### 说明实验的运行环境和模型参数设置值，包括实验运行的计算机系统、软件版本、数据集来源、模型参数设置值、检测算法参数值等？

运行环境： python3.7/3.6 jupyter notebook开发 直接python脚本运行也可以 但是调试理解起来notebook更好

所需python包：

**os pandas numpy random sklearn matplotlib**

计算机系统： windows

数据集来源：网易财经

模型参数设置等参照第4个问题

### 代码是怎么反映投资组合的表现与预测，怎么验证算法的预测准确率、预期投资报酬率、神经网路算法的计算效率(运行时间长短?)

代码搭建了对于三支股票的三个不同的神经网络，位于三个不同的文件的model类中，使用时可运行所有代码，自行训练，训练后的神经网络的权重参数和偏差参数（最佳粒子）位于model.gx变量中，要预测新的数据，可使用model.predict函数进行预测，输出预测的股票价格信息。

**神经网络的输入输出参数：**

输入：x (N,30) 30代表了上周（5天）的 '收盘价','最高价','最低价','开盘价','换手率','成交量’这6种信息

输出：y (N,5) 预测了下周5天的收盘价

**算法的预测准确率：**

average(|真实值 – 预测值|)

**预测的下一周预期回报率：**

R = (预测的下周最后一天收盘价 – 上周收盘价)/上周收盘价

**计算效率：**

可用time函数计时一次预测的时间，这个时间个人认为意义不大。

### 和之前那篇文档的神经网络算法比较，这个算法的优势劣势是什么？

那边文档的神经网络算法是用梯度下降优化损失函数，收敛速度慢，易陷入

局部最优以及泛化能力差。使用粒子群算法优化能尽快收敛至全局最优，训练时间短，效果好。

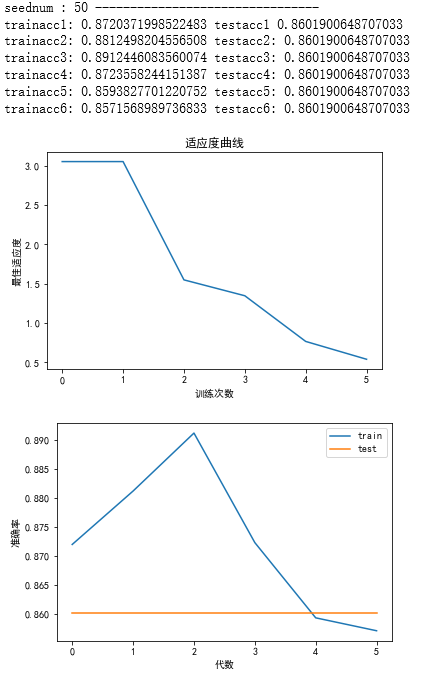
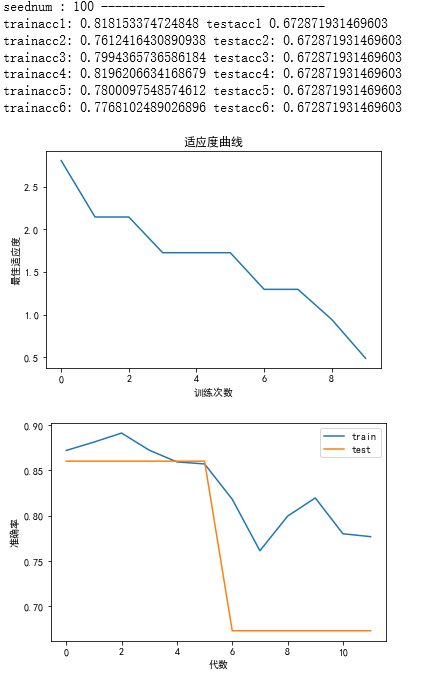
劣势在于不能保证每次训练的收敛结果相同，

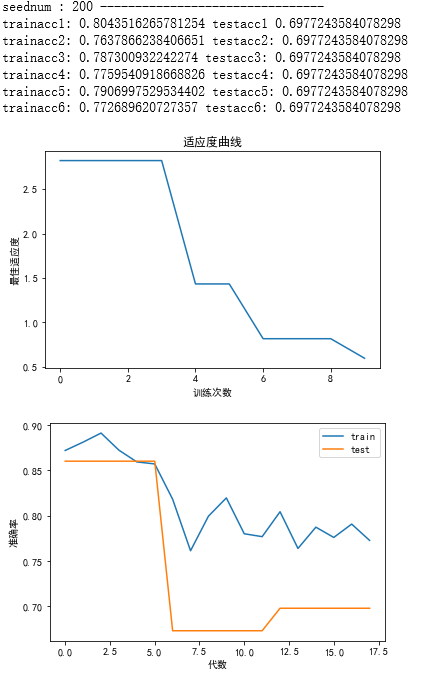
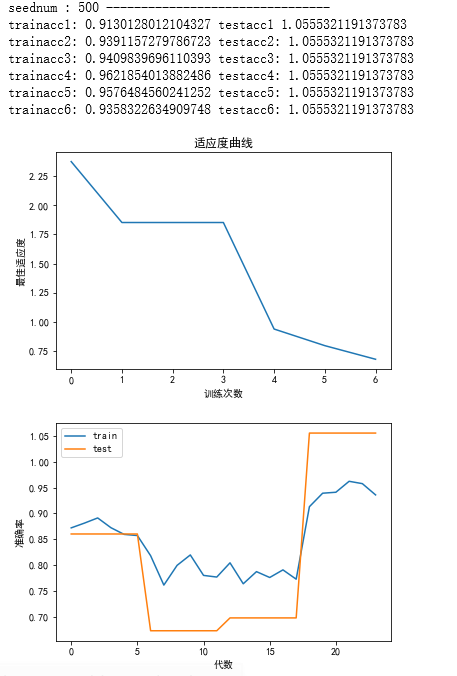
### 算法中设置的参数如何影响组合投资？数据分析(灵敏度分析)参数的调优过程？参数变动范围的选取原因、考虑？

基于粒子群算法的神经网络主要有以下一些参数可以进行调优(基本先粗调：

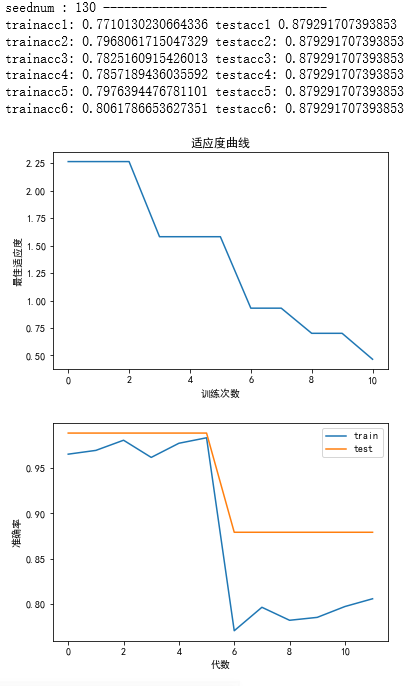
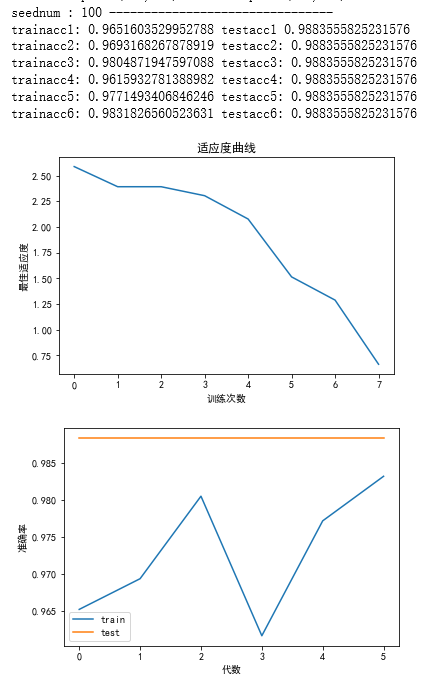
1. 种群数量 选择范围 粗调：[50,100,200,500] 精调: [100,130,150,180,210]

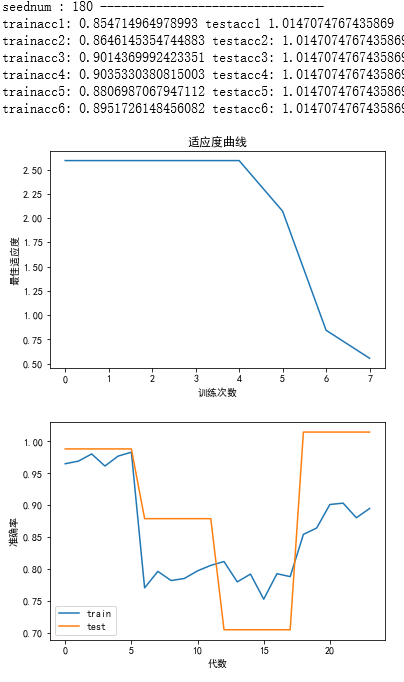
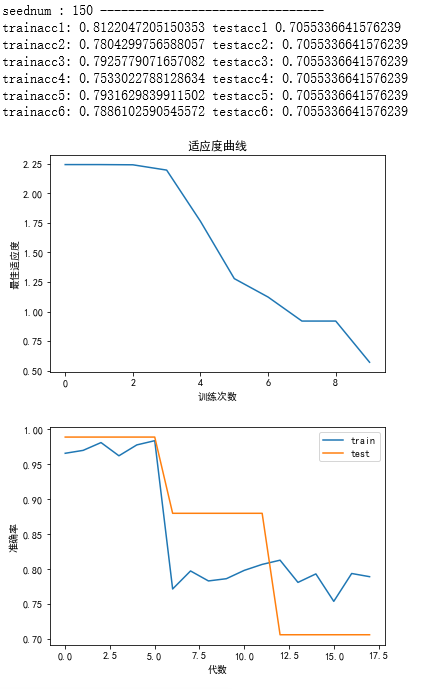
训练例子：粗调[50,100,200,500]

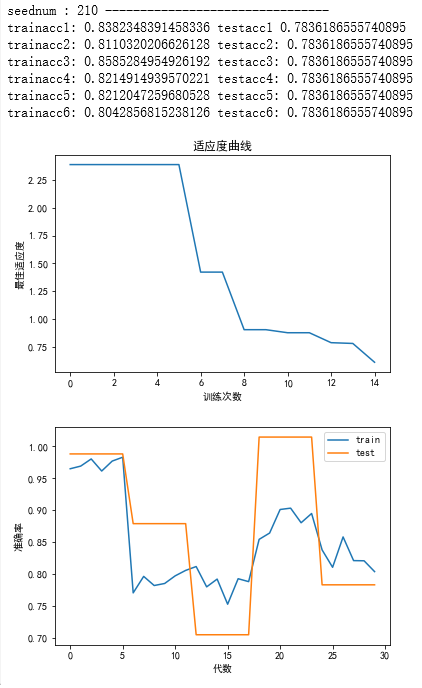
 

然后精调：[100,130,150,180,210]







经过调试，选择默认粒子数量为150。

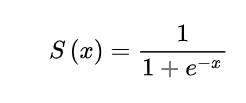
1. 隐藏层个数 [50,100,130,180] 经调试 默认参数为130
2. 速度限制 [0.1,0.2,0.3] 经调试 默认参数为0.2
3. 激活函数 [sigmoid, relu, leaky relu]

经调试 Leaky relu要明显好于 sigmoid 和 relu

原因猜测：

sigmoid会将激活值放到(0,1)中间，缩小了输出参数范围，导致输出误差大。

Relu 会使一部分输出值为0，导致一部分输出神经元死亡。

Sigmoid : 

Relu: S(x) = max(0,z)

Leaky relu: S(x) = max(0.01z,z)

5.损失函数：

[mean\_squared\_error,mean\_absolute\_error,mean\_squared\_log\_error,median\_absolute\_error]

经调试，mean\_squared\_error为最佳的损失函数。

### 训练集和测试集unbalanced的大致介绍，训练集和测试集是否有分开？

训练集： 在对抓取来的数据进行聚类（5天为一组数据）后，将80%的数据作为训练集

测试集：将剩余的20%数据作为测试集

训练集和测试集都为随机选取，并做5次的随机交叉验证，提高训练的准确性（事实效果可能不大，因为粒子群算法在第一次基本收敛）

### 在这个实验的基础上还需要改进的方面？

还需要改进和研究的例如

输入参数中股票的天数是否可以更大或者更小？

其他优化算法（如遗传算法）优化的效果和粒子群相比怎么样？

如果将神经网络的结构加深，效果会更好还是更坏？