

# 用 BP 神经网络预测股票市场涨跌

吴 微<sup>1</sup>, 陈维强<sup>2</sup>, 刘 波<sup>2</sup>

(1. 大连理工大学 应用数学系, 辽宁 大连 116024;

2. 吉林大学 数学系, 吉林 长春 130023)

**摘要:** 利用 BP 网络较好的分类能力, 结合国内股票市场的特性, 对于沪市综合指数涨跌的预测进行了初步探讨. 大量数值实验结果表明, 人工神经网络应用于中国股票市场的预测是可行和有效的, 有着良好的前景.

**关键词:** 神经网络; 在线/ BP 算法; 股票

**中图分类号:** TP183; F830.91 **文献标识码:** A

## 0 引言

多层前馈式神经网络(MLP)是目前应用比较广泛的神经网络, 而 BP 算法是最著名的多层前馈网络训练算法. 早在1974年, Werbos 就在他的博士论文中描述了这种算法, 当时称之为“动态反馈”<sup>[1]</sup>. 尽管随着神经网络科学的发展产生了许多优秀的算法, 而 BP 算法本身又存在收敛速度慢、易陷入局部极小值和推广能力差等不足, 但由于其简单易行、计算量小、并行性强等优点, 目前仍是多层前馈式网络训练的首选算法之一, 并且已被人们广泛地应用于各种实际问题. 以下, 把用 BP 算法作为网络学习算法的多层前馈式神经网络简称为“BP 网络”.

本文将 BP 网络应用于沪市综合指数涨跌的预测, 作了一些初步探讨.

## 1 算法简介

BP 算法的基本思想是利用 LMS<sup>[2]</sup> 学习算法, 在网络的学习过程中使用梯度搜索技术, 利用误差向后传播来修正权, 从而实现网络的实际输出与期望输出的均方差最小化. 图1给出了通常的带有一个隐层的 BP 网络模型. 其中:  $O_i$  表示输出单元,  $V_j$  表示隐单元,  $\xi_k$  表示输入单元. 从输入单元  $k$  到隐单元  $j$  的连接权为  $w_{jk}$ , 从隐单元  $j$  到

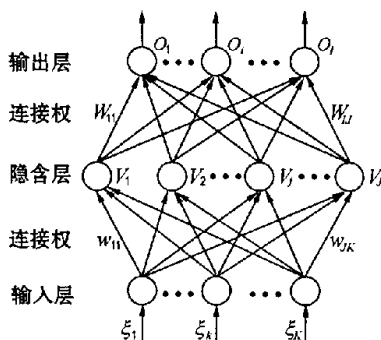


图1 带有一个隐层的 BP 网络模型

Fig. 1 BP model with a hidden layer

输出单元  $i$  的连接权为  $W_{ij}$ ,  $w = \{W, w\}$  表示所有的连接权. 单元下标  $i = 1, 2, \dots, I$  总是针对输出单元的;  $j = 1, 2, \dots, J$  针对隐单元;  $k = 1, 2, \dots, K$  对应输入单元. 上标  $\mu$  表示不同的输入模式,  $P$  表示输入模式数 ( $\mu = 1, 2, \dots, P$ );  $g_1$  和  $g_2$  分别对应于适当选定的隐层和输出层活化函数. 对给定的输入模式  $\mu$ , 隐单元  $j$  的输入为

$$h_j^\mu = \sum_{k=1}^K w_{jk} \xi_k^\mu \quad (1)$$

输出为

$$V_j^\mu = g_1(h_j^\mu) = g_1\left(\sum_{k=1}^K w_{jk} \xi_k^\mu\right) \quad (2)$$

输出单元  $i$  的输入为

$$H_i^\mu = \sum_{j=1}^J W_{ij} V_j^\mu = \sum_{j=1}^J W_{ij} g_1 \left( \sum_{k=1}^K \omega_{jk} \xi_k^\mu \right) \quad (3)$$

最终的输出为

$$O_i^\mu = g_2(H_i^\mu) = g_2 \left[ \sum_{j=1}^J W_{ij} g_1 \left( \sum_{k=1}^K \omega_{jk} \xi_k^\mu \right) \right] \quad (4)$$

对应于任一输入模式  $\mu$  和输出单元  $i$ , 定义在线误差(或瞬时误差)函数为

$$E^\mu(\omega) = \frac{1}{2} (\xi_i^\mu - O_i^\mu)^2 = \frac{1}{2} \left\{ \xi_i^\mu - g_2 \left[ \sum_{j=1}^J W_{ij} g_1 \left( \sum_{k=1}^K \omega_{jk} \xi_k^\mu \right) \right] \right\}^2 \quad (5)$$

对输出单元  $i$ , 总误差函数(又称能量函数)为

$$E(\omega) = \sum_{\mu=1}^P E^\mu(\omega) = \frac{1}{2} \sum_{\mu=1}^P (\xi_i^\mu - O_i^\mu)^2 \quad (6)$$

式中:  $\xi_i^\mu$  表示模式  $\mu$  对应输出单元  $i$  的期望输出。

所谓权的学习, 即对每一当前的  $\omega$ , 选取适当的增量  $\Delta\omega$  得到新的权值  $\omega + \Delta\omega$ , 使得在线误差函数(5)或总误差函数(6)随着迭代的进行而减小, 从而最终达到某一全局的或局部的极小值。显然上述误差函数都是权  $\omega$  的连续可微函数, 可以利用梯度下降算法对权进行学习。以在线误差函数(5)为例, 对于隐单元到输出单元之间的连接权矩阵  $W$  及输入模式  $\mu$ , 利用梯度下降规则得到

$$\Delta W_{ij} = -\eta \frac{\partial E^\mu}{\partial W_{ij}} = \eta \delta_i^\mu V_j^\mu \quad (7)$$

式中:  $\eta > 0$  为适当选定的学习步长,

$$\delta_i^\mu = g_2'(h_i^\mu) (\xi_i^\mu - O_i^\mu) \quad (8)$$

同理, 对于从输入单元到隐单元之间的连接权矩阵  $w$ , 利用链式法则可得

$$\Delta \omega_{jk} = -\eta \frac{\partial E^\mu}{\partial \omega_{jk}} = -\eta \frac{\partial E^\mu}{\partial V_j^\mu} \frac{\partial V_j^\mu}{\partial \omega_{jk}} = \eta \delta_j^\mu \xi_k^\mu \quad (9)$$

式中

$$\delta_j^\mu = g_1'(h_j^\mu) \sum_i W_{ij} \delta_i^\mu \quad (10)$$

这样即可对权  $\omega = \{W, w\}$  实现实时更新:

$$\omega^{\text{new}} = \omega^{\text{old}} + \Delta\omega, \quad \Delta\omega = \{\Delta W, \Delta w\} \quad (11)$$

以上以“在线”形式给出了更新规则, 即在输入端提供模式  $\mu$  之后, 利用在线误差函数(5)实时更新所有的权。这样每一步迭代都减小了在线误差(设  $\eta$  充分小), 也就是说, 每一次更新都是局部的。也可以使用基于总误差函数(6)得到的“批”方式, 即在提供所有的输入模式之后才更新权。在样本数据比较庞杂的情况下, 后者显然要付出更高的计算代价。一般认为, 在线式的更新规则在计算速度、效率和精度上都较批方式优越,

并且有利于在迭代过程中跳出局部极小。

学习步长  $\eta$  是一个重要的参数, 在一定程度上决定了网络的收敛速度<sup>[3]</sup>。  $\eta$  过小会导致权的更新量过小, 因而使收敛速度非常缓慢;  $\eta$  过大却又会导致在极值点附近振荡的可能性加大, 乃至反复振荡而难以收敛。对于这个问题, Rumelhart 和 Hoskins 等人针对在线式更新规则提出了所谓的“矩方法”<sup>[4]</sup>。即在式(7)和式(9)中加上矩量项:

$$\Delta\omega(t+1) = -\eta \frac{\partial E}{\partial \omega} + \alpha \Delta\omega(t) \quad (12)$$

式中:  $\alpha \in (0, 1)$  称为矩参数。这样, 在通过误差函数曲面上的平坦区域时, 可以认为  $\Delta\omega(t+1) \approx \Delta\omega(t)$ , 则网络的学习步长  $\eta$  近似为

$$\eta^{\text{new}} = \eta^{\text{old}} / (1 - \alpha) \quad (13)$$

从而加快了网络在这一区域的学习速度。而在振荡比较剧烈的区域, 矩量项便可忽略不计, 从而相对于平坦区域减小了学习步长, 避免了振荡加剧, 有助于寻找全局最优解。从总体效应来看, 矩方法的引入实现了学习步长的动态变化, 提高了网络的自适应能力, 加快了收敛速度。本文的网络模型引入了这种方法。然而对于不同的研究对象, 仍需通过多次实验才能找到比较理想的  $\eta$  和  $\alpha$ 。在对  $\eta$  和  $\alpha$  进行大量对比实验的基础上, 选择了  $\eta = 0.025, \alpha = 0.01$ 。

关于网络逼近能力的基本结论是: 带有一个隐层的前向网络能够逼近定义在  $\mathbf{R}^n$  中一个有界闭集上的相当任意的非线性函数<sup>[5,6]</sup>, 也就是说, 对于任意非线性决策边界都可以找到一个三层的前向网络对其形成任意接近的逼近。关于循环迭代过程中权值  $\omega$  收敛性的讨论, 可参见文献<sup>[7]</sup>。

## 2 样本与权的规范化

### 2.1 样本的选取和预处理

样本的选取和预处理是模型建立伊始就要解决的一个重要问题, 是研究对象与网络模型的接口。对于股票市场而言, 样本数据要尽可能地正确反映其交易规律, 同时又要顾及网络本身的性能。可以从以下3方面来考虑。

(1) 原始样本的选取。股票交易市场是一个很不稳定的动态变化过程, 不仅受国内外经济因素的影响, 而且人为的作用(“庄家”的作为)、政府的调控等也是影响其未来走势的重要因素。因此, 必须选取正常运作情况下(即没有或少有暴

和暴跌等不稳定现象)的股市样本数据;否则,如果样本选取得很“特殊”,就只能抽取到某种特定的规律,而降低了网络的推广能力. 通过对3 a 内各项股市指标的综合分析和评价,选取了1998-06-15~1999-02-05连续161个交易日的沪市综合指数作为学习和预测样本. 这段时期正是国内证券市场在经受了东南亚经济危机的冲击之后,已经逐步趋于稳定的阶段. 而沪市综合指数作为国内股市行情的重要综合评测指标之一,能够比较准确地反映国内股市行情动态,具有较高的预测价值和较好的可预测性.

(2) 样本向量的确定. 样本向量的各个分量应该选取能充分反映股票市场交易特征的定量指标. 不加选择的选取会使数据庞杂,增加系统负荷,降低网络性能;反之,选取的指标过少又难以刻画股票市场的特点. 同时需要考虑的是,各个时刻  $t$  的指标数据在一定范围内又是相互关联、相互影响的. 也就是说,样本的内部特征是交叉的. 通过对沪市综合指数各项指标的研究,确立输入向量见表1.

表1 确立输入向量

Tab. 1 Input vector

项目	含 义	项目	含 义
$\xi_1$	隐含阈值的系数	$\xi_9$	30 d 内平均成交额
$\xi_2$	今日最高指数	$\xi_{10}$	今日成交量
$\xi_3$	今日最低指数	$\xi_{11}$	30 d 内平均成交量
$\xi_4$	今日开盘指数	$\xi_{12}$	今日涨跌幅
$\xi_5$	今日收盘指数	$\xi_{13}$	昨日涨跌幅
$\xi_6$	今日成交额	$\xi_{14}$	前日涨跌幅
$\xi_7$	昨日成交额	$\xi_{15}$	10 d 内平均涨跌幅
$\xi_8$	前日成交额	$\xi_{16}$	30 d 内平均涨跌幅

这里,不仅考虑了短期因素的影响,而且也兼顾了长期因素的平衡作用. 以此为依据建立的样本融合了股市交易的局部特征和一定范围内的长期特征.

(3) 样本的规范化处理. 由于衡量的指标各不相同,原始样本各个分量数值的数量级有很大的差异. 对于某输入节点  $k$ ,如果  $\xi_k^{\mu}$  过大,则相应的更新量  $\Delta w_{jk}$  过大(见式(9)),从而使  $w_{jk}$  过大;这样在隐单元  $j$  的输出  $V_j$  中,  $w_{jk}\xi_k^{\mu}$  项的影响就要比其他分量大得多(见式(2)),导致其他分量几乎丧失了调控作用. 所以有必要对原始样本进行适度的规范化处理,依据各项指标可参照价值的不同,对其输入幅值重新进行合理的调整(放或缩),

使其变化范围大致均匀分布在某一区间  $(-A,A)$ . 表2列出了几种不同的规范化方案.

表2 几种不同的规范化方案

Tab. 2 Normalization of input vectors

方案	输入单元				
	$\xi_1$	$\xi_2 \sim \xi_5$	$\xi_6 \sim \xi_9$	$\xi_{10} \sim \xi_{11}$	$\xi_{12} \sim \xi_{16}$
A	-1	(0.1,0.15)	(0.1,1)	(0.1,1)	(-0.1,0.1)
B	-1	(0.1,0.15)	(1.0,10)	(1.0,10)	(-0.1,0.1)
C	-1	(0.1,0.15)	(5.0,50)	(1.0,10)	(-0.1,0.1)
D	-1	(0.1,0.15)	(0.5,5)	(0.5,5)	(-0.1,0.1)
E	-1	(2.0,3.00)	(0.5,5)	(0.5,5)	(-3.0,3.0)

图2给出了网络经不同规范化方案处理所得样本进行学习的 MSE(mean square error) 曲线. 实验初始条件  $\eta = 0.025$ ;  $\alpha = 0.01$ ; 权初始域  $(-1,1)$ ; 网络结构:16-7-1; 输入模式数  $P = 100$ ;  $g_1 = \tanh x$ ,  $g_2 = \tanh x$ . 以下各次实验如未作特别说明,初始数据及条件同上. 另外,将由式(6)所得的误差估计记为  $d_{\text{MSE}}$ ; 分类正确率记为  $R_{\text{ver}}$  (valid classification ratio).

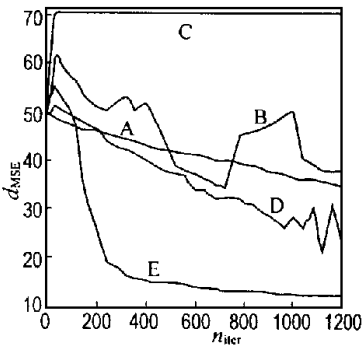


图2 各种方案的规范化效果对比

Fig. 2 Comparison of different normalizations

A(方案 A 简称为 A,下同)和 E 各分量的输入幅值分布比较均匀,所以 MSE 曲线下降得比较平缓. 而在  $\xi_6 \sim \xi_9$  输入幅值过大的情况下,C 很快就达到了饱和. 这是由于神经元的总输入过大使活化函数达到饱和,其一阶导数趋于 0,从而使权的更新量极小(由式(9)及式(10))而导致学习无效,也就是所谓的“饱和现象”;它是导致 BP 算法收敛缓慢的主要原因之一. B 和 D 的  $\xi_6 \sim \xi_{11}$  幅值偏大,成为调节权向量方向变化的主要影响因素,所以在  $\xi_6 \sim \xi_{11}$  波动较大的时候就会引起 MSE 曲线的剧烈振荡而难以收敛. 但由于 D 各分量的偏差比 B 小,波动也要比 B 小一些. 在 D 的基础上,对其他分量进行了适度放大而得到 E

(表2). 这样既在一定程度上遏制了  $\xi_6 \sim \xi_{11}$  项的影响, 又适当加大了其他分量的调节作用, 由图2可以看到, 方案 E 取得了较好的效果. 而通过对比 A 和 E 两条曲线, 可以说明并不是各输入分量间的偏差越小、分布越均匀就越好. A 的各输入分量虽然比 E 分布得更均匀, 但由于各个分量幅值压缩得过小, 一方面导致  $\omega$  的更新量减小(式(9))而降低了收敛速度; 另一方面幅值压缩得过小又难以有效地反映各个指标的变化情况, 样本特征的损失较大, 不但难以得到较高的学习精度而且也降低了网络的推广能力 (generalization performance).

综合以上实验及分析, 样本的规范化处理应在尽量避免“饱和现象”的前提下, 使样本在较大的幅值范围内均匀地分布, E 就是一种较为合适的规范化处理方案.

## 2.2 权及阈值初始值域的确定

尽管从总体来看, 权(含阈值项)是随着迭代的进行而更新的, 并且一般是收敛的, 但权的初始值太大, 可能导致网络很快就达到饱和. 另外, 权的初始值对网络的收敛速度也有一定的影响. 下面的实验(图3)通过对不同的权初始值域的对比, 可以帮助进一步分析问题.

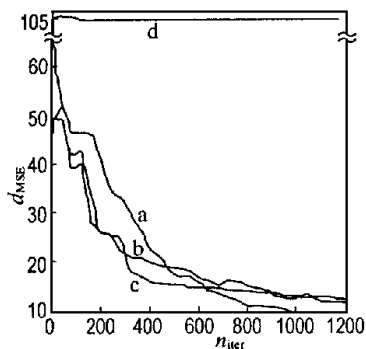


图3 不同权初始值域的对比

Fig. 3 Comparison of different initial weights

权随机初始域: a(-0.05, 0.05), b(-0.2, 0.2), c(-1, 1), d(-10, 10). 样本输入次序相同. 如图3所示, 在输入样本给定的情况下, d 的初始权值过大, 网络很快饱和; a 的初始权值比较合适, MSE 曲线稳定下降; b、c 的初始权值稍大, 但在迭代300次左右, 经过一段迅速下降之后, 也开始趋于平缓. 总体来看, 除了由于初始权值过大而使网络达到饱和外, a、b、c 曲线在经过短

时间的调整以后, 收敛速度大体是一致的, 即各次实验的搜索方向基本相同. 这从另一个角度反映了给定的样本集所张成空间的固有梯度走势. 而且, 实验也表明初始权值只要不是过大, 对网络整体性能的影响并不大.

为了尽可能避免饱和, 并考虑到网络的收敛速度以及股市数据的复杂性, 权的初始值域取  $(-0.05, 0.05)$  比较好, 因为较小的初始值域更利于权均匀地随机初始.

通过以上数值实验分析, 对样本和权进行了适度的规范化处理, 相当于已经完成了从研究对象到网络模型的接口工作.

## 3 隐单元数目的确定

网络拓扑结构对于整个网络的推广能力、计算效率都起着关键作用. 对于股市交易样本数据, 在前面的讨论中, 已经确定了网络的输入单元. 而由于对样本数据进行的是“涨”和“跌”(期望输出分别对应+1、-1)两种情况的分类及预测, 输出层取一个节点即可. 因而, 确定网络拓扑结构的关键就在于确定合适的隐单元数目.

如何确定隐单元的数目, 目前还没有明确而又广泛适用的结论, 但也有学者对此给出了一些经验公式. Mirchadani<sup>[8]</sup>认为隐单元数  $J$  与输入模式数  $P$  的关系为:  $J = \log_2 P$ , Kung 与 Hwang<sup>[9]</sup>也给出了类似的结论. 但是这些结论是否适合于具体的研究对象, 还需进一步验证. 以下几组关于隐单元数目的对比实验, 使之有了较为明确的实际依据, 同时也有助于了解隐单元数目对网络性能的影响.

迭代停止标准:  $R_{\text{ver}} \geq 80\%$ ; 最大循环迭代次数: 10 000; 循环迭代次数超过10 000次仍未达到迭代停止标准, 单独报告实验失败. 为了更好地对比实验结果, 排除随机因素的干扰, 同时也有助于对比不同样本集对网络学习效果的影响, 选取 a、b、c、d 4 个不同的样本集, 每个样本集包含100个样本; 随机排列输入样本; 取5次实验结果平均值. 在样本集和收敛标准相同的情况下, 可以通过比较其循环迭代次数来比较不同隐节点数目 BP 网络的分类能力, 如表3所示. 如4 492.4/1表示其中1次实验陷入局部极值或循环迭代次数超过10 000次仍未达到收敛标准, 即本次实验失败, 取其其余4次实验结果的平均值为4 492.4.

表3 不同隐单元数分类能力的比较

Tab. 3 Classification abilities with different numbers of hidden nodes

样本集	隐 单 元 数							
	3	4	5	6	7	8	9	10
a	4 492.4/1	4 070.4/2	1 860.5	1 532.6	1 641.0	1 611.4	1 646.1	1 535.8
b	952.6	520.7	462.9	467.1	465.2	481.9	482.6	450.1
c	3 836.1	859.4	893.5	796.4	817.5	757.8	776.9	797.1
d	6 533.3/4	7 787.4	5 688.2	4 729.9	4 551.0	4 137.0	4 742.6	4 914.8

实验结果表明,隐单元数目过少严重降低了网络的分类能力.但是当隐单元数目增加到一定程度(5或6)后,迭代次数并不随着隐单元数目的进一步增加而明显地减少.另一方面,过多的隐单元数会增加网络负荷,降低系统效率,更严重的是还可能降低网络的推广能力.表4对比实验说明了这个问题.

表4 不同隐单元数推广能力的比较

Tab. 4 Generalization performance with different numbers of hidden nodes

隐单元数	样本	迭代次数	$R_{\text{ver}}/\%$	$R'_{\text{ver}}/\%$
3	A	9 885	71	53
	B	4 869	79	55
	C	5 277	74	58
	D	7 174	75	47
5	A	8 173	83	60
	B	924	85	57
	C	3 114	82	55
	D	6 165	76	46
7	A	6 904	91	61
	B	613	87	66
	C	2 737	86	58
	D	6 628	88	51
9	A	6 268	96	52
	B	732	95	55
	C	2 973	91	49
	D	6 980	97	45

注: $R_{\text{ver}}$ 表示对给定样本集进行训练后,预测随后连续30个交易日所得的正确分类精度

分析实验结果可知,虽然较多的隐单元数目能够得到很高的学习精度,但却造成了对样本的“过拟合”(overfitting),从而降低了网络的推广能力.

另外,叶东毅<sup>[10]</sup>给出了一种动态修剪隐单元的算法;利用这种算法对不同的样本集和网络初始拓扑结构分别进行了4次修剪,如表5所示.

表5 运用修剪算法修剪隐单元

Tab. 5 Pruning of hidden units

样本集	初始拓扑结构	最终拓扑结构			
		$n = 1$	$n = 2$	$n = 3$	$n = 4$
a	14-10-1	14-8-1	14-7-1	14-6-1	14-7-1
c	14-8-1	14-7-1	14-5-1	14-6-1	14-6-1
d	14-10-1	14-6-1	14-9-1	14-7-1	14-8-1
e	14-9-1	14-9-1	14-7-1	14-7-1	14-6-1

综合表3~5的结果,认为当输入模式总数 $P = 100$ 时,隐单元数目 $J$ 应取6或7为好,这与Mirchadani给出的经验公式基本吻合.

4 活化函数的确定

活化函数 $g(x)$ 的作用是激活神经元,使其对输入产生响应.显然,为了应用梯度下降法对权进行学习,活化函数必须是可微的.在实际应用中,可根据需要选取适当的活化函数,常用的是Sigmoid类型的函数.由于活化函数 $g(x)$ 所具有的非线性特征,使BP算法训练的多层前馈式网络建立了从输入到输出的高度非线性映射,可以表达复杂的客观现象.而且,由于其导数常常可用 $g(x)$ 自身表示,所以在误差反向传播的过程中,不需要另外计算能量函数的导数,大幅度地减少了计算量,提高了网络的效率.但因其同时具有的饱和性,又导致BP算法收敛缓慢.

下面,通过对股市样本数据的学习训练,考虑3种最常见的Sigmoid函数:

$$f_1(x) = \frac{2}{1 + e^{-x}} - 1, f_2(x) = \tanh x,$$
$$f_3(x) = \arctan(x) \frac{2}{\pi}$$

表6 活化函数对比实验  
(取3次单独实验的平均值)

Tab. 6 Comparison of different active functions  
(average of three experiments)

活化函数	循环迭代次数	$d_{\text{MSE}}$	$R_{\text{ver}}/\%$
$f_1$	3 000.0	27.700	81.7
$f_2$	3 000.0	11.540	92.3
$f_3$	3 000.0	15.850	90.3
$f_1$	5 104.7	19.998	90.5
$f_2$	916.3	19.987	89.0
$f_3$	2 189.3	20.000	89.0

表6给出了这几种活化函数的表现.可见,采用 $f_2(x)$ 作为隐层和输出层的活化函数,网络的收敛速度最快.究其原因,可作如下分析.各活

化函数的导数为

$$\begin{aligned}f_1'(x) &= 2e^{-x}/(1 + e^{-x})^2 \\f_2'(x) &= 4/(e^x + e^{-x})^2 \\f_3'(x) &= 2/\pi(1 + x^2)\end{aligned}$$

考查如下极限：

$$\begin{aligned}\lim_{x \rightarrow \pm\infty} \frac{f_1'(x)}{f_2'(x)} &= \lim_{x \rightarrow \pm\infty} \frac{2e^{-x}}{(1 + e^{-x})^2} \cdot \frac{(e^x + e^{-x})^2}{4} = \infty \\ \lim_{x \rightarrow \pm\infty} \frac{f_1'(x)}{f_3'(x)} &= \lim_{x \rightarrow \pm\infty} \frac{2e^{-x}}{(1 + e^{-x})^2} \cdot \frac{\pi(1 + x^2)}{2} = \infty \\ \lim_{x \rightarrow \pm\infty} \frac{f_2'(x)}{f_3'(x)} &= \lim_{x \rightarrow \pm\infty} \frac{4}{(e^x + e^{-x})^2} \cdot \frac{\pi(1 + x^2)}{2} = 0\end{aligned}$$

显然,当  $x \rightarrow \pm \infty$  时,  $f_1'(x) > f_3'(x) > f_2'(x)$ ,这说明  $f_2(x)$  收敛于  $\pm 1$  的速度最慢. 即采用  $f_2(x)$  作为活化函数发生饱和的可能性较小,可在一定程度上提高网络的收敛速度. 实验结果也很好地说明了这一点. 这同时也启发作者进一步讨论这个问题,令

$$f_2(x) = \tanh \beta x$$

为隐层活化函数,可以通过适当调节  $\beta$  参数来调节  $f_2(x)$  的一阶导数值. 输出层活化函数取为  $g_2(x) = \tanh x$ . 表7列出了实验结果.

表7 调节  $\beta$  参数的对比结果  
(取3次单独实验的平均值)

Tab. 7 Comparison of different  $\beta$   
(average of three experiments)

$\beta$	循环迭代次数	$d_{MSE}$	$R_{ver}/\%$
0.10	3 000	24.33	80.3
0.30	3 000	16.90	85.0
0.65	3 000	7.97	94.0
1.00	3 000	13.20	91.7
2.40	3 000	27.33	80.2

分析实验结果可知,适当调节  $\beta$  值在一定程度上加快了网络的收敛速度. 而且随着  $\beta$  值的逐渐减小,加大了活化函数的非饱和区间,增大了权空间的调节范围,这样可以得到更好的精度. 但是,由于股票交易市场样本空间所具有的不规则特性又限制了调节的幅度. 因为过于平缓的活化函数在理论上虽然可以得到较好的速度和精度,却要付出相当大的计算代价. 从表8的结果来看,可以选取  $\beta = 0.65$ .

既然 Sigmoid 类型活化函数的饱和性是导致网络收敛缓慢的主要因素之一,那么是否可以考虑在输出层采用线性活化函数来减少饱和现象的

发生呢?令

$$f_4(x) = x$$

实验结果见表8.

表8 输出层采用线性活化函数的结果  
(取3次单独实验的平均值)

Tab. 8 Linear output active functions  
(average of three experiments)

隐层	输出层	循环迭代次数	$d_{MSE}$	$R_{ver}/\%$
$f_1$	$f_4$	5 000	38.27	73.3
$f_2$	$f_4$	5 000	26.39	78.3
$f_3$	$f_4$	5 000	23.77	82.0
$f_1$	$f_4$	20 290	24.97	86.6
$f_2$	$f_4$	8 152	24.99	84.2
$f_3$	$f_4$	8 336	24.99	83.4

分析表8的实验结果发现,输出层采用线性函数后,在学习精度和收敛速度上都有所降低. 进一步分析可知,采用线性函数虽然可以逼近输出值在  $(-\infty, +\infty)$  的任意函数,但是对本文所讨论的符号分类问题而言,显然输出值在  $(-1, +1)$  的 Sigmoid 型函数更优越,更容易收敛. 而尽管输出层采用线性函数对于减少饱和现象有一定好处,但在样本波动比较剧烈的区域,其自适应能力却要比 Sigmoid 型函数弱. 所以采用线性输出函数不但未能对网络性能进行优化,反而有所降低.

以上的实验和分析已经表明,在采用  $f_2(x)$  作为隐层及输出层活化函数的基础上,适度地调节  $\beta$  值,能够得到较好的学习精度和较快的收敛速度. 而输出层采用线性函数并不能提高网络的性能.

5 结果和讨论

至此,已经初步确定了比较适合于股票市场预测的网络结构和初始条件. 总结如下:学习步长  $\eta = 0.025$ ; 矩参数  $\alpha = 0.01$ ; 权及阈值的初始域  $(-0.05, 0.05)$ ; 对原始样本依据表2中的方案E进行规范化处理,网络拓扑结构(相应于100个输入样本):  $16(\xi_1 \sim \xi_{16})-7-1$ , 隐层活化函数  $g_1(x) = \tanh 0.65x$ , 输出层活化函数  $g_2(x) = \tanh x$ .

利用所得的网络结构及初始条件,在选定的沪市综合指数的数据中,取1998-07-27~1998-12-21连续100个交易日作为训练学习样本,对1998-12-23~1999-02-04的30个交易日进行预测,训练精度达到了98%,预测精度达到70%.初步说明了人工神经网络应用于股票市场的可行性及实用

价值.

对于股票市场走势的预测,国外已经有了一些这方面的工作<sup>[11]</sup>;尽管运用的方法和预测的指标各不相同,但都处于定性研究阶段,而且还没有得到令人满意的结果. 本文对应用 BP 网络预测国内股市的涨跌进行了初步的尝试,并给出了比较详细的应用方法和步骤. 通过大量的数值实验和分析对网络的性能进行了特定的优化后,得到了较好的预测精度. 但在样本处理、预测方法及网络的自适应能力等方面还有待进一步的改进,应该能得到更好的结果.

参考文献：

[1] WERBOS P. Beyond regression;New tools for prediction and analysis in the behavioral sciences[D]. Cambridge MA;Harvard University, 1974.

[2] 徐秉铮,张百灵,韦 刚. 神经网络理论与应用[M]. 广州:华南理工大学出版社, 1994.

[3] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning internal representations by error-propagation[A]. **Parallel Distributed Processing**;Vol 1[M]. Cambridge MA;MIT Press, 1986.

[4] HOSKINS J C, HIMMELBLAU D M. Process control via artificial neural networks and reinforcement learning[J]. **Comput Chem Eng**, 1992, 16(4):

141-151.

[5] HORNIK K, STINCHCOMBE M, WHITE H. Universal approximation of an unknown mapping and its derivatives using multilayer feedword networks[J]. **Neural Networks**, 1990, 3:551-560.

[6] HU D W , WANG Z Z. The approximation of arbitrary functions with multilayer BP neural networks[A]. **Proceedings of International Joint Conference on Neural Network'92**[C]. Beijing: [s n], 1992.

[7] 吴 微,许跃生. 用于神经网络的在线梯度学习算法的收敛性[A]. 第六届全国计算数学年会论文集[C]. 上海:[s n], 1999.

[8] MIRCHADANI G, CAO W. On hidden units in neural nets[J]. **IEEE Trans Circuits Syst**, 1989, 36(15):661-664.

[9] KUNG S Y, HWANG J N. An algebraic projection analysis for optimal hidden units size and learning rates in back-propagation learning[A]. **Proceedings of IEEE International Joint Conference on Neural Network'88**[C]. [s 1]:[s n], 1988. 363-370.

[10] 叶东毅. 前馈神经网络隐层节点设计的一个学习算法[J]. 电子学报, 1997, 25(11), 25-29.

[11] TAKASHI K, KAZUO A. Stock market prediction system with modular neural networks[A]. **International Joint Conference on Neural Network'89**[C]. [s 1]:[s n], 1989. 101-107.

Prediction of ups and downs of stock market by BP neural networks

WU Wei<sup>1</sup>, CHEN Wei-qiang<sup>2</sup>, LIU Bo<sup>2</sup>

( 1. Dept. of Appl. Math. , Dalian Univ. of Technol. , Dalian 116024, China;  
2. Dept. of Math. , Jilin Univ. , Jilin 130023, China )

**Abstract:** The authors utilize the good classification capability of multilayer feedforward back-propagation neural networks to forecast the ups and downs of the Shanghai Stock Index. The theoretical results and the numerical experiments show that artificial neural networks are feasible and efficient for the prediction of the Chinese stock market.

**Key words:** neural networks; on line/BP algorithms; stocks