**利用前馈神经网络预测大盘**

**一、 问题描述和设置**

通过SMA，WMA，MOM，STCK，STCD，MACD，RSI，WILLR，CCI，MFI，OBV，ROC，CMO等技术指标作为训练模型的特征预测大盘的涨跌。采用2007-1-4至2016-6-2的数据做训练集（2288个样本），2016-6-3至2017-1-3做测试集（143个样本）。

实验设置有一点小区别。因为离散型变量本身只有相互区别的性质，没有大小可比的性质。如果直接赋值，就默认了离散型变量具有大小可比性。这可能会造成实验的不精准。在此，做如下设置：

* 对每一个特征值(以MACD为例)，采用两个输入量：

x1,x2x1,x2

* 如果，MACD看多，则：
* x1=1,x2=0x1=1,x2=0
* 否则：

x1=0,x2=1x1=0,x2=1

* 这样，在原来13个指标的基础上，一共定义了26个输入。
* 同样，对于输出值，采用两个输出量：

y1,y2y1,y2

* 如果，上涨，则：
* y1=1,y2=0y1=1,y2=0
* 否则：

y1=0,y2=1y1=0,y2=1

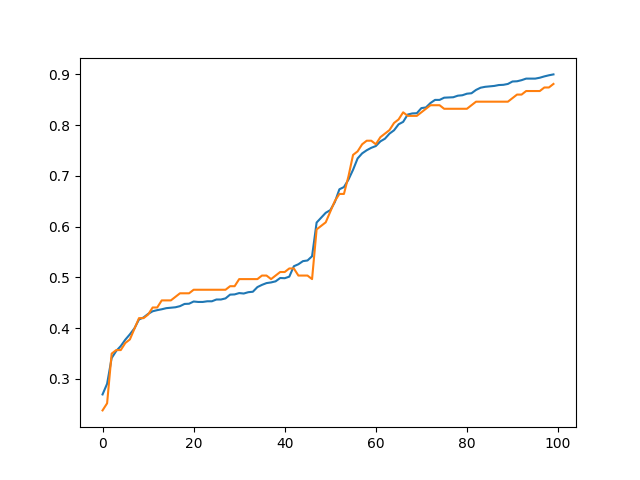
* 因此，输出量有两个。

基于上述设置，神经网络的输入变量和输出变量个数分别为26和2，这是确定的。关于其他参数的影响，实验如下。

**二、实验**

**2.1. 实验一**

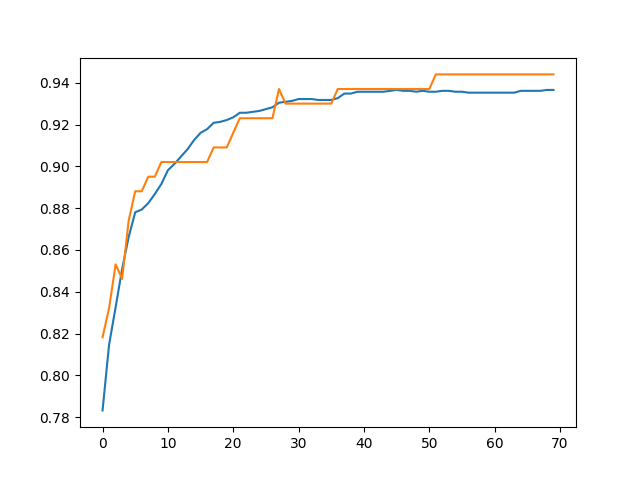
设置：二次目标函数，隐藏层1层（50个神经元），学习率0.001，没有正则化项。

实验结果如下：  


上图中，纵轴是正确率，横轴是训练的轮数，蓝色的线是训练集正确率，橘色的线是测试集正确率。可以发现，在50轮之前的训练比较慢，接近一百轮之后达到0.9以上。下面考虑通过Corss-Entropy函数来加快训练速度。

**2.2. 实验二**

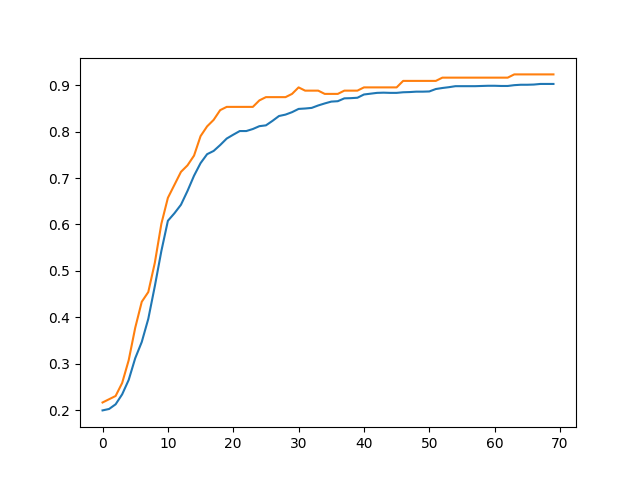
设置：Corss-Entropy目标函数，隐藏层1层（50个神经元），学习率0.001，没有正则化项。



可以看到，采用了Cross-Entropy函数后，训练明显加快。不到20轮就进入了0.9的正确率。但是测试集相对于训练集的波动还是比较大。我们进一步加入正则化项看看。

**2.3. 实验三**

设置：Corss-Entropy目标函数，隐藏层1层（50个神经元），学习率0.001，有正则化项（参数0.1）。

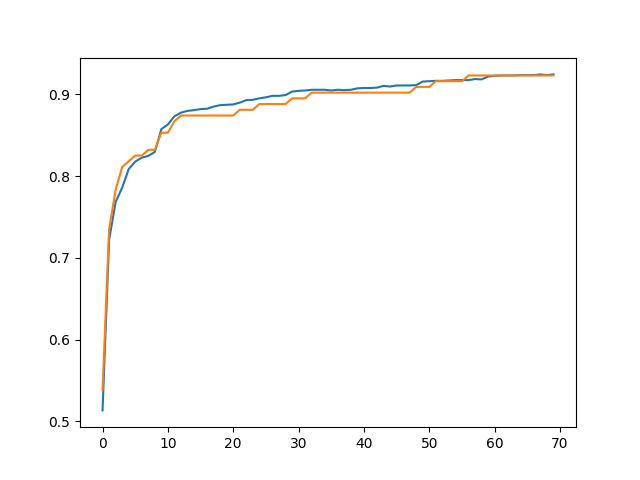


加入正则化项之后，测试集正确率和训练集正确率变动都比较平稳。理论上训练集的正确率应该在测试集之上。这里导致测试集具有更好的正确率的原因是测试集本身太小。每一个正例对正确率的贡献比训练集大。

**2.4. 实验四**

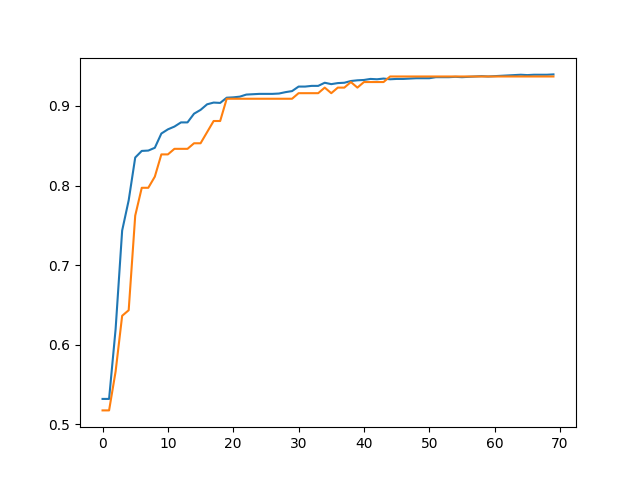
看看添加更多隐藏层的效果怎么样。

设置：Corss-Entropy目标函数，隐藏层2层（每层50个神经元），学习率0.001，有正则化项（参数0.1）。



效果还可以，没有明显的过拟合。再添加一些呢？

设置：Corss-Entropy目标函数，隐藏层5层（每层50个神经元），学习率0.001，有正则化项（参数0.1）。

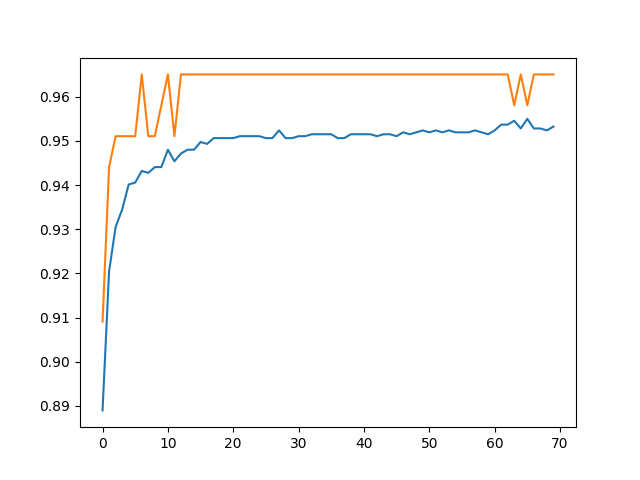


可以看出，随着隐藏层层数的增加，并没有明显的过拟合发生。说明深度神经网络具有稳定的泛化性。

**2.5. 实验五**

最后看看增加学习率会发生什么。

设置：Corss-Entropy目标函数，隐藏层1层（每层50个神经元），学习率0.01，有正则化项（参数0.1）。



可以看到，学习速度变得更快。在第一轮正确率就进入了0.8以上。

但是可以通过实验发现，在本实验中设置中，学习率>1的情况下，正确率始终保持在50%左右。说明梯度下降时，步子迈得太大，始终不能找到最优值。

**三、总结**

从实验过程中看，最大的问题还是训练集太小。上千条数据其实不一定采用神经网络，通过SVM或者非线性回归也可以达到效果。另外本实验测试集太小，导致测试集准确率比训练集要高。另外，实验其实是将所有的样本看成是独立的。这个不太合理，证券市场的变化是具有时间累积效应：今天的涨跌对明天是有一定影响的。所以假设样本独立不太合理。证券市场应该是一个马尔科夫过程，应该引入时间效应。从神经网络的角度来看，可以采用递归神经网络（Recurrent Neural Network）来改进。