**基于BP神经网络的股价预测**

1. **可行性分析**

大量检验表明股价波动是非线性的，这就为非线性预测模型提供了用武之地。人工神经网络的基本原理是模拟人脑思维，而行为金融理论认为股价运行轨迹是人们心理变化的结果，因此人的思维方式在很大程度上可以体现到股价的变化上。所以由于人工神经网络与股价变化都与人的思维密切相关，利用非线性的人工神经网络来模拟股价运行在理论上是可行的。

人工神经网络分为前馈型网络和反馈型网络。前馈型网络用于解决函数逼近和分类的问题，反馈型网络用于解决联想记忆和聚类方面的问题。预测实践中广泛采用的是前馈型神经网络，这是基于前馈型网络良好的模拟性能和预测实践的需要。运用神经网络进行股价预测，途径是通过对股价运行规律进行学习，然后将网络学习到的规律外推。这是一个函数逼近的问题。因此我们**选用前馈型网络预测股价**更合理。

BP神经网络属于单隐层前馈型网络的一种，具有全局响应特性，相对来说结构简单。于是我们考虑将BP网络定为预测模型。

1. **BP神经网络股价预测模型结构设计**

网络结构包括确定输入变量的数目，隐含层数以及隐含层的节点数，输出节点数。对于输入和输出节点数可由应用对象的实际情况来决定，**输入节点数在数据预处理环节决定**，输出为下一交易日的收盘价、最高价、最低价，故**输出节点数为3**。一般来说，BP网络至少包含一个隐层，但对于一个网络究竟要包含几个隐层才能达到良好的性能，并没有一个严格的标准。但是有人已经证明了具有一个隐层的BP网络可以在任意精度内逼近闭区间内的任何一个连续函数。因此，我们采用的BP网络结构可以**采用一个隐含层**，从而完成股价运行规律的函数逼近。

对于隐层节点个数，大家普遍都靠经验决定。这里有几种学者提出的原则：1、隐层节点数=，其中X为输入层的节点数。2、隐层节点数=sqrt(X+Y)其中X为输入层的节点数，Y为输出层的节点数。3、隐层节点数=(X+Y ) /2。但这些方法均来自经验，并没有得到理论上的论证，由于神经网络的复杂性和数据特征的差异性，最好的**隐层节点数确定方法还是根据实验的结果来确定**，通过选取不同的节点数来训练网络，选取性能最佳时的节点数即可。

1. **指标选取**

EXCEL中原始数据为上证50指数2015年2月9日到2017年2月7日一共484个交易日的收盘价、成交量、最高价、最低价以及一系列相关指数数据的时间序列。其中2015年2月9日到2016年9月27日共400个交易日的数据作为训练样本（由于选取的指标有最近20日数据平均值，所以我们实际开始计算的交易日为2015年3月13日，于是总共381个交易日的数据作为训练样本），2016年9月28日到2017年2月7日共84个交易日的数据作为预测检验样本。我们的最终目的是利用训练所得的结果未来的SZ50指数收盘价、最高价、最低价。

基于所提供的数据，本次预测选取的变量如下：

因变量Y：明日上证50指数收盘价、最高价、最低价

自变量：

:今日上证50指数收盘价

:今日上证50指数最高价

:今日上证50指数最低价

:昨日上证50指数收盘价

:昨日上证50指数最高价

:昨日上证50指数最低价

:最近5日上证50指数收盘价平均值

:最近5日上证50指数最高价平均值

:最近5日上证50指数最低价平均值

:最近20日上证50指数收盘价平均值

:最近20日上证50指数最高价平均值

:最近20日上证50指数最低价平均值

:今日上证IVIX

:今日美指VIX

:今日标普500

:今日美元汇率

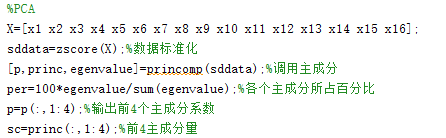
1. **数据预处理**

**4.1输入变量优化组合**

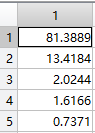
主成分分析：

对于原始的具有一定相关性的多个变量，主成分分析将其重新组合成几个互不相关的指标，重新组合后的指标可称之为主成分，各个主成分包含了原始数据的绝大部分信息。数学上的通常做法就是将原有的多个变量做线性组合，作为新的主成分。对于信息的包含量用主成分的方差来表示，方差越大，主成分包含的信息越多。一般原则上要求信息损失不能高于I S%，即各个主成分按方差大小排序后的累计方差不能小于85%，否则不能较大程度包含原有信息。

利用MATLAB中的princomp函数实现主成分分析如下：



百分比：



我们不拘束于累计方差贡献率大于85%的一般准则，前4个主成分累计占比达到98.4482%，包含信息的完整程度很好，因此我们选择4个主成分。

值得一提的是，我们在主成分分析之前对数据做了标准化处理。即新数据，由于本次预测并不需要回归分析，而是优化神经网络的输入层数据，因此无需在后面进行变量还原，直接采用4个主成分的值（即代码中sc）作为神经网络的初始输入值。

**4.2数据的归一化**

本文运用的神经网络模型采用的传递函数为S型函数（），其取值区间为[0,1]，因而该S型函数的适用范围也为[0,1]，对于适用区间之外的数据，在神经网络训练时会造成网络瘫痪，无法得到良好训练结果，于是我们需要对数据归一化处理使之落在[0,1]上。

我们采用如下归一化方法：

其中为数据中X的最大值，表示数据中X的最小值。这样归一化之后既保持了数据的原有特征，又避免数据落入到S型函数不适用的区域。在MATLAB中我们可以直接使用mapminmax函数实现。

1. **训练过程与模型确定**

**5.1神经网络模型初始参数设计**

MATLAB神经网络工具箱中构建BP网络的函数为newff()，构建网络之前需要对括号内参数进行初始化。

这些参数包括：

PR：输入向量的取值范围

Si：第i层的神经元个数，一共N层

Tfi：第i层的传递函数，缺省值为“tansig”

BTF：BP网络的训练函数

BLF：BP网络权值和阈值学习函数，缺省值为“learngdm”

PF：性能函数，缺省值为“mse”

1. 前文已对输入向量作归一化处理，因而输入向量取值范围为[0,1]。
2. 本文构建的神经网络结构为三层的BP网络，输入、隐层、输出节点数分别为4，x，3。为适合输入和输出，本文构建的网络中输入层和隐层之间传递函数采用logsig函数，隐层和输出层之间的传递函数采用purelin函数。
3. BP函数的训练函数有很多，本文采用具有动量和自适应学习率的梯度下降法。工具箱中traingdx训练算法可以实现将动量法和调整学习率的梯度下降法有效结合起来。与traingdx有关的参数有epochs、goal、lr、max-fail、mc、min-grad、show、time、Ir-inc、Ir-dec。

epochs:最大训练(迭代)次数，由于本文原始数据较多，为避免网络训练失败，在初始选择时可以选取较大的次数，实验中定为5000。

goal:训练要求的精度，一般取较小值，本文参考一般标准1e-5 。

Ir:学习率，学习率取值在0到1之间。对于BP网络，选择合适的学习率是非常重要的，学习率过大会导致网络难以收敛，学习率过小又会增加训练的时间，降低网络的效率。学习率的大小在实践中一般偏好维持网络的稳定性，从而选取较小的学习率(0.01-0.5 )。由于本文采用的网络训练方法traingdx包含了学习率的自适应调整过程，因而最佳学习率会在实验中通过比较得出。初始值选0.1。

max-fail:最大失败次数，缺省为5。应用缺省值。

mc:动量因子，缺省为0.9。动量因子处于0到1之间，当时，权值修正只与当前负梯度有关系，当时，权值修正就完全取决于上一次循环的负梯度了。较大的能支持较大的学习率，从而加快网络的训练，但同时可能会导致网络的不稳定。较小的a适用于比较平坦的样本空间，能提高网络训练的精度，但对于网络训练的速度是不利的。本文实践中先取较大的动量因子，根据网络的训练误差确定最佳值。

min-grad:最小梯度要求，缺省为1e-10。应用缺省值。

show:显示训练迭代过程。可以在实验后看到训练过程。

time:最大训练时间。缺省为inf,可以采用缺省值，不限制训练时间。

Ir-inc:学习率1r增长比，缺省为1.05。应用缺省值。

lr-dec:学习率1r下降比，缺省为0.7。应用缺省值。

1. 本文中BP网络权值和阈值学习函数应用缺省值learngdm。
2. 性能函数采用最小均方误差标准，应用缺省值mse。

**5.2隐节点数的确定**

隐节点数的选择范围为1到4，对不同的隐节点数分别进行实验。这里我们令动量因子、学习率。检验标准为测试集上的均方误差。

|  |  |
| --- | --- |
| 隐节点数 | 测试集上均方误差（CLOSE HIGH LOW） |
| 1 | 1.0e+06 \*  6.8399 7.5405 6.0832 |
| 2 | 1.0e+06 \*  7.2527 7.4073 7.3966 |
| 3 | 1.0e+06 \*  7.3549 8.5078 6.3682 |
| 4 | 1.0e+07 \*  1.1019 1.1307 1.1484 |

由表中数据可知，我们选择均方误差相对较小时的隐节点数1。

**5.3动量因子的确定**

令隐节点数为1，学习率，去不同的值进行实验。检验标准仍然为测试集上的均方误差。

|  |  |
| --- | --- |
| 动量因子 | 测试集上均方误差（CLOSE HIGH LOW） |
| 0.1 | 1.0e+06 \*  7.4896 8.2562 6.6585 |
| 0.2 | 1.0e+06 \*  7.0658 7.7906 6.2818 |
| 0.3 | 1.0e+06 \*  6.9372 7.6483 6.1684 |
| 0.4 | 1.0e+06 \*  6.8862 7.5922 6.1233 |
| 0.5 | 1.0e+06 \*  6.8453 7.5465 6.0875 |
| 0.6 | 1.0e+06 \*  6.9018 7.6085 6.1382 |
| 0.7 | 1.0e+06 \*  6.8497 7.5514 6.0915 |
| 0.8 | 1.0e+06 \*  6.8630 7.5660 6.1036 |
| 0.9 | 1.0e+06 \*  6.6848 7.3685 5.9466 |

由表中数据可知，我们选择均方误差最小的动量因子0.9。

**5.4学习率的确定**

令隐层节点数为1，动量因子，取不同的值进行实验。检验标准为测试集上均方误差。

|  |  |
| --- | --- |
| 学习率 | 测试集上均方误差（CLOSE HIGH LOW） |
| 0.05 | 1.0e+06 \*  6.7664 7.4589 6.0187 |
| 0.1 | 1.0e+06 \*  6.7686 7.4615 6.0205 |
| 0.2 | 1.0e+06 \*  6.8998 7.6062 6.1364 |
| 0.3 | 1.0e+06 \*  6.7139 7.4017 5.9714 |
| 0.4 | 1.0e+06 \*  6.8476 7.5485 6.0905 |
| 0.5 | 1.0e+06 \*  6.7082 7.3953 5.9664 |
| 0.55 | 1.0e+06 \*  6.6899 7.3748 5.9505 |
| 0.6 | 1.0e+06 \*  6.7047 7.3915 5.9634 |

我们发现学习率为0.5和0.6时预测结果均方误差十分接近，于是我们补测了的情况，发现均方误差更小，于是我们选择学习率为0.55。

1. **结论**

我们最终通过训练得到了最终的神经网络模型，但是从预测的角度来看，该模型还十分欠完善。通过在预测集上的表现来看，虽然有时能预测较准，但很多时候预测偏差达到15%左右，这是不能接受的。或许在建立模型的过程中有所疏漏，也或许是设定的变量不能反应真实情况，今后我会找出其中的原因。