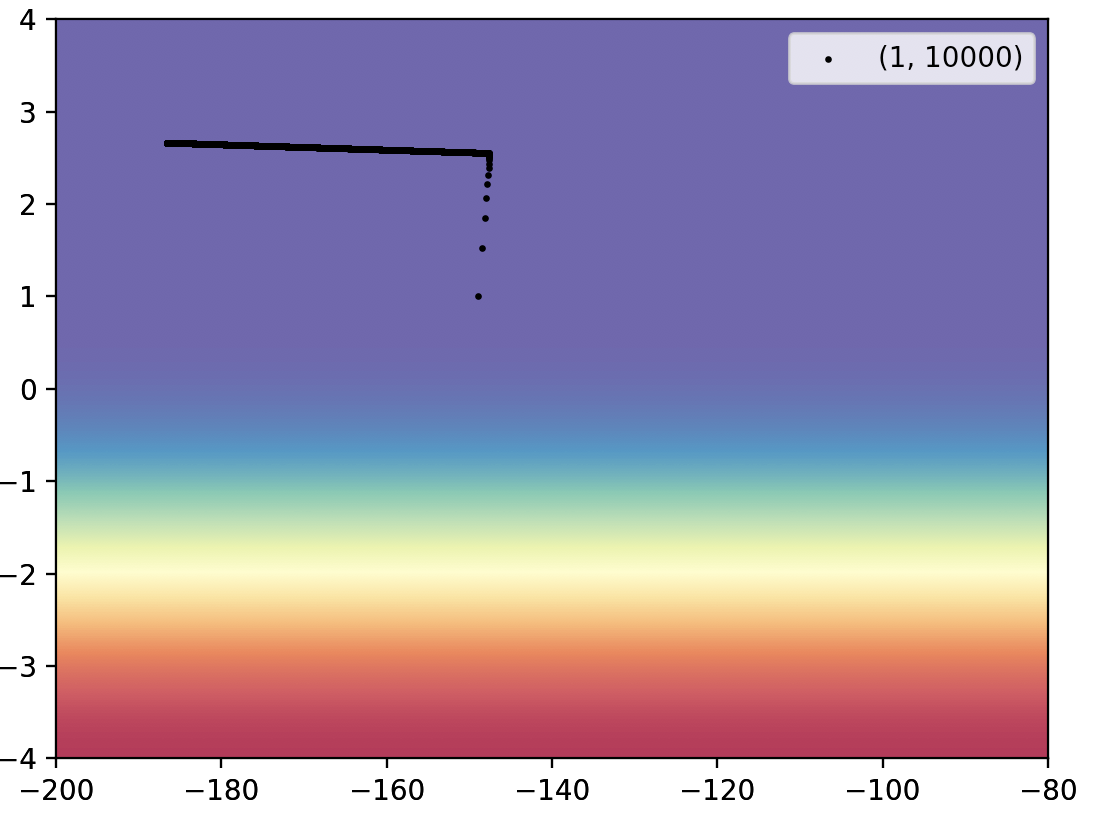


学习率：0.000001 迭代次数：500000

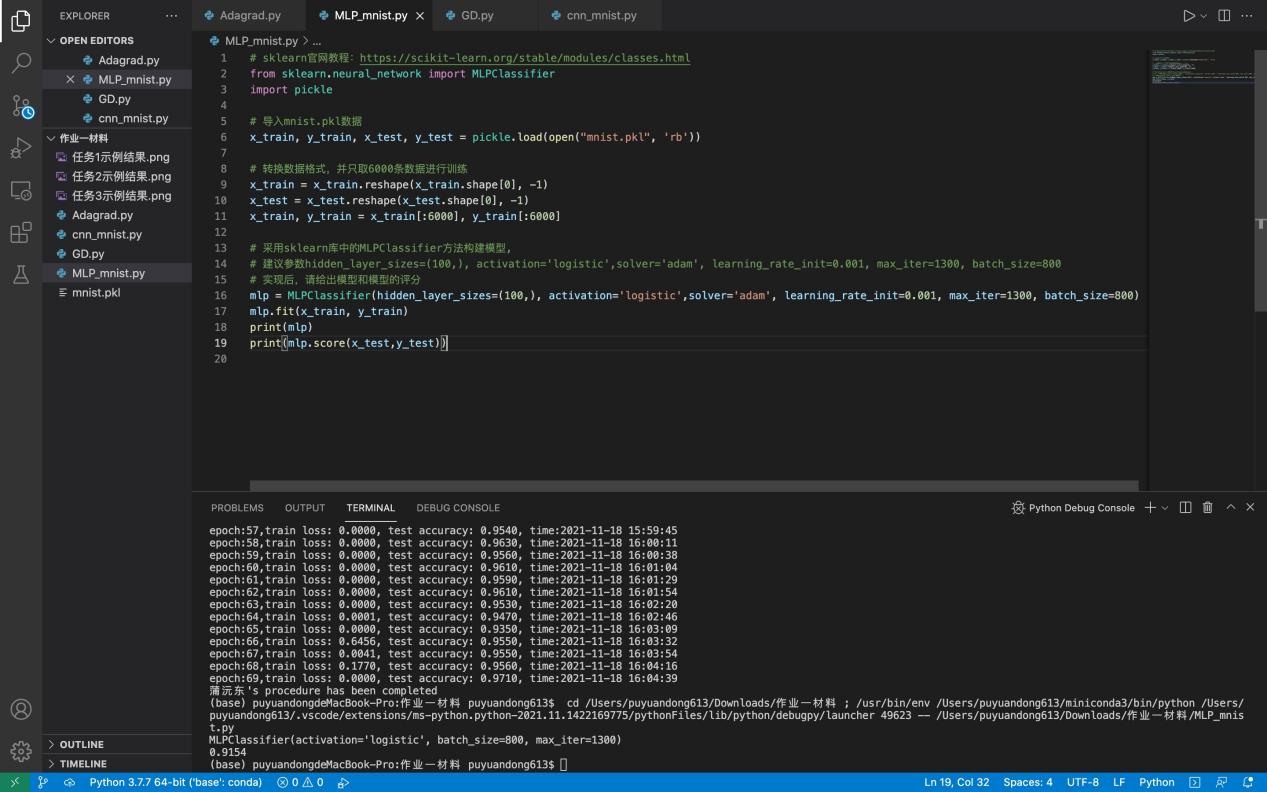


Adagrad算法

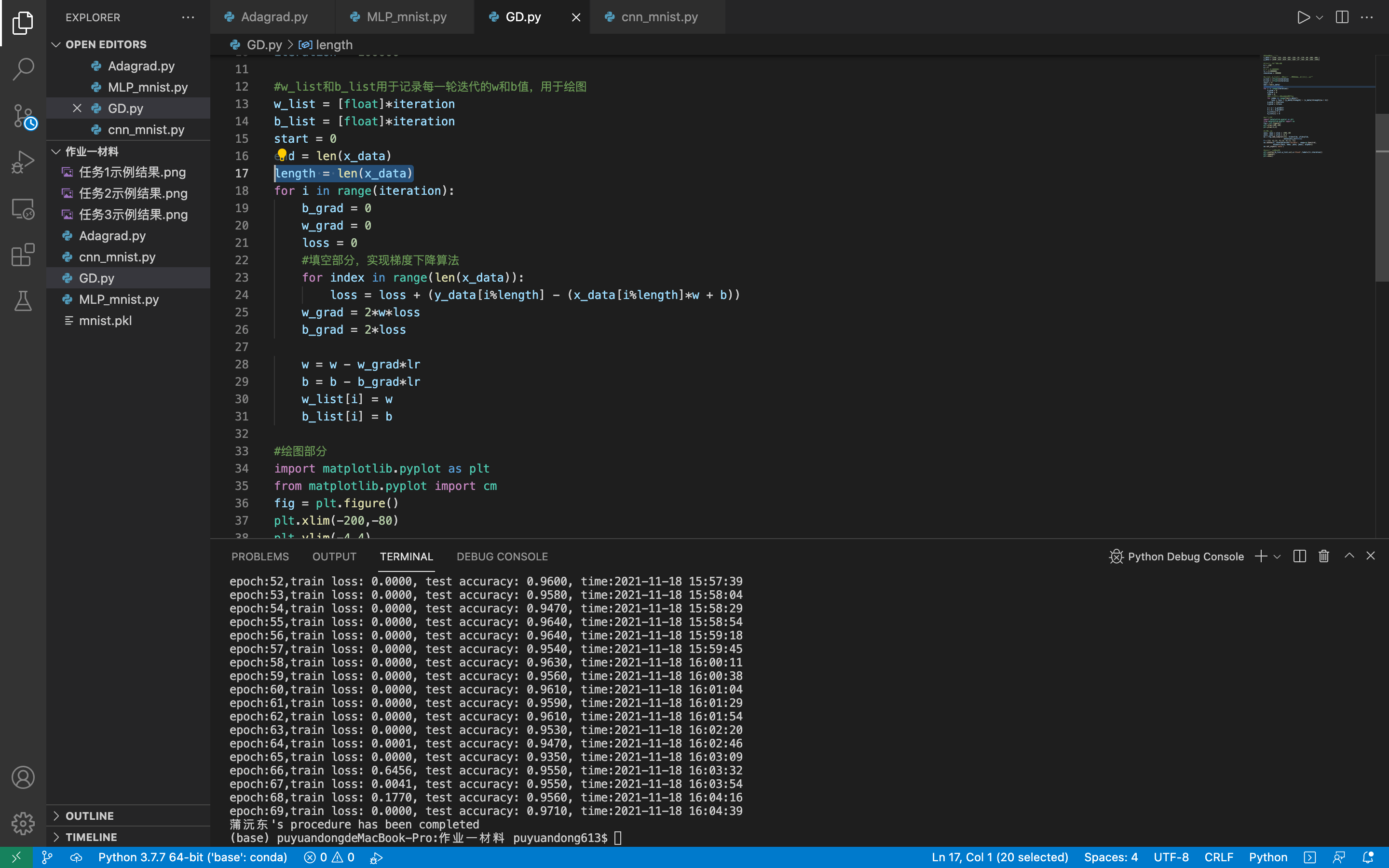
学习率：1，迭代次数：10000 的绘图结果



任务2：



任务3：



开放问题：

1. 梯度下降算法和Adagrad算法中参数设置对算法性能的影响；

学习率变小会放慢梯度下降的速度，学习率变大会提高梯度下降的速度，但是有可能错过全局最小点，这样会导致结果震荡。

迭代次数少可能导致模型收敛不了，迭代次数多可能会导致模型速度变慢。

1. MLP模型中学习率、隐藏层大小、训练轮次对模型性能的影响；

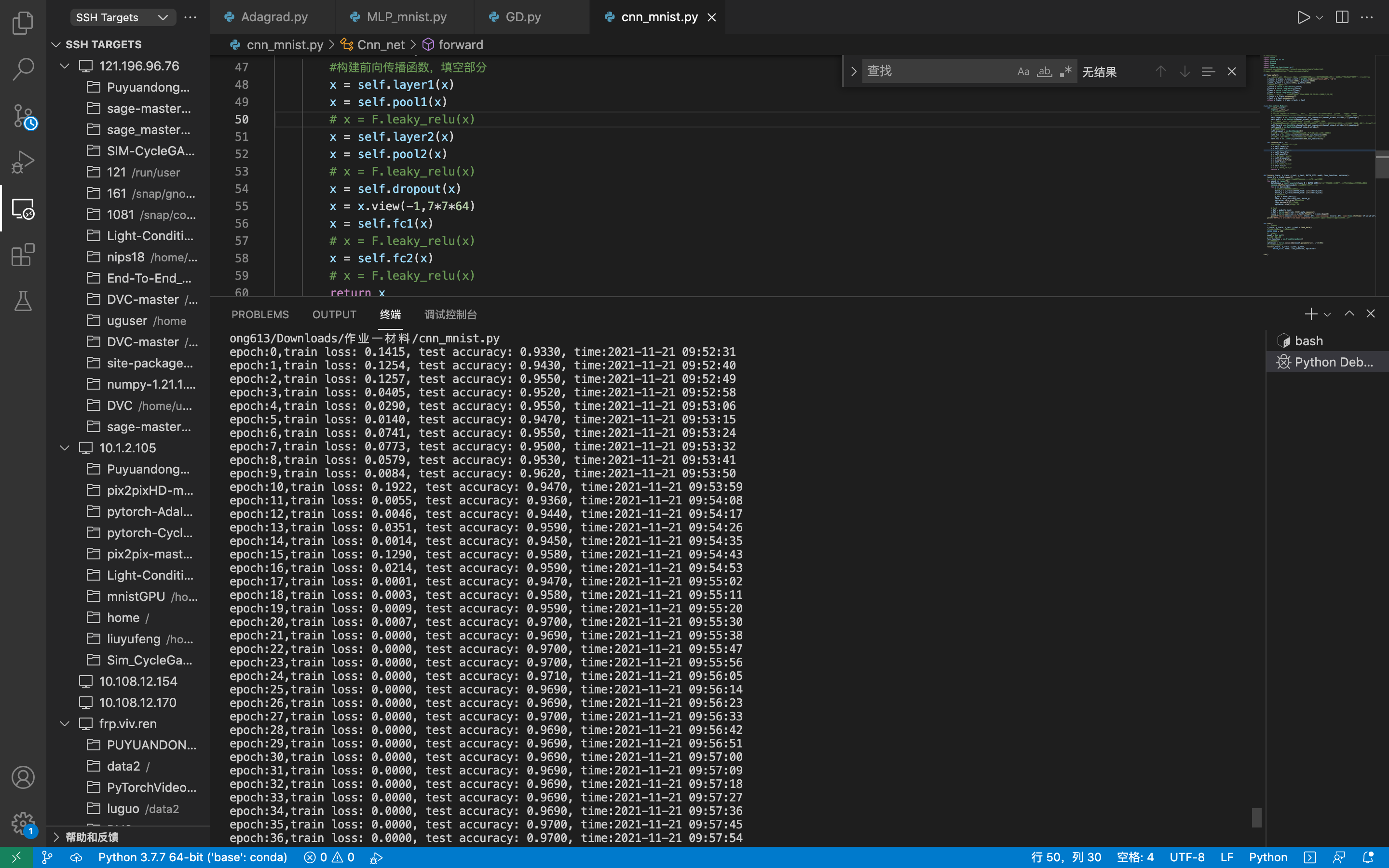
学习率变小会减缓梯度下降的速度，学习率变大会加速梯度下降的速度，但是有可能错过全局最小点，这样会导致结果不准确。

训练轮次太少会导致模型不能够有效收敛，训练轮次太多会导致模型学习速度较慢。

隐藏层大小导致改变模型的网络结构，会改变最终模型的训练结果。往往增加隐藏层数，加深网络结构会比增加隐藏层纬度，加宽网络结构有更好的效果（当然太深也不行，参考Resnet）

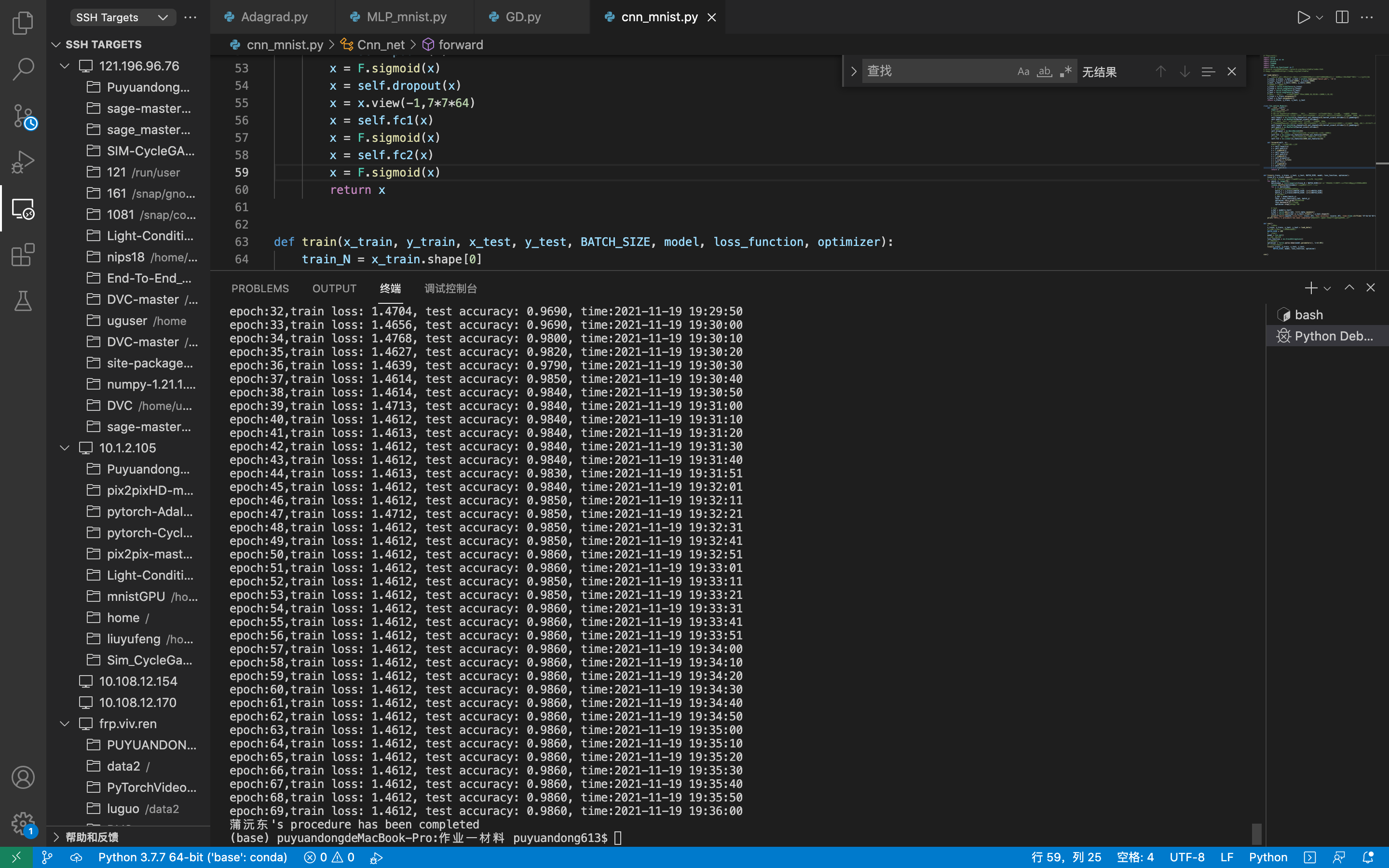
1. 尝试对CNN模型中的激活函数、卷积层参数、池化层参数进行修改，并分析其影响。
   * 1. 激活函数无激活函数

结果：97.10%



* + 1. Sigmoid

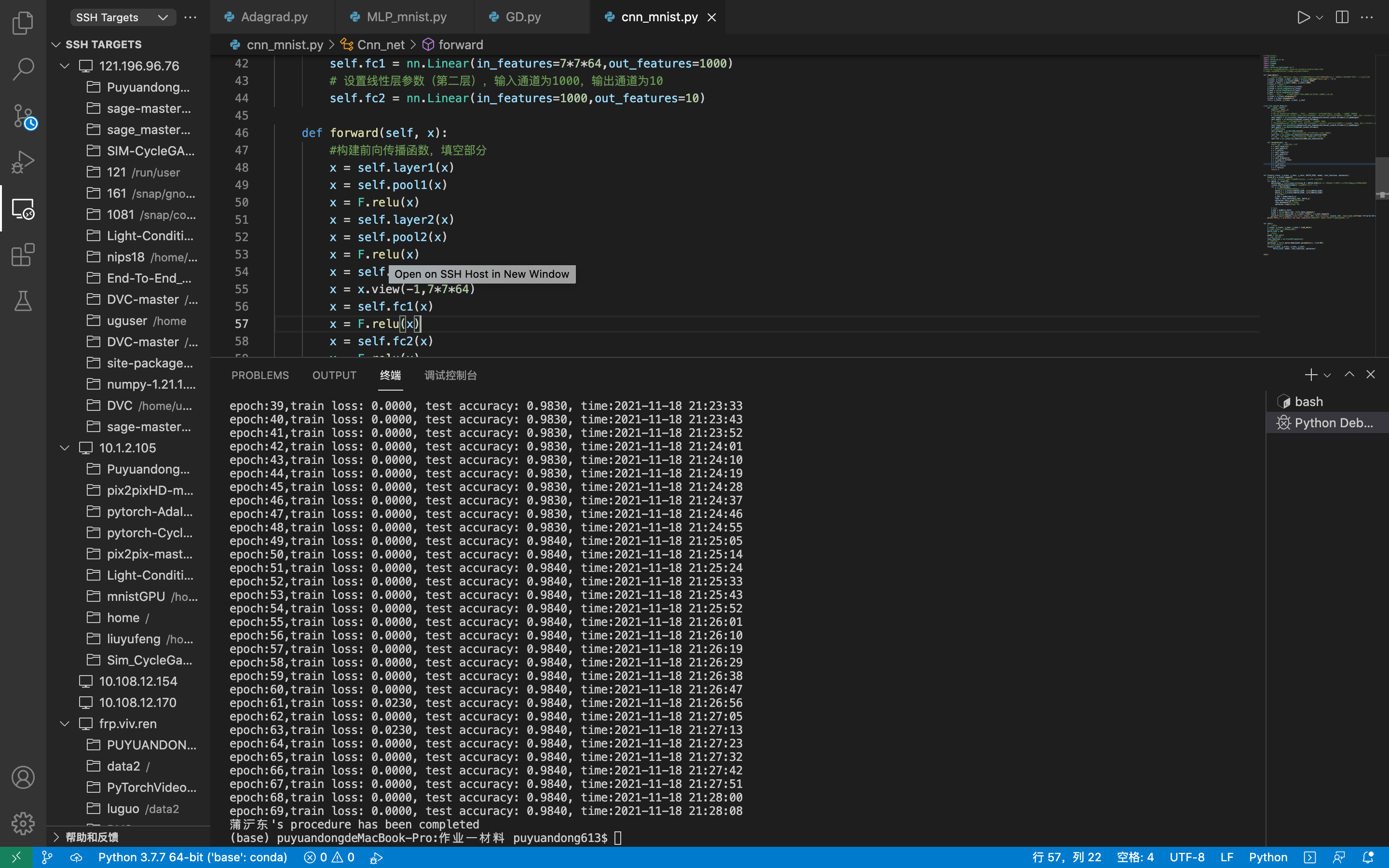
结果：98.60%



将激活函数换为sigmoid过后，可以看出最后的训练结果有了提升，但是在训练过程当中有略微的震荡

* + 1. Relu

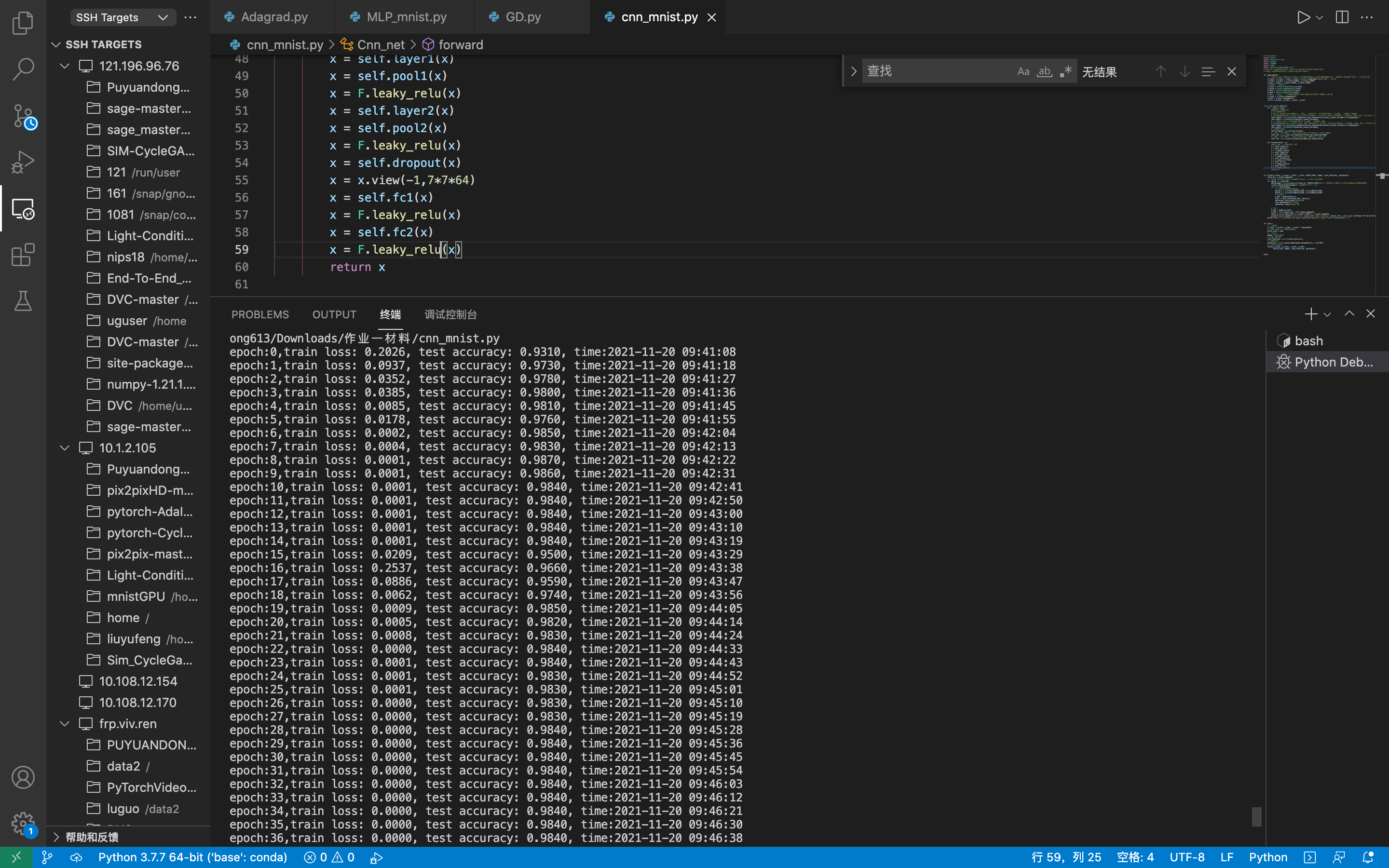
结果：98.40%



将激活函数改为relu过后，训练效果相比于无激活函数有提升，但是效果不如sigmoid。在训练过程当中比较平稳

* + 1. leakyRelu

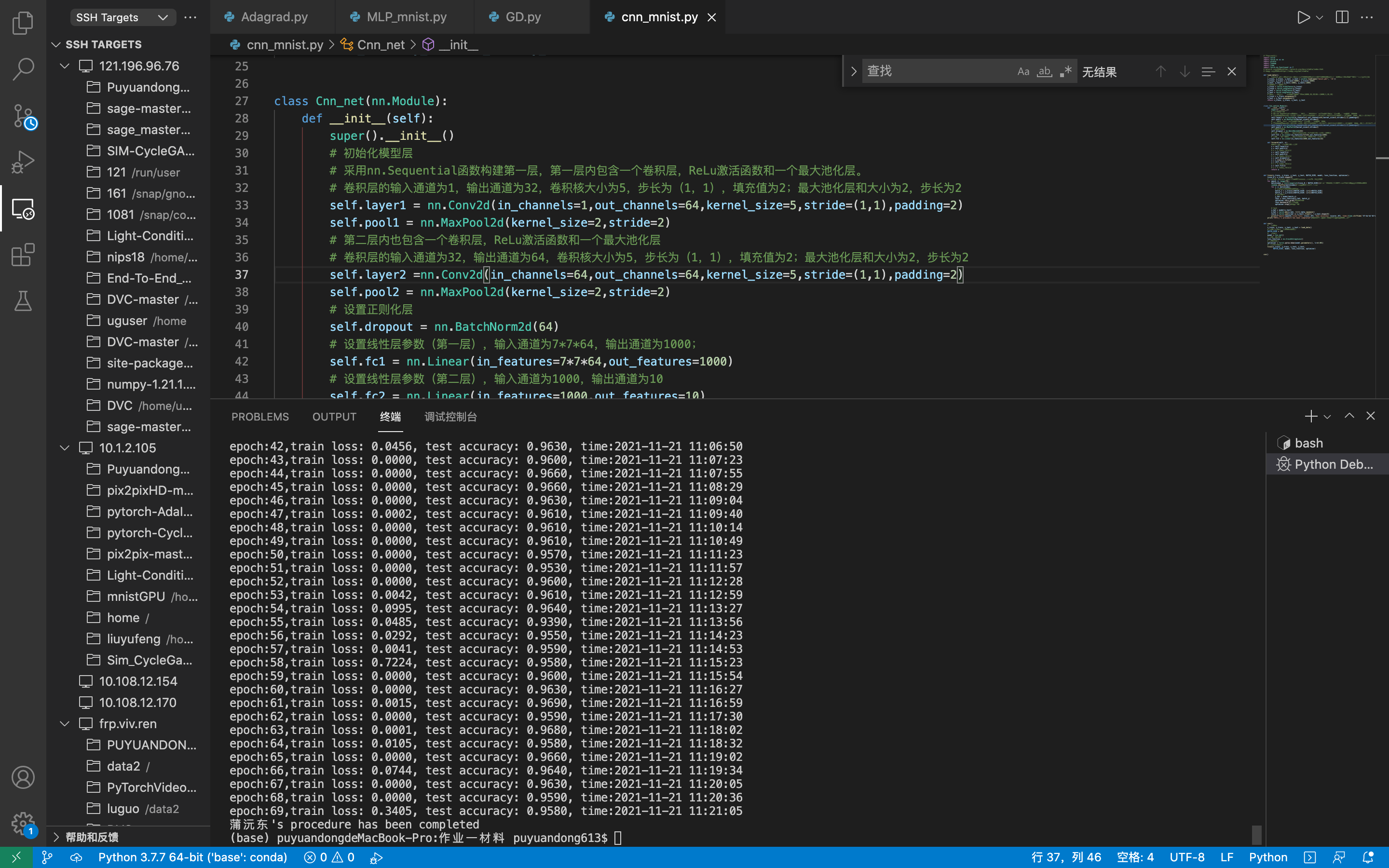
结果：98.50%



将激活函数改为leakyRelu过后，训练效果比无激活函数有了提升，但是效果仍不比sigmoid

* 1. 卷积层参数

我将卷积层的第一层参数的输出改为64，则结果为



可以看出训练效果有一定的减弱，并且可以发现模型训练并未收敛

* 1. 池化层参数

将池化层的参数改为：kernel\_size = 3 ， stride = 3过后，发现训练效果有所提升，收敛速度更加快，最后的准确率为98.5%

