中国科学技术大学

硕士学位论文

基于深度学习之股指期货交易

作者姓名：杨杰群

学科专业：概率论与数理统计

导师姓名：庄玮玮 副教授

完成时间：二〇一五年四月

University of Science and Technology of China

A dissertation for master's degree

Stock Index Futures Trading Based on Deep Learning

Author :	Yang Jiequn
Speciality :	Statistics
Supervisor :	Asso Prof. Zhuang Weiwei
Finished Time :	April, 2015

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：杨志强　　签字日期：2015.6.2

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

☑公开　☐保密 _____ 年

作者签名：杨志强　　导师签名：[illegible]

签字日期：2015.6.2　　签字日期：2015.6.2

摘 要

自 20 世纪 80 年代初以来, 人工神经网络技术在全球范围迅速发展传播, 人工神经网络是在模拟人类大脑的构造与思维的基础上而建立起的数学模型, 能够处理类似图像识别、语音识别、自然语言等诸多问题, 引起了国内外学者的高度广泛关注.2006 年 Hinton 等人基于人工神经网络提出了深度学习的概念, 深度学习是蕴含了很多个隐含层的深度神经网络, 具备更优异的特征学习的本领, 能够更本质的抽象和表达数据, 通过逐层的数据初始化来优化数据模型训练过程, 进而提升模型预测分类的准确性. 近两年利用深度学习的方法来处理金融高频数据掀起了一股方兴未艾的研究及应用浪潮, 但是如何针对数据特征来选择合适的初始化模型是研究的关键问题, 本文分别利用自编码器和受限波尔兹曼机理论结合人工神经网络建立了一个研究股指期货的模型并作出了比较分析, 最后根据交易策略搭建了一个交易系统. 模型可以大致分为四个模块: 数据预处理模块、神经网络初始化模块、深度学习模块、交易策略模块.

关键词: 人工神经网络, 深度学习, 自编码器, 受限波尔兹曼机

ABSTRACT

Since the early 1980*s*,the artificial neural network technology has been developing rapidly around the whole world,it was the mathematical model that is established on the basis of the simulation of the human brain structure and thinking.It also can solve many problems such as image recognition,speech recognition,natural language processing,which have caused the high attention of scholars in the domestic and abroad. 2006 Hinton and others proposed the concept of deep learning on the base of artificial neural network.Deep learning is the neural network of the depth that contains a lot of hidden layer.It has more excellent ability about characteristics of learning and can abstract and expression data more essential.Deep learning optimize the training process of the data model by initializing data one by one,and then improve the accuracy of the model prediction classification. In the past two years,the use of deep learning methods to deal with the high-frequency financial data sets off a wave of the rise of research and application.However the point is how to choose the appropriate initialization model according to the data features. It established a model of stock index futures and made a comparative analysis by using the auto-encoder machine and Restricted Boltzmann machine theory that combined the artificial neural network in the paper. Finally it build a trading system according to the trading strategy in the paper.The model can be roughly divided into four parts:data preprocessing module,neural network initialization module,deep learning module,the trading strategy module.

Keywords: The Artificial Neural Network, Deep Learning, Auto-Encoder Machine, Restricted Boltzmann Machine

目 录

主要符号对照表

RBM	受限玻尔兹曼机 (Restricted Boltzmann Machine)
AEM	自编码器 (Auto-Encoder Machine)
ANNs	人工神经元网络 (Artificial Neural Networks)
CTR	广告点击率 (Click Through Rate)

第一章 绪论

1.1 量化投资

时至今日, 随着计算机科学和技术的蓬勃发展, “云计算”, “大数据” 这些热门词汇越来越引起学术界和工业界的关注, 在金融领域, 如何从庞大的交易数据中挖掘出信息, 产生交易策略是时下的热点研究问题. 量化投资策略就是利用数量化方法, 对金融市场进行分析、判断和交易的策略、算法的总称.

量化投资起码可以追溯到 1971 年美国富国银行的信托投资平台创建的定量投资模型. 历经 40 年的发展, 量化投资的规模逐渐扩大, 如今在美国至少有 60% 的交易是通过量化策略由计算机自动进行的, 在华尔街, 有很多公司在从事量化投资, 詹姆斯. 西蒙斯 (*James.Simmons*) 的文艺复兴科技公司是当中的佼佼者, 该团队是由顶尖的数学家、统计学家、物理学家等构成, 它发起的基金自 1988 年后, 平均年回报率高达 20% 左右, 即使在 1997 年和 2008 年两次金融危机时期依旧保持较高的回报率, 西蒙斯本人也被赞誉为 “最赚钱基金经理” (Yu,2013).

经济的全球化促使了量化投资观念也在中国兴起, 由于中国资本市场的种种特殊性, 我国的量化投资起步较晚. 2010 年中国融资融券和沪深 300 股指期货的出现, 带来了两种做空机制, 打破只能做多的束缚.2015 年 3 月 12 日 IF1503 单日交易量达 132 万手, 成交额超过 1.4 亿元人民币, 股指期货上市以来流动性好, 其交易活跃, 在这种情况下, 量化投资开始在国内流行起来, 大量的机构和个人从事期货交易, 量化产品逐渐在市场上立足, 而且呈现壮大之势. 新发行的嘉实量化阿尔法基金就是量化投资产品, 量化交易具有很高的研究价值和应用价值. 量化投资视角广阔, 严格执行数量化投资模型, 更具理性, 其模型是根据历史数据来建立的, 因此要依赖历史数据.

1.2 股指期货

沪深 300 股指期货于 2014 年 4 月 16 日在中金所挂牌正式交易, 并且以超市场预期的表现成功迅速在证券市场站稳, 成为证券市场上一个重要的期货品种. 股指期货是我国首个场内标准化的金融衍生产品, 它的平稳上市满足了国家证券监管部门 “高标准, 稳起步” 的设计要求. 从市场来看 $IF1503$ 交易量每天达到十几万张, 交易金额上千亿, 是一个成熟的交易活跃的期货品种. 随着国家监管和实施政策的进一步放开, 机构投资者大规模参与股指期货交易的步伐也逐渐加快. 股指期货的价格影响因素跟很多因素有关, 与宏观经济、股票市场、经济预期、金融货币政策等息息相关. 由于证券市场的逐利性, 采取各类方式对期货建模研究从未停止 (Zhang,2011).

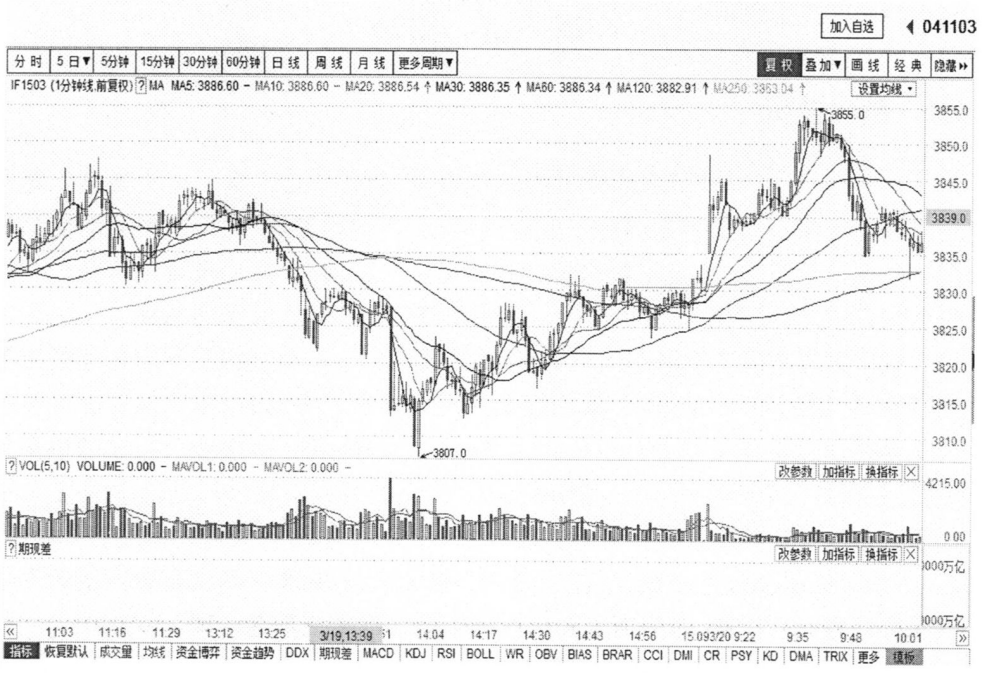


图 1.1: IF1503 日内 60 分钟 K 线图

1.3　人工神经网络与深度学习

1.3.1　神经元结构

人的大脑构成十分复杂, 神经元的数量级在亿, 神经元通过合理的结构组成网络, 发挥功能. 人能够以非常高的速度理解感觉器官传达的信息, 人脑具有强大的识别能力, 高稳定性, 卓越的联想记忆能力以及强大的思维分析能力等. 如果能制作出类似于人的大脑那样具有强大功能的机器, 那将是一件十分伟大的事情. 随着科学技术的发展, 尤其是计算机技术的高速发展, 科学家们通过模拟人脑神经网络的原理和结构创造出了人工神经网络, 用来解决科学领域和工业领域日趋复杂的难题, 人工神经网络 (Artificial Neural Networks,ANNs) 需要研究的实质内容相当的广泛, 包括生物原型、建立模型、算法、应用等. 在这些理论模型基础上构建的神经网络 (计算机模拟), 相比传统方法, 具备并行计算、容错、可硬件实现和自主学习等优点.

人工神经元作为人工神经网络的基本处理单元如图1.4, 是通过模仿生物神经元而发明, 类似一个多输入单输出非线性功能器件, 这里, $X = [X_1, X_2 \cdots X_n]^T$ 表示其他神经元的输出作为该神经元的输入;$W = [W_1, W_2 \cdots W_n]^T$ 表示其他神经元与该神经元连接的强弱重要程度, 即该神经元输入对应的权值; 输入信号的加权和 $\sum x_i w_i \geq \theta$ 时, 神经元被激活; θ 为阈值, θ 一般并不是一个固定常数, 它是根据模型拟合数据而调整的; 当加权和大于 θ 时, 神经元被激活.$f(\cdot)$ 表示神经元的输出函数, 即激活函数, 又称激励函数;O 为神经元输出, 神

合约标的	沪深 300 指数
合约乘数	每点 300 元
合约价值	股指期货指数点乘以合约乘数
报价单位	指点数
最小变动单位	0.2 数
合约月份	2015 年 3 月合约
交易时间	上午 9:15 - 11:30，下午 13:00 - 15:15
最后交易日	上午 9:15 - 11:30，下午 13:00 - 15:00
最大波动限制	上一个交易日结算价的正负 10%
交易保证金	3 月合约的 12%
交割方式	现金交割
最后交易日	2015 年 3 月 20 日
交割日期	同最后交易日
手续费	交易手续费暂定为成交金额的万分之零点五，交割手续费标准为交割金额的万分之一。
交易代码	IF1503

图 1.2: IF1503 股指期货合约

经元输出函数值可以表示为 $y = f(\sum_{i=1}^{i=n} x_i w_i - \theta) = f(W^T X - \theta)$，它模拟了生物神经元的基本功能. 通常, 我们假定神经元有 $n-1$ 个突触结构, 其实际输入变量为 $X_1, X_2, \cdots, X_{n-1}$. 那么可设 $X_n = -1$, $W_n = \theta$, 这样阈值就直接加入了 $f(\sum_{i=1}^{i=n-1} x_i w_i) = W^T X$.

常用的激活函数有:

(1) 线性函数 (Purelin), 函数表达式如下:

$$y = f(x) = x. \tag{1.1}$$

(2)Sigmoid 函数, 也称为 S 形函数. 该函数严格单调递增且连续可微, 其函数表达式如下:

$$y = f(x) = \frac{1}{1 + e^{-\lambda x}}. \tag{1.2}$$

本文中采用的是此函数作为激活函数, 其基本图像如图 1.6:

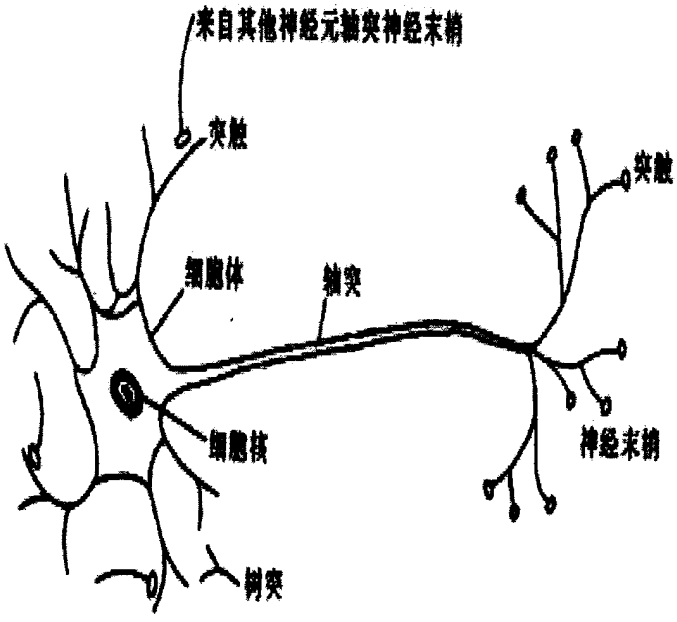


图 1.3: 生物神经元

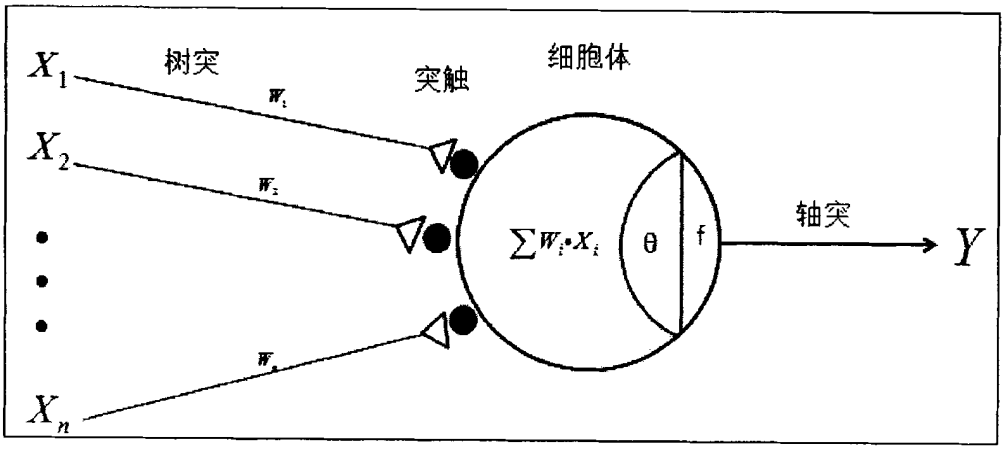
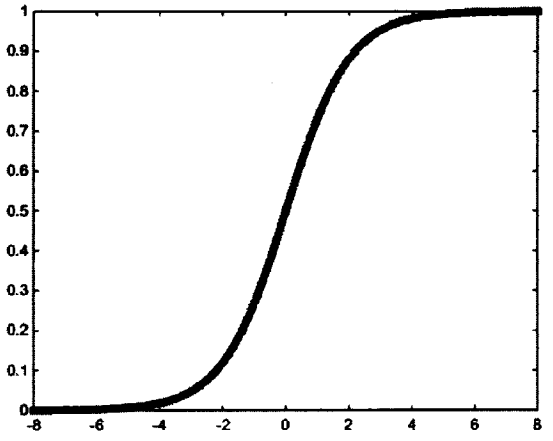


图 1.4: 人工神经元图像

图 1.5: Sigmoid 函数图像



(3) 硬限值函数 (Hardlim), 其函数表达式如下:

$$y = \begin{cases} 0 & x < 0 \\ +1 & x \geq 0 \end{cases}. \tag{1.3}$$

人工神经元并没有包括生物神经元的延迟效应以及不应期反应, 人工神经元的输出仅仅对应当前的输入, 如果想要达到延时效果可以通过附加延时单元来实现. 总结发现, 人工神经元具有如下特点:

(1) 神经元是一个多输入且单输出 (类似于函数) 的功能元件.

(2) 它具有非线性的输入和输出.

(3) 它是可以调节的, 通过权值 W 和阈值 θ 的变化来实现调节.

(4) 神经元的输出受到多个输入值的影响, 是输入节点的综合作用.

(5) 输入可以分为正值 (兴奋型) 和负值 (抑制型) 两种.

唯有过亿个生物神经元连接形成生物神经网络, 才能够对外部感知 (神经末梢) 的信息进行处理加工. 类似地, 人工神经网络也是由大量的基本神经元按一定规则组合而成, 并让网络中的每个神经元的权值 W 和阈值 θ 按一定规则调整, 满足设计神经网络的功能要求.

1.3.2 人工神经网络发展过程

人工神经网络最早在 1943 年被提出, 来自美国芝加哥大学的心理学家 $W.S.McCulloch$ 与数理逻辑学家 $W.A.Pitts$ 最早创建了一个神经网络的数学模型,($McCullch-Pitts$, $M-P$ 模型), 它标志着人们对于神经网络研究的开端, 对后续相关研究发挥着巨大的启发作用. 2006 年. 机器学习的泰斗, 来自多伦多大学的教授 $Hinton$, 与他的学生 $Salakhutdinov$ 在《科学》杂志上发表论

文:“深度信念网络的一种快速学习算法”,解决了包含很多隐层的神经网络的问题,使数据训练更有效,并提出了新的概念——“深度学习”.深度学习的实质是通过构建具备很多隐层的人工神经网络模型和训练大批数据逐层来学习对输出层有帮助的特征,从而使分类和预测更准确.

百度公司自 2012 年开始改变原有以逻辑回归作为预测模型对广告点击率 CTR 进行预估,采用深度神经网络模型,大幅提高了模型学习和抽象特征的能力,使得 CTR 的评估更有效.2013 年 1 月百度 CEO 李彦宏提出建立深度学习研究院,聘请了许多著名的科学家,将深度学习视为核心技术创新方向.同年,科大讯飞公司也将深度学习视为未来研发的重点项目.2012 年 6 月,《纽约时报》报道了 *Google* 公司最为神秘的一个部门 *GoogleX* 的一个“谷歌大脑”项目成果,吸引了很多人的眼球.这个项目是由斯坦福大学著名的机器学习方面的教授吴恩达 (Andrew Ng) 与在大规模计算机系统领域的世界顶尖级大师 Jeff Dlianhe 联合主导,利用包含 16000 个 CPU Core 的强大并行计算平台来为一种名为“深层神经网络”的机器学习模型进行模拟,在语音识别与图像识别等方面获得了巨大的突破.在公司的一次最新成果公开展示中,谷歌大脑从一千万个随机抽取的没有经过任何关联标注处理的视频里,自动将猫脸识别出来.之后,*Hinton* 加盟谷歌公司,创建人工智能研究所 (DNN reasearch),共同研究深度学习.近日,京东公司也破译了“千人千面”,当京东客服刚刚拿起客户打进的电话的时候,根据大数据分析,他就能基本把握客户的情绪状态,性格和心理,可以提前做好准备来应对,使服务更令人满意.这项技术的背后不仅仅是大数据支撑,更重要的是基于深度学习技术的分析模型,如今深度学习理论被广泛运用,成为时下的热点.

1.3.3 深度学习介绍

深度学习相关的研究和应用如此火爆,那究竟什么是深度学习?

机器学习的目标可以是图像识别、语音识别、模式识别、预报天气、股价预测、基因表达、内容推荐等.以图象识别为例,目前我们通过机器学习去解决这些问题的思路一般是如图1.6所示.

开始时,由传感器 (例如 CMOS) 来获得原始数据.之后依次是数据预处理——特征提取——特征选择,再到推理、预测或识别.最后一个部分“推理、预测或识别”,通常所说的机器学习的部分,主要是通过数学和统计方法来建立学习模型.“特征表达”包括中间的三部分.只有优良的特征表达,才能保证最后算法的准确性,而且特征表达是整个机器学习系统建立过程中主要的计算和测试工作.但实际操作中,一般这一块都是依赖人工完成的,靠人工提取特征.而深度学习能够自主的提取特征,节省了人工提取的过程,这就是深度学习的主要作用.

深度学习是模拟大脑皮层的胡贝尔 -魏塞尔模型,采用一层层“抽象化”的方式来对数据或者信号进行表达.类似于大脑皮层对图像的分辨,深度学习模型首先从原始信号 (与人脸识别模型中的像素相似) 中从低层特征隔离 (类似人脸

图 1.6: 机器学习的一般流程图

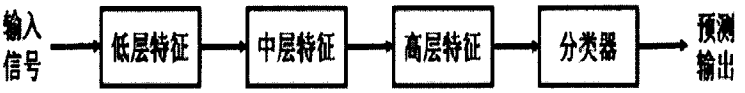


图 1.7: 深度学习层次结构

识别系统中物体的边), 再从低层特征中获得高一层的特征 (近似于人脸识别系统中由边线构成的轮廓), 然后获得更高一层的表达 (类似人脸识别中的人脸), 最后在高层特征上建立起分类器, 获得模型的预测输出, 如图1.7.

感知机的学习是有监督的学习, 权值是通过学习不断调整直到达到想要的效果. 学习步骤如下：

（1）给输入变量权值设置初值, 通常为随机的非零常数.

（2）选定训练集, 即输入和对应的输出变量.

（3）求出感知机对应输入变量得到的输出 y.

（4）根据输出调整权值.

（5）如果某一项感知机输出等于训练集输出, 学习停止; 如果不等于, 返回继续调整权值.

在从低层往高层特征的学习过程中, 一方面是要对数据进行“好”的表达, 另一方面需要尽量不损失信息. 怎么理解一个“好的表达”呢? 我们可以用一个例子来简单说明“表达”的重要性. 在十进制的表示形式下, 我们一般认为乘法比加法复杂, 比如 235144 和 30324 的和, 只要各位对齐, 一位一位地相加并处理好进位就好了, 口算能力一般的人也可以心算出来; 但是如果是做乘法计算的话, 口算是一件不怎么实际的事情. 因为我们平时习惯用十进制. 但是, 假如我们换一种表达方式: 将每一个数字替换成它的素数因子的集合, 例如:

$$235144 \triangleq \{2, 2, 2, 7, 13, 17, 19\}.$$

$$30324 \triangleq \{2, 2, 3, 7, 19, 19\}.$$

这个乘法可以写成:

$$235144 * 30324 \triangleq \{2, 2, 2, 2, 2, 3, 7, 7, 13, 17, 19, 19, 19\}.$$

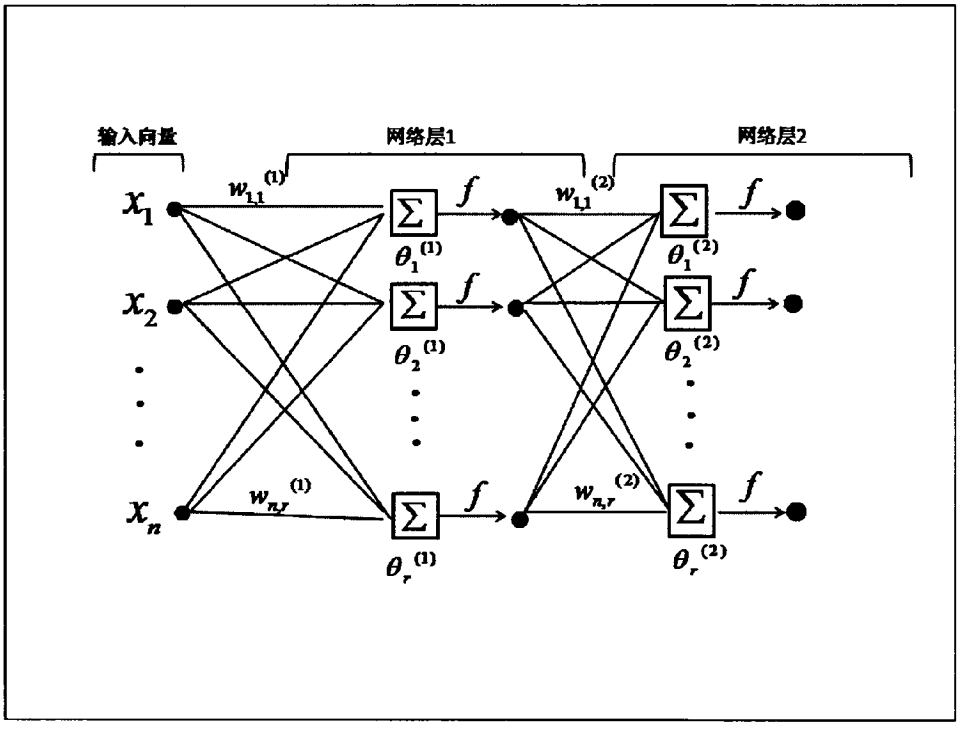


图 1.8: 神经网络结构

深度学习过程中, 先对确定好的特征利用逐层学习贪婪式算法进行提取, 这是一种无监督的学习. 逐层学习时, 有很多种可选择方式获取初始值, 本文进行学习时用的模型是用自编码器和受限玻尔兹曼机, 在后面将简单介绍这两种模型的基本原理. 它们的实现方式都是基于人工神经网络.

1.3.4 MATLAB 工具箱中神经网络结构

图形 1.4 中的符号说明; W 表示连接权重, θ 表示阈值, X 表示输入, f 表示激励函数. 神经网络可以将复杂的问题简单解决. 一个完整的神经网络模型 (如图 1.8 所示) 一般将节点三层, 分别是输入层、输出层和隐层: 输入层的每个节点对应一个预测变量; 输出层的节点对应目标变量 (可有多个); 在输入层和输出层之间的是隐层. 迄今, 隐层层数和其节点数的选取仍然没有相关的理论支持, 只是靠经验来选取, 在本文中隐层层数为 2 层, 每层 100 个节点.

输入层即要输入的自变量 X , 输出层即我们想“预测”的应变量 Y , 隐含层可当做是网络系统的状态. 对于分类神经网络而言, 输出层节点的个数一般是是可能的分类别数.

单个神经元的数学表达式为：

$$y = f(\sum_{i=1}^{D} w_i \cdot x_i), \tag{1.4}$$

其中 D 表示上一层节点数. 不妨假设神经网络第 i 层第 k 个神经元的输入有 n 个，则其输出的数学表达式为：

$$y_i^{(k)} = f(\sum_{j=1}^{n} w_{i-1,j}^{(k)} y_{i-1}^{(k)}), \tag{1.5}$$

其中 $w_{i-1,j}$ 表示相对应的权重，对于函数 f，隐层的激励函数可能会与输出层的输出函数有所不同, 在公式的当中未作区分，实际应用中应做注意. 神经网络模型的学习即用我们已存在的输入输出数据（训练集）, 对参数 w 和 θ 的优化, 使得输出 y_i 尽可能的接近于其真实值 t_i (注意这里研究的是输出层，m 表示的是输出层的节点数), 即要使得如下的预测误差（即损失函数）最小化. 假设一共有 N 个数据集，对于第 i 个数据有：

$$E_i(W) = \frac{1}{m}\sum_{k=1}^{m}(y_k - t_k)^2, \qquad k = 1, 2\cdots, m; i = 1, 2\cdots, N, \tag{1.6}$$

那么整个系统的误差为：

$$E(W) = \sum_{i=1}^{N} E_i(w) = \sum_{i=1}^{N}\frac{1}{m}\sum_{k=1}^{m}(y_{ki} - t_{ki})^2. \tag{1.7}$$

一般使用梯度下降法来获取最优的参数 W ：

$$w_{ij} := w_{ij} - \alpha\frac{\partial}{\partial w_{ij}}E(W). \tag{1.8}$$

这里 w_{ij} 为第 i 层第 j 个节点对应的权重. 其中参数 α 为学习率, 反映的是每一次迭代的步长. 梯度下降法效率随着神经网络训练样本的数据量的增大而降低, 应该采用计算效率比较高的随机梯度下降（Stochastic gradient descent）方法或者是迷你批量方法（Mini-Batch）进行优化. 深度学习中, 神经网络的训练一般采取迷你批量的方式进行优化.

由于 $E(w)$ 是来自训练集所有数据获得的预测均方误差, 所以 $\frac{\partial}{\partial w_{ij}}E(W)$ 的计算也需要用到数据集里所有的数据. 而本文的数据量高达 15 万之多, 在很多情况下, 数据量会更大, 对于这样的大规模数据问题, 梯度下降算法的迭代速度非常缓慢.

$$\begin{aligned} w_{ij} :&= w_{ij} - \alpha\frac{\partial}{\partial w_{ij}}E(W) \\ &= w_{ij} - \alpha\frac{\partial}{\partial w_{ij}}\sum_{I=1}^{N}E_I(W) \\ &= w_{ij} - \sum_{I=1}^{N}\alpha\frac{\partial}{\partial w_{ij}}E_I(W). \end{aligned} \tag{1.9}$$

从上式可以得出, 每一次迭代需要获得的梯度实际上是把单个样本的预测误差 $E(w)$ 对 w_{ij} 的偏导数进行累加得到的结果. 故而等价于分别将 w_{ij} 依此针对每个样本做更新的结果.

随机梯度下降法是在对单个样本做参数更新后, 用更新之后的参数来计算下一个样本的预测误差, 用于下一个样本的参数更新计算. 因此公式 (1.9) 可改写成:

$$w_{ij} := w_{ij} - \sum_{I=1}^{N} \alpha' \frac{\partial}{\partial w_{ij}} E_I(W). \tag{1.10}$$

其实就是当第 I 个样本来更新参数时, $E_I(w)$ 中的参数 w 用上一个样本更新之后的值 $W^{(I-1)}$ 来计算, 该等式表示的是用右边的数值来更新左边权重, 下面的也用此法表示. 而对于单个样本在迭代时, 其待优化的参数都在不断更新, 当迭代计算的样本量很大时, 随机梯度下降法的收敛速度会明显优于梯度下降法. 而且, 随机梯度下降法有更大的可能避免局部最优解.

相对于梯度下降法, 随机梯度下降法的问题在于: 由于样本之间的差异性很大, 每次迭代的参数更新并不是朝着“最优”的方向进行, 而且在固定学习率下, 算法通常不会收敛, 最后会在最优值附近振荡. 深度学习模型训练中常用的迷你批量（Mini-Batch）下降法是在随机梯度下降法的基础上建立起来的一种优化迭代算法. 其算法是每次参数更新不是按照单个样本, 而是按照由若干个样本组成的一个批次来计算参数更新的梯度 (将 $1, 2\cdots, N$ 个样本分成子集 $n_1, n_2\cdots, n_k$).

$$w_{ij} := w_{ij} - \alpha'' \frac{\partial}{\partial w_{ij}} \sum_{n_k \in Banch(N)} E_{n_k}(W^{(n_{k-1})}), \tag{1.11}$$

$W^{(n_{k-1})}$ 表示 n_{k-1} 批次的权重, 因此, 迷你批量方法可以认为折中了普通的梯度下降和随机梯度下两种方法. 一方面它保证了算法迭代的高效性; 另一方面, 在每一批次内各类样本选取较为均衡的情况下, 每次迭代时的参数更新都是近似朝着“最优”的方向进行的. 同时, 与随机梯度下降方法相比, 迷你批量下降法的每一次迭代都采用多个样本进行计算, 充分利用了 Matlab 等科学计算软件在向量化计算上的高效性. 迷你批量方法中, 每个批次内的样本数量一般在 2 到 200 之间.

第二章　相关理论基础

人工神经网络和生物神经网络类似, 需要经过学习才会具有智能, 其调节权值和阈值的过程实际上就是学习过程. 根据模仿人类的学习过程, 神经网络的学习形式主要有三种：有监督学习、无监督学习和强化学习. 有监督的学习, 就是利用提供的训练集进行模仿或分类. 无监督的学习, 只规定学习的方式和规则, 具体的学习内容会随训练库的不同而不同, 系统可以根据学习来发现特征. 在人工神经网络的设计与使用过程中, 从互联结构来看, 包括前馈网络, 反馈型局部互联网络, 有反馈的前馈网络, 全反馈型网络 (如 Hopfidd 网络和玻尔兹曼机), 以及自组织网络等等. 我们主要介绍与本文相关的 BP 神经网络.

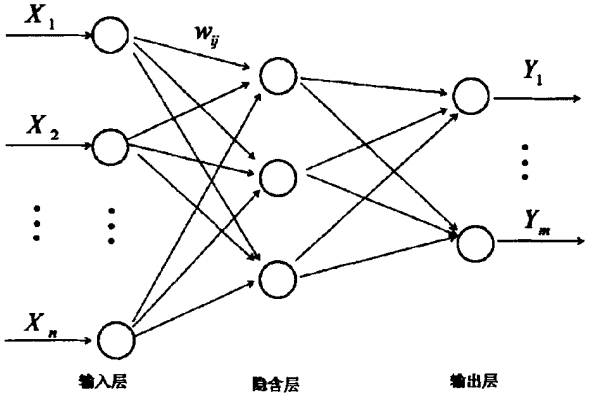
2.1　BP 神经网络及其求解方法

和其它神经元类似, 区别是非线性函数是 BP 神经元的激励函数, 最常用的是 $tansig$ 和 $logsin$ 函数, 线性函数也被部分输出层采用. 其输出为 $y = logsin(WX - \theta)$.

假如多层 BP 网络的输出层采用的是 S 型激励函数, 那么它的输出值就会限制在一个较小的 $(0, 1)$ 区间内, 而线性函数是可以取得任意值的，取值范围较广. 本文采用的是两层 BP 神经网络模型如图 2.1.

BP 网络的训练方式是有监督学习, 当数据训练集赋予网络后, 其神经元的激活值就会由输入层经过各中间层往输出层正向传播, 最终在输出层的各神经元输出对应于输入数据的网络呼应. 继而依照最小化希望输出和实际输出的均方误差的原则 (均方误差最小化), 从输出层经过各中间层、最后回到输入层反向逐层修

图 2.1: 两层 BP 神经网络图



正权值, 经过每一层是将误差分摊给前一层, 前一层的单元获得误差信号并据此修改单元权值. 由于这种修正过程是逆向的, 它也被叫做" 误差逆传播算法". 网络对输入数据的回应的准确程度也会伴随着误差的逆向传播的不断地进行调整, 从而准确率不断提高. BP 网络不仅有位于中间位置的隐含层, 而且还有对应的学习规则可以遵循, 因而可以通过训练, 使得它可以识别非线性模式.

BP 网络的学习过程主要由以下四部分构成:

(1) 输入数据顺传播 (输入从输入层经过中间层往输出层传播计算).

设输入向量为 $X_k = (x_1^k, x_2^k, \cdots, x_n^k)$ (k 为学习模式层数,n 层内节点数), 对应的输出向量为 $Y_k = (y_1^k, y_2^k, \cdots, y_m^k)$ (m 为输出层数), 计算中间层各神经元的激活值:

$$s_j = \sum_{i=1}^{n} W_{ij} \cdot x_i - \theta_j, \quad (j = 1, 2, \cdots, p), \tag{2.1}$$

式中: W_{ij} 表示连接权值; θ_j 表示中间层节点的阈值; 以 S 型函数作为激活函数, 由于它连续可微, 且和生物神经元的信号输出方式接近, 则将上面的激活值代入可得到中间层 j 单元的输出值为

$$b_j^K = f(s_j) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} W_{ij} \cdot x_i + \theta_j)}. \tag{2.2}$$

阈值 θ_j 和权值 w 在学习的进程中一直被修正. 同样, 能够获得输出端的激活值 l_t 和输出值 c_t :

$$c_t = f(l_t), \tag{2.3}$$

$$l_t = \sum_{i=1}^{p} W_{ij} \cdot x_i - \theta_t, \tag{2.4}$$

其中 p 表示有 p 个输出,f 仍为 S 型函数, θ_t 为单元阈值, W_{ij} 为中间层至输出层的权重.

(2) 输出误差逆向传播

在第一步顺序传播过程中, 当实际输出与期望的输出值的误差大于给定的数值时, 网络模型的校正从后面向前面, 从输出层开始计算, 再计算中间层, 最后计算输入层, 根据后一层的计算结果来校正前一层, 校正的误差为:

$$d_t^k = (y_t^k - c_t^k) f'(l_t^k), \tag{2.5}$$

其中 y_t^k 为希望输出. c_t^k 为实际输出; $f'(l_t^k)$ 为输出层函数的导函数.

关于输出层到中间层连接权和输出层阈值的校正量是:

$$\Delta W_{ij} = \alpha \cdot d_t^k \cdot b_j^k. \tag{2.6}$$

$$\Delta\theta_t = \alpha \cdot d_t^k. \tag{2.7}$$

其中调整量关于误差正比, 若误差越大则调整幅度越大; 且调整量正比于输入值, 要想使学习过程活跃就应该加大连接权值的调整幅度; α 为学习系数, 调整量与之成正比, α 的取值一般在 $0.1-0.8$ 之间, 一般地, 学习初期时选取比较较大的学习系数, 随后随着学习过程的进行, 学习系数也逐渐减小.

(3) 循环记忆训练 (计算调整过程重复交替循环进行)

为了使网络的输出误差趋于最小值,BP 网络的每一组训练模式要经历成百上千次甚至更多次的循环记忆训练, 也就是重复 (1)、(2) 这两个过程.

(4) 学习结果判别 (鉴定全局误差是否趋于极小值)

当每次循环训练结束之后, 对于学习结果, 模型都要作判别分析, 其目的是检验输出误差是否已达到可允许的程度. 如果误差已经在允许范围内, 则结束学习过程, 否则继续循环训练. 对于 BP 网络有两个很大的缺陷: 一是收敛速度慢, 二是存在" 局部极小值" 问题. 对于第一个问题本文采用的是" 梯度下降法" 和" 批量 Mini" 法来改进收敛速度; 局部最小值的问题一般是改变通过初始值和训练步长. 本文采用了自编码器和受限玻尔兹曼机来改善初始值, 对于步长也就是本文的学习效率 α, 主要是通过经验来调整.

2.1.1 BP 网络学习算法图

1. 初始化, 给各连接权 W_{ij} 及阈值 θ 赋予区间 $[-1,1]$ 之前的随机值; 2. 选取一对输入输出训练模式 $X_k = [x_1^k, x_2^k, \cdots, x_n^k]$, $Y_k = [y_1^k, y_2^k, \cdots, y_m^k,]$ 提供给网络; 3. 根据激励函数 计算中间层、输出层各单元的输入与输出; 4. 计算输出层、中间层各个单元的校正误差; 5. 调整各层之间的连接权重和阈值、更新数据输入; 6. 判断模式训练是否完成, 若否, 回到第一步, 若是则进行下一步; 7. 更新学习次数, 判断误差是否在允许范围内, 若是, 学习结束, 否则返回第一步. 具体流程图可见 2.2.

2.2 初始化训练值的方法

BP 网络的收敛过程有两个很大的缺陷: 一是收敛速度很慢; 二是存在" 局部极小点" 的问题. 学习过程中偶尔会产生这样一个问题, 当学习反复到一定次数之后, 即使网络的实际输出和期望输出之间还存在很大的误差, 这是由于存在" 局部极小值" 的问题, 无论再怎么学习都无法使得误差走出局部极值范围. 如图2.3所示.

若出事条件是从 A 点位置开始, 这很有可能只达到局部最小点, 这不是我们想要的结果. 试想从 B 点位置开始则可以达到全局最小值, 或者可以改变训练时候的迭代步长, 给每个连接权 w_i 加上一个非常小的随机数来改变步长, 或许就能在收敛过程中避开局部极小点. 我们尝试了" 自编码器" 和受限玻尔兹曼机"

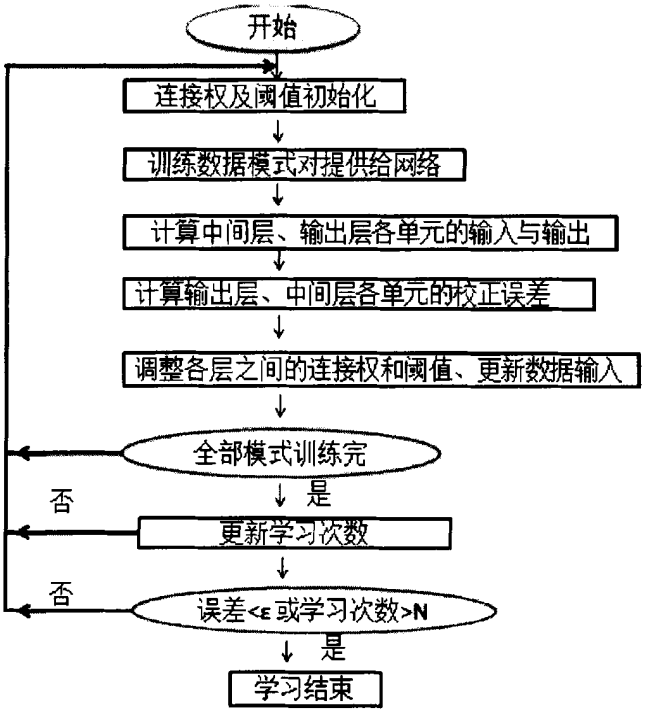


图 2.2: BP 网络学习过程框图

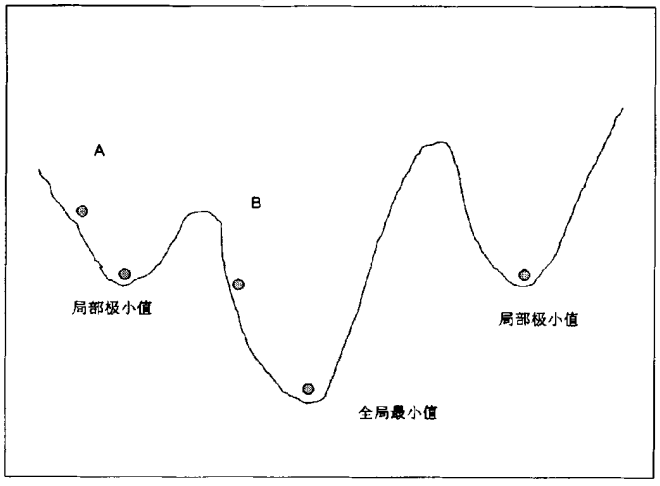


图 2.3: 多个极小值时全局误差范围示意图

的方法来尽可能地使均方误差减小. 隐含层处于 BP 网络的中间位置, 且还有对应的学习规则可循, 因此可以用来训练这种网络, 使得它具备对非线性模式的处理能力. 尤其是它的数字意义明确、步骤分明的学习算法, 使它的应用价值非常广泛.

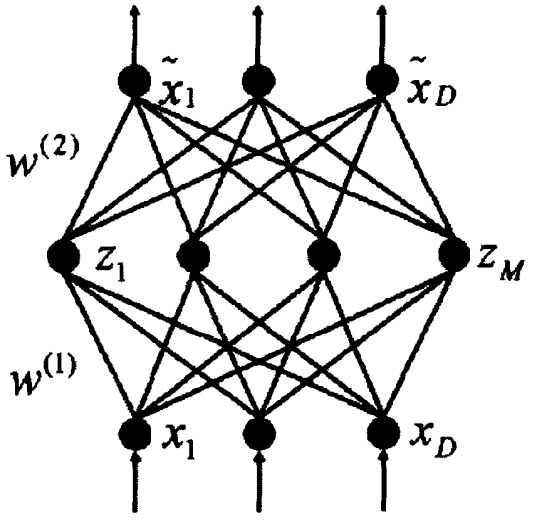


图 2.4: 自编码器示意图

2.2.1 自编码器

自编码器是一种常用的无监督网络学习方法, 用来获取 BP 网络的初始连接权重 $W^{(0)}$ 与阈值 $\theta^{(0)}$ 继而提高模型训练速度, 是一种特殊的神经网络模型. 该网络模型的输出层等于它的输入层 $X = \tilde{X}$, 如图 2.4 所示. 通过自编码器 $W^{(1)}$ "编码" 获得隐层 Z 之后, 可以通过 $W^{(2)}$ 对隐层进行 "解码", 将原始数据重新还原. 模型的优化目标函数为

$$L_{AE} = \frac{1}{N}\sum_{1}^{N}(x_i - \tilde{x}_i)^2. \tag{2.8}$$

也就是要使得模型的预测输出尽可能的等于输入的一种编码方式. 自编码器的深度学习模型思路一般是: 首先对原始的数据, 经过自编码器的学习, 取得编码层 H1; 再对编码层 H1 进行学习, 取得编码层 H2; 类似于这样的逐层的获得网络的各个隐含层. 最后利用学习好的编码器系数和结果作为深层神经网络模型的初始值, 进一步学习深层神经网络.

2.2.2 受限玻尔兹曼机

受限玻尔兹曼机是一种只需要输入数据集就能学习概率分布的生成随机神经网络. 它只能运用于求最优解, 无论从任何初始状态出发, 都非常有可能收敛到全局最小值. 而各个局部最小值不能用来记忆多个输入模式, 但可以用来记忆概率分布.

假设有一个二部图, 一层是输入层 V (即可视层), 一层是隐含层, 每层内的节点之间没有连接. 在已知 V 时, 全部的隐藏节点之间都是条件独立的 (因为这个

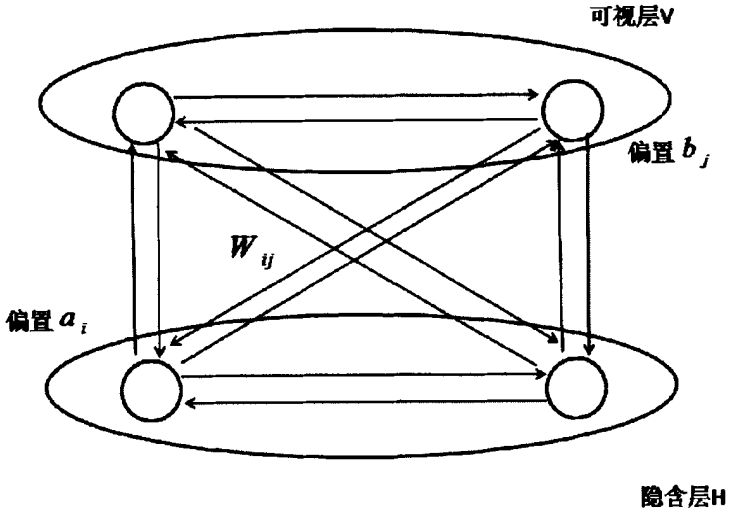


图 2.5: 玻尔兹曼机示意图

模型是二部图), 即

$$p(h|v) = p(h_1|v)p(h_2|v)\cdots p(h_n|v). \tag{2.9}$$

同样, 在知道隐层 H 的情况下, 可视节点也是条件独立的, 又因为全部的 h 和 V 满足玻尔兹曼分布, 所以当输入 V 的时候, 通过 $p(h|v)$ 可得到隐含层 h, 得到 h 之后通过 $p(h|v)$ 又可以重构可是层 V , 通过参数的调整, 使得从隐层计算得到的可视层与原来的可视层有相同的分布. 这样的话, 得到的隐层就是可视层的另外一种表达, 即可视层的特征表示. 作为一种随机网络模型, 受限玻尔兹曼机中包含了能量函数, 能量函数是整个系统状态的测度, 一般情况下, 网络越是有秩序或者说概率分布越是集中, 网络的能量就越小, 反之, 网络越杂乱或者概率分布越不集中, 网络能量越大, 当网络趋于稳定时, 能量函数取得最小值. 能量函数的定义如下：

$$E(v,h|\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}h_iv_j - \sum_{j=1}^{m} b_jv_j - \sum_{i=1}^{n} a_ih_i. \tag{2.10}$$

式中, $\theta = W, b, a$ 是 RBM 模型参数、 m 为可视节点数、 n 为隐层节点数、向量 v 为可视节点的状态、向量 v 为隐藏节点的状态、 b_j 为可视节点 j 的偏置、 a_i 为隐藏节点 i 的偏置、 W_{ij} 为连接权重矩阵. 能量函数反映了, 每个可视节点

与隐节点间都有一个能量连接, 表明可视节点的每组取值与隐层节点的每组取值都有一个能量函数值. 当 θ 已知时, 可以得出 (v,h) 的联合分布:

$$P(v,h|\theta)=\frac{\exp\left((-E(v,h|\theta)\right)}{Z(\theta)}. \tag{2.11}$$

其中 $Z(\theta)=-\sum_{v,h}\exp -E(v,h|\theta)$, $P(v,h|\theta)$ 是玻尔兹曼分布函数, $Z(\theta)$ 是归一化常数, 因此 RBM 模型分配给每个可视节点 v 的概率是:

$$P(v|\theta)=\frac{1}{Z(\theta)}\sum_{h}\exp\left((-E(v,h|\theta)\right). \tag{2.12}$$

给定隐层 (可视层) 节点状态时, 可视层 (隐层) 节点间的状态条件独立:

$$P(v|h)=\prod_{i=1}^{n}P(v_i|h). \tag{2.13}$$

$$P(h|v)=\prod_{j=1}^{m}P(h_j|v). \tag{2.14}$$

当可视节点的状态给定时, 第 j 个隐层节点的激活概率为:

$$P(h_j=1|v,\theta)=f(b_j+\sum_{j}v_iW_{ij}). \tag{2.15}$$

当全部隐层节点状态被求解出来后, 第 i 个可视节点激活的概率为:

$$P(v_i=1|v,\theta)=f(a_i+\sum_{j}h_jW_{ij}). \tag{2.16}$$

其中, $f(\cdot)$ 为 $Sigmoid$ 激活函数. 运用受限玻尔兹曼机模型的目的是求出参数 $\theta=W,b,a$ 值拟合数据训练集, 其理论部分参见文献 [9], 在 MATLAB 中有其工具箱.

第三章　实证分析

3.1　实证分析

基于深度学习的股指期货交易策略, 首先使用深度学习的模型对短期内股指期货的涨跌进行预测, 然后根据预测结果确定股指期货的买卖信号. 我们利用 t 分钟内的开盘价、收盘价、最高价、最低价及成交量等信息对应 $t+1$ 分钟内的收盘价的价格变化构建相应的深度学习模型, 预测的目标是价格变动方向 (即是涨还是跌) 和变动幅度是否大于万分之四 (我们考虑的交易成本是万分之二, 我们认为大于两倍交易成本的交易在实际操作中才有价值).

首先是深度学习预测模型, 一般的, 预测时间间隔越小的话, 深度学习模型的预测本领就越强. 本文研究的是股指期货的以分钟为单位的 1 分钟级高频数据在预测模型输入的选择上, 选择的是短期内上述五个数据, 以提高预测的准确性. 预测模型的输出是有交易价值的未来短期内的涨跌方向考虑与交易相关的涨跌, 本文选取了过去涨跌幅度较大的数据集作为深度学习模型训练的样本.

当深度学习预测模型训练好之后, 我们可以根据预测模型的输出判断未来的涨跌但是这里的预测结果不能直接用于建立交易策略, 原因有两点：1、涨跌只是一个方向性的预测, 并没有说明涨跌的几率具体有多大, 幅度是否足够大；2、高频下股指期货的振幅有限, 持仓时间很短的交易难以带来超额的回报.

基于这两点考虑, 本文中的交易策略有两个特点：第一, 给定阈值, 只有达到或超过阈值的预测得分才会触发买卖信号；第二, 交易中的开仓和平仓都由买卖信号触发.

虽然深度学习模型是一个分类模型, 但同时也可以给出样本属于各个类别的得分值（得分是逻辑函数, 如图1.5所示）对于股指期货价格变化预测模型, 可以获得上涨的预测得分 Score1 和下跌的预测得分 Score2 . 以判断未来股价是否上涨为例, 得分 Score1 越大的样本, 未来上涨的概率越大相应的, 得分 Score2 越大的样本, 未来下跌的概率越大. 我们可以根据得分的大小, 设定阈值, 从而当得分超过阈值时, 触发买卖信号记做多信号触发阈值为 BuyTrigger, 做空信号触发阈值为 SellTrigger, 则买卖信号 Signal 如下：

$$Signal = \begin{cases} 1 & Score1 > BuyTrigger \\ -1 & Score2 > SellTrigger \\ 0 & else \end{cases}. \tag{3.1}$$

我们的交易策略由买卖信号触发每天开盘时都是空仓状态由买入、卖出信号的触发分别建立多仓和空仓, 持仓时间不定. 如果在持有多仓时, 有空头信号触发, 立即平仓并反向建立空仓, 否则 (如继续有多头信号触发时) 不改变持有头寸.

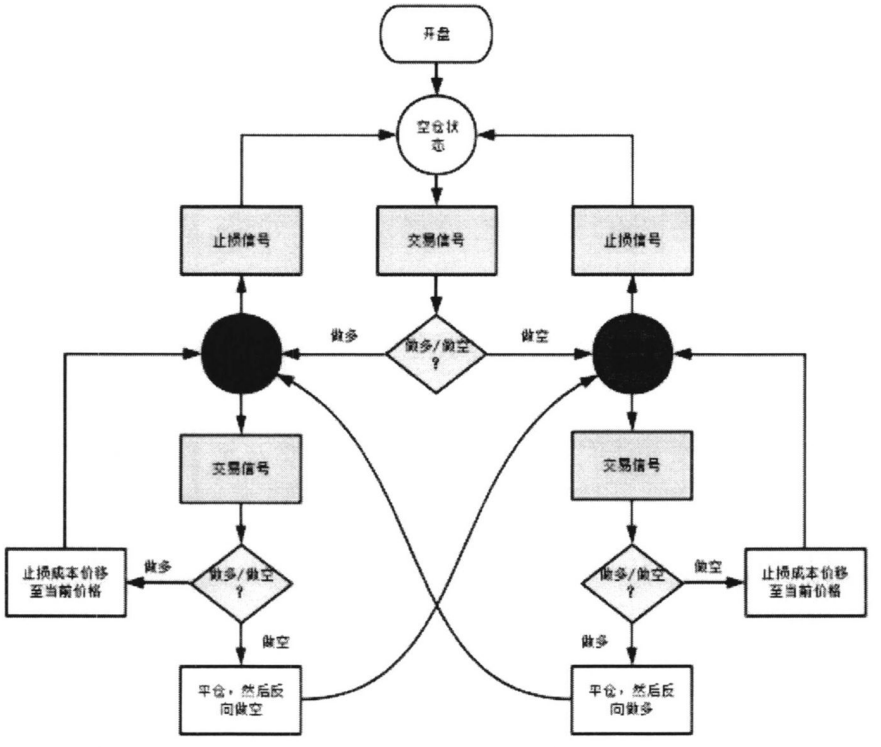


图 3.1: 深度学习股指期货交易策略

如果在持有空仓时, 有多头信号触发, 立即平仓并反向建立多仓, 否则 (如继续有空头信号触发时) 不改变持有头寸. 由于市场噪声和突变的情况, 预测模型对只能够准确预测一部分股价变化, 所以在持仓的时候, 要设置止损策略, 当实时价格突破止损线 r 时, 立即进行止损平仓, 止损价格以最近一次发出交易信号时的股票价格为基准进行计算具体的交易策略如图 3.1 所示.

3.2　数据介绍

本文选取了股指期货 "IF1503" 2014 年 8 月 29 号到 2015 年 3 月 6 号的约 15 万个交易数据, 其中 2014 年 11 月 30 日之前的数据用做模型训练拟合, 之后的数据用来回测. 模型研究变量包含每分钟的开盘价、收盘价、最高价、最低价和成交量五个变量, 利用 MATLAB 软件建立人工神经网络模型, 根据前五分钟的交易数据对下一分钟收盘价的变化进行滚动预测并建立交易策略, 该模型的预测的准确率达到 59%. 模型采用的是 BP 网络模型, 初始化权重和阈值采用的是自编码器和受限玻尔兹曼机模型.

日期	开盘价 (元)	最高价 (元)	最低价 (元)	收盘价 (元)	成交量
2014/8/29 10:00	2371.8	2372	2371.8	2372	9
2014/8/29 10:02	2372.6	2372.6	2372.6	2372.6	1
2014/8/29 10:03	2371.8	2371.8	2371.4	2371.4	2
2014/8/29 10:04	2371	2371	2370.2	2370.2	5
2014/8/29 10:05	2370	2370	2369.6	2369.6	3

3.3 数据预处理

我们将 t 时刻所考察样本 X_t 与其前五个时刻的样本组成输入向量 $\bar{X}_t = [X_t, X_{t-1}, X_{t-2}, X_{t-3}, X_{t-4}]$, (前一个交易日的收盘价是同一个变量, 不用扩展) 在数据的预处理阶段, 首先将股价数据的标准化 (将 t 时刻输入向量 $\bar{X}_t$ 中所有股价数据除以 t 时刻的收盘价格之后, 再取对数 $x_t = ln(x_s)$, 对于极端数据的平滑, 以变量的 99.9% 和 0.1% 为限定, 超过限定的数据用限定内的数值来替代), 归一化 (全部数据都按照公式3.2转化为 $[0,1]$ 之间的数据).

$$x_t^i := \frac{x_t^i - \min x_t^i}{\max x_t^i - \min x_t^i}. \tag{3.2}$$

3.3.1 交易算法

深度学习网络的第一个隐层有 200 个节点, 第二个隐层有 100 个节点, 输出层有 2 个输出节点（输出 Score1 和 Score2 依次表示预测价格是上涨或者下跌的得分）无监督学习的隐层训练迭代次数为 110 次, 有监督学习的迭代次数为 168 次. 在 DELL 笔记本, 主频 2.1GHZ 内存为 2G 的处理器下, 模型的训练时间为 15 分钟模型应用时, 单个样本的预测耗时在 1ms 左右, 如图3.2所示.

对于数据训练时, 由图 3.2 和图 3.3 可以明显地对比出, 使用无监督的自编码器对初始权重和阈值进行设定比受限玻尔兹曼机的效果要好, 具体体现在两方面. 一方面: 自编码器在第 110 次就能使得 MSE 达到最小值, 而受限玻尔兹曼机需要 168 次才能使得使得 MSE 收敛到最小值, 从图中也可以直观地看出来自编码器使得 MSE 下降得较快; 另一方面: 采用自编码器的收敛误差要比使用受限玻尔兹曼机的误差要小. 因此, 在后续的分析中, 采用了自编码器的结果作为初始化设置进行数据分析.

在选取 2014 年 12 月 1 号之后的数据作为回测数据, 可以得到深度学习股价预测结果如下表3.1:

样本以外的数据回测结果如表所示, 其中 Y 代表实际股价变化 Y>0 代表实际股价上涨, 模型以 59% 的概率会给出价格上涨的信号, 建立多头仓位; Y<0 代表实际股价下跌, 模型以 59.4% 的概率给出价格下跌的信号, 建立空头仓位因此模型预测的准确率超过 59%. 如果不考虑交易成本, 完全按照深度学习模型给

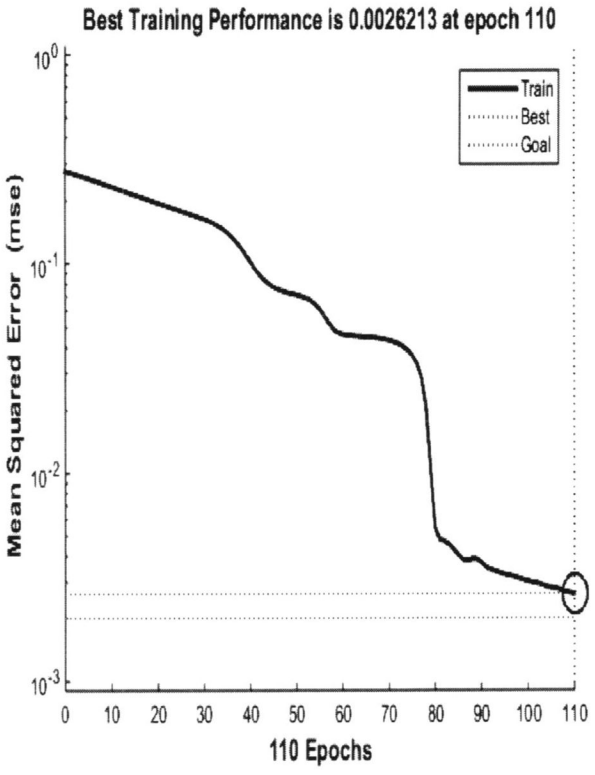


图 3.2: 自编码器深度学习模型训练情况

表 3.1: 预测结果

	Y>0	Y<0
Score1>Score2	3357 59%	2237 41.6%
Score1<Score2	2233 40.5%	3272 59.4%

出的信号进行交易（Score1>Score2 则做多,Score1<Score2 则做空, 下一分钟立即平仓）

事实上, 仅有极少数交易机会中, 股票价格会有大幅波动, 因此我们策略中选取的交易机会 α 在 1% 左右（多头空头机会加起来约 2%）, 对于其他金融期货数据可以选择根据数据选择合适的 α 根据训练样本所有数据的预测得分, 定义做多信号阈值 BuyTrigger 和做空信号阈值 SellTrigger 如下

$$BuyTrigger := P(Score1 > BuyTrigger) = \alpha. \tag{3.3}$$

$$SellTrigger := P(Score2 > SellTrigger) = \alpha. \tag{3.4}$$

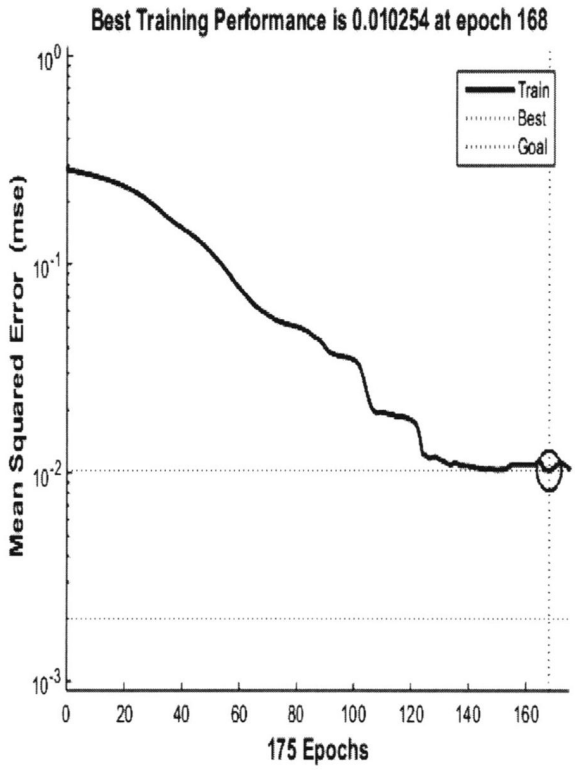


图 3.3: 受限玻尔兹曼机深度模型训练情况

即以 α 的概率, 上涨预测得分 Score1 会大于 BuyTrigger, 触发做多信号；有 α 的机会, 下跌预测得分 Score2 会大于 SellTrigger, 触发做空信号.

在一般情况下, $\alpha = 1\%$, 以 $r = 0.5\%$ 为止损线, 以 2014 年 12 月 1 日为例, 一共发出 9 次做多和做空信号, 如图所示3.4, 红色向上箭头表示做多信号, 绿色向下箭头表示做空信号.

5 次做多信号发出后一分钟, 股指期货价格有 4 次上涨,1 次下跌；5 次做空信号发出后一秒, 股指期货价格有 4 次下跌,1 次上涨. 如表 3.5 所示.

通过预测模型给出的做多做空信号进行交易, 由于多空信号是间隔发出的, 实际执行的交易次数也一共有 10 次, 其中 2 次交易亏损,8 次交易获得正的收益, 如表所示.

3.4　总结

本文利用深度学习的方法对股指期货进行建模分析, 在一分钟级的数据基础上对期货的收盘价进行了预测，其预测准确率约为 59%，考虑到交易费用以及一分钟内出现大的振幅可能会达到止损点等因素，更重要的是市场风格、其他

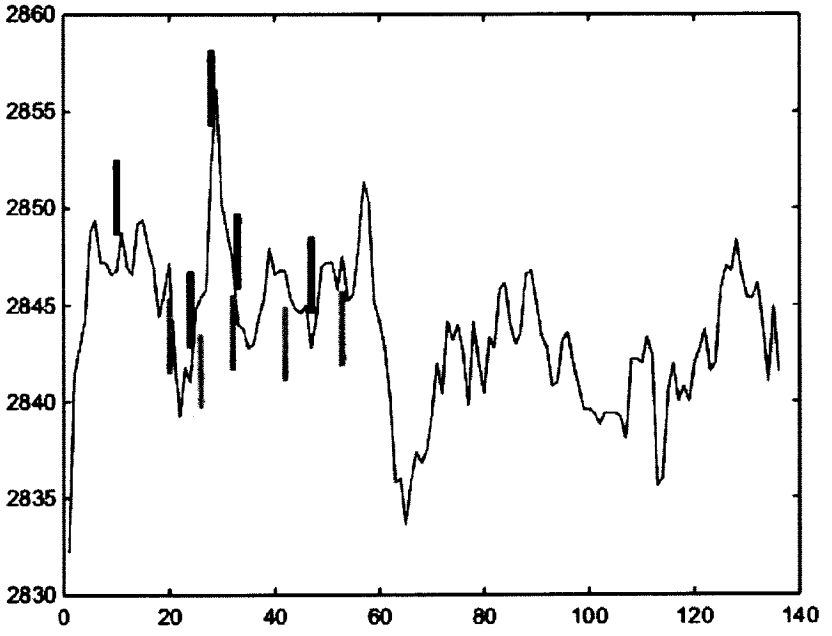


图 3.4: 12 月 1 日根据模型交易情况

交易行为都有可能会对模型产生影响, 模型并不能保证绝对有效，仅作为一种处理股指期货的参考方法.

总的来说, 深度学习功能十分强大, 对于其他的期货品种, 首先要考虑其数据特征是否与本模型的数据特征有类似之处, 交易是否活跃，参于者主要采取的交易方式等信息来判断是否可以适用，可能调整模型参数 (选择合适的迭代次数和学习系数 α) 及网络层数, 对数据做一定的训练. 对于其他的股指期货品种, 例如 $IF1506$ 和 $IF1509$，他们都是基于沪深 300 股指, 具有较强的相关性, 只是在合约到期时期表现会略有不同，本模型是可以适用的.

建仓时间	平仓时间	建仓时股价	一分后股价	平仓时股价	建仓方向	交易盈亏	平仓备注
9:23	9:33	2846.8	2848.8（涨）	2847.2	多头	0.01%	平仓并反向建仓
9:33	9:37	2847.2	2842.2(跌)	2841	空头	0.22%	平仓并反向建仓
9:37	9:39	2841	2844.8（涨）	2845.4	多头	0.23%	平仓并反向建仓
9:39	9:41	2845.4	2845.8（涨）	2852.4	空头	-0.25%	平仓并反向建仓
9:41	9:45	2852.4	2856.2（涨）	2847.4	多头	-0.18%	平仓并反向建仓
9:45	9:46	2847.4	2844（跌）	2844	空头	0.12%	平仓并反向建仓
9:46	9:55	2844	2843.8（跌）	2846.8	多头	0.01%	平仓并反向建仓
9:55	9:59	2846.8	2845.4（跌）	2842.8	空头	0.14%	平仓并反向建仓
9:59	10:06	2842.8	2844.4（涨）	2847.6	多头	0.17%	平仓并反向建仓

图 3.5: 12 月 1 日交易明细

参考文献

[1] 白斌飞. 基于神经网络理论的线性时间序列预测研究【D】 成都. 西南交通大学,2005.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*【M】 Springer 2006.

[3] Hinton G ,Salahutdinov R. *Reducing the dimensionality of data with neutral networks*【J】 Science,2006.313(504),Doi:10.1126/science.1127647.

[4] 冯定. 神经网络专家系统【M】 科学出版社. 北京 2006:50-68.

[5] 蒋艳凰, 赵强利. 机器学习方法【M】. 电子工业出版社, 北京,2009.

[6] 马冬梅. 基于深度学习的图像检索研究【D】 内蒙古. 内蒙古大学,2014.

[7] 乔高秀. 沪深 *300* 股指期货价格发现、定价偏差及波动率研究【D】 成都, 西南财经大学,2012.

[8] 王旭，王宏，王文辉. 人工神经元网络原理与应用【M】. 第二版. 山东大学出版社, 书号: 129830.1516, 山东, 2007.

[9] 余凯、贾磊、陈雨强、徐伟. 深度学习的昨天、今天和明天【J】 计算机科学与发展,2013.50(9): 1799-1804.

[10] 周开利，康耀红. 神经网络模型及其 *MATLAB* 仿真程序设计【M】. 清华大学出版社, 书号: ISBN7-302-10829-3, 北京, 2005.

[11] 张亮. 中国证券市场量化对冲方法及实证分析【D】 合肥, 中国科学技术大学,2011.

附录 A　MATLAB 程序代码

```
%数据预处理
zhangdiefu=diff(data(5:5387,4));
zhangdiefu=zhangdiefu./data(5:5386,4);
xuhao=find(abs(zhangdiefu)>4*10^(-4));
xuhao2=xuhao+4;
input=[data(xuhao2,:),data(xuhao2-1,:),data(xuhao2-2,:),
    data(xuhao2-3,:),data(xuhao2-4,:)];
target=[(zhangdiefu>0).*(1-(zhangdiefu<0)).*(1-(
    zhangdiefu==0)),(zhangdiefu<0).*(1-(zhangdiefu>0))
    .*(1-(zhangdiefu==0)),(zhangdiefu==0).*(1-(zhangdiefu
    >0)).*(1-(zhangdiefu<0))];
biaohao=find(zhangdiefu==0);
biaohao2=biaohao+4;
input=[input;[data(biaohao2,:),data(biaohao2-1,:),data(
    biaohao2-2,:),data(biaohao2-3,:),data(biaohao2-4,:)]];
target=target([xuhao;biaohao],:);
[m,n]=size(input);
Dinput=(input-repmat(min(input),m,1))./repmat(max(input)-
    min(input),m,1);
XX=max(input);
NN=min(input);
%自编码器获得初值
pr1(1:30,1)=0;
pr1(1:30,2)=1;
bpnet1=newff(pr1,[200,30],{'logsig','logsig'},'traingdx
    ','learngdm');
bpnet1.trainParam.epochs=11000;
bpnet1.trainParam.goal=0.002; bpnet1.trainParam.show=10;
bpnet1.trainParam.lr=0.2;
bpnet1=train(bpnet1,Dinput',Dinput');
w1=bpnet1.IW{1};
b1=bpnet1.b{1};
%
bpnet4=newff(pr1,200,{'logsig'},'traingdx','learngdm');
```

```
output1=sim(bpnet4,Dinput');


pr2(1:200,1)=0;
pr2(1:200,2)=1;
bpnet2=newff(pr2,[100,200],{'logsig', 'logsig'}, '
   traingdx', 'learngdm');
bpnet2.trainParam.epochs=11000;
bpnet2.trainParam.goal=0.002;
bpnet2.trainParam.show=10; ?
bpnet2.trainParam.lr=0.3; bpnet2=train(bpnet2,output1,
   output1);

w2=bpnet2.IW{1};
b2=bpnet2.b{1};

%网络训练
pr(1:30,1)=0;
pr(1:30,2)=1;
bpnet=newff(pr,[15,3],{'logsig','logsig'}, 'traingdx', '
   learngdm');
bpnet.IW{1}=w1;
bpnet.b{1}=b1;
bpnet.LW{2,1}=w2;
bpnet.b{2}=b2;
bpnet.trainParam.epochs=11000; 2?
bpnet.trainParam.goal=0.001; bpnet.trainParam.show=10;
bpnet.trainParam.lr=0.3; bpnet=train(bpnet,Dinput',target
   ');
```

交易策略程序

```
cangwei=0;
X=data(5:-1:1);
i=5;
n=length(data);
```

```
yingkui=[];
jilu=[];
while i<=n
X=data(i:-1:i-4);
score= sim(bpnet,X');
if score(1)>score(2)&&score(1)>Buytrigger

jilu=[jilu,i];
if cangwei==0
buyprice=X(1);
cangwei=1;
else if cangwei==-1
yingkui=[yingkui,sellprice-X(1)];
buyprice=X(1);
cangwei=1;
end
end
end
if score(1)<score(2)&&score(2)>Selltrigger

jilu=[jilu,-i];
if cangwei==0
sellprice=X(1);
cangwei=-1;
else if cangwei==1
yingkui=[yingkui,X(1)-buyprice];
sellprice=X(1);
cangwei=-1;
end
end
end
i=i+1;
end
```

致 谢

在中国科技大学完成硕士学业的日子里，我所从事的学习和研究工作，都是在导师以及系里其他老师和同学的指导和帮助下进行的. 在完成论文之际，请容许我对他们表达诚挚的谢意.

首先感谢导师庄玮玮副教授多年的指导和教诲，她严谨的研究态度及忘我的工作精神，认真细致的治学态度及宽广的胸怀，将使我受益终身.

感谢胡治水、崔文泉、苏淳、吴耀华、胡太忠等老师，他们的指导给我研究生阶段的研究工作打下了基础.

感谢科大，感谢一路走过来的兄弟姐妹们，在最宝贵年华里，是你们伴随着我的成长.

最后，感谢我家人一贯的鼓励和支持，你们是我追求学业的坚强后盾.

杨杰群

2015 年 6 月 2 日