**摘 要**

本文首先总结了人工神经网络的研究背景、发展和意义，介绍了人工神经网络的基本理论，为理解BP神经网络打下了基础。然后对BP神经网络进行论述，其中着重介绍了BP网络算法，用数学推导的方式讲解了标准BP算法的思想，将正向传播和误差反传这两个过程分别做了阐述，同时也介绍了传统BP网络存在的缺陷与不足以及针对这些不足所提出的改进算法。最后，利用MATLAB中的神经网络工具箱设计出一个网络，对BP神经网络的实例应用进行了仿真。从仿真结果可以看出，BP神经网络具有强大的自学习能力、完美的逼近能力和自适应能力。

**关键字:** 神经网络;BP神经网络;自学习能力;逼近能力

**Abstract**

In this paper, the background, development and significance of ANN are summarized, and the basic theory of artificial neural network is introduced, which lays the foundation for understanding BP neural network.Then discussed the BP neural network, which focuses on the BP neural network algorithm, explained the mathematical derivation of the standard BP algorithm, the forward propagation and error reverse spread the two process were expounded on. At the same time, it also describes the defects of the traditional BP network and not enough and in the light of these problems, the proposed improved algorithm.At last, a network is designed by using the neural network toolbox of MATLAB, and the application of BP neural network is simulated.From the simulation results, BP neural network has powerful self-learning ability, perfect approximation ability and adaptive ability.

**Keywords:** neural network; BP neural network; Self-learning ability; approximation ability

**目 录**

[1 绪 论 1](#_Toc421460382)

[1.1 课题背景 1](#_Toc421460383)

[1.2 神经网络发展及研究意义 1](#_Toc421460384)

[1.2.1 神经网络发展 1](#_Toc421460385)

[1.2.2 研究意义 3](#_Toc421460386)

[1.3 论文安排 3](#_Toc421460387)

[2 人工神经网络 5](#_Toc421460388)

[2.1 人工神经网络概述 5](#_Toc421460389)

[2.2 人工神经网络基本处理单元 6](#_Toc421460390)

[2.2.1 生物学基础 6](#_Toc421460391)

[2.2.2 人工神经元 7](#_Toc421460392)

[2.3 人工神经网络的基本要素 8](#_Toc421460393)

[2.3.1 神经元激活函数 8](#_Toc421460394)

[2.3.2 神经元之间的连接形式 10](#_Toc421460395)

[2.3.3 神经网络的学习 11](#_Toc421460396)

[2.4 人工神经网络的应用 13](#_Toc421460397)

[3 BP神经网络 14](#_Toc421460398)

[3.1 BP神经网络概述 14](#_Toc421460399)

[3.2 BP网络的基本原理 14](#_Toc421460400)

[3.2.1 BP网络模型 14](#_Toc421460401)

[3.2.2 标准BP网络算法 15](#_Toc421460402)

[3.2.3 BP网络的主要能力 19](#_Toc421460403)

[3.3 BP网络存在的问题 20](#_Toc421460404)

[3.4 BP网络算法的改进 20](#_Toc421460405)

[4 BP神经网络函数的程序设计 22](#_Toc421460406)

[4.1 BP神经网络工具箱中的函数 22](#_Toc421460407)

[4.1.1 BP网络创建函数 22](#_Toc421460408)

[4.1.2 神经元上的传递函数 22](#_Toc421460409)

[4.1.3 BP网络学习函数 23](#_Toc421460410)

[4.1.4 BP网络训练函数 24](#_Toc421460411)

[4.2 BP算法的编程步骤及流程图 25](#_Toc421460412)

[5 BP网络在模式识别中的应用 27](#_Toc421460413)

[5.1 问题的提出 27](#_Toc421460414)

[5.2 数据选择 27](#_Toc421460415)

[5.3 建立网络 28](#_Toc421460416)

[5.4 使用含噪声的数字样本进行训练 28](#_Toc421460417)

[5.5 测量网络容错性 29](#_Toc421460418)

[5.6 对污染数字进行识别 30](#_Toc421460419)

[5.7 仿真分析 32](#_Toc421460420)

[5.7.1 隐含层神经元数目对仿真的影响 32](#_Toc421460421)

[5.7.2 训练函数对仿真的影响 33](#_Toc421460422)

[5.7.3 传递函数对仿真的影响 33](#_Toc421460423)

[结 论 35](#_Toc421460424)

[致 谢 36](#_Toc421460425)

[参考文献 37](#_Toc421460426)

[附 录 38](#_Toc421460427)

[数字识别完整代码 38](#_Toc421460428)

# **1 绪 论**

## **1.1 课题背景**

现代计算机的运行速度是人脑中的神经元运行速度的几百万倍，对于那些有明确特征以及有清楚的推理或运算规则的可编程问题，可以十分高速有效地求解，具有运行速度快、计算精度高、强大的记忆功能、逻辑判断能力以及可以实现自动控制等优势，从而在信息处理和控制决策等方面为人们带来了智能化和自动化。但现有计算机是按照冯·诺依曼原理进行工作的，其结构模式和运行机制并没有跳出传统的逻辑运算规则，因此在许多方面还远不能达到人类大脑的智能水平，传统的人工智能方法对于直觉，如模式、语音识别、非结构化信息处理方面存在相当大的缺陷。

人工神经网络，是20世纪80年代以来在人工智能领域兴起的研究热点。它是由大量具有适应性的节点（或称神经元）按照一定的连接方式组成的广泛并行互连的网络，其从信息处理的角度对人脑神经元网络进行模仿，反映了人脑功能的许多特征，是一个高度复杂的非线性、自适应信息处理系统。人工神经网络具有大规模的并行、分布式存储与处理、自组织、自适应和自学习能力，因而在解决模糊的、随机的、非线性的问题方面具有相当大的优势。BP神经网络是一种最典型的神经网络，其存在非常多的优点，如非线性映射能力、自学习能力、结构简单、可操作性强等，因此在众多领域中得到了广泛的应用。

## **1.2 神经网络发展及研究意义**

### 1.2.1 神经网络发展

神经网络的研究可以追溯到19世纪末期，其发展历史大约可分为三个时期：

(1) 启蒙时期:

1890年，美国心理学家William James（1842—1910）出版了第一部详细论述人脑结构及功能的专著《心理学原理》，对和学习、联想记忆相关的基本原理做出了开创性的研究。半个世纪后，生理学家W.S.McCulloch和数学家W.A.Pitts在1943年发表了一篇神经网络方面的著名文章。在文章中，他们在已知的神经细胞生物学的基础上，从信息处理的角度出发，提出形式神经元的数学模型，称为M-P模型，该模型被认为开创了神经科学理论研究的新时期。随后，心理学家Donald Olding Hebb 在1949年出版了一本名为《行为构成》的书，在该书中他建立了一种连接权训练算法，即Hebb算法。Hebb认为，如果源和目的神经元同时被激活兴奋时，它们之间突触的连接强度将会增强，这便是Hebb训练算法的生理学基础。1958年计算机学家Frank Rosenblatt （1928-1969）提出了一种具有三层网络特性的神经网络结构，成为“感知机”，被认为世界上第一个真正优秀的人工神经网络。1960年，Bernard Widrow和Marcian Hoff提出了自适应神经元，可用于构成自适应系统。

(2) 低潮时期

1969年，人工智能的创始人之一M.Minsky和S.Papert出版了一本评论人工神经网络的书——《感知器》。该书指出，简单的神经网络仅仅可以运用于线性问题的求解，对于非线性问题的解决需要具有隐层的网络，然而，从理论上来看，将感知机模型扩展到多层网络是否具有意义还无法证明。由于Minsky在学术界的地位和影响，其悲观情绪极大地影响了当时的人工神经网络研究，很多领域的专家纷纷放弃了对神经网络课题的研究，开始了神经网络发展史上长达10年的低潮时期。

在之后的十年中，神经网络领域的研究人员数量大幅度减少，但仍然有一些学者致力于神经网络的研究，为日后神经网络研究的复兴和发展奠定了理论基础。1969年，S.Grossberg教授和他的夫人G.A.Carpenter提出了著名的自适应共振理论。1972年，芬兰的T.Kohonen教授提出了自组织映射（SOM）理论，称其神经网络结构为“联想存储器”。同年，美国的神经生理学家和心理学家J.Anderson提出了一个类似的神经网络，称为“交互存储器”。

在10年低潮时期，上述研究成果及其他有意义的工作虽然没有引起当时人们的普遍重视，但是其科学价值是不可磨灭的，在以后的神经网络的发展中起到了重要的作用。

(3) 高潮时期

1982年，美国加州理工学院的[物理学家](http://baike.baidu.com/view/67012.htm)J.J.Hopfield博士提出了Hopfield神经[网格](http://baike.baidu.com/view/10755.htm)模型，对网络引用了物理力学的分析方法，给出了网络稳定性判断。 1984年，他又提出了连续时间Hopfield神经网络模型，为研究[神经计算机](http://baike.baidu.com/view/1215856.htm)的理论做出了开拓性的工作，开创了神经网络用于联想记忆和优化计算的新途径，有力地推动了神经网络的研究。1985年，Rumelhart、McClelland提出了一个基于梯度下降算法的神经网络模型，采用BP算法，即误差反向传播学习算法。该算法已成为当今影响最大的一种网络学习方法。1987年6月，首届国际神经网络学术会议在美国加州圣地亚哥召开，标志着在世界范围内掀起了一股神经网络开发研究的热潮，此后，人工神经网络的发展正式进入了高潮时期。

### 1.2.2 研究意义

对于那些仅需要考虑较少因素、有确定的运算规则以及必须进行精确解答的问题，电子计算机可以很优秀、很快速的将其解决，其处理速度和准确度远远高于人类的大脑。但是，大量不确定、模糊的非线性信息处理问题，却并不存在某种线性运算规则，此时电子计算机便无法发挥其强大的计算功能，失去了其运算和处理优势。相反的，人脑很善于处理这类模糊性问题。人脑中的神经网络是一种高度并行的非线性信息处理系统，具有学习认知能力、信息加工能力以及记忆与联想能力，这些能力使得人类能够从大量的现有实例中产生解决问题的经验，可以模糊地、直觉地对一个问题作出判断。

人工神经网络是对人脑神经网络的一种抽象，是从信息处理的角度出发，采用数学和物理方法构建起的某种简单模型。其对人脑生物神经网络进行了简化、抽象与模拟，能够反映出人脑的许多基本特性。神经网络具有并行处理、分布式存储与容错性等结构特征，同时能够自学习、自组织和自适应，在模式识别、系统辨识、信号处理、自动控制、组合优化、预测评估、故障诊断、医学与经济学等领域得了广泛的应用，解决了许多电子计算机难以解决的实际问题。人工神经网络具有良好的智能特性和广阔的应用前景，对其进行更为深入的研究，将会推动人工智能技术、信息处理技术和神经计算机的快速发展，具有深远的研究意义。

## **1.3 论文安排**

该课题首先是对神经网络以及BP网络的基本原理进行了分析、阐述，做一些简单的研究，将BP网络的优点和缺点做了介绍。然后利用MATLAB软件设计了BP网络，体现了BP神经网络在模式识别方面的应用，分析其性能。共包括5个部分，下面阐述每个部分的主要内容。

第1部分：绪论。主要介绍了神经网络的研究背景、发展历程、研究意义等内容

第2部分：人工神经网络。主要对人工神经网络进行了简单介绍，包括人工神经网络的概述、人工神经网络的构成要素以及按照不同规则对它的分类情况。

第3部分：BP神经网络。主要介绍了BP神经网络的基本原理，阐述了它的优点和缺点以及简单介绍几种针对BP网络缺点所做的改进算法。

第4部分：BP神经网络函数的程序设计。主要简单介绍了MATLAB神经网络工具箱中的各种功能函数。

第5部分：BP神经网络在模式识别中的应用。这部分主要是使用MATLAB创建了一个BP神经网络用来解决数字识别问题，并进行了深入分析，体现出了神经网络解决此类问题的能力。

# **2 人工神经网络**

## **2.1 人工神经网络概述**

神经生理学和神经解剖学的研究结果表明，神经元（neuron）是人脑组织的基本单元，是实现人脑各种功能的最小单元。据统计，人类大脑大约包含有个神经元，这些神经元之间通过各种形式的相互连接组成了一个极为复杂和庞大的网络，根据连接强弱、外部激励、兴奋与抑制状态的不同，实现不同的信息处理方式，从而完成人脑神经对外部刺激的反应过程。

人工神经网络是基于这项生物神经学理论设计出的一种数学模型，旨在对人脑神经网络进行抽象、简化和模拟。它是由大量具有适应性的节点（或称神经元）按照一定的连接方式组成的广泛并行互连的网络，其从信息处理的角度对人脑神经元网络进行模仿，反映了人脑功能的许多特征，是一个高度复杂的非线性、自适应信息处理系统。人工神经网络具有大规模的并行、分布式存储与处理、自组织、自适应和自学习能力，因而在解决模糊的、随机的、非线性的问题方面具有相当大的优势。

神经网络性能的好坏取决于三个重要的元素：激活函数、拓扑结构、学习方式。不同的激活函数使得神经元具有不同的信息处理特性，它反映的是神经元输入和输出之间的关系，以此来调整不同神经元的状态。神经元的连接方式决定了整个神经网络的拓扑结构，按照连接方式的不同，神经网络可以分为层次型结构和互连型结构。若结合网络内部信息的传递方向，可将网络划分为前馈型网络和反馈型网络两种。神经网络的拓扑结构是决定神经网络特性的第二大要素。最后一个要素是神经网络的学习方式，其最核心的是学习规则或者说学习算法，即改变权值的规则。学习算法可以分为三类：有导师学习、无导师学习以及灌输式学习。根据神经网络要实现的功能来可以选择不同的学习算法，使得所设计的网络性能达到最优。

经过几十年的发展，神经网络理论在模式识别、自动控制、信号处理等众多研究领域已经取得了广泛的成功。然而现如今的神经网络仅仅是对人脑的简单模仿，其智能性仅相当于人类四、五岁的儿童，还未能达到人类对新一代人工智能的期望，因此仍然需要进一步的深入研究，提高人工神经网络各个方面的性能。

## **2.2 人工神经网络基本处理单元**

### 2.2.1 生物学基础

科学研究结果表明，脑组织的基本单元是神经元，是神经系统结构与功能的单位。据估计，人类大脑大约包含有个神经元，这些生物神经元以确定的方式和拓扑结构相互连接起来形成了生物神经网络，它是一种相当灵巧、复杂的生物信息处理系统。网络中的每一个生物神经元都是相当简单的个体，仅能对生物刺激做出简单的反应，然而，由于众多神经元之间的相互连接，协同合作，使得人脑能够对从生物系统内外环境所接收到的信息进行综合分析处理，从而调动机体对环境做出适当的反应，这种复杂的信息处理能力是现代计算机所不具有的。

人脑中神经元的形态、功能并不是完全相同的，但是其基本的组成结构是一样的，主要由细胞体、树突、轴突、突触四部分组成，如图2.1所示。在生物神经元中，突触是细胞体之间进行通信连接的接口，相当于输入输出接口，建立起神经元的神经末梢和其他神经元的树突或者细胞体之间的连接通道。树突类似于输入端，用于接收信号。对接收到的信号的处理则由细胞体负责，相当于计算机中的微处理器，对来自其他神经元的输入信号进行整合之后，产生输出。轴突的作用是传递信息，将输出信号传递至神经末梢，之后再继续传递到其他的神经元。

经过一系列的信号传递，生物神经网络中各神经元之间连接的强弱将会按照接收到的信号进行自适应的变化，同时每个神经元会根据多个输入信号的综合结果呈现出兴奋或抑制状态。大脑最终对信息的处理结果由所有神经元的状态所决定，而神经元之间连接强度的自适应变化便是大脑的学习过程。

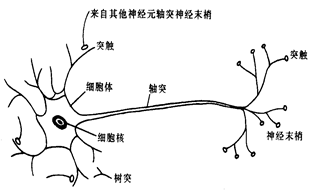


图2.1 生物神经元

### 2.2.2 人工神经元

由神经生物学基础了解到，神经元是人脑信息处理系统的最小单元，是神经网络的基本器件，所以对于人工神经网络来说，首先，也是最重要的，是对生物神经元的模拟。根据对生物神经元的研究，人们提出了许多神经元模型，其中最早的是心理学家McCulloch 和数学家W.Pitts提出的M-P模型，如今已成为应用最广泛的一种模型。该模型认为，人工神经元应具有以下特点：

(1) 神经元输入输出之间存在时滞。

(2) 神经元的输入分为兴奋性和抑制性输入。

(3) 神经元的输出信号是其输入信号的加权和经过综合处理后得出的。

(4) 神经元是一个多输入、单输出的处理器，具备多输入单输出特性。

(5) 神经元具有阈值特性。只有当总的输入信号大于阈值时，神经元才能被激活。

(6) 神经元本身是非时变的，即其突触时延和突触强度均为常数。以上述特点为基础所建立的神经元模型如图2.2所示。

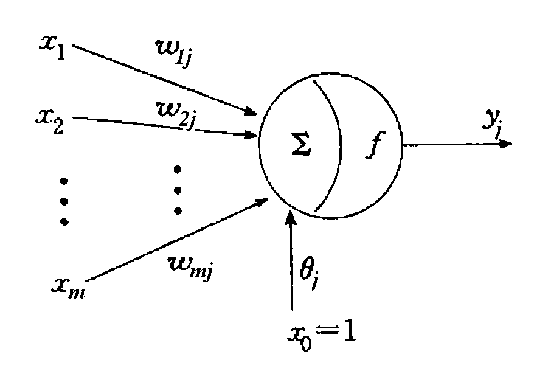


图2.2 人工神经元示意图

为简单起见，该模型并未将时间考虑在内。用数学表达式来描述此人工神经元模型，如下式：

其中——神经元的输出；

——神经元的输入；

——神经元到的连接权重；

——神经元的阈值；

——神经元激活函数；

## **2.3 人工神经网络的基本要素**

影响神经网络信息处理性能的三个主要因素分别为神经元激活函数、神经元之间的连接方式以及网络的学习算法，因此，在对神经网络进行设计和研究应用时，通常需要考虑这3个方面的内容，围绕这3个方面进行展开。

### 2.3.1 神经元激活函数

激活函数，又称为传递函数，是为了限制神经元输出信号值的范围而设定的。神经元输入信号经过加权求和后，通过激活函数将计算结果压制到允许的范围之内进行输出。激活函数反映了输入和输出信号之间的函数关系，通过改变这种输入输出关系，可以使神经网络实现不同的信息处理特性，从而解决不同的问题。

常见的神经元激活函数有如下几种，其中为神经元激活函数，为输入信号经过连接权值加权后的输出。

(1) 简单线性函数

线性函数的连续取值，数学表达式如下：

如图2.3所示：



图2.3 线性激活函数的函数图

(2) 阈值型函数

常用的阈值型激活函数有硬限幅函数和对称硬限幅函数两种，其函数图如图2.4所示。图2.4(a)为硬限幅函数，函数表达式为



图2.4(b)为对称硬限幅函数，函数表达式为



以上函数属于非线性的，输出只有两种结果，当净输入大于阈值T时，输出为1，反之，输出为0或-1。输出结果代表了神经元的状态，结果为正表明神经元为兴奋状态，结果为0或负表明神经元为抑制状态。



（b）对称硬限幅函数

（a）硬限幅函数

图2.4 阈值型函数

(3) S型函数

在神经网络中最常见的S型函数为sigmoid函数，其输出限制在两个有限值之间，是一个连续可微的函数，且其导数同样是可微的，表达式为



这是一个单极性的S型函数，其输出取值范围在[0 1]之间。如果将其向下平移，能够得到双极性的S型函数。在神经网络中有时也采用这种函数作为激活函数，此时输出范围在[-1 1]之间，表达式如下



S型函数的图像如下：



（a）单极性S型函数

（b）双极性S型函数

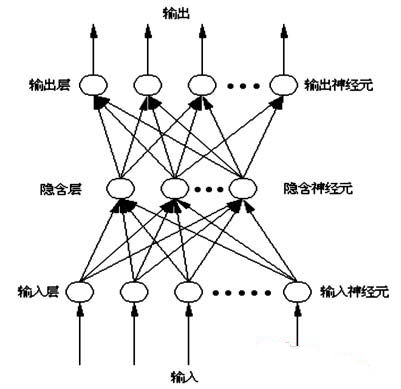


图2.5 S型激活函数

### 2.3.2 神经元之间的连接形式

单个神经元的功能是十分有限的，无法完成复杂的信息处理过程，因此需要将神经元通过广泛互连的方式组成一个庞大复杂的神经网络。连接形式决定了人工神经网络的网络结构，而不同的网络结构将会带来不一样的网络特性和功能。根据神经元之间连接形式的不同，神经网络大体上能分为两类：前向网络和反馈型网络。

(1) 前向网络

在前向网络中，网络可以分为若干个“层”，每层都是由多个神经元构成。一般来说，可以将这些“层”分为三类：输入层、输出层和隐层。输入层由若干个没有计算功能的输入节点构成，仅仅用来输入信息，位于网络的最前层。输出层处于网络的最后面，对接收到的信息进一步处理后输出。输入层和输出层统称为“可见层”，除此之外的中间层则称为隐层。隐层不和外界环境直接进行联系，它从输入层直接接收信息，输出也只作用于和它直接相连的其它层神经元。根据隐层的有无，可以将前向网络继续细分为单层前向网络和多层前向网络。图2.6是一前向无环有向图，可用它来表示前向网络的一般结构。

输出

输入

输出层

隐含层

输入层

图2.6 前向网络结构

输出神经元

隐含神经元

输入神经元

可以看出，在前向网络中，信息从输入层开始按照固定的方向一层一层的向后传递，第层神经元只接收第层神经元的输出信号，神经元自身及神经元之间并不存在反馈连接，整个前向网络中没有信息反馈。最典型的前向网络是BP神经网络，即误差反向传播神经网络。

(2) 反馈网络

Hopfield网络是一种典型的反馈神经网络。与前向网络不同，反馈神经网络的每一个神经元都具有计算功能，所以每一个神经元都可以作为信息的输入节点，也可以作为信息的输出节点，即可以存在反馈环。事实上，在反馈神经网络中至少有一个反馈连接，信息从一个节点输出后会重新被输送到自身节点或前面各层的其它节点。网络的初始状态由输入信号决定，在信息传递和反馈过程中，状态发生改变并逐步趋于稳定，稳定状态下的结果就作为反馈网络的最终输出。图2.7是一个典型的反馈型网络，图中用圆圈表示的节点代表一个神经元，每个节点都具有信息处理能力。

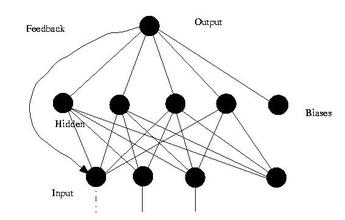


图2.7 反馈网络模型

这两种网络只是最基本的神经网络，在实际的应用中所使用的神经网络要更为复杂。有些网络不仅仅是层与层之间的神经元相互连接，在每一层内部也存在着互连的节点，这导致了神经网络结构的复杂多变，使得神经网络的信息处理功能变得多样化。

### 2.3.3 神经网络的学习

某个神经网络只有拓扑结构是没有智能特性的，只有当它和一套完整的规则相互配合时才是真正意义上的神经网络。人工神经网络的智能特性主要体现在网络的学习能力上，根据神经网络的结构特点可知，网络的学习过程即是网络神经节点之间连接权值的改变过程。神经网络的初始连接权值从整体反映了该网络对于要解决问题的知识存储，在网络学习过程中，连接权值不断发生改变从而影响神经网络的输出，使其不断逼近期望输出结果。神经网络的学习方式很多，可以大致分为三类：有导师学习、无导师学习和强化学习。

(1) 有导师学习

有导师学习也称作有监督学习，在这种学习方式中，训练样本有与之对应的正确的输出作为期望输出。将训练样本的数据加到网络输入端，同时将网络输出与相应的期望输出做比较，得到误差信号，用这个信号对连接权值进行相应的调整，反复多次后将权值收敛到一个较为理想的结果，训练结束。实际应用时，网络可以自行学习修改权值来适应不同的样本，得到期望的结果。

(2) 无导师学习

无导师学习事先不给定标准样本，需要不断地向网络提供输入信息，神经网络根据相应的学习算法，从输入的信息中找到任何可能存在的规律，然后根据找到的规律来调整权值或其结构。这是一种自组织的形式，不需要外部信进行指导。在有导师学习中，神经网络获得的外部信息越多，其解决问题的能力就越强。但是，对于很多实际问题来说并没有太多的先验信息可以提供，这种时候就只能依靠无导师学习，因此，无导师学习更具有实际意义。

(3) 强化学习

强化学习介于有导师学习和无导师学习之间，通常这种算法只是给出一个评价来作为对整个网络的评价。例如对和错，对于一系列的样本输入好或者坏就是评价这个网络性能的指标。

无论是哪种学习方式，实质都是通过改变权值来达到学习的目的。改变神经节点连接权值的规则称为神经网络的学习算法，它对网络的收敛（输出结果接近期望输出）速度有着很重要的影响。随着神经网络研究的深入，许多学习算法已经被提了出来，比较经典的学习算法有Hebb学习算法、Perceptron学习算法、σ学习算法、BP学习算法等等。该课题重点在于BP神经网络的基本原理，因此在下章节中会详细介绍BP学习算法，在这里仅简要介绍一下Hebb学习算法。

20世纪40年代末，D.O.Hebb首先提出了一种神经网络学习算法，即Hebb学习算法，在此之后出现的各种学习算法多是以此为基础提出的。

在Hebb学习算法中，学习信号简单地等于神经元的输出：

权向量的调整公式为：

上面的式子表明，连接权值调整量和输入输出的乘是成正比的。很明显，若是某一输入模式经常出现将会对权向量产生较大的影响。若出现这种情况，Hebb学习算法则需要提前设置权饱和值，防止输入输出正负总是一致时权值无约束的增长。

Hebb学习算法代表了一种纯前馈、无导师学习，该学习算法在许多神经网络模型中至今仍起着十分重要的作用。

## **2.4 人工神经网络的应用**

人工神经网络是对人脑神经网络的一种抽象，是从信息处理的角度出发，采用数学和物理方法构建起的某种简单模型。由于神经网络具有并行处理、分布式存储与容错性等结构特征，同时能够自学习、自组织和自适应，因此在信息处理、自动化、经济、数学、计算机科学等较多领域都得到了广泛且成功的应用。

下面列出一些主要应用领域：

(1) 信号处理。在自适应信号处理方面，神经网络已得到了广泛的应用，例如自适应滤波、谱估计、噪声消除等。

(2) 神经控制器。神经网络具有自学习自适应等智能特性，比较适合处理复杂的非线性系统，在实时控制系统中可以做到比常规控制器更好的控制效果，因此广泛被采用。

(3) 气象工程。无论是地球环境的气候变化还是现代的天气预报，都是十分复杂的非线性系统，神经网络在气象模式识别、环境气候分析方面都有很好的应用。

(4) 市场预测。影响市场供求关系变化的因素非常多，而且这些因素的综合作用是相当复杂的，因此传统方法对市场走势的预测误差很大而且极其困难。从复杂的问题中找出潜在的规律是神经网络的强项，利用神经网络将能够建立较为准确的市场模型。

# **3 BP**[**神经网络**](http://lib2.nuist.edu.cn/opac/openlink.php?title=%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C%E6%A8%A1%E5%9E%8B%E5%8F%8A%E5%85%B6MATLAB%E4%BB%BF%E7%9C%9F%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1)

## **3.1 BP神经网络概述**

1986年Rumelhant和McClelland提出了一种多层网络，该网络采用的是“逆推”学习算法，这便是BP (Back Propagation) 神经网络。BP神经网络如今已成为应用最为广泛和成功的神经网络之一，其基本思想是，在学习过程中，信号的正向传播和误差的反向传播共同产生作用，不断调整网络连接权值和阈值，实现网络的自学习特性。

BP网络采用的是有导师学习方式，学习开始时需要设定样本以及与之相对应的正确输出作为教师信号。样本从输入层传入，经过隐层时被逐层处理，之后传递给输出层得到一个实际输出，这是正向传播过程。将期望输出和实际输出进行比较后若发现误差不可接受，则进行误差的反向传播。误差反向传播的实质是利用误差信号来调整各个神经节点连接权值。在这个过程中，输出误差按照BP算法作用于各层的所有神经节点，从而可以得到各层节点的误差信号，以此来修正连接权值。由于是利用后一级网络层的权值和输出来计算前一级网络权值和输出，一层层的向前修正，所以将这一过程称为反向传播。需要注意的是，这两个过程并不是执行一次就能够实现网络的全部训练过程，需要周而复始地进行。权值不断地进行调整，直到网络的输出误差或学习次数达到设定值为止。

## **3.2 BP网络****的基本原理**

### 3.2.1 BP网络模型

BP网络是一个采用BP算法的多层感知器，它的拓扑结构由三部分组成：输入层、隐层、输出层。信息从输入层一层一层的向后传递，单向传播，第层神经元只接收第层神经元的输出信号，神经元自身及神经元之间并不存在反馈连接，整个前向网络中没有反馈。

图3.1是一个单隐层的BP网络模型，从图中看出网络结构并不复杂。其中为输入向量，为隐层输出向量，  为输出层输出向量，为期望输出向量。网络中的全部连接权值可以用矩阵来表示，令为输入层和隐层之间的 阶权值矩阵 ，表示隐层第个神经元的权向量；为隐层和输出层之间的阶权值向量，表示输出层第个神经元的权向量。。

图3.1 三层BP网络



对输出层：

其中为输出层激活函数，连续可微，为输出层阈值，表示输出层节点数。

对隐层：

其中为隐层激活函数，连续可微，为隐层阈值，、分别表示输入层和隐层节点数。

若令 ，则式可改写为

式可改写为

以上数学表达式构成了一个简单的BP神经网络数学模型。

### 3.2.2 标准BP网络算法

BP算法可以十分系统地解决多层前向网络中隐含层节点的连接权值的许多学习问题，使得以此为基础的BP神经网络是目前应用最为广泛和成功的神经网络之一。标准BP网络算法是最初的BP算法，如今使用的其他改进算法是在它的基础上改善而来的，下面用数学推导的方式对其进行介绍。

以上节中的单隐层BP网络为例，假设现在共输入R个学习样本，第P个学习样本的输出误差为,令平方型误差作为输出误差，则第P个学习样本的误差为：

这R个学习样本的全局误差为：

BP网络连接权值是以各个样本的误差为根据进行调整的，先来讨论第R个学习样本。为使误差不断地减小，一种方法是令权值的修正量和误差的梯度下降成正比，即：

其中， ，在训练中代表学习速率的大小。负号表示沿着梯度下降的方向，因此称为误差的梯度下降算法。

由式、、可得：



再由式、可进一步得到：



对于输出层，将式带入到式中 ，得到结果：

对隐层，将式带入到式中 ，得到结果：

对输出层和隐层各定义一个误差信号，令：

可以计算出：

因此，输出层和隐层的权值调整公式可以改写为：

由此可以看出，权值的调整和三个因素有关，即学习率、该层输出的误差信号以及这层的输入信号。

下面来推导输出层和隐层的误差函数。

输出层：

由式、进一步得：

  隐层：

由式、进一步得：

根据式进一步化简：

误差函数推导完毕。

将式、代入到式得：



以上便是对第P个样本权值调整计算公式的全部推导，下面推广至R个样本，式改写为：



将时间因素考虑进去，最终权值表达式为：



从最终的结果可以看出，与隐层相比较，输出层权值调整公式只和输出误差有关，而隐层则要受到输出层的影响，其误差信号是从输出层反传过来的。同时也可以看出，各层的激活函数必须是连续可微的，这样在计算过程中才能得出期望的调整结果，在BP网络中一般使用S型函数。

对于有多个隐层的BP神经网络，存在相同的结论，这里就不再详细论述。算法的整个过程如图3.2所示。

*V*

*f2(VTX-T2)*

*W*

*f1(WTY-T1)*



隐层

输出层



图3.2 BP算法的信号流向

BP算法由信号的前向计算和误差的反向传播两部分组成，从图中可以看出，前向计算过程是：输入信号*X*从输入层进入后，通过隐层各节点的权值向量*Vj*得到该层的输出信*Y*；该信号向前输入到输出层，通过其各节点权向*Wk*得到该层输*O*。反向过程是：在输出层期望输出*d*与实际输出*O*相比较得到误差信号，由此可计算出输出层权值的调整量；误差信号通过隐层各节点的新权值向量反传至隐层各节点，得到隐层的误差信号，由此可计算出隐层权值的调整量。

### 3.2.3 BP网络的主要能力

由于BP算法成功的解决了非线性连续函数的多层感知器权值修正问题，因此BP网络的实用价值得到了大大的提升，目前已经成为了应用最广泛的神经网络之一。相比于其它的神经网络来说，BP神经网络具有相当强的非线性映射能力、泛化能力以及容错能力，这成为了BP网络成功的重要原因。

（1）非线性映射能力

对于BP网络来说，只要能够提供足够多的样本，它便可以学习和存储其中隐藏的输入-输出模式映射关系，完成从输入空间到输出空间任意维度的非线性映射。在工程中往往有大量的输入输出数据，找出其中蕴含的规律是主要的难题，使用BP网络是一个很好的选择。

（2）泛化能力

通俗一点来说，泛化能力指的是神经网络的普遍适用性。在网络进行学习时不可能将所有的数据全都作为样本输入进行训练，因此，好的网络应该在训练结束后能够对未曾见过的非样本数据做出正确的处理。BP网络在以后的工作中，对于输入的陌生数据能够完成从输入空间向输出空间的映射。这是衡量BP网络性能优劣的重要因素。

（3）容错能力

BP网络允许输入模式及样本中存在较大的误差，甚至允许出现个别错误。因为BP网络对权值的调整结果是从大量的样本数据中统计出来的，个别样本的中的误差并不影响网络对整个样本中所蕴含规律的正确反映。

## **3.3** **BP网络存在的问题**

虽然BP网络目前得到了十分广泛的应用，但是其自身也存在以下一些主要问题。

(1) 网络的收敛速度慢。

由于学习速率是固定的，因此，BP神经网络学习次数过多，需要较长时间的训练过程，学习收敛速度太慢。对于一些复杂的问题，标准BP算法可能需要成百上千次的学习才能收敛。

(2) 存在局部极小点。

由上节的推导结果来看，BP神经网络的误差函数的极小点并不是唯一的，标准BP算法能够使权值在训练过程中逐步收敛到某个最小点，但是无法保证这点就是[误差](http://baike.baidu.com/view/40051.htm)平面的全局最小点。若其陷入到某个局部最小点，训练过程将无法继续进行。

(3) 隐含层层数和节点数选取困难。

对于BP神经网络来说，网络的隐含层具有十分重要的意义。但是目前网络隐含层的层数和节点数还没有十分有效的选取方法，缺乏理论上的指导，大多是通过反复实验或者根据经验确定。因此，网络具有很大的冗余性，延长了网络的学习过程，在一定程度上也对网络的学习增加了负担。

(4) 网络的学习和记忆具有不稳定性。

如果给一个训练结束的BP神经网络增加了学习样本，那么训练好的网络就需要重新进行训练，它对于已经训练好的权值和阈值是不会记忆的。也就是说，它已经调整好的网络连接权值会被破坏，导致已经记住的模式信息消失。

## **3.4 BP网络算法的改进**

由于BP神经网络存在以上缺陷与不足，在神经网络的整个发展过程中，许多专家学者提出了改进的BP网络算法，这些算法都在一定程度上改善了标准BP算法，使得BP神经网络的应用更为广泛。下面简单介绍两个比较典型的改进算法。

(1) 增加动量项的BP算法

标准BP算法在修正权值时，只考虑了当前时间误差的梯度下降方向，却忽视了当前时间以前的梯度方向。这使得训练过程中权值的变化动荡、不平滑，从而收敛缓慢。为了加快BP网络的收敛速度，一些学者提出了可以在权值修正公式中加入动态修正量：

式中，W为某层权值矩阵，O为该层输出向量，α∈(0,1)称为动量系数。这使得如果在训练过程中误差梯度出现局部极小时，动量项可以使其跳出局部极小区域。

(2) 可变速率的BP算法

学习速率对于学习算法是比较重要的，如果学习速率太大有可能导致算法产生震荡，而学习速率太小却使算法收敛速度太慢。在标准的BP算法和增加动量项的BP算法里，学习速率是固定不变的，不能保证在任何情况下都是最合适的，因此便有了可变学习速率的BP算法。

它基本的思想是：观察整个样本集中均方误差的变化，当均方误差的变化是逐渐减小时，说明权值和阈值的调整方向是正确的，可以令学习速率η乘以增量因子α来增大学习速率，适当加快收敛速度；当均方误差逐渐增加最终超过某个值时，这说明权值调整的方向是错误的，需要减小学习速率，此时令学习速率η乘以减量因子ρ。通常情况下，增量因子α可以取为1.05，减量因子ρ可以取为0.75。学习速率的修正方式可用如下数学式子来表示：



以上学习算法仅仅是众多改进学习算法中比较典型的两个，除此之外还有弹性BP算法、共轭梯度法、拟牛顿算法、L­\_M算法等等。这些算法大致可以分为两类，一类是启发式学习算法，其中增加动量项的BP算法、可变速率的BP算法、弹性BP算法便属于此类；另一类是数值优化方法，共轭梯度法、拟牛顿算法、L­\_M算法属于这类。这些方法之间并不存在优劣之分，只是应用的场合和背景不同，每种算法都存在这样或那样的优势和不足，我们需要做的是在合适的情况下使用合适的算法来解决我们的问题。

# **4 BP神经网络函数的程序设计**

MATLAB软件的快速发展则提供了一种便利的仿真手段来对神经网络理论进行实现。MATLAB神经网络工具箱的出现，将大量的计算工作交给了计算机，提高了计算的准确度和精度，使研究人员从冗杂的工作中解放了出来，专心于网络理论部分的研究。这在很大程度上提高了工作效率，神经网络的应用范围由此也更加广泛。下面简单介绍下MATLAB神经网络工具箱中的一些主要函数及标准BP算法的程序实现。

## **4.1 BP神经网络工具箱中的函数**

### 4.1.1 BP网络创建函数

在MATLAB神经网络工具箱中有两个函数可以创建BP神经网络，下面分别做一下介绍。

(1) newff函数

该函数可以创建一个BP网络。调用的格式为：

net=newff(PR,[S1 S2..SN],{TF1 TF2..TFN},BTF,BLF,PF)

其中，

net为创建出的BP神经网络；

PR为网络输入向量元素的最大值和最小值组成的矩阵；

[S1 S2…SN]中Si表示第i层神经元的个数，共N层；

{TFl TF2…TFN}表示网络各层的传输函数，默认为‘tansig’；

BTF表示网络的训练函数，默认为‘trainlm’；

BLF表示网络的权值学习函数，默认为‘learngdm’；

PF表示性能数，默认为‘mse’。

(2) newcf函数

newcf函数用于创建级联前向BP网络，其调用格式及参数与newff相同。

### 4.1.2 神经元上的传递函数

传递函数是BP神经网络的重要组成部分。传递函数即为激活函数，需要是连续可微的。BP网络经常使用的传递函数是S型对数或正切函数和线性函数。

(1) logsig函数

该传递函数为S型对数函数。其调用格式为：

A=logsig(N,FP)

其中，

N：Q×S维的输入列向量矩阵；

FP：性能参数(可忽略)；

A：函数的返回值，位于区间(0,1)中；

(2) tansig函数

tansig函数为双曲正切S型传递函数。其调用格式为：

A=tansig(N,FP)

其中，

N：Q×S维的输入列向量矩阵；

FP：性能参数(可忽略)；

A：函数返回值，位于区间(-1,1)之间。

(3) purelin函数

该函数为线性传递函数。其调用格式为：

A=purelin(N,TP)

其中，

N：Q×S维的输入列向量矩阵；

FP：性能参数（可忽略）；

A：函数返回值，A=N。

### 4.1.3 BP网络学习函数

在MATLAB神经网络工具箱中有两个函数可以创建BP神经网络，下面分别做一下介绍。

(1) learngd函数

该函数为梯度下降权值/阈值学习函数，它通过神经元的输入和误差，以及权值和阈值的学习速率来计算权值或阈值的变化率。其调用格式为：

[dW,LS]=learngd(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)

其中，

W：S×R维的权值矩阵；

P：Q×R维的输入向量；

Z：Q×S维的加权输入向量；

N：Q×S维的输入向量；

A：Q×S维的输出向量；

T：Q×S维的层目标向量；

E：Q×S维的层误差向量；

gW、gA：与性能相关的S×R维梯度；

D：S×S维的神经元距离矩阵；

LP：学习参数，用来设置学习速率。

LS：学习状态，初始状态下为空。

输出参数dW：S×R维的权值或阈值变化率矩阵；

输出参数LS：新的学习速率。

(2) learngdm函数

learngdm函数为梯度下降动量学习函数，它利用神经元的输入和误差、权值或阈值的学习速率和动量常数，来计算权值或阈值的变化率。其调用格式及参数与learngd函数相同。

### 4.1.4 BP网络训练函数

在MATLAB神经网络工具箱中提供了train函数及adapt函数实现BP神经网络训练。下面分别对这两个函数作介绍。

(1) train函数

该函数用于训练一个神经网络。网络训练函数是通用的学习函数，训练函数重复地把一组输入向量应用到一个网络上，每次都更新网络，直到达到某种准则。停止准则可能是最大的学习步数、最小的误差梯度或误差目标等。该函数的调用格式为：

[net,tr,Y,E,Pf,Af]=train(NET,P,T,Pi,Ai)

其中，

net：训练后的网络；

tr：训练记录；

Y：网络输出向量；

E：误差向量；

Pf：训练终止时的输入延时状态；

Af：训练终止时的层延时状态；

NET：训练前的网络；

P：网络输入向量矩阵；

T：网络的目标矩阵，默认为0；

Pi：初始输入延时，默认为0；

Ai：初始的层延时，默认为0；

(2) adapt函数

该函数为学习自适应函数，其在每一个输入时间阶段更新网络时仿真网络。其调用格式及参数与trian函数相同。

## **4.2 BP算法的编程步骤及流程图**

目前神经网络的实现主要以软件编程为主，下面介绍一下标准BP算法的编程步骤。

(1) 初始化 对权值矩阵、随机地赋予较小的非零数值，给定阈值，将样本计数器、训练次数设为1，学习率设为区间内的小数，网络训练后达到的精度设为一正的小数。

(2) 输入训练样本对，计算各层输出 将当前样本输入到网络中，计算和中各分量。

(3) 计算网络输出误差 设共有对训练样本，网络对第个样本的误差

将全部样本输出误差进行累加作为总输出误差。

(4) 计算各层误差信号 应用式、计算和。

(5) 调整各层权值 应用式、计算、中各分量。

(6) 检查是否所有的样本都经过了训练 若，计数器、增1，返回步骤(2)，否则转步骤(7)。

(7) 检查网络总误差是否达到精度要求 当总误差，训练结束，否则置1，返回步骤(2)。

标准BP算法流程如图4.1所示。

初始化V、W、T

计数器q=1，p=1

输入样本，计算各层输出：





计算输出误差：



计算各层误差信号：





调整各层权值：







结束

*P*增1，*q*增1

*P*=1



图4.1 标准BP算法流程

# **5 BP网络在模式识别中的应用**

## **5.1 问题的提出**

利用MATLAB设计一个BP神经网络，使用0-9这十个阿拉伯数字对该网络进行训练，将训练好的网络对阿拉伯数字进行识别。其中，阿拉伯数字用二进制数来表示。要求设计的网络不仅能对理想的输入向量进行识别，也能够识别被噪声所污染的数字。

## **5.2 数据选择**

每个代表数字的输入向量均用35个二进制元素来表示，这样就组成了一个输入向量矩阵number。而期望输出有10个元素，因此使用10位的二进制数来表示。十个数字的顺序为1,2，……,9,0，在二进制数中，数字所在的位置处元素为1，其他处为0。例如，因为数字2是第二个数字，所以其期望输出向量为（0,1,0，……）。

因此，输入输出向量如下：

number=

[0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0; %1

1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1; %2

1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1; %3

1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0; %4

1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1; %5

1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1; %6

1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1; %7

1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1; %8

1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1; %9

1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1];%0

输出目标向量为：

targets=

[1 0 0 0 0 0 0 0 0 0

0 1 0 0 0 0 0 0 0 0

0 0 1 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0

0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 0 0 1];

## **5.3 建立网络**

使用newff（）函数创建该BP网络。根据输入输出向量可知，该网络具有35个输入节点，10个输出节点。隐含层神经元数目可以改变，根据经验和猜测，将隐层节点数定为12。如果训练结果不理想，可以对隐含层神经元个数进行适当的调整。隐含层和输出层的神经元传递函数均使用logsigmoid函数，该函数将输入数据映射到（0,1）区间内，符合输出向量的要求。该网络结构如图5.1所示：

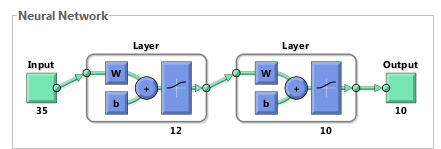


图5.1 BP网络结构

其实现的MATLAB代码为：

>> clear all;

[R,Q]=size(number);

P=number;

T=targets;

net=newff(minmax(P),[12,10],{'logsig','logsig'},'traingdx');

net.LW{2,1}=net.LW{2,1}\*0.01;%调整权值

net.b{2}=net.b{2}\*0.01;%调整阈值

## **5.4 使用含噪声的数字样本进行训练**

为了使产生的网络具有一定的容错能力，在训练网络时不能只输入理想的数字信号，应该同时兼顾输入信号受到噪声污染时的情况。因此具体的训练过程如下：

（1）使用理想的数字信号对网络进行训练，令网络具备识别理想信号的能力。

（2）使用10组带有噪声的信号对网络进行训练，同时，依然要输入理想信号。这是为了防止网络只能正确识别有噪信号而影响了其对理想信号的分辨。

其MATLAB代码如下：

>> %无噪训练

net.trainParam.goal=0.001;

net.trainParam.epochs=5000;

net=train(net,P,T);

>> %有噪训练

netn=net;

netn.trainParam.goal=0.001;

netn.trainParam.epochs=6000;

T=[targets targets targets targets];

for i=1:10

P=[number,number,(number+randn(R,Q)\*0.3),(number+randn(R,Q)\*0.2)];

netn=train(netn,P,T);

end

## **5.5 测量网络容错性**

接下来对网络识别系统的可靠性进行测量。输入上百个数字信号且对信号进行不同程度的噪声污染，绘制网络识别误差和噪声的比较曲线。

其实现的MATLAB代码如下：

noise\_range=0:0.05:0.5;

T=targets;

max\_test=100;

for i=1:11

error1(i)=0;

error2(i)=0;

for j=1:max\_test

P=number+randn(R,Q)\*noise\_range(i);

A=sim(net,P);

AA=compet(A); error1(i)=error1(i)+sum(sum(abs(AA-T)))/2

An=sim(netn,P);

AAn=compet(An);

error2(i)=error2(i)+sum(sum(abs(AAn-T)))/2

end

end

plot(noise\_range,error1,'r-',noise\_range,error2,'-')

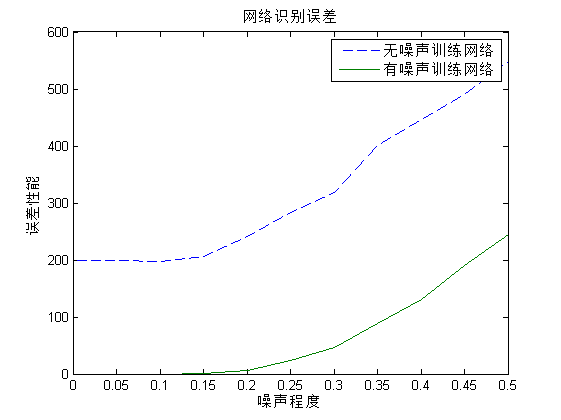
输出结果如图5.2所示：

图5.2 识别误差比较

从图中可以看出，就容错性而言，有噪声信号训练的网络明显比无噪声信号训练的网络好的多。

## **5.6 对污染数字进行识别**

以噪声水平为0.4的噪声信号对数字进行污染，使用有噪声训练网络对所污染的数字进行识别。

其实现MATLAB代码如下：

>> %对污染数字进行识别

for i=1:10

number\_noise=number(:,i)+randn(R,1)\*err;%对某一列加噪

subplot(4,5,i);

plotchar(number\_noise);%画出污染数字

%画出识别结果

A3=sim(netn,number\_noise);

A3=compet(A3);

answer=find(A3==1);

subplot(4,5,i+10);

plotchar(number(:,answer));

end

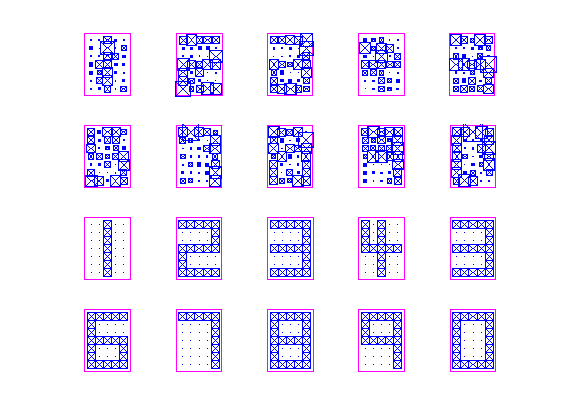
建立的BP神经网络对污染数字识别结果如图5.3所示：

图5.3 数字识别结果

识别结果：

从图5.3可以看出，10个污染数字中，训练后的网络正确识别出了9个,这说明了经过训练后的BP网络确实有较好的模式识别能力。但是正确率并不是100%，需要进一步改进。

## **5.7 仿真分析**

### 5.7.1 隐含层神经元数目对仿真的影响

BP网络的隐含层神经元数目对BP网络预测精度影响较大：神经元个数过少，网络需要增加训练次数，训练的精度会受到影响；神经元个数太多，训练时间增加，网络容易过拟合。最佳隐含层节点数选择可参考如下公式：



式中，n为输入层节点个数；为隐层节点个数；为输出层节点个数；为0 ~ 10之间的常数。根据参考公式，本例中隐层节点数大致在6 ~ 16之间。不改变网络传递函数、学习函数和训练函数的情况下，不同隐层节点数得到的仿真结果如下：



（a）节点数为2

（b）节点数为12

（c）节点数为16

（d）节点数为300

图5.4 不同节点数误差图

从图中可以看出随着节点数很小时误差很大，而节点数增大时，误差随之减小。但

是，当节点数过于大的时候虽然误差比较小，但是训练时间却比较长，出现过拟合现象。

### 5.7.2 训练函数对仿真的影响

比较不同的训练函数trainlm、traingdx、traingdm和traingd对网络性能的影响。在网络结构和权值、阈值一样的情况下，BP神经网络预测误差和训练函数之间的仿真结果如下图所示：



（a）traingd

（b）traingdm

（c）traingdx

（d）trainlm

图5.5 不同训练函数误差图

图5.5表明训练函数的选择对BP神经网络预测精度有较大影响，在本例中训练函数traingdx、trainlm明显优于traingd、traingdm，训练出来的网络误差较小。其实，在实际的网络训练中，traingdx和trainlm虽然预测误差相差不大，但是trainlm的训练速度比traingdx要快很多，所以在不太复杂的实例中一般多使用trainlm作为网络训练函数。

### 5.7.3 传递函数对仿真的影响

在隐含层神经元的个数为12的情况下，使得隐含层和输出层的传递函数为（logsig、logsig）、（logsig、tansig）、（tansig、logsig）、（tansig、tansig），仿真结果如下：



图5.6 不同传递函数误差图

从图中明显可以看出，隐含层和输出层的传递函数为（logsig、logsig）时网络性能最好。虽然logsig和tansig都是S型函数，但是在本例中的数据均是在（0,1）区间内，所以使用logsig函数更好。

# **结 论**

本文首先介绍了人工神经网络以及BP神经网络的基本知识和原理，然后使用MATLAB神经网络工具箱进行了实例仿真，分别验证了BP神经网络在模式识别中的应用效果。仿真结果显示，BP神经网络确实具有强大的信息处理能力。由于BP神经网络的并行分布式结构、万能逼近能力，因此在处理模糊性问题上具有无可比拟的优势，十分适合应用在模式识别等问题中。

虽然论文对BP神经网络的各个方面都做了介绍，包括算法原理、局限性以及改进算法，并且在最后还给出了实例分析，但是仍然存在许多不足之处。在对BP算法的推导过程中，仅仅考虑了单隐层的BP网络，虽然得出了正确的结果，但是由于取得是特例，所以说服力并不是很强。而且对改进算法的介绍太简略，实际上在网络应用中，采用的都是改进的BP算法，本文却忽略了这点，其实用性不够强。在实例应用中对于隐含层的层数和节点数的确定也没有讨论，而是根据经验公式直接得出的数值。这些都是需加强提高的，所以论文的完成并不能说明自己真正掌握了这门科学，还是需要不断学习相关的知识、理论以及研究成果。

经过几十年不断的深入研究，人工神经网络已经取得了令人瞩目的发展和相当广泛的应用，在很多地方都发挥着其自身的作用，几乎渗透到了人类社会的各个学科领域之中。但是目前神经网络的研究远没有达到人们理想中的期望，基于神经网络的人工智能仅能达到人类五、六岁孩子的水平，因此还有很大的研究空间，这需要广大神经网络爱好者和研究人员继续努力和创新。未来的神经网络必将对人类社会产生意义深远的重大影响，改变人们对很多东西的认知。

# **致 谢**

通过几个月的不懈努力，自己的毕业论文终于圆满的完成了。在此，首先要感谢我的指导老师赵鸿图教授。赵老师宽容的性格、渊博的知识令我获益匪浅，让我在书写论文期间能够把握住自己的节奏，扎扎实实的完成了自己的第一篇论文。

其次，我也要感谢在这四年中一直关心、照顾我的辅导员老师、教授我课堂知识的各科任教老师、朝夕相处的同学们以及在学校里认识的学长、学姐和朋友们，他们在我的学习、生活和校园活动中默默地影响和指导着我，让我在美好的四年时光里没有因为放纵而堕落、虚度而悔恨。感谢我所遇到过的人，感激每个帮助过我的人，是他们充实了我整个的大学生活，真的谢谢！

最后，向评审论文的老师们表示深深的敬意！向答辩委员会的老师们表示衷心的感谢！

# **参考文献**

[1] 韩立群.人工神经网络理论、设计及应用. 北京: 化学工业出版社,2007

[2] 赵鸿图，茅艳等. 通信原理MATLAB仿真教程.北京：人民邮电出版社，2010

[3] 葛哲学，孙志强等.神经网络理论与MATLAB R2007实现.北京：电子工业出版社，2007

[4] 陈明等.MATLAB神经网络原理与实例精解.北京：清华大学出版社，2013

[5] Simon Haykin著；叶世伟等译.神经网络原理（原书第2版）.北京：机械工业出版社，2004

[6] 朱大奇，史慧. 人工神经网络原理及应用.北京：科学出版社，2006

[7] 丁士圻，郭丽华. 人工神经网络基础.哈尔滨：哈尔滨工程大学出版社，2008

[8] 张德丰. MATLAB神经网络应用设计.北京：机械工业出版社，2009

[9] 史峰，王小川等. MATLAB神经网络30个案例分析.北京：北京航空航天出版社，2010.04.01

# **附 录**

## **数字识别完整代码**

clear all;

%网络创建

load data;

[R,Q]=size(number);

P=number;

T=targets;

%不同隐含层节点

% NodeNum=2;

NodeNum=12;

% NodeNum=16;

% NodeNum=300;

%不同训练函数

% BTF='traingd';

% BTF='traingdm';

% BTF='trainlm';

BTF='traingdx';

%不同传递函数

TF1='logsig';%隐含层传递函数

TF2='logsig';%输出层传递函数

% TF1='tansig';

% TF2='tansig';

% TF1='logsig';

% TF2='tansig';

% TF1='tansig';

% TF2='logsig';

BLF='learngdm';

TypeNum=10;%输出层节点

net=newff(minmax(P),[NodeNum,TypeNum],{TF1,TF2},BTF,BLF);

net.LW{2,1}=net.LW{2,1}\*0.01;%调整权值

net.b{2}=net.b{2}\*0.01;%调整阈值 %使得初始权值足够小，加快学习速率

%网络训练

net.trainParam.goal=0.001;

net.trainParam.epochs=5000;

net=train(net,P,T);

%网络测试

A=sim(net,P);

AA=compet(A)；

%有噪声输入来训练网络,共四组输入，两组未加噪两组加噪，10次训练

netn=net;

netn.trainParam.goal=0.001;

netn.trainParam.epochs=6000;

T=[targets targets targets targets];

for i=1:10

P=[number,number,(number+randn(R,Q)\*0.3),(number+randn(R,Q)\*0.2)];

netn=train(netn,P,T);

end

%使用加入不同程度噪声的输入向量进行网络性能测量，训练100次

noise\_range=0:0.05:0.5;

T=targets;

max\_test=100;

for i=1:11

error1(i)=0;

error2(i)=0;

for j=1:max\_test

%未加噪输入进行网络性能测试

P=number+randn(R,Q)\*noise\_range(i);

A=sim(net,P);

AA=compet(A);%compet竞争传递函数使得矩阵每一列中最大的为1，其他的为零。

error1(i)=error1(i)+sum(sum(abs(AA-T)))/2

%加噪输入进行网络性能测试

An=sim(netn,P);

AAn=compet(An);

error2(i)=error2(i)+sum(sum(abs(AAn-T)))/2

end

end

figure;

plot(noise\_range,error1,'--',noise\_range,error2,'-')

title('网络识别误差'),xlabel('噪声程度'),ylabel('误差性能')

legend('无噪声训练网络','有噪声训练网络')

%对污染数字进行识别 以噪声指标为0.5的噪声信号对数字进行污染

err=0.4;

figure;

for i=1:10

number\_noise=number(:,i)+randn(R,1)\*err;%对某一列加噪

subplot(4,5,i);

plotchar(number\_noise);%画出污染数字

% 画出识别结果

A3=sim(netn,number\_noise);

A3=compet(A3);

answer=find(A3==1);

subplot(4,5,i+10);

plotchar(number(:,answer));

end