

河北师范大学

本科生毕业论文（设计）册

学 院： 软件学院

专 业： 软件工程

班 级： 2009 级软件 5 班

学 生： 刘东

指导教师： 李文斌

河北师范大学本科毕业论文（设计）任务书

论文（设计）题目：_____基于神经网络的股指预测系统_____

学 院：_____软件学院_____专业：_____软件工程_____班级：_____2009 级软件 5 班_____

学生姓名：_____刘东_____学号：_____2009010231_____指导教师：_____李文斌_____职称：_____教授_____

1、论文（设计）研究目标及主要任务

- 研究目标：利用神经网络理论，建立一个合理的模型，对证券市场未来相关指标数据进行预测；并开发一个系统，对预测功能进行封装。
- 主要任务：开发一个系统，该系统能根据股票运行的历史数据，预测出未来 M 天趋势。比如：根据历史上某段时期的收盘价数据，预测未来 5 天的收盘价趋势。

2、论文（设计）的主要内容

- 采集数据，对数据进行预处理。
- 根据处理后的数据，生成训练样本。
- 研究神经网络理论，设计基本网络结构。
- 利用样本和网络进行实验，确定最优的网络结构。
- 基于结论，设计并实现系统。
- 使用系统进行预测，并与实际结果进行对比分析。

3、论文（设计）的基础条件及研究路线

- 基础条件：本文研究的基础是对股票知识和反向传播神经网络理论进行学习了解，而后将神经网络模型应用到股票指数预测中。
- 研究路线：要对股票指数进行预测，设计出合理的网络结构是关键。因此本文重点研究确定网络参数及对其进行优化的问题。当这些确定以后，就可利用 MATLAB 神经网络工具箱进行网络训练和应用。

4、主要参考文献

[1]曾华军译. 机器学习. 机械工业出版社.

[2] MATLAB 中文论坛. MATLAB 神经网络 30 个案例分析. 北京航空航天大学出版社.

5、计划进度

	阶段	起止日期
1	资料收集、综述报告撰写、理解系统需求	2012.12.01-2012.12.30
2	学习神经网络原理、掌握 MATLAB 神经网络工具箱的使用	2013.01.01-2013.01.30
3	系统实现	2013.02.01-2012.03.30
4	数据分析、论文撰写	2013.04.01-2012.04.30
5	论文答辩	2013.05.02-2013.05.08

指 导 教师：_____李文斌_____

_____2012 年_____12 月_____30 日_____

教研室主任：_____单博_____

_____年_____月_____日_____

河北师范大学本科生毕业论文（设计）开题报告书

软件学院

软件工程 专业 2013 届

[illegible]

河北师范大学本科生毕业论文（设计）文献综述

摘要：本文首先对神经网络的历史和发展状况做了一些总结；随后，就反向传播（BP）神经网络的基本原理做了简述，列举了几种改进方法；最后，提出了基于 BP 算法的神经网络的设计方法和原则。

1 神经网络及其研究历史

神经网络（Neural Networks）是对人脑或生物神经网络的抽象和建模，具有从环境学习的能力，以类似生物的交互方式适应环境。神经网络是智能科学的重要部分，以脑科学和认知神经科学的研究成果为基础，拓展智能信息处理的方法，为解决复杂问题和实现自动控制提供了有效途径。

1943 年，心理学家麦克洛奇（McCulloch）和数理逻辑学家皮兹（Pitts）提出了 M-P 神经网络模型，为神经元和网络结构的描述提供了形式化方法。1949 年心理学家赫布（Hebb）提出神经元突触连结强度可变的假设，认为学习过程最终发生在神经元之间的突触部位。赫布的学习规则为神经网络的学习算法奠定了基础。1957 年罗森博拉特（F.Rosenblatt）提出了感知器模型，第一次把神经网络从纯理论研究付诸工程实现，掀起神经网络研究的第一次高潮。20 世纪 70 年代末以来，神经网络的研究进展迅速；1982 年物理学家霍普菲尔特（Hopfield）提出的离散神经网络模型给出了网络稳定的判据，标志着神经网络研究高潮的又一次到来。

自 20 世纪 80 年代中期以来，许多国家都掀起神经网络研究的热潮，各种学术会议的规模逐步扩大；由于半导体电子器件、光学器件、分子器件的发展应用，神经网络计算机的开发也日趋广泛。现在，神经网络仍是颇具前景的研究课题。

2 反向传播（BP）神经网络

误差反向传播思想最早由 Bryson 等人于 1969 年提出，直到 1986 年，Rumelhart 研究小组在 *Nature* 发表其研究成果后才引起人们注意。BP 神经网络是一种基于反向传播思想的多层感知器，自出现以来，一直是神经网络中最为流行的神经计算模型。

BP 神经网络是全连接的，即在任意层上的一个神经元与它前一层上的所有神经元都连接起来，信号一层接一层的逐步传播。在网络中，有两种信号流通：函数信号和误差信号。函数信号是从输入层末端而来的输入信号（刺激），通过网络传播，到达网络输出层末端形成一个输出信号（响应）。误差信号产生于网络的一个输出神经元，并通过网络反向传播。输出神经元构成网络的输出层，余下的神经元构成网络的隐层。第一隐层的信号

是从感知单元（源节点）组成输入回馈给的，而它的结果信号又应用于下一个隐层。网络的其余部分以此类推。

BP 网络实现了一个从输入到输出的映射功能，而数学理论已经证明它具有实现任何复杂非线性映射的功能。它能从样本数据中学习知识，抽象出一般规律，具有泛化能力。但 BP 网络也存在诸多问题，如学习速度慢、易陷入局部极值、网络结构的选择尚无统一而完整的理论指导等。

针对这些问题，国内外已提出不少有效的改进方法。改进的方向传播算法大致分为两类：启发式算法和数值优化算法。启发式算法较容易理解和实现，但不容易分析其性能，如增加动量项算法和自适应学习率算法。数值优化算法以增加网络计算复杂度为代价，是网络收敛速度有重大改进，如共轭梯度算法和 Levenberg-Marquardt 算法。

3 BP 神经网络的设计方法

由于网络的开发设计还没有一套完善的理论做指导，应用中采取的主要设计方法是，在充分了解待解决问题的基础上将经验与试探相结合，选出较好的设计方案。

- 确定训练样本数

对于确定的样本数，网络参数太少则不足以表达样本中蕴含的全部规律，网络参数太多则不能使其得到充分训练。当不能提供较多训练样本时，应减少样本维数。

- 输入、输出量的选择

输入、输出量可分为数值变量和语言变量，当语言变量作输入、输出量时，需转换为离散的数值量。当不确定某个量是否适合作输入、输出量时，可分别训练含有和不含有该量的两个网络，对其效果进行对比。

- 输入量得提取与表示

很多时候，网络的输入量无法直接获得，需要用信号处理与特征提取技术从原始数据中提取能反映其特征的若干参数作为输入量。提取方法与实际问题相关。

- 输入、输出数据的预处理

由于各输入、输出数据的意义和量纲可能不同，其变化范围差异较大；如果输入数据的绝对值较大，则神经元输出饱和，权值调整进入误差曲面平坦区，因此，可用线性尺度变换将输入、输出数据所有分量限制在 0~1 或 -1~1 之间。当样本分布不合理时，可用对数变换或平方根变换改变其分布规律，使样本分布较均匀。

- 训练集的设计

一般来说，训练样本越多，网络精度越高，但样本数多到一定程度时，网络精度很难再提高。实践表明，输入、输出之间的非线性关系越复杂，为保证一定的网

络精度，需要的训练样本数越多。样本选取时，一定要有代表性，并注意样本类别的均衡，尽量使每个类别的样本数大致相同。样本的组织要注意不同类别样本的交叉输入或随机选择样本输入。

- 隐层数和隐结点数的设计

设计网络时，一般先考虑设一个隐层，当一个隐层的隐结点数很多而仍不能改善网络性能时，考虑设两个隐层，一般最多只需两个隐层。隐结点取决于样本的多少、网络蕴含规律的复杂程度等。确定最佳结点数可用试凑法，先设较少的隐结点训练网络，然后逐渐增加隐结点数。

- 训练次数的选择

实践表明，在某个训练次数之前，随着训练次数的增加，网络测试误差下降；而超过这个次数后，随着训练次数的增加，网络测试误差开始上升。为了获得较好的网络泛化能力，防止训练不足和过度，应通过多次试验确定训练次数。

参考文献

[1]史忠植. 神经网络[M]. 北京：高等教育出版社，2009.

[2]施彦，韩力群，廉小亲. 神经网络设计方法与实例分析[M]. 北京：北京邮电大学出版社，2009.

[3]周志华，陈世福. 神经网络集成[J]. 计算机学报，2002，25(01)：1-8.

[4]吴建鑫，周志华，沈兴华，陈兆乾. 一种选择性神经网络集成构造方法[J]. 计算机研究与发展，2000，37(09)：1039-1044.

人工神经网络

它们是什么？

它们如何工作？

它们在哪些领域应用？

森特路易斯大学商业与管理学院 Daniel Klerfors

1 写作目的

此篇报告想要展示什么是人工神经网络，如何工作，以及它们的最新应用，以便读者理解。该报告不是纯粹技术性报告，因此，不讨论深度数学表示，仅仅给出一种整体性的描述。

1.1 使用的方法

为了报告的客观性，本文采用描述的方式。本文中所用到的数据都是通过调研、查阅书籍、网上搜索获得的二手数据，还有一部分从 Terry L. Huston 教授讲授的课程中获得。

2 什么是人工神经网络

人工神经网络是一种大致模仿人脑的系统。与该领域相关的概念有很多，如联系系统、平行分配过程、神经计算、自然智能系统、机器学习算法和神经网络。它在特殊硬件和复杂软件内对人脑进行模拟，构成多层结构的简单单元称作神经元。每个神经元通过不同的效率的联系与周围特定神经元连接，联系代表连接关系的强度。学习是通过调整连接关系的强度以产生适合的网络输出来实现的。

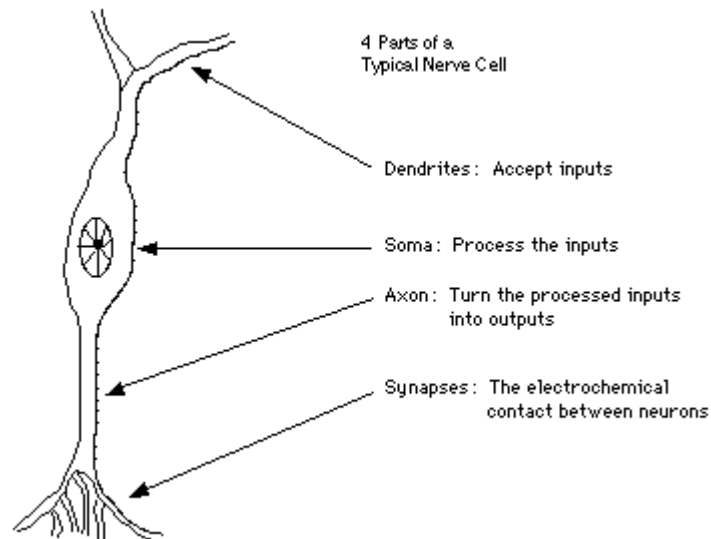
2.1 与人脑的类比

神经网络最基本的组成部分是模仿人脑结构建立的。一些神经网络结构并不与人脑中的结构十分接近，也有一些结构在人脑中没有对应的生理结构。但是，神经网络与生物大脑相似度很高，因此，很多术语都借鉴于神经科学。

2.2.1 生物神经元

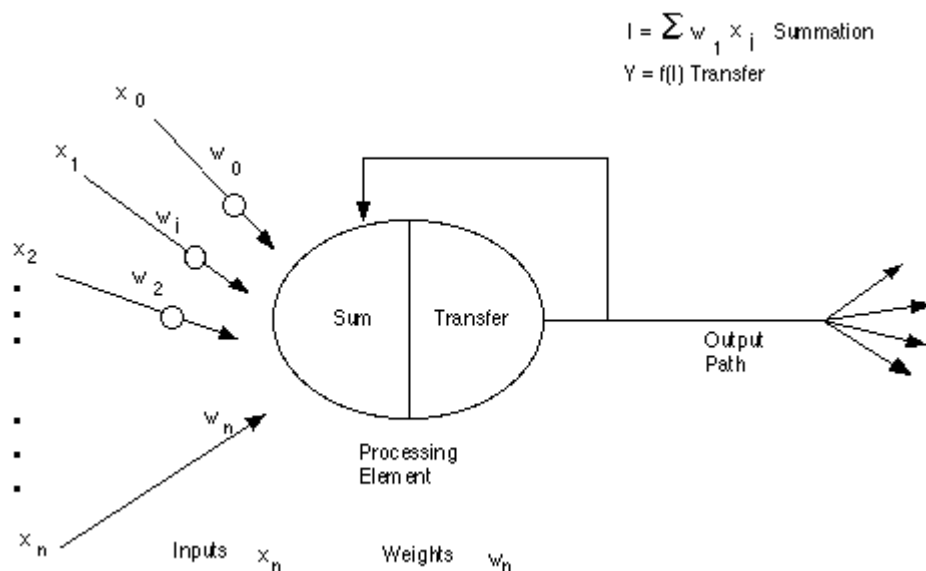
人脑中最基本的单元是一种特殊的细胞，这些细胞能提供给我们对每个动作进行记忆、思考和应用以往经验的能力。这些细胞被称作神经元，每个神经元可与多达 200000 个其他神经元相连。大脑的能力来自这些基本单元的数量和它们之间的复杂联系。

所有生物神经元都有 4 个基本组成部分，它们是树突、细胞体、轴突和突触。最基本的方式是，生物神经元从其它源头接收输入，将它们以某种方式组合起来，对结果做非线性处理，然后将最终结果输出。下面的图展示了简化了的生物神经元以及它的 4 个组成部分之间的关系。



2.1.2 人工神经元

神经网络最基本的单位——人工神经元——模仿生物神经元的 4 个基本功能。人工神经元比生物神经元简单的多；下面的图片展示了人工神经元的基本原理。



我们注意到，网络的各输入分量用数学符号 $x(n)$ 表示；各输入量乘以一个连接权值，连接权值用 $w(n)$ 表示。在最简单的情形中，将以上结果简单相加，然后通过变换产生一个结果，再输出。

尽管所有人工神经网络都是由这些基本的单元构造的，但基本构造单元或有所不同。

2.2 网络设计

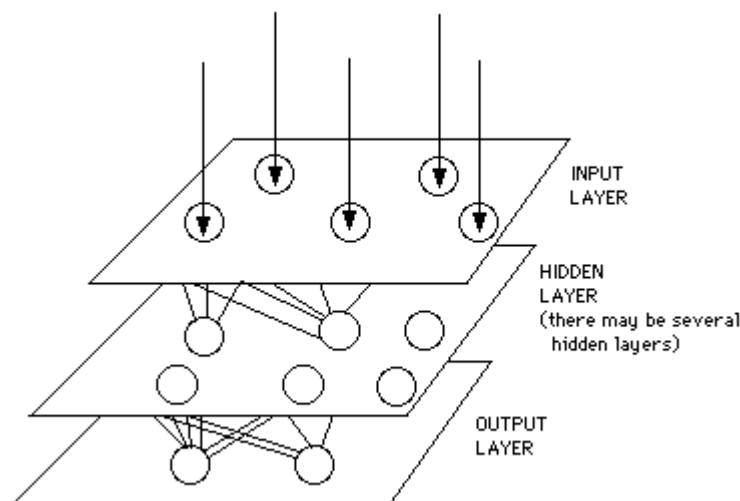
开发者在做出满意的设计之前，必须要经历一段试验性的和错误的设计过程。神经网络的设计是复杂的，也是系统开发人员主要考虑的。

设计神经网络的步骤包括：

- 在各层排列神经元；
- 确定层之间和同一层内神经元的连接类型；
- 确定神经元接收输入和产生输出的方式；
- 通过训练数据集使网络学习到合适的权值，从而确定网络连接强度；

2.2.1 神经网络的层

在微观构成上，生物神经网络以三维的方式构建，神经元之间似乎能够无限制的交流。但这在人工网络中并不成立。人工神经网络是原始人工神经元的简单聚类。聚类通过产生相互连接的层实现。层间如何连接也有多重方式。但是，所有人工神经网络都有一个相似的拓扑结构。一些神经元接收外界输入，另一些向外界输出，所有其它神经元在外部不可见。



如上图所示，神经元按层分类。输入层的神经元接收外部输入，输出层的神经元向用户或外界提供系统输出。在这两层之间通常还有一些隐层，上图展示的是只有一个隐层的简单结构。

当输入层接收输入，其神经元产生输出，这些输出又会成为其它层的输入。这个过程会一直进行直到某个条件得到满足或输出层提出请求并停止向外界输出。

要确定最佳的隐层神经元数量，通常要经历试验与错误。如果隐层神经元数量太多，

会使网络过拟合，影响泛化能力；训练集细节会被网络记牢，而不适用于新的数据集。

2.2.2 交流和连接的类型

神经元通过网络路径连接，这些路径携带某一神经元的输出信息，并以此作为下一神经元的输入。这些路径一般都是单向的，但神经元间也有可能存在双向连接，因为有可能存在反向的另一条路径。一个神经元从其它很多神经元接收输入信息，但只产生一个输出。

同一层的神经元可能会相互交流，也有可能毫无联系。一层的神经元至少与其它一层的神经元相连接。

2.2.2.1 层际连接

层间有很多不同类型的连接，这些连接称为层际连接。

- **全连接：**
第一层的每个神经元都与第二层的所有神经元相连接；
- **部分连接：**
第一层的神经元不必与第二层的所有神经元都相连接；
- **单向连接：**
第一层神经元的输出作为第二层神经元的输入，但第一层神经元不接收任何来自第二层神经元的信息；
- **双向连接：**
单向连接的基础上第二层神经元的输出反馈给第一层神经元；
单向连接和双向连接可以是全连接，也可以是部分连接。
- **分级连接：**
如果神经网络具有分级结构，低层神经元只能与下一层的神经元交流；
- **同步连接：**
层间采用双向连接，路径上能持续传播信息直到某一条件满足；

2.2.2.2 层内连接

在更为复杂的结构中，同一层内的神经元相互交流，这被称作层内连接。有两种类型的层内连接。

- **递归连接：**
同一层内的神经元为全连接或部分连接。当这些神经元从其它层接收到输入之后，它们会反复输出信息到另一层。通常在满足一些条件后，它们才能改变输出对象。
- **中心/周边型连接：**

在一层内的神经元对自身和紧邻的神经元的连接兴奋度较高，而抑制对其它神经元的连接。我们可以将这种连接想象成一个神经元的竞争群体。每个群体对自身和其成员产生兴奋，而抑制其它群体的成员。经过几轮的信号交换，具有活跃输出值的神经元在竞争中获胜，它能更新自己和群体其它成员的权值。（两个神经元间有两种连接方式：兴奋型和抑制型。在兴奋型连接中，神经元输出能提高与它连接神经元的活跃性。当连接时抑制型时，输出能降低接收神经元的活跃性或动作潜能。一个神经元引发另一个神经元的兴奋度增加，同时其它神经元使其兴奋度下降。这样就会一个神经元兴奋而另一个抑制。）

2.2.3 神经元学习

大脑基本是从经验中学习。神经网络有时又叫机器学习算法，因为改变连接权值（训练）能使网络学到解决问题的方法。神经元间的连接强度被存储为特定连接的权值。系统通过调整连接权值学习。神经网络的学习能力由它的结构和训练算法决定。

训练方法通常包含以下三种方案之一：

- **非监督型学习：**

隐层神经元必须不依赖于外界帮助而找到一种自我组织方式。对这种方式，没有为给定的输入向量提供用于检测网络的预测效果的输出样本。网络在应用中学习。

- **加强型学习：**

该方法基于外界的加强因素。隐层神经元的连接权值是随机安排的，然后根据与目标网络的差距重新调整。加强型学习又叫监督型学习，因为它需要一个教师。

这个教师可能是训练数据集，或是评价网络表现的观察者。

无论是非监督型学习还是加强型学习都由于随机的初始权值分配而导致找合适的权值的过程相对缓慢和低效。

- **反向传播：**

该方法已被证明对训练多层神经网络十分成功。该方法不仅给出了加强方法，错误信息也被反向传回，用于调整层间的连接权值和性能优化。它是一种监督型学习方法。

2.2.3.1 离线学习和在线学习

我们可以按另一种分类方式对学习方法进行分：离线和在线。当系统以利用输入数据改变权值的方式学习特定知识时，系统进入训练或学习模式。当系统用来指导决策时，就进入执行模式，有时也叫做回归模式。

- **离线学习：**

在离线学习方式中，系统一旦进入执行模式，则权值固定，不再改变。多数网络是离线学习方式。

- **在线学习：**

在在线或实时学习方式中，当系统进入执行（回归）模式，在做决策的同时继续学习。这种学习方式需要更复杂的网络结构设计。

2.2.3.2 学习法则

有多种较为常用的学习法则，这些法则都是用于更新连接权值的数学算法。多数法则都是最早的和最著名的 Hebb 规则的变型。人类关于神经系统运作方式的了解十分有限，学习过程自然比通过现有学习法则简单描述的过程更为复杂。专业出版物上有一些关于各种学习函数的新研究。以下是一些主要的学习法则的例子：

- **Hebb 规则：**

第一个也是最著名的法则来自 Donald Hebb，在 1949 年发表于《行为的组织》。基本规则是：如果一个神经元从另一个神经元获得输入，且如果这两个神经元都是高度活跃的（即在数学上有相同表示），则它们之间的权值应当被加强。

- **Hopfield 法则：**

该法则与 Hebb 规则类似，不同的是，该法则规定加强和减弱的最大程度。它可这样描述：如果期望输出和输入都处于激活或抑制状态，则通过学习率增大连接权值，否则通过学习率减小连接权值。（多数学习函数都有关于学习率或学习常数的规定，通常它是 0 到 1 之间的正数。）

- **差量规则：**

差量规则是 Hebb 规则更深刻的变型，它是最常用的。它的基本思想是通过不断修改神经元各输入的权值使期望输出与实际输出间的差距（差量）越来越小。该规则通过最小化网络均方误差的方式修改连接权值。每次误差都反向传回网络前一层，该反向传播过程持续到误差传到第一层。其网络类型为递归网络，反向传播得名于它计算误差项的方式。该规则也被称作 Windrow-Hoff 学习规则和最小均方学习规则。

- **Kohonen 法则：**

此算法由 Teuvo Kohonen 发明，是受到生物系统中的学习的启发。在该方法中，神经元竞争学习或更新权值的机会。具有最大输出的神经元在竞争中获胜并具有

抑制竞争者和激活相邻神经元的能力。只有获胜者能够输出，只有获胜者和其相邻神经元能够更新连接权值。**Kohonen** 法则不需要期望输出，因此它是非监督型学习。**Kohonen** 曾经用该法则和中心/周边型层内连接方式构造了具有非监督型学习法则的自组织神经网络。

2.3 神经网络的应用

神经网络在识别和匹配复杂、模糊或不完整的模式方面有着其他方法无可比拟的优势，已经被用来解决很多问题。

最常见的就是用于预测。在很多领域，预测可帮助确定优先权。例如，医院急救室气氛紧张，弄清谁最需要帮助可以实现更好的救护。一般所有单位都需要建立优先权，从而更好地分配资源。神经网络作为专家系统的学习机构，在股票预测方面有令人吃惊的准确度。神经网络也被用来为信用机构做破产预测。

尽管神经网络可以用来解释、预测、诊断、计划、管理、调试、修正、说明和控制，最成功的应用是分类和模式识别。系统从很多可能的分类中确定研究对象（例如，一种疾病、模式、图片、化学合剂、文字、顾客经济状况等）的种类，反过来提出一些建议（如治疗计划、经济计划）。

Nestor 公司利用神经网络做保险抵押风险评估，判断贷款风险是好是坏。神经网络也可用来将文字转化为语音，**NETtalk** 就是这样的一个系统。图像处理和模式识别可能是神经网络最活跃的研究领域之一。

神经网络的另一应用是特征识别和手写识别，被银行、信用机构和其它需要正确识别文件上字体的财经服务所采用。神经网络的识别功能在检测中被用来读取文字，这些文字通常只能人工书写。**HNC** 公司曾为 **BankTec** 银行开发过一个自动检测系统，该系统能加快检测速度并减少错误。

一个最著名的应用就是一些美国机场安装的炸弹侦查器。这种设备叫做 **SNOOPE**，它能根据物质的化学组态确定其组成成分。

在相关国际会议的文献中，能找到神经网络在很多领域应用的报告，如自动控制、语音、信号处理、作曲中的特征识别、心脏病和癫痫的检查、鱼的分类、优化和预定计划。多数应用尚处于研究阶段。

多数神经网络的应用基本分 5 类：

- 预测：

利用输入预测输出，例如股市选股、天气预报和识别癌症高危人群。

- **分类:**

利用输入确定分类，例如确定输入为字母 A，或确定输入为飞机录像视频的一部分并给出飞机类型。

- **数据联系:**

与分类类似，另外还识别出错误数据，例如不仅判断出所扫描数据的特点，还指出扫描仪何时不能正常工作。

- **数据概念化:**

分析各输入数据，确定它们的隐含关系，例如从数据库中提取最可能买特定商品的顾客的姓名。

- **数据过滤:**

使输入信号更平滑，例如除去电话信号中的噪声。

原文出处: <http://osp.mans.edu.eg/rehan/ann/Artificial%20Neural%20Networks.htm>

Artificial Neural Networks

What are they?

How do they work?

In what areas are they used?

Author: Daniel Klerfors, Saint Louis University,
School of Business & Administration

1 Purpose

This report is intended to review and help the reader understand what Artificial Neural Networks are, how they work, and where they are currently being used. This project is a result of an assignment in AI. The report is a non-technical report, thereby it does not go into depth with mathematical formulas, but tries to give a more general understanding

1.1 Method

To achieve the objectives with this report, the report is done by a descriptive approach. The data used in this report is secondary data gained by studying, reviewing books, Internet publications, and information gained in AI-lectures taught by Dr. Terry L. Huston.

2 What are Artificial Neural Networks?

Artificial Neural Network is a system loosely modeled on the human brain. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results.

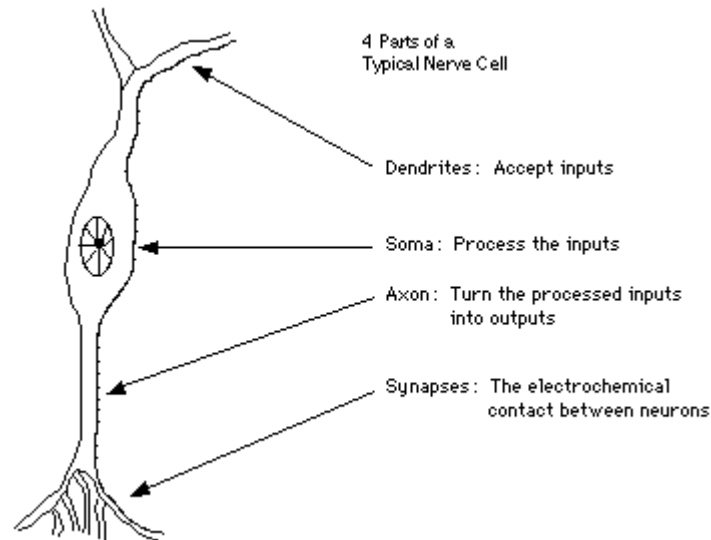
2.1 The Analogy to the Brain

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience.

2.1.1 The Biological Neuron

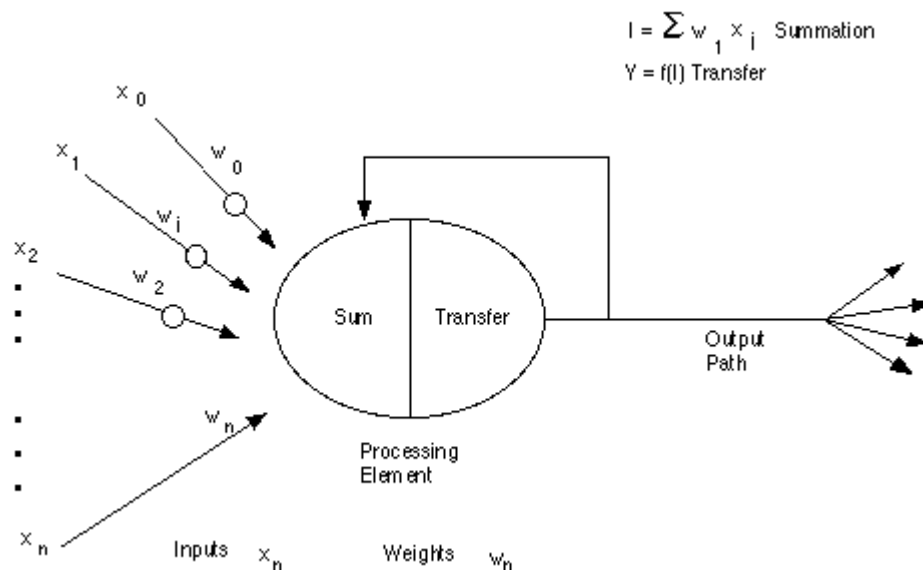
The most basic element of the human brain is a specific type of cell, which provides us with the abilities to remember, think, and apply previous experiences to our every action. These cells are known as neurons, each of these neurons can connect with up to 200,000 other neurons. The power of the brain comes from the numbers of these basic components and the multiple connections between them.

All natural neurons have four basic components, which are dendrites, soma, axon, and synapses. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then output the final result. The figure below shows a simplified biological neuron and the relationship of its four components.



2.1.2 The Artificial Neuron

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron; the figure below shows the basics of an artificial neuron.



Note that various inputs to the network are represented by the mathematical symbol, $x(n)$. Each of these inputs are multiplied by a connection weight, these weights are represented by $w(n)$. In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output.

Even though all artificial neural networks are constructed from this basic building block the fundamentals may vary in these building blocks and there are differences.

2.2 Design

The developer must go through a period of trial and error in the design decisions before coming up with a satisfactory design. The design issues in neural networks are complex and are the major concerns of system developers.

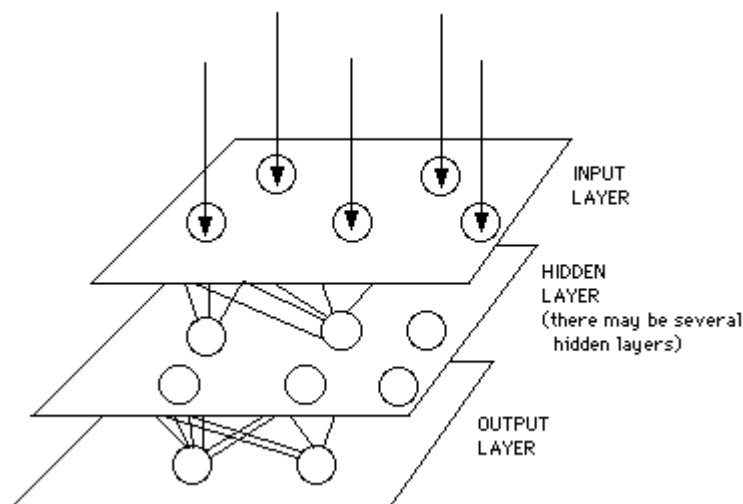
Designing a neural network consist of:

- Arranging neurons in various layers.
- Deciding the type of connections among neurons for different layers, as well as among the neurons within a layer.
- Deciding the way a neuron receives input and produces output.
- Determining the strength of connection within the network by allowing the network learns the appropriate values of connection weights by using a training data set.

The process of designing a neural network is an iterative process; the figure below describes its basic steps.

2.2.1 Layers

Biologically, neural networks are constructed in a three dimensional way from microscopic components. These neurons seem capable of nearly unrestricted interconnections. This is not true in any man-made network. Artificial neural networks are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers, which are then connected to one another. How these layers connect may also vary. Basically, all artificial neural networks have a similar structure of topology. Some of the neurons interface the real world to receive its inputs and other neurons provide the real world with the network's outputs. All the rest of the neurons are hidden from view.



As the figure above shows, the neurons are grouped into layers. The input layer consists of neurons that receive input from the external environment. The output layer consists of neurons that communicate the output of the system to the user or external environment. There are usually a number of hidden layers between these two layers; the figure above shows a simple structure with only one hidden layer.

When the input layer receives the input, its neurons produce output, which becomes input to the other layers of the system. The process continues until a certain condition is satisfied or until the output layer is invoked and fires their output to the external environment.

To determine the number of hidden neurons the network should have to perform its best, one is often left out to the method trial and error. If you increase the hidden number of neurons too much, you will get an overfit, that is, the net will have a problem to generalize. The training set of data will be memorized, making the network useless on new data sets.

2.2.2 Communication and types of connections

Neurons are connected via a network of paths carrying the output of one neuron as input to another neuron. These paths are normally unidirectional, there might however be a two-way

connection between two neurons, because there may be another path in reverse direction. A neuron receives input from many neurons, but produce a single output, which is communicated to other neurons.

The neuron in a layer may communicate with each other, or they may not have any connections. The neurons of one layer are always connected to the neurons of at least another layer.

2.2.2.1 Inter-layer connections

There are different types of connections used between layers, these connections between layers are called inter-layer connections.

- **Fully connected**

Each neuron on the first layer is connected to every neuron on the second layer.

- **Partially connected**

A neuron of the first layer does not have to be connected to all neurons on the second layer.

- **Feed forward**

The neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back from the neurons on the second layer.

- **Bi-directional**

There is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.

Feed forward and bi-directional connections could be fully- or partially connected.

- **Hierarchical**

If a neural network has a hierarchical structure, the neurons of a lower layer may only communicate with neurons on the next level of layer.

- **Resonance**

The layers have bi-directional connections, and they can continue sending messages across the connections a number of times until a certain condition is achieved.

2.2.2.2 Intra-layer connections

In more complex structures the neurons communicate among themselves within a layer, this is known as intra-layer connections. There are two types of intra-layer connections.

- **Recurrent**

The neurons within a layer are fully- or partially connected to one another. After these neurons receive input from another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to another layer. Generally some conditions among the neurons of the layer should be achieved before they communicate their outputs to another layer.

- **On-center/off surround**

A neuron within a layer has excitatory connections to itself and its immediate neighbors, and has inhibitory connections to other neurons. One can imagine this type of connection as a competitive gang of neurons. Each gang excites itself and its gang members and inhibits all members of other gangs. After a few rounds of signal interchange, the neurons with an active output value will win, and is allowed to update its and its gang member's weights. (There are two types of connections between two neurons, excitatory or inhibitory. In the excitatory connection, the output of one neuron increases the action potential of the neuron to which it is connected. When the

connection type between two neurons is inhibitory, then the output of the neuron sending a message would reduce the activity or action potential of the receiving neuron. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. One excites while the other inhibits.)

2.2.3 Learning

The brain basically learns from experience. Neural networks are sometimes called machine learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights.

The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training.

The training method usually consists of one of three schemes:

- **Unsupervised learning**

The hidden neurons must find a way to organize themselves without help from the outside. In this approach, no sample outputs are provided to the network against which it can measure its predictive performance for a given vector of inputs. This is learning by doing.

- **Reinforcement learning**

This method works on reinforcement from the outside. The connections among the neurons in the hidden layer are randomly arranged, then reshuffled as the network is told how close it is to solving the problem. Reinforcement learning is also called supervised learning, because it requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results.

Both unsupervised and reinforcement suffer from relative slowness and inefficiency relying on a random shuffling to find the proper connection weights.

- **Back propagation**

This method is proven highly successful in training of multilayered neural nets. The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance. A form of supervised learning.

2.2.3.1 Off-line or On-line

One can categorize the learning methods into yet another group, off-line or on-line. When the system uses input data to change its weights to learn the domain knowledge, the system could be in training mode or learning mode. When the system is being used as a decision aid to make recommendations, it is in the operation mode, this is also sometimes called recall.

- **Off-line**

In the off-line learning methods, once the systems enters into the operation mode, its weights are fixed and do not change any more. Most of the networks are of the off-line learning type.

- **On-line**

In on-line or real time learning, when the system is in operating mode (recall), it continues to learn while being used as a decision tool. This type of learning has a more complex design structure.

2.2.3.2 Learning laws

There are a variety of learning laws which are in common use. These laws are mathematical algorithms used to update the connection weights. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplification represented by the learning laws currently developed. Research into different learning functions continues as new ideas routinely show up in trade publications etc. A few of the major laws are given as an example below.

- **Hebb's Rule**

The first and the best known learning rule was introduced by Donald Hebb. The description appeared in his book *The organization of Behavior* in 1949. This basic rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

- **Hopfield Law**

This law is similar to Hebb's Rule with the exception that it specifies the magnitude of the strengthening or weakening. It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate." (Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one.)

- **The Delta Rule**

The Delta Rule is a further variation of Hebb's Rule, and it is one of the most commonly used. This rule is based on the idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a neuron. This rule changes the connection weights in the way that minimizes the mean squared error of the network. The error is back propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed forward, Back-propagation derives its name from this method of computing the error term.

This rule is also referred to as the Windrow-Hoff Learning Rule and the Least Mean Square Learning Rule.

- **Kohonen's Learning Law**

This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the neurons compete for the opportunity to learn, or to update their weights. The processing neuron with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted output, and only the winner plus its neighbors are allowed to update their connection weights.

The Kohonen rule does not require desired output. Therefore it is implemented in the unsupervised methods of learning. Kohonen has used this rule combined with the on-center/off-surround intra-layer connection (discussed earlier under 2.2.2.2) to create the self-organizing neural network, which has an unsupervised learning method.

2.3 Where are Neural Networks being used?

Neural networks are performing successfully where other methods do not, recognizing and matching complicated, vague, or incomplete patterns. Neural networks have been applied in solving a wide variety of problems.

The most common use for neural networks is to project what will most likely happen. There are many areas where prediction can help in setting priorities. For example, the emergency room at a hospital can be a hectic place, to know who needs the most critical help can enable a more successful operation. Basically, all organizations must establish priorities, which govern the allocation of their resources. Neural networks have been used as a mechanism of knowledge acquisition for expert system in stock market forecasting with astonishingly accurate results. Neural networks have also been used for bankruptcy prediction for credit card institutions.

Although one may apply neural network systems for interpretation, prediction, diagnosis, planning, monitoring, debugging, repair, instruction, and control, the most successful applications of neural networks are in categorization and pattern recognition. Such a system classifies the object under investigation (e.g. an illness, a pattern, a picture, a chemical compound, a word, the financial profile of a customer) as one of numerous possible categories that, in return, may trigger the recommendation of an action (such as a treatment plan or a financial plan).

A company called Nestor, have used neural network for financial risk assessment for mortgage insurance decisions, categorizing the risk of loans as good or bad. Neural networks has also been applied to convert text to speech, NETtalk is one of the systems developed for this purpose. Image processing and pattern recognition form an important area of neural networks, probably one of the most actively research areas of neural networks.

Another of research for application of neural networks is character recognition and handwriting recognition. This area has use in banking, credit card processing and other financial services, where reading and correctly recognizing handwriting on documents is of crucial significance. The pattern recognition capability of neural networks has been used to read handwriting in processing checks, the amount must normally be entered into the system by a human. A system that could automate this task would expedite check processing and reduce errors. One such system has been developed by HNC (Hecht-Nielsen Co.) for BankTec.

One of the best known applications is the bomb detector installed in some U.S. airports. This device called SNOOPE, determine the presence of certain compounds from the chemical configurations of their components.

In a document from International Joint conference, one can find reports on using neural networks in areas ranging from robotics, speech, signal processing, vision, character recognition to musical composition, detection of heart malfunction and epilepsy, fish detection and classification, optimization, and scheduling. One may take under consideration that most of the reported applications are still in research stage.

Basically, most applications of neural networks fall into the following five categories:

- **Prediction**

Use input values to predict some output. E.g. pick the best stocks in the market, predict weather, identify people with cancer risk.

- **Classification**

Use input values to determine the classification. E.g. is the input the letter A, is the blob of the video data a plane and what kind of plane is it.

- **Data association**

Like classification but it also recognizes data that contains errors. E.g. not only identify the characters that were scanned but identify when the scanner is not working properly.

- **Data Conceptualization**

Analyze the inputs so that grouping relationships can be inferred. E.g. extract from a database the names of those most likely to buy a particular product.

- **Data Filtering**

Smooth an input signal. E.g. take the noise out of a telephone signal.

原文出处: <http://osp.mans.edu.eg/rehan/ann/Artificial%20Neural%20Networks.htm>

河北师范大学

本科生毕业论文设计

基于神经网络的股指预测系统

作者姓名：刘东

指导教师：李文斌

所在学院：软件学院

专业（系）：软件工程

班级（届）：2013 届软件 5 班

二〇一三年四月三十日

目 录

基于神经网络的股指预测系统	1
1 引言	2
1.1 研究背景及意义	2
1.2 研究内容	2
2 基于神经网络的预测模型	4
2.1 BP 神经网络基本理论	4
2.1.1 BP 神经网络的模型	4
2.1.2 反向传播 (BP) 算法	5
2.2 对股票运行指标预测进行建模	7
2.2.1 输入、输出变量的选定	7
2.2.2 隐层结构的设置	7
2.2.3 训练样本的选择	7
2.2.4 数据的预处理	8
2.3 基于神经网络集成的预测	9
2.3.1 神经网络集成的方法	9
2.3.2 对性能的分析	10
3 预测模型的实验比较	12
3.1 一次性预测未来 N 天数据的情形	12
3.2 迭代方式预测未来 N 天数据的情形	13
3.3 集成的方式预测未来某天数据	13
3.4 实验总结及分析	14
4 系统设计	15
4.1 设计综述	15
4.2 数据的存储方式	15
4.3 神经网络算法设计	16
4.4 关键功能模块的设计	17
4.4.1 历史数据展示	17
4.4.2 收盘价预测	17

4.4.3	预测结果比较分析	17
5	系统实现	18
5.1	系统功能展示	18
5.2	预测结果与实际结果的对比分析	19
6	总结	23
	参考文献	24
	A Stock Index Forecasting System Based on Neural Networks	25

基于神经网络的股指预测系统

软件学院 软件工程专业

指导教师 李文斌

作 者 刘东

摘 要：自股票市场诞生以来，人们就以各种方式对其运行规律进行研究，提出了很多股票指数预测模型。人工神经网络作为一种新兴的智能技术，在这类预测中展现出巨大优势。但神经网络技术仍处在发展阶段，在实际应用中，如何设计网络结构仍缺乏完善的理论做指导，需要多次实验与尝试。

本文意在研究上证指数收盘价的短期预测。首先，本文基于 **BP** 神经网络理论提出了几种可行的预测模型，并用实验对它们的预测结果加以比较。随后，基于较优方案，利用 **MATLAB** 的图形界面开发平台及神经网络工具箱作为开发工具，开发了股指预测系统。最后，利用该系统进行多次收盘价预测，并与实际结果进行对比分析。

关键词：BP 神经网络 神经网络集成 股票指数 收盘价 时间序列

1 引言

1.1 研究背景及意义

股票市场作为一个高风险、高回报的领域，一直倍受人们关注。自其诞生以来，人们以各种方法研究股市运行规律，期望预测其未来走势，从而获得更大收益。同时，随着经济全球化，股票市场对世界经济的反应更加灵敏，其运行状况成为经济状态的重要指标。预测股票市场未来走势对经济的调控有重要意义。

为了准确地预测股票价格，经济学家基于动力学、统计学理论提出了一系列模型，如定价理论、投资组合模型、时间序列模型等。其中时间序列模型的基本思想都是将股票价格看作一组随时间变化的统计量，即时间序列，然后建立该序列的主观模型，利用模型进行计算和预测。但这些既定的模型有时并不能很好地反映股票价格与时间的关系。

近年来，随着计算机科学技术，特别是人工智能技术的发展，对时间序列的分析和预测提出了新方法，也为股票市场建模预测带来了新的活力。

神经网络（Neural Networks）又称人工神经网络（Artificial Neural Networks, ANN），是对人脑或生物神经网络的抽象和建模，是由大量处理单元组成的大规模非线性自适应系统，具有模仿大脑风格的信息处理能力和从环境学习的能力。神经网络以脑科学研究成果为基础，实现智能信息处理，是智能科学的重要组成部分。神经网络可通过对样本的学习，解决分类和回归估计等问题。

对于股票价格的预测，神经网络最大的优势是不依赖于主观模型，价格影响因素与股票价格之间关系的确定是一个通过学习而逐渐逼近的过程，这很适合股票市场相关影响因素多、非线性动力学特征复杂的特点。

1.2 研究内容

在股票市场数据中，有 6 个基本指标：开盘价、收盘价、最高价、最低价、成交量和成交金额。其中收盘价是指某种证券在证券交易所一天交易活动结束前最后一笔交易的成交价格，是当日行情的标准，又是下一个交易日开盘价的依据，可据以预测未来证券市场行情。因此，本文选定的预测价格为上证指数收盘价。

反向传播（BP）神经网络为神经网络的一类，能够表达各类复杂映射，且其算法较为简洁。BP 神经网络在各个领域内的应用都较为广泛，已经成功解决大量实际问题。因此本文选取 BP 神经网络作为预测模型。

利用 BP 网络对上证指数收盘价进行预测也是基于时间序列的递推预测。

定义时间序列：

$$\{\varphi_i\} \quad i=1,2,\dots \quad (1.1)$$

其中 φ_i 为历史某个交易日后第 i 个交易日的上证指数收盘价；假设对 $i \leq N$ ， φ_i 为已知数据，要研究的是以数据 $\{\varphi_{N-n+1}, \varphi_{N-n+2}, \dots, \varphi_N\}$ ($n \ll N$) 推测 $\{\varphi_{N+1}, \varphi_{N+2}, \dots, \varphi_{N+m}\}$ 。

从其他有关上证指数收盘价预测的资料来看，历史数据日期距被预测日期越远，预测误差越大，故本文选择收盘价短期预测，即 $m < 10$ 。

以上预测将收盘价的影响因素局限于收盘价历史数据，而股票价格受政治、经济、心理等多方面因素影响，这种预测有很大局限性。在上述的 6 个指标中，最高价、成交量等能一定程度上反映股市参与者的心理状况，有一定参考价值，且这些数据也比较容易获得。

因此，本文也尝试将这些变量添加到基于时间序列的预测中，预测结果以神经网络集成的方式呈现，具体在 2.3 节中介绍。

2 基于神经网络的预测模型

2.1 BP 神经网络基本理论

BP (Back-Propagation) 神经网络, 又称反向传播网络, 是一种基于误差反向传播理论的多层前馈网络, 其学习过程由函数信号正向传播过程和误差信号反向传播过程组成。下面分别以形式化的方式描述 BP 神经网络模型及反向传播算法。

2.1.1 BP 神经网络的模型

为了叙述方便, 这里给出相关符号的定义:

- BP 神经网络共 $m+1$ 层, 第 0 层为输入层, 第 m 层为输出层, 中间各层为隐层; 第 i 层神经元个数为 n_i ($i=0,1,\dots,m$)。
- 第 i 层第 j 个神经元的输入信号为 net_{ij} , 输出信号为 x_{ij} , 定义阈值 $x_{i0}=-1$, 第 i 层输入向量为 $net_i = (net_{i1}, net_{i2}, \dots, net_{in_i})$, 输出向量为 $x_i = (x_{i0}, x_{i1}, \dots, x_{in_i})$ ($j=1,2, \dots, n_i, i=0,1,\dots,m$); 输出层第 j 个输出信号为 $o_j = x_{mj}$, 网络输出向量为 $o = (o_1, o_2, \dots, o_{n_m})$ 。
- 第 $i-1$ 层到底 i 层间的权值矩阵为 $W_{(n_{i-1}+1) \times n_i}(i)$ ($i=1,2,\dots,m$)。
- 第 i 层转移函数为 $f_i(x)$ ($i=0,1,\dots,m$), 对输入层 $f_0(x)=x$, f_i 为光滑函数。

$m+1$ 层 BP 神经网络的数学模型为:

$$x_{ij} = f_i(net_{ij}) \quad (j=1,2,\dots,n_i; i=0,1,\dots,m) \quad (2.1)$$

$$net_i = x_{i-1} W(i) \quad (i=1,2,\dots,m) \quad (2.2)$$

即

$$net_{ij} = \sum_{k=0}^{n_{i-1}} x_{i-1k} w_{kj}(i) \quad (j=1,2,\dots,n_i; i=1,2,\dots,m) \quad (2.2^*)$$

由此可知, 输入信号 x_0 (也即输入层输出信号) 从输入层传入, 刺激各神经元产生相应, 逐层传递, 最终形成输出信号 o , 形成函数信号正向传播过程, 如图 2.1 所示。

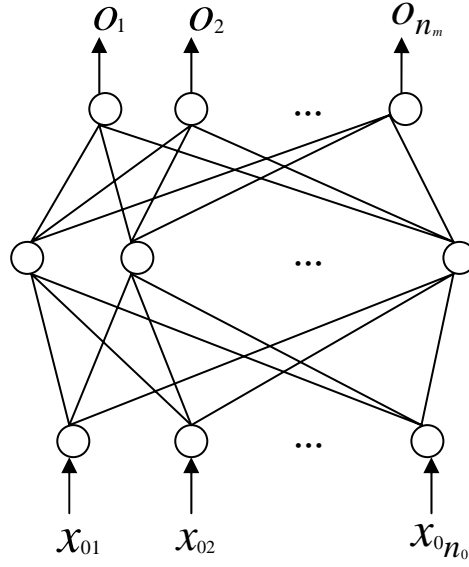


图 2.1 三层 BP 网络

2.1.2 反向传播 (BP) 算法

首先我们定义网络平方误差:

$$E = \frac{1}{2}(d - o)^2 = \frac{1}{2} \sum_{i=1}^{n_m} (d_i - o_i)^2 \quad (2.3)$$

其中 $d = (d_1, d_2, \dots, d_{n_m})$ 为期望输出向量; 对给定的 x_0 、 d , E 可由权值矩阵 $W(i) (i=1, 2, \dots, m)$ 确定, 即 E 是各层权值 $w_{ij}(k)$ 的函数。为了使网络输出更接近期望输出, 要通过调整各权值使 E 更小。BP 学习算法基本原理就是沿梯度负方向调整权值, 使 E 以最快速度下降, 即

$$\Delta w_{ij}(k) = -\eta \frac{\partial E}{\partial w_{ij}(k)} \quad (i=0, 1, \dots, n_{k-1}; j=1, 2, \dots, n_k; k=1, 2, \dots, m) \quad (2.4)$$

其中 $\eta \in (0, 1)$ 为学习率。

下面推导具体权值的调整式。

由式 (2.1) 和 (2.3) 得:

$$E = \frac{1}{2} \sum_{k=1}^{n_m} (d_k - x_{mk})^2 = \frac{1}{2} \sum_{k=1}^{n_m} (d_k - f_m(\text{net}_{mk}))^2 \quad (2.5)$$

又由式 (2.1) 和 (2.2*) 得:

$$\text{net}_{ij} = \sum_{k=1}^{n_{i-1}} f_{i-1}(\text{net}_{i-1k}) w_{kj}(i) - w_{0j}(i) \quad (2.6)$$

即为各层输入信号的递推关系式, 并容易看出对 $i > 0$, E 可表示为 net_{ij} ($j=1,2,\dots,n_i$)

和 $w_{pq}(k)$ ($k > i$) 的函数, 因此

$$\frac{\partial E}{\partial w_{ij}(k)} = \frac{\partial E}{\partial \text{net}_{kj}} \cdot \frac{\partial \text{net}_{kj}}{\partial w_{ij}(k)} = \frac{\partial E}{\partial \text{net}_{kj}} \cdot x_{k-i} \quad (2.7)$$

由递推式 (2.6) 得

$$\frac{\partial E}{\partial \text{net}_{i-1j}} = \sum_{k=1}^{n_i} \frac{\partial E}{\partial \text{net}_{ik}} \cdot \frac{\partial \text{net}_{ik}}{\partial \text{net}_{i-1j}} = \sum_{k=1}^{n_i} \frac{\partial E}{\partial \text{net}_{ik}} \cdot f'_{i-1}(\text{net}_{i-1j}) \cdot w_{jk}(i) \quad (2.8)$$

令 $\delta_{ij} = -\frac{\partial E}{\partial \text{net}_{ij}}$, 为局部梯度, 由 (2.5) 得 $\delta_{mj} = (d_j - o_j) \cdot f'_m(\text{net}_{mj})$; 综上, 各层权

值调整量由

$$\Delta w_{ij}(k) = \eta \delta_{kj} \cdot x_{k-i} \quad (i=0,1,\dots,n_{k-1}; j=1,2,\dots,n_k; k=1,2,\dots,m) \quad (2.9)$$

给出, 其中 δ_{ij} 由递推式

$$\begin{aligned} \delta_{i-1j} &= \sum_{k=1}^{n_i} \delta_{ik} \cdot f'_{i-1}(\text{net}_{i-1j}) \cdot w_{jk}(i) \quad (i=2,3,\dots,m; j=1,2,\dots,n_{i-1}) \\ \delta_{mj} &= (d_j - o_j) \cdot f'_m(\text{net}_{mj}) \quad (j=1,2,\dots,n_m) \end{aligned} \quad (2.10)$$

从而反向传播算法可由以下步骤组成:

步骤 1: 初始化各权值、阈值、计数器;

步骤 2: 接收一组输入信号, 由式 (2.1)、(2.2) 计算各层输入、输出信号;

步骤 3: 接收相应期望输出, 由式 (2.3) 计算网络平方误差;

步骤 4: 判断是否误差达到精度或计数器计数达到最大值; 若是, 则训练结束; 否则,

按式 (2.9)、(2.10) 调整各层权值, 计数器增 1, 返回步骤 2;

由式 (2.9)、(2.10) 可看出, 局部梯度由高层到低层依次得出, 权值矩阵也依据其由高层到低层依次调整, 可看作误差信号由高层到低层传播, 即反向传播。

2.2 对股票运行指标预测进行建模

本节介绍利用 BP 神经网络对上证指数收盘价进行预测的方法。从神经网络的结构可以看出,对实际问题进行建模,首先应明确输入、输出的实际意义,还要确定网络输入、输出变量的数量以及网络层数、各层神经元数。下面就分别讨论这些问题。

2.2.1 输入、输出变量的选定

既然是基于时间序列对上证指数收盘价进行预测,输入、输出量自然应为上证指数收盘价数据。

对于输入、输出变量个数,如果太少,则难以充分反映输入、输出变量之间的函数关系;如果太多,则其关系过于复杂,致使网络在训练时难以收敛。参考相关文献,对于收盘价,当输入变量个数为 10 时,达到饱和状态,故选定输入变量个数为 10。为了做短期预测,可选定为预测未来 5 天数据。

对于预测未来 5 天数据,又有多种方式。例如:设置网络输出变量个数为 5,一次性输出未来 5 天数据;设置网络输出变量个数为 1,当网络训练好以后,预测未来 1 天数据,并将预测数据作为输入数据迭代回网络,预测出未来第二天的数据,以这种迭代的方式,得出未来 5 天数据。在第 3 章中将以实验的方式比较这两种预测方式的效果。

综上,输入信号: $x = (x_1, x_2, \dots, x_n)$, $n=10$, 为连续 10 天上证指数收盘价; 输出信号: $o = (o_1, o_2, \dots, o_m)$, $m=1$ 或 5, 为随后连续 m 天上证指数收盘价。

2.2.2 隐层结构的设置

Hornik 等人在 1990 年证明,对于含一个隐层的三层 BP 网络,当隐层转移函数为 sigmoid 非线性函数,输出层转移函数为线性函数时,它能够实现任何复杂的非线性映射,也即三层 BP 网络具有泛化能力。就目前资料来看,三层 BP 网络的应用也十分广泛,故本文也采用三层 BP 网络。

对于隐层神经元个数,如果太少,网络就不能充分学习样本中的信息,从而不能充分反映样本规律;如果太多,网络就会过分学习非一般规律的样本细节,导致泛化能力下降。因此,隐层神经元具体个数 p 将在第 3 章中以实验方式确定,其原则是:先选取较少隐层节点数进行实验,然后逐渐增加,直至网络预测误差不再明显下降。

2.2.3 训练样本的选择

由于影响收盘价的因素较为复杂,而这些因素又是随时间变化的,因此未来收盘价与作为网络输入的变量之间的关系也是随时间变化的,有一定时间特征,故训练数据的日期

与要预测的数据的日期不应相距太远，训练样本也不宜太多。

则可用以下取样方法：收盘价序列为 $\{\varphi_i\}$ ，第一个输入、输出样本为

$$\begin{aligned} &(\varphi_1, \varphi_2, \dots, \varphi_n), \\ &(\varphi_{n+1}, \varphi_{n+2}, \dots, \varphi_{n+m}) \end{aligned} \quad (2.11)$$

第 i 个输入、输出样本为

$$\begin{aligned} &(\varphi_i, \varphi_{i+1}, \dots, \varphi_{i+n-1}), \\ &(\varphi_{i+n}, \varphi_{i+n+1}, \dots, \varphi_{i+n+m-1}) \end{aligned} \quad (2.12)$$

共 N 个样本， N 取 20；用于预测的输入样本为

$$(\varphi_{N+m}, \varphi_{N+m+1}, \dots, \varphi_{N+n+m-1}) \quad (2.13)$$

这样训练样本比较紧凑，距预测点也较近。

2.2.4 数据的预处理

为了使实际数据能够更合理的输入网络，现考虑对输入数据进行归一化处理。

为了使网络有更好的泛化能力，现取隐层、输出层转移函数都为非线性 Sigmoid 函数：
双曲正切函数

$$f(x) = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.14)$$

该函数取值在 $(-1,1)$ 内变动且满足

$$f' = 1 - f^2 \quad (2.15)$$

假设隐层神经元输出信号为 $y = (y_1, y_2, \dots, y_p)$ ，对于三层 BP 网络，由式 (2.9)、(2.10)

可得各层权值调整量为：

$$\Delta w_{ij}(1) = \eta \left(1 - y_j^2\right) x_i \sum_{k=1}^m (d_k - o_k) \left(1 - o_k^2\right) w_{jk}(2) \quad (i=0,1,\dots,n; j=1,2,\dots,p) \quad (2.16)$$

$$\Delta w_{ij}(2) = \eta (d_j - o_j) \left(1 - o_j^2\right) y_i \quad (i=0,1,\dots,p; j=1,2,\dots,m) \quad (2.17)$$

上证指数收盘价的数据都为比较大的数，如果直接做输入量，可能会导致隐层输入值较大，由 (2.14) 函数的性质知隐层输出量 y_i 接近 1，神经元输出饱和；又由式 (2.16) 知输入层到隐层间的权值调整量趋于 0，该层权值训练停滞。为此，对输入变量作如下变

换：

$$x_0 = \frac{\max\{x_i\} + \min\{x_i\}}{2}$$

$$x_i = \frac{x_i - x_0}{\max\{x_i\} - \min\{x_i\}} \quad (2.18)$$

其中 $\max\{x_i\}$ 、 $\min\{x_i\}$ 为整个输入样本的最大、最小值。这可将输入量控制在 $[-0.5, 0.5]$ 内，有效地防止输出饱和。

由函数 (2.14) 的性质知网络输出在 $(-1, 1)$ 内，因此对输出量也要做此变换。由于输入、输出样本都是收盘价数据，输入、输出量可做统一变换，因此只要将样本中所有收盘价数据 φ_i 都按上述变换处理即可，即

$$\varphi_0 = \frac{\max\{\varphi_i\} + \min\{\varphi_i\}}{2}$$

$$\varphi_i = \frac{\varphi_i - \varphi_0}{\max\{\varphi_i\} - \min\{\varphi_i\}} \quad (i=1, 2, \dots, M) \quad (2.18^*)$$

其中 $\max\{\varphi_i\}$ 、 $\min\{\varphi_i\}$ 为数据样本中收盘价数据的最大、最小值， M 为数据个数。要进行预测时，只需将输出量作逆变换，即：

$$o_i = o_i \cdot (\max\{\varphi_i\} - \min\{\varphi_i\}) + \varphi_0 \quad (i=1, 2, \dots, m) \quad (2.19)$$

2.3 基于神经网络集成的预测

虽然神经网络在很多领域都有广泛应用，但对于特定问题，网络结构的确定缺乏严格的理论做指导，这就使人们只能凭经验和不断实验来设计网络。如果网络结构不合理，其泛化能力将大大下降。

神经网络集成是人们提出的一种开创性方法，其算法比较简洁，且能显著提高神经网络泛化能力。

2.3.1 神经网络集成的方法

神经网络的集成的一般过程是：先训练多个神经网络，再以某种方式对各个网络的输出进行结合。

对于用于集成的个体神经网络的生成方法，一般的原则是尽量增加它们的不相关。在

这方面，有 Bagging、Boosting 等算法；还有学者利用遗传算法，对一组个体神经网络进行筛选，剔除个别网络，做选择性集成。

本文要对上证指数收盘价进行预测，又要将该功能封装到系统中，可获取的对股价影响的量化因素较少，因此可用于集成的不相关的个体网络数量较少，故决定训练 3 个不同的神经网络作为集成个体，且不再对其进行筛选。

同 1.2 节所述，利用基本指标中的最高、低价，成交量、金额数据，3 个网络输入如下：

$$x(1) = (x_1, \dots, x_n, x_{n+1}(1), \dots, x_{n+4}(1)) \quad (2.20a)$$

其中个分量依次为连续 1 到 n 个交易日收盘价、第 n 日最高价、1 到 n 日最高价、第 n 日最低价、1 到 n 日最低价；

$$x(2) = (x_1, \dots, x_n, x_{n+1}(2), \dots, x_{n+4}(2)) \quad (2.20b)$$

其中个分量依次为连续 1 到 n 个交易日收盘价、第 n 日成交量、1 到 n 日平均成交量、第 n 日成交金额、1 到 n 日平均成交金额；

$$x(3) = (x_1, \dots, x_n) \quad (2.20c)$$

为 1 到 n 个交易日收盘价时间序列。

考虑到集成要训练多个网络，运行代价加大，所以仅在单个网络预测效果较差的第 5 天进行集成预测，故 3 个网络的输出 $o(1)$ 、 $o(2)$ 、 $o(3)$ 都为上述 n 个交易日后第 5 个交易日的收盘价。

对于最高价、最低价、成交量、成交金额，其数据范围可能与收盘价有所差异，为了使各个输入变量处于同等地位，对训练样本中以上 4 组数据也做类似于 (2.18*) 的变换，使得各个输入变量处于相同变化区间。

对于各个神经网络输出的结合方法，可采用简单算术平均或加权平均。虽然理论已证明，合理的使用加权平均可使集成的网络具有更好的泛化能力，但简单算术平均不需要更复杂的分析，也能获得较好的效果，且在实际问题中应用也较多，故对上述 3 个网络的输出采用简单算术平均的方式结合。

2.3.2 对集成性能的分析

假设共有 n 个网络用于集成，每个网络都有 m 个输出，第 i 个网络的第 j 个输出（预测值）为 $o_j(i)$ ；第 j 个期望输出（实际值）为 d_j 。

对简单算术平均，各个网络的权值 $\omega_j = \frac{1}{n}$ ，则神经网络集成的第 j 个输出为

$$o_j = \frac{1}{n} \sum_{i=1}^n o_j(i) \quad (2.21)$$

预测值与实际值的平方误差为

$$\begin{aligned} SE &= (o_j - d_j)^2 = \frac{1}{n} \sum_{i=1}^n (o_j - o_j(i) + o_j(i) - d_j)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (o_j(i) - d_j)^2 + \frac{1}{n} \sum_{i=1}^n (o_j - o_j(i))^2 - 2 \frac{1}{n} \sum_{i=1}^n (o_j(i)^2 - o_j(i) \cdot o_j) \\ &= \frac{1}{n} \sum_{i=1}^n (o_j(i) - d_j)^2 - \frac{1}{n} \sum_{i=1}^n (o_j - o_j(i))^2 \\ &\leq \frac{1}{n} \sum_{i=1}^n (o_j(i) - d_j)^2 = \frac{1}{n} \sum_{i=1}^n SE_i \end{aligned} \quad (2.22)$$

预测值与实际值的绝对误差为

$$\begin{aligned} AE &= |o_j - d_j| = \left| \frac{1}{n} \sum_{i=1}^n o_j(i) - d_j \right| = \frac{1}{n} \left| \sum_{i=1}^n (o_j(i) - d_j) \right| \\ &\leq \frac{1}{n} \sum_{i=1}^n |o_j(i) - d_j| = \frac{1}{n} \sum_{i=1}^n AE_i \end{aligned} \quad (2.23)$$

由此可见，无论是平方误差，还是绝对误差，集成后的误差都小于原来各网络误差的平均值。因此，在不确定选定的 3 个网络的预测误差孰大孰小的情况下，简单算术平均的结合方式还是比较有效的。

3 预测模型的实验比较

本章以实验的方式对预测模型进行研究比较。

以下实验中实验数据，即上证指数收盘价均来自搜狐财经网 (<http://q.stock.sohu.com/zs/000001/lshq.shtml>)，选取2012年5月至2013年3月的日数据，其原始数据格式如下：“日期 开盘价 收盘价 涨跌额 涨跌幅 最低 最高 成交量(手) 成交金额(万) 换手率”。

MATLAB 神经网络工具箱是目前一种应用比较广泛的神经网络设计开发平台，它提供了各种网络的训练与仿真函数，并对算法进行了优化，大大方便了编程实现。因此，本文选用 MATLAB 7.0 中的 BP 神经网络工具箱作为实验平台。

3.1 一次性预测未来 N 天数据的情形

一次性预测未来多天，即将未来多天收盘价预测数据作为网络输出。按照 2.2 节中的模型，3 层 BP 网络输入层结点个数为 10，输出层结点个数为 5。

按照 2.2.3 小节的取样方式，多次取样进行实验，并由少到多增加隐层结点数，最终确定隐层结点数为 12。

现取 5 组不处于同时间段的数据，每组形成 20 个样本，进行训练和预测，并与实际数据做比较，其相对误差百分比 ($100 \times (\text{预测值} - \text{实际值}) / \text{实际值}$) 统计情况如表 3.1 所示。

表 3.1 一次性预测的误差统计表

	第 1 组	第 2 组	第 3 组	第 4 组	第 5 组	各分量相对 误差的平方 和
	-0.1546	7.0203	4.7198	4.226	-5.0636	115.0841
	-0.5418	9.0309	4.865	5.2446	-3.405	144.6188
	-2.84	6.713	5.2704	8.6901	-5.8335	190.4546
	-3.5888	1.5411	6.1806	6.7444	-0.9275	99.80148
	-2.8463	-1.1318	-2.8935	7.8465	1.9249	83.02754
组相对误差平方和	29.36396	179.5621	120.294	227.9372	75.82929	总计： 632.9866

3.2 迭代方式预测未来 N 天数据的情形

迭代方式预测即：通过训练后的网络预测最近未来 1 天的数据；然后将该预测数据加上前 $n-1$ 天的历史数据作为网络输入，预测未来第 2 天的数据；以此类推，将未来 4 天的预测数据加上前 $n-4$ 天的历史数据作为网络输入，预测未来第 5 天的数据。

3 层 BP 网络输入层结点数为 10，输出结点数为 1，多次取样后进行实验并由少到多增加隐层结点数，最终确定隐层结点数为 7。

现取同 3.1 节中的 5 组数据，每组形成 24 个样本，进行训练并预测，与实际数据比较后，5 个分量的相对误差百分比统计情况如表 3.2 所示。

表 3.2 迭代预测的误差统计表

	第 1 组	第 2 组	第 3 组	第 4 组	第 5 组	各分量相对 误差的平方 和
	-2.7162	-0.6835	-3.7688	1.1199	-1.4469	25.39646
	-2.2118	-3.9553	-2.5618	2.2825	-0.6	32.66908
	-1.2725	-6.2575	-0.9859	3.0172	-6.1795	89.03603
	-0.4415	-5.5603	6.5011	1.5685	-12.4319	230.3885
	2.1995	-2.4984	7.0964	3.3522	4.3636	91.71695
组相对误差 平方和	18.92178	92.42556	114.3619	29.26492	214.2329	总计： 469.207

3.3 集成的方式预测未来某天数据

按照 2.3 节的叙述，建立 3 个网络，其输入、输出结点个数分别为(14,1)、(14,1)、(10,1)。将实验数据中收盘价、最高价、最低价、成交量、成交金额数据都进行（2.18*）的变换，并多次取样后进行实验，可确定各网络隐层结点数都为 7 即可。

仍选取 3.1 节中的 5 组数据，每组形成 20 个样本，对 3 个网络分别进行训练和预测，并将其预测结果集成。现将 3 个独立网络和集成后的相对误差都做统计，见表 3.3。

表 3.3 独立网络与集成网络预测误差比较表

网络类型	第 1 组	第 2 组	第 3 组	第 4 组	第 5 组	各网络的相对误差的平方和
Net1	-0.2852	6.1131	-3.6605	0.3768	-3.1335	60.81139
Net2	0.3994	10.4865	-3.6303	2.5433	2.0041	133.7901
Net3	-0.1739	-2.2968	-2.8173	2.1069	0.2135	17.72732
集成网络	-0.0199	4.7676	-1.7384	1.6757	-0.3053	28.65362

3.4 实验总结及分析

从表 3.1 中可以看到，一次性预测方式的预测误差没有显著偏向，各输出分量的预测误差相当，即对未来 5 天收盘价预测的准确率无明显差别。

从表 3.2 来看，迭代的预测方式对天数较短的预测（未来 1 天）准确率较高，而对天数较长的预测可能有较大误差。而且，迭代式预测的总体预测效果优于一次性预测。

从表 3.3 中看出，集成后的网络误差变化不大，比较平滑；虽然集成网络预测结果可能不及各独立网络中的最优预测结果，但当某个网络的预测误差较大时，集成网络仍有较小误差。

3.1 节与 3.2 节的实验结果表明，迭代方式预测对预测较短天数的数据效果较佳，且其总体预测效果优于一次性的预测；故对于未来 4 天的收盘价预测，采用迭代方式预测。

对于可能出现较大误差的第 5 天收盘价预测，采用集成的方式进行预测，使预测误差不至于太大，作为未来 5 天收盘价趋势的一个加强点。

4 系统设计

4.1 设计综述

本文致力于上证指数收盘价的预测，进行了一系列研究后，期望以一种快捷的方式应用其结果，并进行进一步研究。因此，要设计一个系统将预测方法做封装，以提供更便捷的使用方式和更直观的展示方式。

本系统最主要的功能就是利用历史数据对未来数据进行预测；其次，为了方便对比研究，系统还提供了简单的历史数据展示及预测结果与实际结果对比的功能。

第3章中BP神经网络的设计利用的是MATLAB7.0提供的神经网络工具箱；为了方便系统实现，仍选用它做BP算法实现。对于整体系统的开发，采用MATLAB提供的图形界面开发平台GUIDE(GUI Builder)。MATLAB的编程语言比较擅于数值运算，这大大方便了系统实现；而且这使整个系统的开发与关键算法的实现都利用同一平台，避免了混合编程的麻烦。

4.2 数据的存储方式

由于系统实现的功能比较单一，其内部不含有任何类结构，内部数据结构不过是用于存储训练样本、预测样本和预测结果等数据，则采用相应大小的矩阵即可。

其外部数据即为用于预测的上证指数收盘价等数据。考虑到这些数据需能够通过较为方便的方式获得，故采用第3章中搜狐财经网提供的数据。网上这些数据为开源数据，且实时更新。

考虑到系统需要，这些数据要能够比较方便地进行更换，且没有保密需要，则没有必要采用数据库存储，只需用文件存储即可，现选定为记事本文档(.txt)。

网上的原始数据含多个字段，而只有部分是系统需要的，因此需要对数据进行筛选。由于每次进行训练与预测的数据不是太多，大概30个，则可将数据按照其原始格式存储，将筛选工作放在网络训练时进行，即样本库是临时建立的，这样并不会花费太多时间和存储空间。

总之，数据存储方式如下：

- 每条数据的格式为：“日期 开盘价 收盘价 涨跌额 涨跌幅 最低 最高 成交量(手) 成交金额(万) 换手率”，各数据项间用空格隔开；
- 每条数据在.txt文件中占一行，数据按日期递减的方式由上到下排列；
- 文件中不含表头，其中第1行数据为要预测的交易日之前一个交易日的数据；

这种数据存储格式与数据在网页上显示的格式一致,因此只要在网页上将需要的数据选中后,经复制、粘贴即可完成数据获取。

4.3 神经网络算法设计

系统中用到的最主要的算法就是 BP 神经网络算法,而这些算法由 MATLAB 神经网络工具箱提供。

要进行网络训练,首先应建立网络对象,可用函数 `newff()` 新建前馈 BP 网络:

$$\text{net} = \text{newff}(\text{LIMIT}, [\text{N1}, \text{N2}, \dots, \text{Nn}], \{\text{F1}, \text{F2}, \dots, \text{Fn}\}, \text{BTF}, \text{BLF}, \text{PF})$$

`LIMIT` 为输入量得范围矩阵; `[N1,N2,⋯, Nn]` 为各隐层和输出层结点数; `{F1,F2,⋯,Fn}` 为各隐层和输出层的转移函数; `BTF` 为网络的训练函数; `BLF` 为网络权值学习函数; `PF` 为用于评定网络性能的函数。

2.2 节已提到,隐层与输出层都采用非线性 Sigmoid 函数作为转移函数,选用函数 `tansig()`,其函数原型为双曲正切函数 (2.14)。

这里选用批处理的方式进行权值学习,即所有训练样本输入后,计算网络总误差,依据总误差进行权值调整。因此,只要设置训练函数为 `trainlm()`。它利用对 BP 学习算法改进的 Levenberg-Marquardt 算法,能极大地提高学习速度。

网络性能函数采用默认的均方误差函数 `mse()`,即 $\sqrt{\frac{2E}{n_m}}$, E 为 (2.3) 中定义的平方误差。

由于初始化函数 `init()` 被 `newff()` 自动调用,各权值、阈值按默认参数初始化,因此不必再显式调用初始化函数。

网络训练前,设置最大训练次数为 5000,目标误差为 0.001,其它参数为默认值。网络训练后,用 `sim()` 函数做仿真模拟,即由预测数据得出预测结果。

具体过程如下:

```
net = newff(LIMIT,[hide_num out_num],{'tansig','tansig'},'trainlm');
net.trainParam.epochs = 2000;
net.trainParam.goal = 0.001;
net = train(net,X,D);
O = sim(net,SX);
```

其中 `hide_num`、`out_num` 为隐层、输出层节点数; X 为输入矩阵, D 为期望输出矩阵;

SX 为预测输入向量，O 为预测结果向量。

4.4 关键功能模块的设计

4.4.1 历史数据展示

历史数据展示设计如下：

- 数据获取：根据需要从搜狐财经网筛选出相应范围的日数据，将其复制，并粘贴到数据文件 lib.txt，保存。
- 数据展示：系统启动后，自动从文件读取数据并在主界面显示。显示有两种方式。
列表显示：一行为一个交易日数据，以日期递减的方式共显示多个连续交易日的数据，数据格式为“日期/收盘价”。图表显示：以日期为横轴，收盘价为纵轴，用蓝线绘制上述多个数据点的折线图。在主界面点击“Update”按钮可重新读取数据并显示。

4.4.2 收盘价预测

收盘价预测设计如下：

- 数据准备：按 4.4.1 中的方法获取数据，该数据要包含要预测的首个交易日之前连续的至少 30 个交易日的数据，否则，进行预测时系统会提示训练数据不足。
- 预测：在主界面，选中列表中一行（即一个）数据，该数据作为要预测的首个交易日之前一个交易日的的数据；点击“Forecast”按钮，系统自动生成训练样本进行训练，并预测选中数据后 5 天的数据。
- 结果展示：预测后，系统会在主界面显示预测结果。显示也有两种方式：以静态文本框显示 5 个预测结果；以日期为横轴，收盘价为纵轴，用红线绘制预测结果数据点的折线图。

4.4.3 预测结果比较分析

预测结果分析设计如下：

- 数据准备：按 4.4.1 中的方法获取数据，该数据要包含要预测的 5 个交易日的实际数据，否则，进行比较时系统会提示用于比较的数据不足。
- 比较分析：预测完成后，主界面上的“Compare”按钮会由不可用状态变为可用状态。点击“Compare”按钮，主界面上会显示比较分析结果：以静态文本框显示预测数据的残差、相对误差及均方误差；在同一图表中，以日期为横轴，收盘价为纵轴，用红线绘制预测数据点折线图，用蓝线绘制实际数据点折线图。

5 系统实现

5.1 系统功能展示

- 系统主界面如图 5.1:

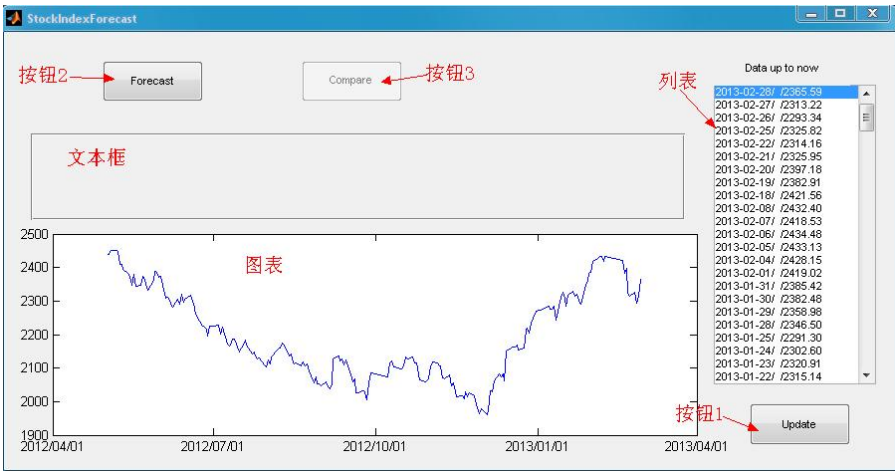


图 5.1 系统主界面

主界面中各部件描述如表 5.1:

表 5.1 主界面功能描述表

部件名称	描述
按钮 1	点击后，根据存储文件，更新图表、列表中数据；
按钮 2	点击后，依据列表中所选数据位置，对未来数据进行预测；
按钮 3	预测完成后点击，将预测数据与实际数据做比较，通过图表、列表显示；
图表	横轴为日期，纵轴为收盘价，显示实际数据与预测数据；
列表	显示实际数据，并允许用户选中一行；
文本框	显示预测数据和预测误差相关数据；

- 收盘价预测功能如下:

在数据列表中，选中一行数据，作为预测前最后一个交易日的数据，点击“Forecast”按钮，系统将做未来收盘价预测。图 5.2 展示了收盘价预测情况:



图 5.2 预测完成后的界面

- 预测结果比较分析功能如下：

预测完成后，“Compare”按钮可用；点击“Compare”按钮，系统将预测结果与实际数据做比较。误差相关数据展示如图 5.3：



图 5.3 比较完成后的界面

5.2 预测结果与实际结果的对比分析

为了验证系统的预测能力，现进行仿真实验，对 4 个不同历史时期的上证指数收盘价进行预测，并与实际数据进行比较。

- 时期 1：2012 年 11 月 2 日——2012 年 11 月 8 日

实验结果分析如表 5.2 和图 5.4，可以看出，未来 5 天数据的预测相对误差绝对值都在 1%之内，预测数据整体走势与实际数据走势也基本一致，预测已有相当高的准确度。

表 5.2 预测 1 误差数据表

日期	实际值	预测值	残差	相对误差
2012-11-02	2117.05	2131.88	-14.83	0.70%
2012-11-05	2114.03	2121.00	-6.97	0.33%
2012-11-06	2106.00	2092.06	13.94	-0.66%
2012-11-07	2105.73	2108.42	-2.69	0.13%
2012-11-08	2071.51	2081.61	-10.10	0.49%

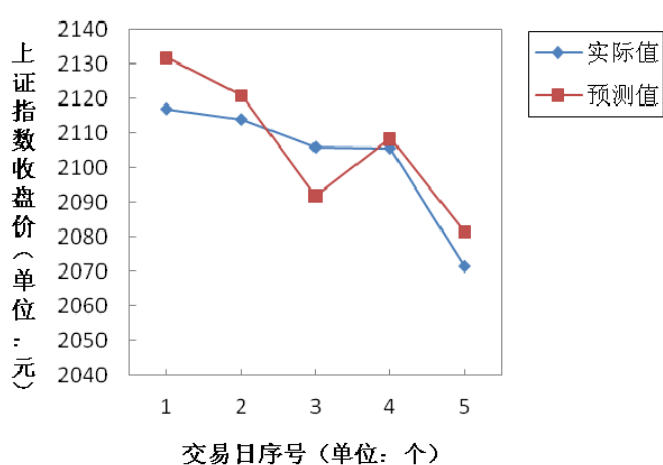


图 5.4 预测 1 数据比较折线图

- 时期 2:2012 年 12 月 3 日——2012 年 12 月 7 日

实验结果分析如表 5.3 和图 5.5，从结果看出，除第 5 天的数据外，其它数据的预测值与实际值已比较接近，它们的走势也大致一致。

图 5.3 预测 2 误差数据表

日期	实际值	预测值	残差	相对误差
2012-12-03	1959.77	1980.60	-20.83	1.06%
2012-12-04	1975.14	1996.16	-21.02	1.06%
2012-12-05	2031.91	2030.61	1.30	-0.06%
2012-12-06	2029.24	2043.06	-13.82	0.68%
2012-12-07	2061.79	1989.30	72.49	-3.52%

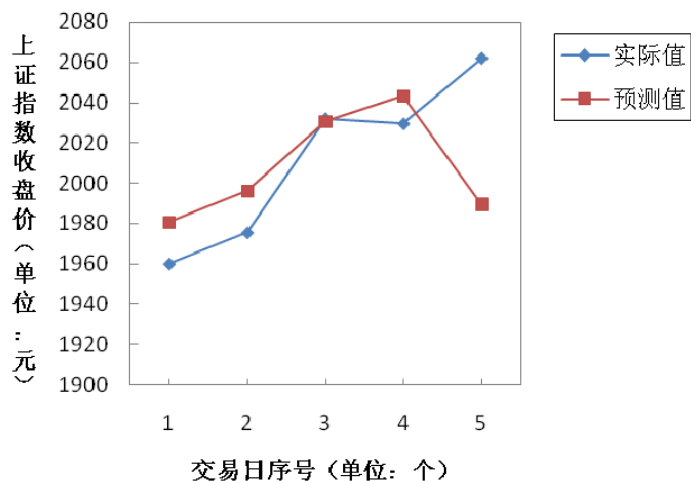


图 5.5 预测 2 数据比较折线图

- 时期 3:2013 年 1 月 4 日——2013 年 1 月 10 日

实验结果分析如表 5.4 和图 5.6, 同样, 前 4 天数据的预测结果较满意。可以看出, 实际值比预测值更平滑, 预测值震荡较大, 故可能网络训练出现了问题。

表 5.4 预测 3 误差数据表

日期	实际值	预测值	残差	相对误差
2013-01-04	2276.99	2276.64	0.35	-0.02%
2013-01-07	2285.36	2281.20	4.16	-0.18%
2013-01-08	2276.07	2291.10	-15.03	0.66%
2013-01-09	2275.34	2291.79	-16.45	0.72%
2013-01-10	2283.66	2214.62	69.04	-3.02%

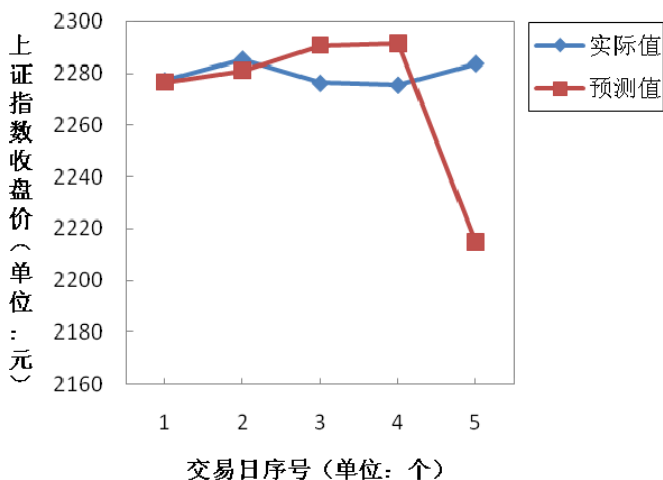


图 5.6 预测 3 数据比较折线图

- 时期 4:2013 年 2 月 1 日——2013 年 2 月 7 日

实验结果分析如表 5.5 和图 5.7，从实验结果看出，预测相对准确，预测数据的整体趋势与实际相符合。

表 5.5 预测 4 误差数据表

日期	实际值	预测值	残差	相对误差
2013-02-01	2419.02	2444.86	-25.84	1.07%
2013-02-04	2428.15	2463.72	-35.57	1.46%
2013-02-05	2433.13	2438.59	-5.46	0.22%
2013-02-06	2434.48	2386.68	47.80	-1.96%
2013-02-07	2418.53	2338.32	80.21	-3.32%

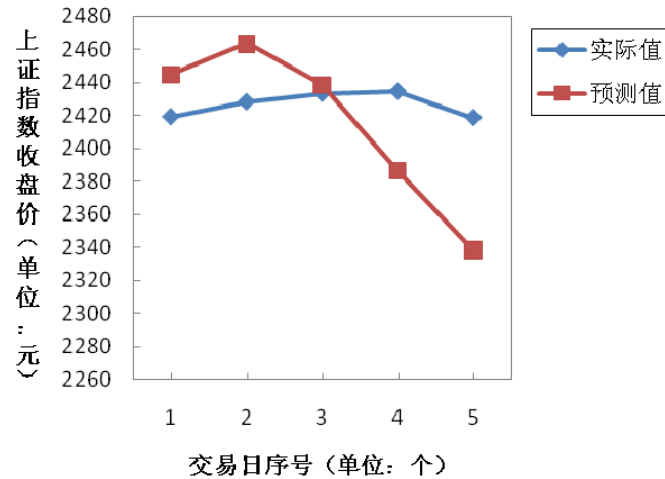


图 5.7 预测 4 数据比较折线图

综上所述，系统的总体预测结果还是比较准确的，所测的收盘价走势与实际走势也大致一致；但不排除对个别数据，有较大误差，其相对误差在 5%之内，仍处于可以接受的范围内。

6 总结

本文通过实验确立了一个较优的预测模型，即总体上用迭代方式、个别点处用网络集成的方式预测。主要完成了以下工作：

- 对原始数据进行了合理的归一化处理，并根据时间序列生成了网络训练样本。
- 通过实验确立了较为合理的神经网络结构，并通过简单的神经网络集成，进行了优化预测结果的尝试。
- 通过预测结果与实际结果的比较分析，一定程度上论证了神经网络用于股票指数预测的可行性与准确性。

但限于时间和作者知识水平等因素，预测模型存在一些问题：

- 设置神经网络的输入结点个数为 10 为参考文献得出的，该数据是否适用于所有时期的收盘价预测还有待研究。
- 在决定未来收盘价的因素中，该模型仅考虑了历史收盘价、成交量和最高、低价等少数几个易获得的量，这样就很大地限制了预测的准确性。
- 对于网络集成方式的预测，3 个个体网络的输入、输出较为相似，网络间的相关性大，导致集成后的网络误差不会大幅减小；此外，简单算术平均的网络输出结合方式也不能使集成效果达到最优。
- 本文仅对该模型预测上证指数收盘价的能力进行了实验验证，而该模型未必适用于其他股票指数，其应用范围就有了限制。

参考文献

- [1]史忠植. 神经网络[M]. 北京: 高等教育出版社, 2009.
- [2]施彦, 韩力群, 廉小亲. 神经网络设计方法与实例分析[M]. 北京: 北京邮电大学出版社, 2009.
- [3]董振海. 精通 MATLAB 7 编程与数据库应用[M]. 北京: 电子工业出版社, 2007.
- [4]周志华, 陈世福. 神经网络集成[J]. 计算机学报, 2002, 25(01): 1-8.
- [5]吴建鑫, 周志华, 沈学华, 陈兆乾. 一种选择性神经网络集成构造方法[J]. 计算机研究与发展, 2000, 37(09): 1039-1044.
- [6]程跃, 车永才, 魏毅. 基于 MATLAB 的改进 BP 在煤炭产量预测中的应用[J]. 江西煤炭科技, 2006, (3): 79-81.
- [7]路志强, 李书全, 赵良英, 等. 改进 BP 网络模型在年用水量预测中的应用[J]. 水科学与工程, 2005, (1): 18-20.
- [8]田云丽, 周利华. 基于 BP 神经网络的煤与瓦斯突出预测方法的研究[J]. 系统工程理论与实践, 2005, (12): 102-106.
- [9]施彦, 黄聪明, 侯朝桢. 选择性神经网络二次集成方法在定量构效关系建模中的应用研究[J]. 计算机与应用化学, 2005, (2): 153-156.
- [10]Adrian E D. Electrical Study of Neuron[M]. Philadelphia: Univ. of Pennsylvania Press, 1932.
- [11]Adrian E D. Physical Background of Perception[M]. Oxford: Oxford University Press, 1947.
- [12]Anderson J A. Introduction to Neural Networks[M]. Cambridge, MA: MIT Press, 1995.
- [13]Denker J S. Neural Networks for Computing[C]. New York: American Institute of Physics, 1986:17-22.
- [14]Michael A.Arbib. The Handbook of Brain Theory Neural Networks[M]. Massachusetts: The MIT Press, 2003.

A Stock Index Forecasting System Based on Neural Networks

Abstract: People have explored the laws of the stock market in any ways since it was born. And they have established many models to forecast the stock index. As a kind of new intelligence technology, Artificial Neural Networks showed its great advantages in this kind of forecasting. But the theory is still in developing period, there's no mature theory of how to design the network structure in practical use. Many experiments must be done.

This paper is aimed to study short period forecasting of Shanghai Composite Index. At first, several forecasting models based on BP Neural Networks are given and I compare their effects by experiments. And then, I develop a stock index forecasting system based on the best model using the Neural Network toolbox and GUI provided by MATLAB. At last, I make several closing prices predictions using this system and compare the results with actual data.

Key Words: BP Neural Networks integration of neural networks stock market index
closing prices time sequences

河北师范大学本科生毕业论文（设计）评议书

姓 名	刘东	学 院	软件学院	专 业	软件工程	年 级（班）	2009 软件 5
论 文 题 目	基于神经网络的股指预测系统					完成时间	4 月 30 日
论 文 内 容 摘 要	<p>本文主要研究了利用神经网络预测股票指数的方法，提出了可行设计方案，并设计系统对可行方案进行封装。</p> <p>本文首先依据股票指数预测的背景，确定了要预测的股票指数类型和基本预测方法；而后综述了 BP 神经网络的基本理论和相关算法。针对股指预测问题，本文提出了可行网络结构和数据处理方法，并做了简单的网络集成的尝试；而后通过实验，选出较优方案，并确定了较为合理的网络参数。最后，提出了图形化界面系统的设计方案，将较优预测方案封装，并利用设计的系统进行了预测，并与实际数据进行了比较分析。</p>						
指 导 教 师 评 语	年 月 日						
指 导 教 师		职 称		初评成绩			

答辩小组	姓名		职称	教研室
	组长			
	成员			
答辩记录：				
记录人签字： 年 月 日				
答辩小组意见：				
组长签字： 年 月 日				
学院意见：				
评定成绩： _____ 签章				
年 月 日				