1. 构建模型与说明

训练中采用了PyTorch模块的Sequential模块进行神经网络的构建，本实验报告中会对不同的网络层数、不同的神经元个数、使用不同的激活函数分别采用不同的训练次数以测试网络性能，在所有训练中学习率取0.001，每次测试进行3次随机划分训练集和测试集后的训练和预测，并取平均值作为最终

1. 测试
   1. 激活函数ReLU，神经元个数64

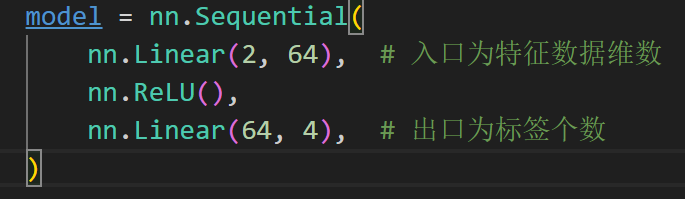
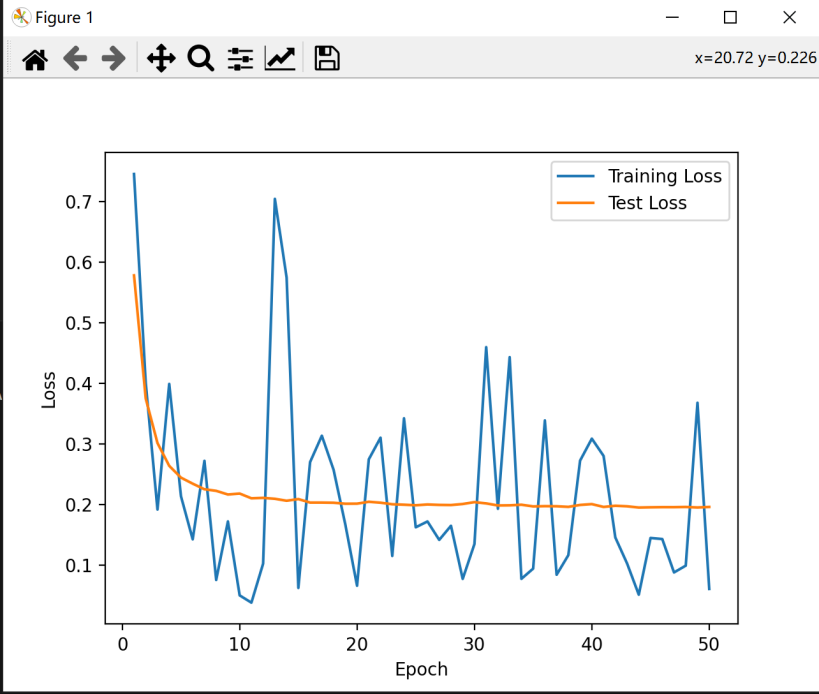
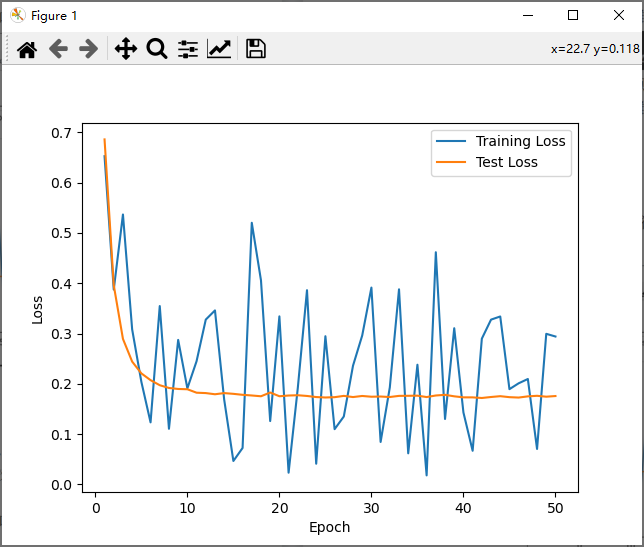
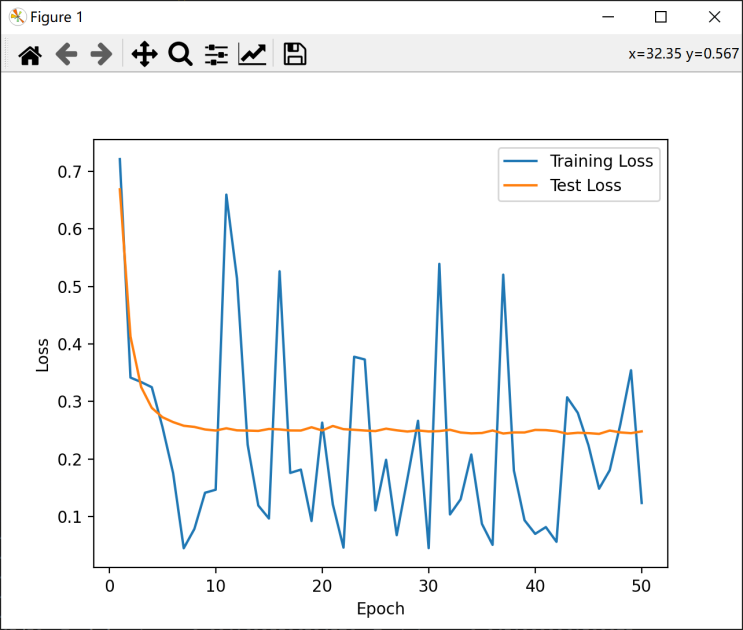


图 1 模型

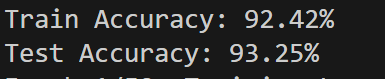
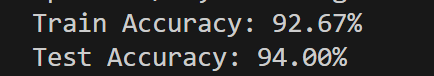
* + 1. 第一次测试：训练次数50，测试集与训练集损失如下图所示

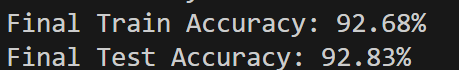
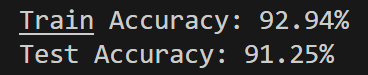




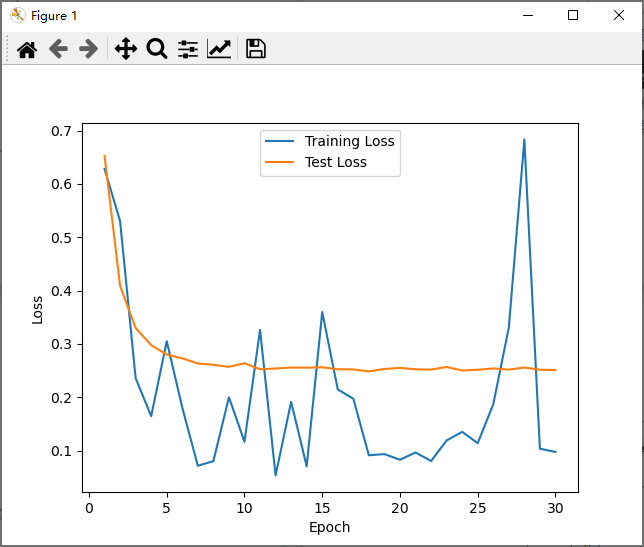


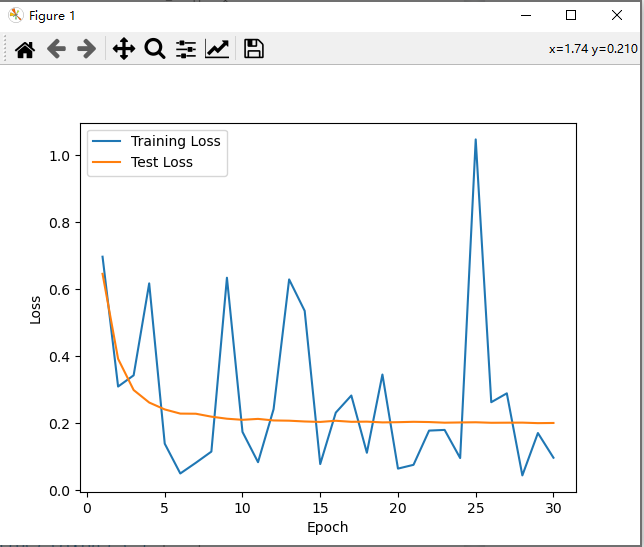
最终训练集准确率92.68%，测试集准确率92.83%，如下图所示

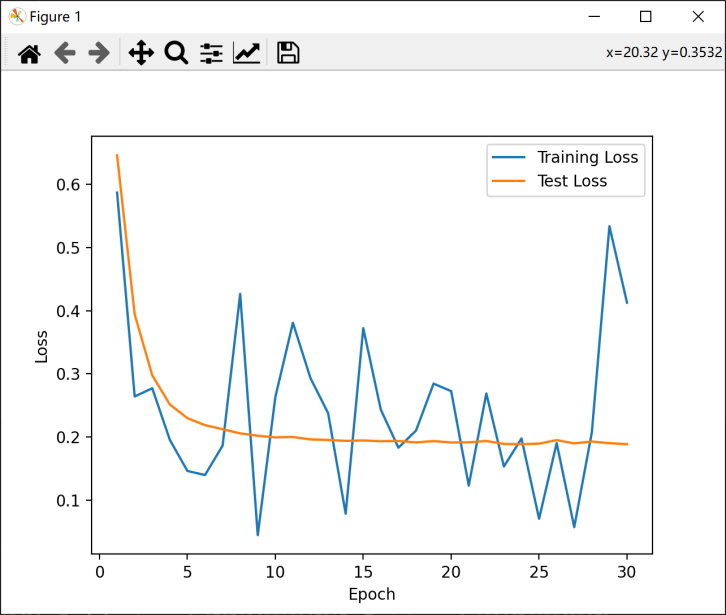


可以看到，测试集损失在20轮以内就趋近最小值，并基本平缓，而训练集损失一直在跳变，再增加训练轮数意义不大，故第三次训练将训练次数下调至30轮

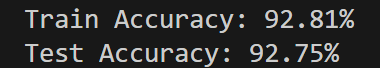
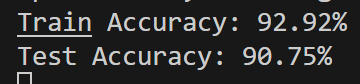
* + 1. 第二次训练，训练次数30，测试集与训练集损失如下图所示

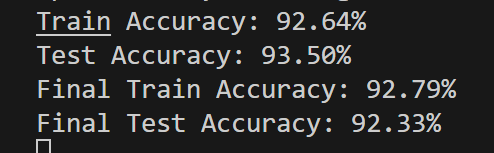






最终训练集准确率92.79%，测试机准确率92.33%，如下图所示

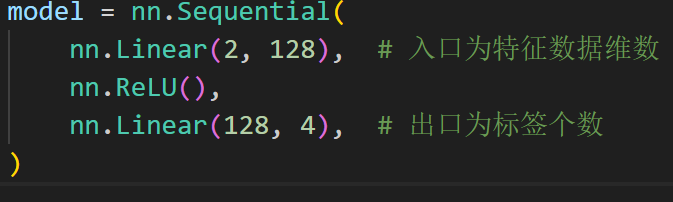




总结：

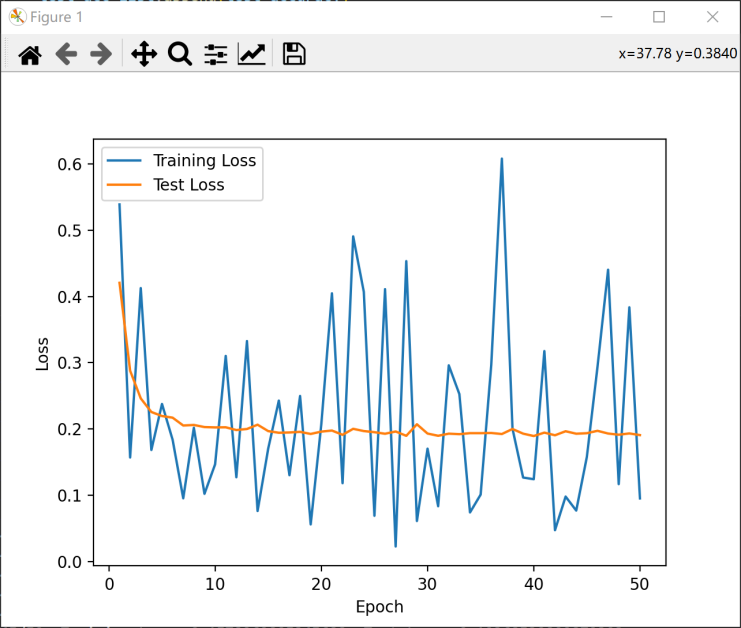
可以看到，测试集损失在30轮训练内以趋向最小值，并已经平缓，最终测试集与训练集准确率也与前两次训练相似

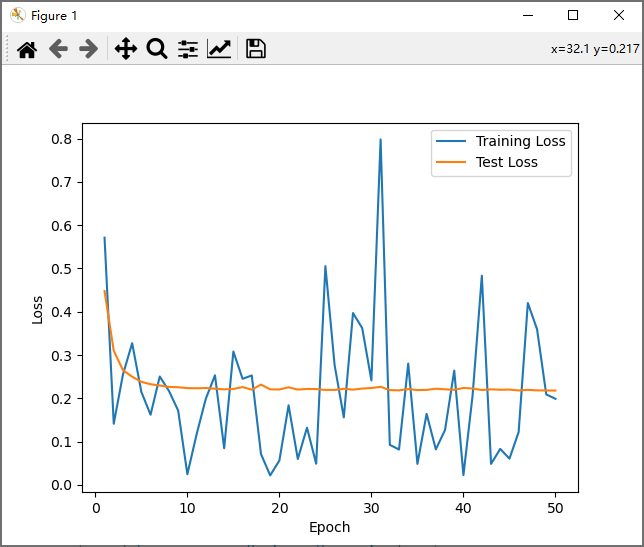
* 1. 激活函数ReLU，神经元个数128

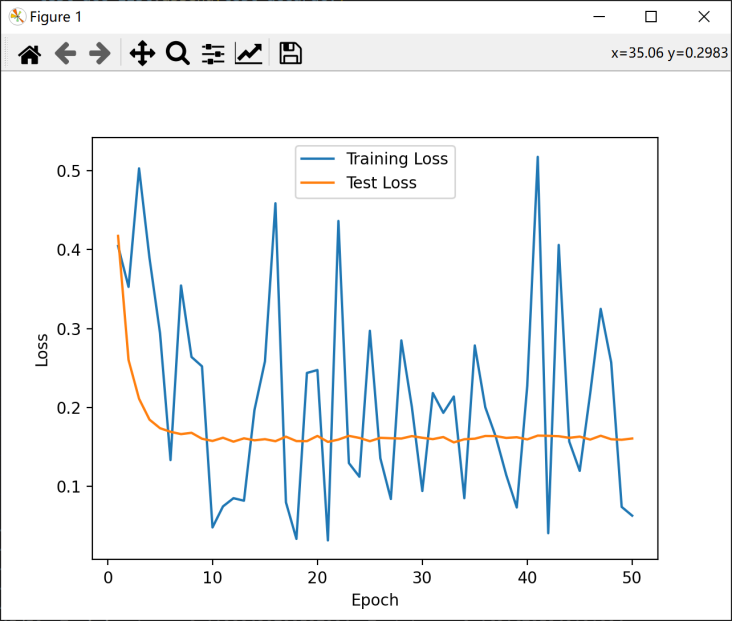


因训练轮数的影响在前文探讨过，故选只选取50轮训练作为训练次数

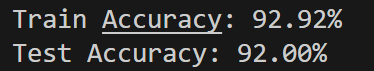
训练次数50，测试集与训练集损失如下图所示

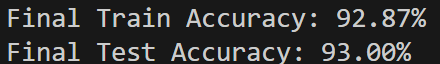






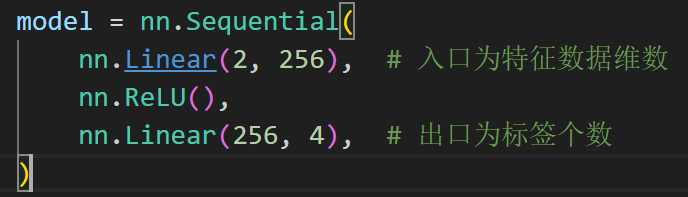
进行三次训练最终训练集准确率92.87%，测试集准确率93.00%，如下图所示



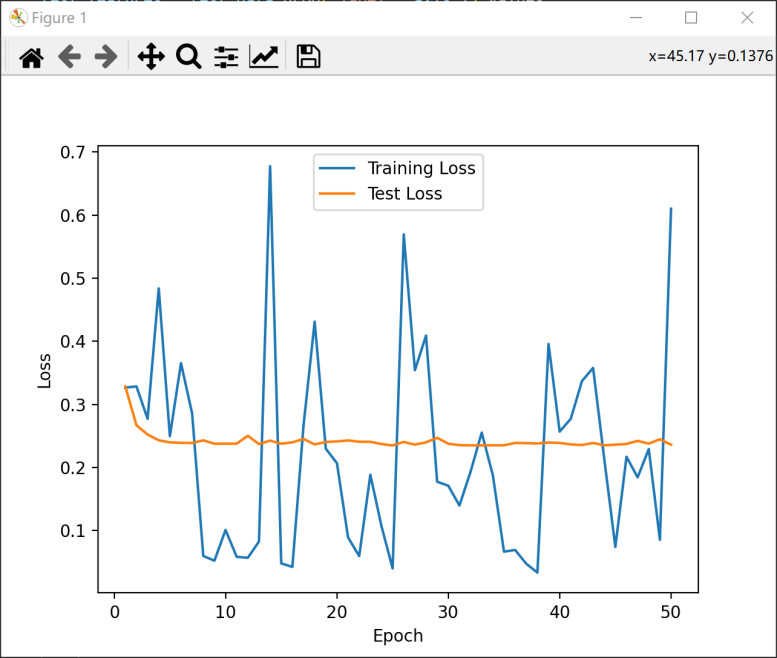


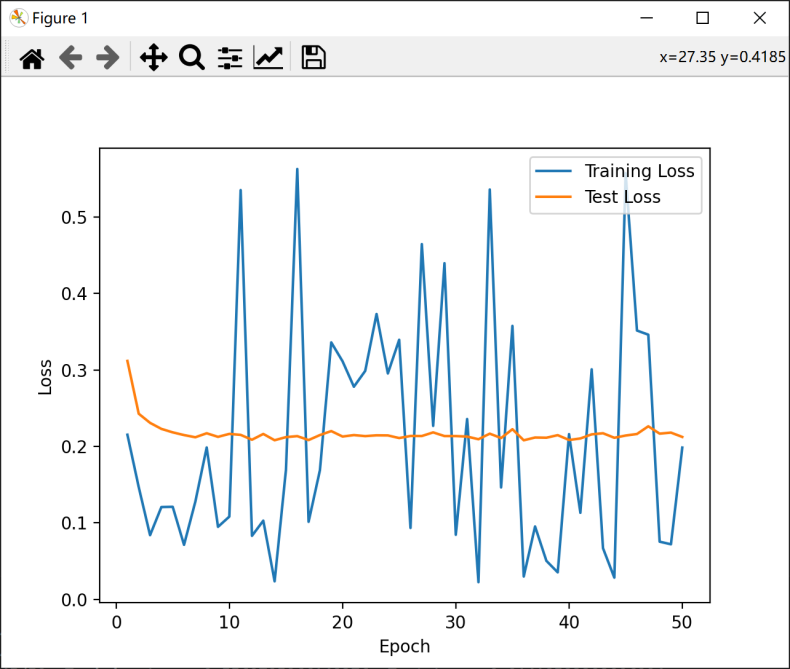
与前几次测试相差不大，并且测试集损失已接近平缓，故不再修改训练轮数

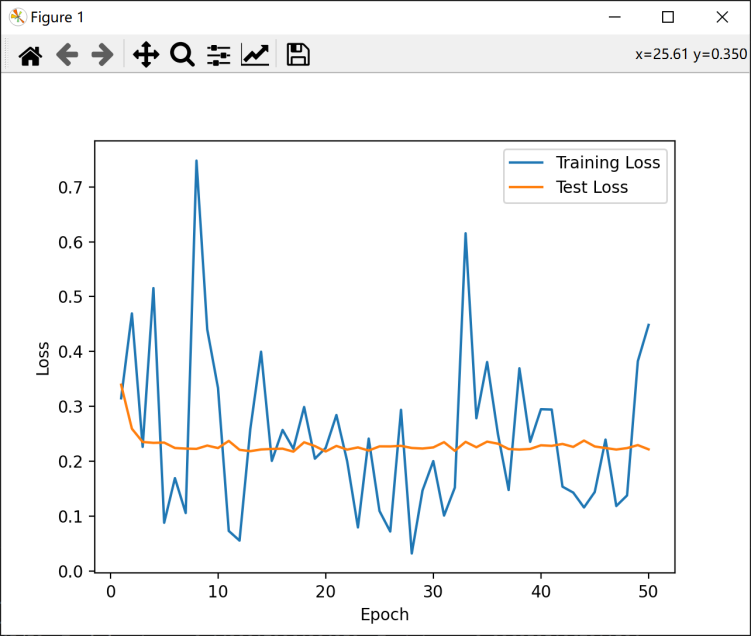
* 1. 激活函数ReLU，神经元个数256



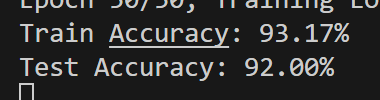
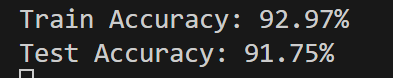
训练次数50，测试集与训练集损失如下图所示

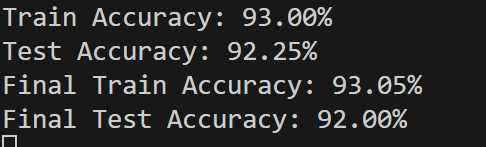






进行三次训练最终训练集准确率93.05%，测试集准确率92.00%，如下图所示

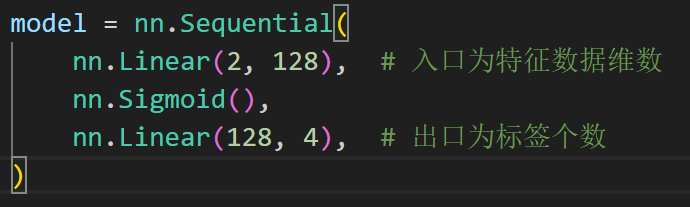




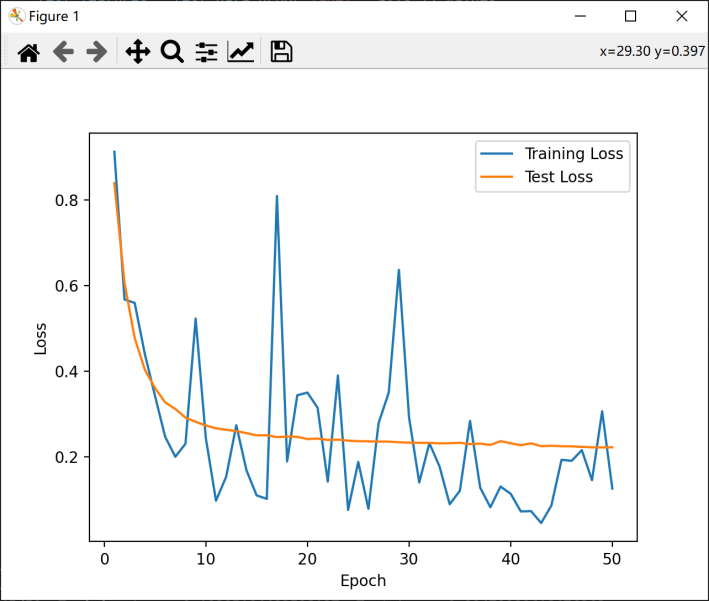
总结：

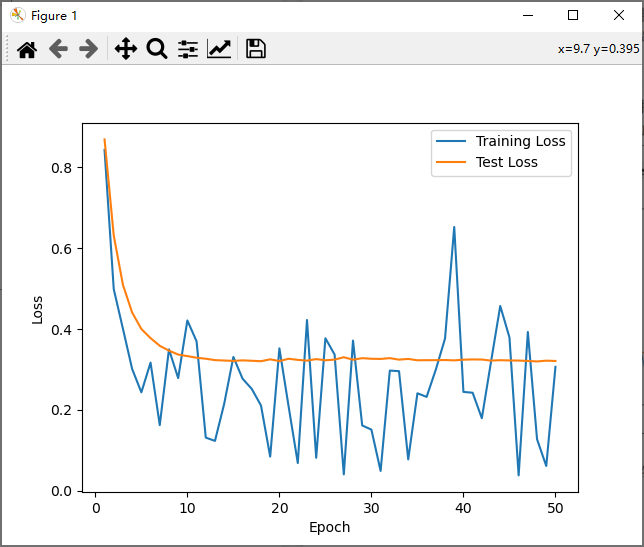
可以看出，随着神经元个数的增加，测试集损失下降得更快，但是对训练集损失并没有什么太大影响，对最终测试集与训练集准确率影响较小，但神经元个数为128时训练效果较好

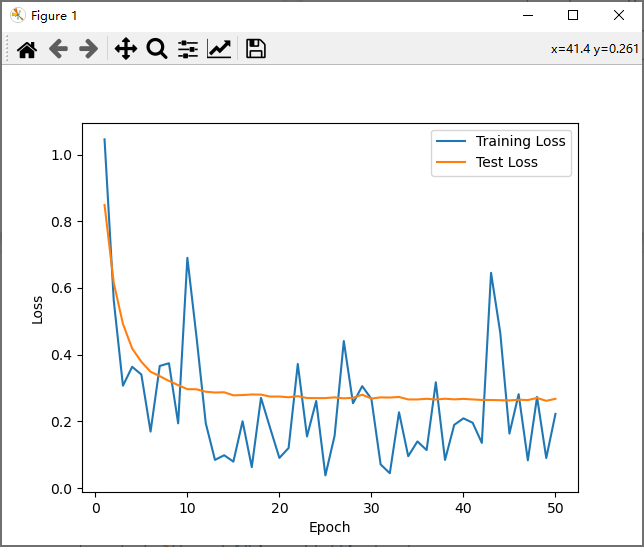
* 1. 激活函数Sigmoid，神经元个数128



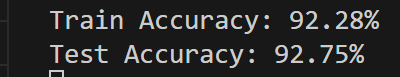
因在前面的实验中，证明了神经元个数为128时效果较好，取训练轮数为50，测试集与训练集损失如下图所示

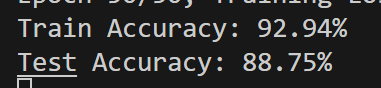


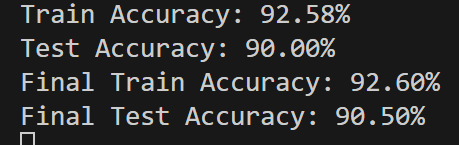




进行三次训练最终训练集准确率92.60%，测试集准确率90.50%，如下图所示



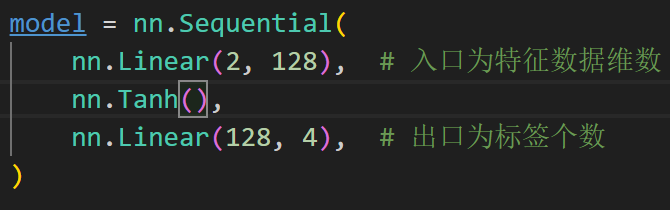




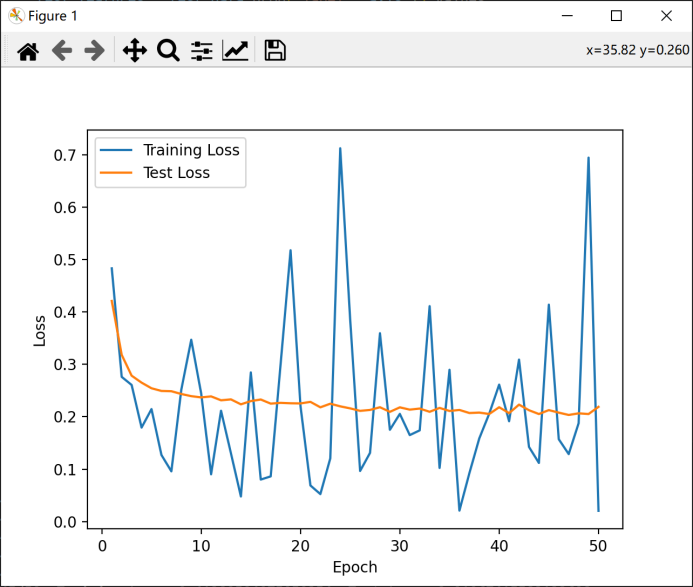
总结：

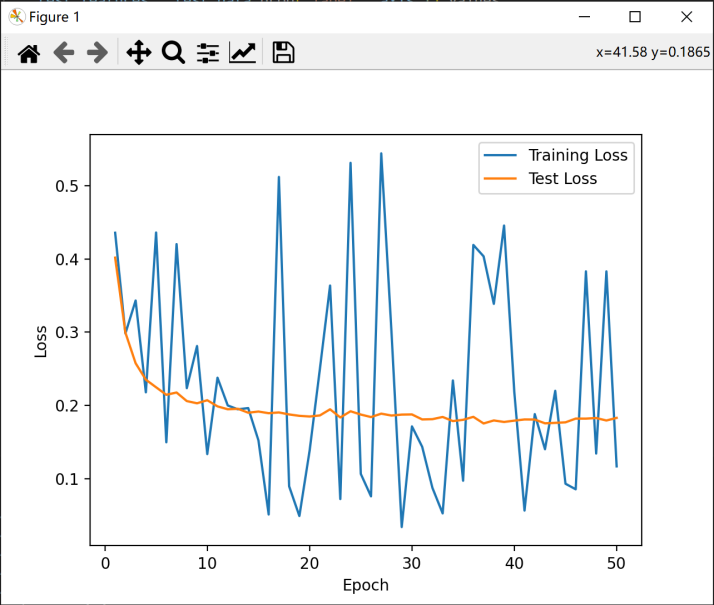
可以看出测试集损失下降较ReLU函数慢，但是训练集损失波动较大，最后的测试集与训练集准确率也不够ReLU函数好

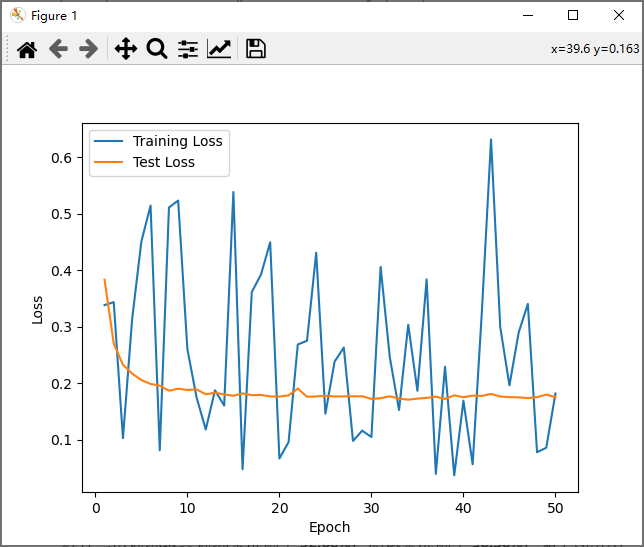
* 1. 激活函数Tanh，神经元个数128



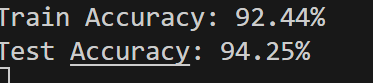
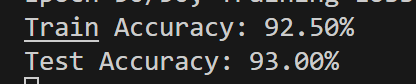
因在前面的实验中，证明了神经元个数为128时效果较好，取训练轮数为50，测试集与训练集损失如下图所示

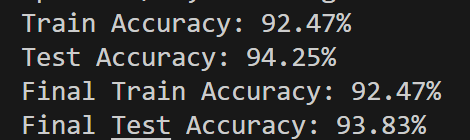






进行三次训练最终训练集准确率92.47%，测试集准确率93.83%，如下图所示

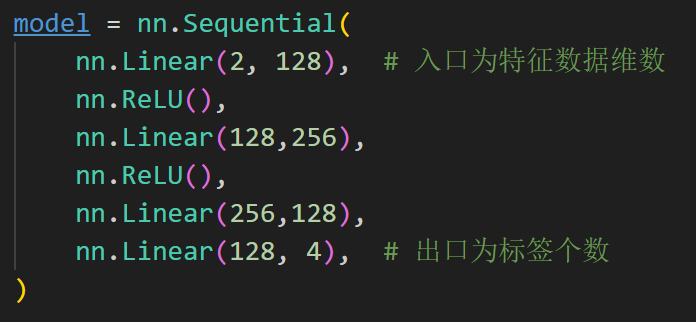




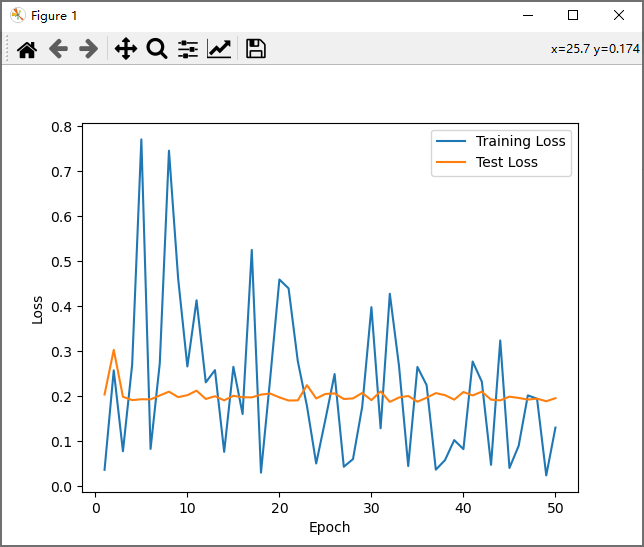
总结：

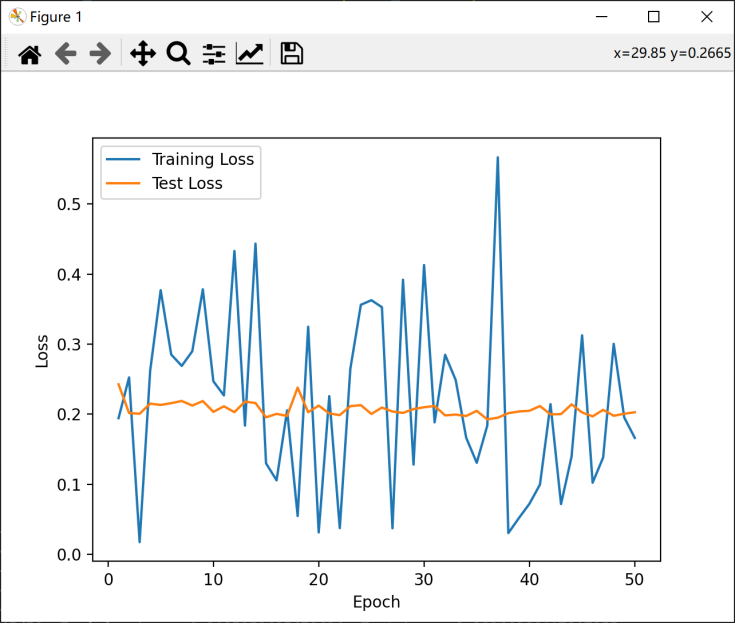
使用Tanh函数进行激活时，测试集损失下降较慢，但最终测试集准确率较高。

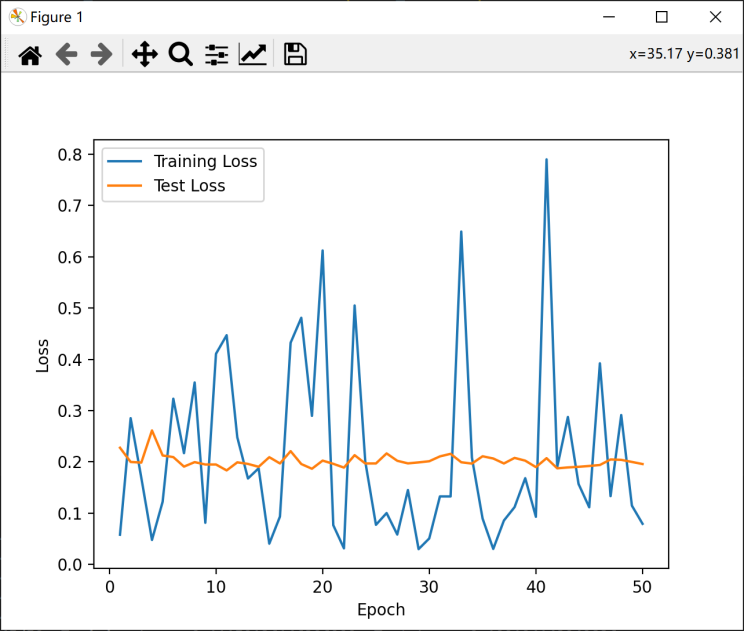
* 1. 改变神经网络层数，再插入两个全连接层和一个隐藏层使用下图的网络层



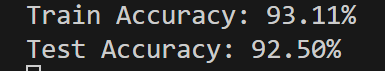
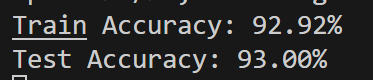
取训练轮数为50，测试集与训练集损失如下图所示

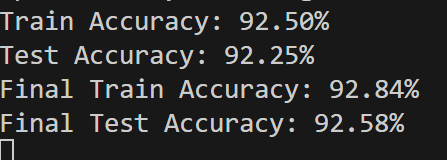






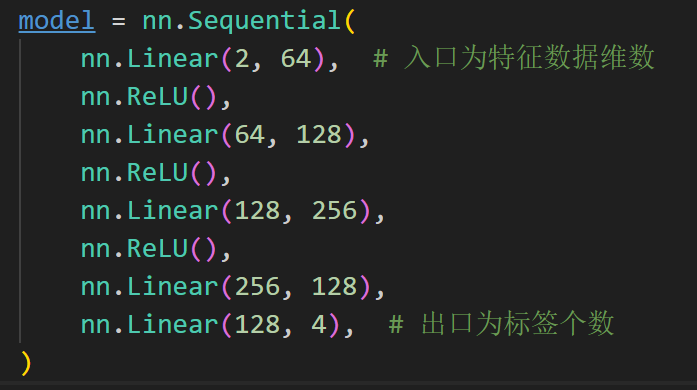
进行三次训练最终训练集准确率92.84%，测试集准确率92.58%，如下图所示



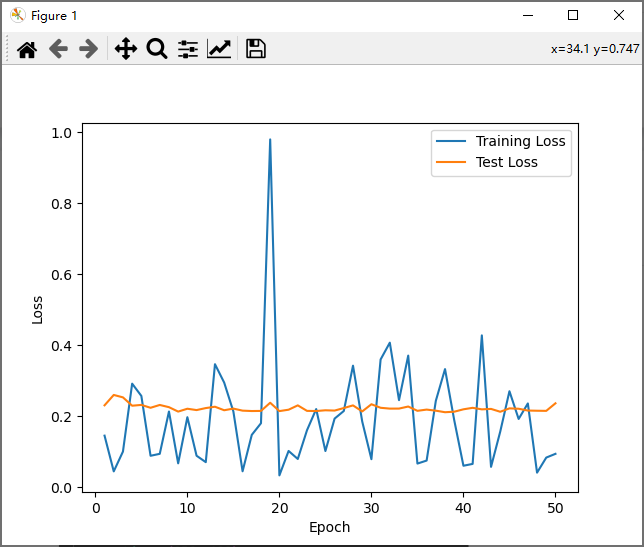


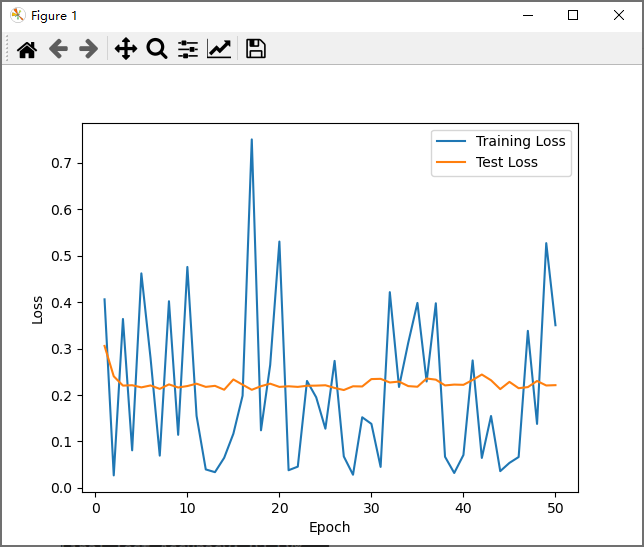
总结：增加网络层数后，测试集损失基本1轮训练后就到达最小值，波动趋于平缓，训练集损失差异不大，训练集与测试集准确率差异不大，但是训练中，运算量大了很多

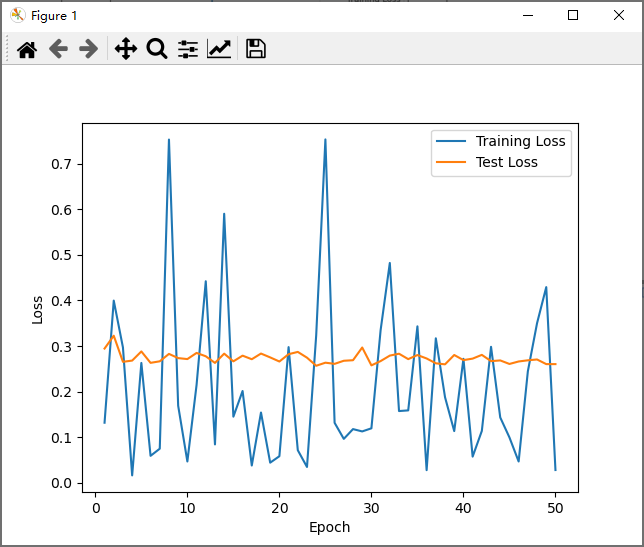
* 1. 改变神经网络层数，再插入两个全连接层和一个隐藏层使用下图的网络层



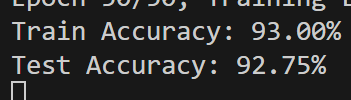
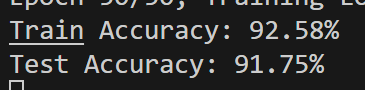
取训练轮数为50，测试集与训练集损失如下图所示

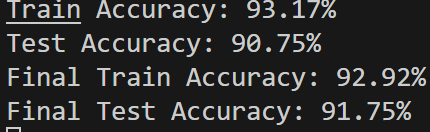






进行三次训练最终训练集准确率92.92%，测试集准确率91.75%，如下图所示

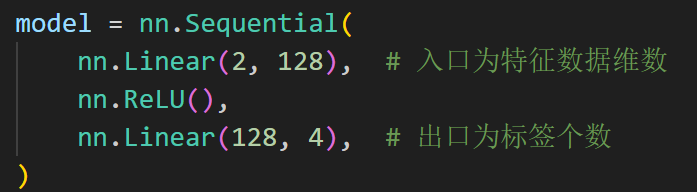




总结：增加网络层数后，测试集损失基本1轮训练后就到达最小值，波动趋于平缓，训练集损失差异不大，训练集与测试集准确率反而较之前下降，训练中，运算量大了很多

1. 总结

在本次实验中，我尝试了不同的激活函数、网络层、神经元个数、训练次数组合，最终选定ReLU激活函数，训练次数50、两层全连接层和一层隐藏层、神经元个数128作为我的训练网络，如下图所示



因为这种方法兼顾了计算量和最终预测准确度。

在这次实验中，我掌握了如何取构建一个简单的神经网络训练模型，也学会了如何去调试我的模型，使其能更好地运行，预测准确率更高。