

基于 MATLAB 的 BP 神经网络应用

蒋亮亮

南京信息工程大学滨江学院自动化专业，南京 210044

摘要：本文首先说明课题研究的目的和意义，评述课题的国内外研究现状，引出目前存在的问题。然后分析了神经网络算法的基本原理，给出经典神经网络算法的具体实现方法，总结神经网络算法的特点，并给出神经网络算法的基本流程。采用 **Matlab** 软件编程实现 BP 神经网络算法。将神经网络算法应用于函数逼近和样本含量估计问题中，并分析相关参数对算法运行结果的影响。最后对 BP 神经网络算法进行了展望。

关键词：神经网络；BP 神经网络；函数逼近

目 录

1 绪论	1
1.1 人工神经网络的研究背景和意义	1
1.2 神经网络的发展与研究现状	2
1.3 神经网络的研究内容和目前存在的问题	3
1.4 神经网络的应用	4
2 神经网络结构及 BP 神经网络	4
2.1 神经元与网络结构	4
2.2 BP 神经网络及其原理	7
2.3 BP 神经网络的主要功能	9
2.4 BP 网络的优点以及局限性	9
3 BP 神经网络在实例中的应用	10
3.1 基于 MATLAB 的 BP 神经网络工具箱函数	10
3.2 BP 网络在函数逼近中的应用	12
3.3 BP 网络在样本含量估计中的应用	17
4 结束语	23
参考文献:	24
英文摘要	25
致 谢	26

基于 MATLAB 的 BP 神经网络应用

蒋亮亮

南京信息工程大学滨江学院自动化专业，南京 210044

摘要: 本文首先说明课题研究的目的和意义，评述课题的国内外研究现状，引出目前存在的问题。然后分析了神经网络算法的基本原理，给出经典神经网络算法的具体实现方法，总结神经网络算法的特点，并给出神经网络算法的基本流程。采用 Matlab 软件编程实现 BP 神经网络算法。将神经网络算法应用于函数逼近和样本含量估计问题中，并分析相关参数对算法运行结果的影响。最后对 BP 神经网络算法进行了展望。

关键词: 神经网络；BP 神经网络；函数逼近

1 绪论

人工神经网络（Artificial Neural Networks, NN）是由大量的、简单的处理单元（称为神经元）广泛地互相连接而形成的复杂网络系统，它反映了人脑功能的许多基本特征，是一个高度复杂的非线性动力学系统^[1]。神经网络具有大规模并行、分布式存储和处理、自组织、自适应和自学习能力，特别适合处理需要同时考虑许多因素和条件的、不精确和模糊的信息处理问题。神经网络的发展与神经科学、数理科学、认知科学、计算机科学、人工智能、信息科学、控制论、机器人学、微电子学、心理学、微电子学、心理学、光计算、分子生物学等有关，是一门新兴的边缘交叉学科。

神经网络具有非线性自适应的信息处理能力，克服了传统人工智能方法对于直觉的缺陷，因而在神经专家系统、模式识别、智能控制、组合优化、预测等领域得到成功应用^[2]。神经网络与其他传统方法相组合，将推动人工智能和信息处理技术不断发展。近年来，神经网络在模拟人类认知的道路上更加深入发展，并与模糊系统、遗传算法、进化机制等组合，形成计算智能，成为人工智能的一个重要方向。

MATLAB 是一种科学与工程计算的高级语言，广泛地运用于包括信号与图像处理，控制系统设计，系统仿真等诸多领域。为了解决神经网络问题中的研究工作量 and 编程计算工作量问题，目前工程领域中较为流行的软件 MATLAB，提供了现成的神经网络工具箱（Neural Network Toolbox，简称 NNbox）^[3]，为解决这个矛盾提供了便利条件。神经网络工具箱提供了很多经典的学习算法，使用它能够快速实现对实际问题的建模求解。在解决实际问题中，应用 MATLAB 语言构造典型神经网络的激活传递函数，编写各种网络设计与训练的子程序，网络的设计者可以根据需要调用工具箱中有关神经网络的设计训练程序，使自己能够从烦琐的编程中解脱出来，减轻工程人员的负担，从而提高工作效率。

1.1 人工神经网络的研究背景和意义

人工神经网络是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系

统对真实世界物体所作出的交互反应^[5]。

人工神经网络就是模拟人思维的一种方式，是一个非线性动力学系统，其特色在于信息的分布式存储和并行协同处理。虽然单个神经元的结构极其简单，功能有限，但大量神经元构成的网络系统所能实现的行为却是极其丰富多彩的。

近年来通过对人工神经网络的研究，可以看出神经网络的研究目的和意义有以下三点：（1）通过揭示物理平面与认知平面之间的映射，了解它们相互联系和相互作用的机理，从而揭示思维的本质，探索智能的本源。（2）争取构造出尽可能与人脑具有相似功能的计算机，即神经网络计算机。（3）研究仿照脑神经网络的人工神经网络，将在模式识别、组合优化和决策判断等方面取得传统计算机所难以达到的效果。

人工神经网络特有的非线性适应性信息处理能力，克服了传统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的缺陷，使之在神经专家系统、模式识别、智能控制、组合优化、预测等领域得到成功应用。人工神经网络与其它传统方法相结合，将推动人工智能和信息处理技术不断发展。近年来，人工神经网络正向模拟人类认知的道路上更加深入发展，与模糊系统、遗传算法、进化机制等结合，形成计算智能，成为人工智能的一个重要方向，将在实际应用中得到发展。将信息几何应用于人工神经网络的研究，为人工神经网络的理论研究开辟了新的途径。神经计算机的研究发展很快，已有产品进入市场。光电结合的神经计算机为人工神经网络的发展提供了良好条件。

1.2 神经网络的发展与研究现状

1.2.1 神经网络的发展

神经网络起源于 20 世纪 40 年代，至今发展已半个多世纪，大致分为三个阶段^[7]。

1) 20 世纪 50 年代-20 世纪 60 年代：第一次研究高潮

自 1943 年 M-P 模型开始，至 20 世纪 60 年代为止，这一段时间可以称为神经网络系统理论发展的初期阶段。这个时期的主要特点是多种网络的模型的产生与学习算法的确定。

2) 20 世纪 60 年代-20 世纪 70 年代：低潮时期

到了 20 世纪 60 年代，人们发现感知器存在一些缺陷，例如，它不能解决异或问题，因而研究工作趋向低潮。不过仍有不少学者继续对神经网络进行研究。

Grossberg 提出了自适应共振理论；Kohonen 提出了自组织映射；Fukushima 提出了神经认知网络理论；Anderson 提出了 BSB 模型；Webos 提出了 BP 理论等。这些都是在 20 世纪 70 年代和 20 世纪 80 年代初进行的工作。

3) 20 世纪 80 年代-90 年代：第二次研究高潮

进入 20 世纪 80 年代，神经网络研究进入高潮。这个时期最具有标志性的人物是美国加州工学院的物理学家 John Hopfield。他于 1982 年和 1984 年在美国科学院院刊上发表了两篇文章，提出了模拟人脑的神经网络模型，即最著名的 Hopfield 模型。Hopfield 网络是一个互连的非线性动力学网络，它解决问题的方法是一种反复运算的动态过程，这是符号逻辑处理方式不具备的性质。20 世纪 80 年代后期到 90 年代初，神经网络系统理论形成了发展的热点，多种模型、算法和应用被提出，研究经费重新变得充足，使得研究

者们完成了很多有意义的工作。

1.2.2 神经网络的现状

进入 20 世纪 90 年代以来,神经网络由于应用面还不够宽,结果不够精确,存在可信度问题,从而进入了认识与应用研究期。

- 1) 开发现有模型的应用,并在应用中根据实际运行情况对模型、算法加以改造,以提高网络的训练速度和运行的准确度。
- 2) 充分发挥两种技术各自的优势是一个有效方法。
- 3) 希望在理论上寻找新的突破,建立新的专用/通用模型和算法。
- 4) 进一步对生物神经网络进行研究,不断地丰富对人脑的认识。

1.3 神经网络的研究内容和目前存在的问题

1.3.1 神经网络的研究内容

神经网络的研究内容相当广泛,反映了多科学交叉技术领域的特点。目前,主要的研究工作集中在以下四方面^[6]:

- (1) 生物原型研究:从生理学、心理学、解剖学、脑科学、病理学、生物科学方面研究神经细胞、神经网络、神经系统的生物原型结构及其功能机理。
- (2) 建立理论模型:根据生物圆形的研究,建立神经元、神经网络的理论模型,其中包括概念模型、知识模型、物理化学模型、数学模型等。
- (3) 网络模型与算法研究:在理论模型研究的基础上构成具体的神经网络模型,以实现计算机模拟或准备制作硬件,包括网络学习算法的研究。这方面的工作也称为技术模型研究。
- (4) 神经网络应用系统:在网络模型与算法研究的基础上,利用神经网络组成实际的应用系统,例如,完成某种信号处理或模式识别的功能、构成专家系统、制成机器人等。

1.3.2 神经网络研究目前存在的问题

人工神经网络的发展具有强大的生命力。当前存在的问题是智能水平还不高,许多应用方面的要求还不能得到很好的满足;网络分析与综合的一些理论性问题还未得到很好的解决。例如,由于训练中稳定性的要求学习率很小,所以梯度下降法使得训练很忙动量法因为学习率的提高通常比单纯的梯度下降法要快,但在实际应用中还是很慢^[7]。针对千变万化的应用对象,各类复杂的求解问题,编制一些特定的程序、软件求解,耗费了大量的人力和物力。而这些软件往往只针对某一方面的问题有效,并且在人机接口、用户友好性等诸多方面存在一定的缺陷。在微机飞速发展的今天,很多都已不能满足发展的需要。

1.4 神经网络的应用

神经网络理论的应用取得了令人瞩目的发展，特别是在人工智能、自动控制、计算机科学、信息处理、机器人、模式识别、CAD/CAM 等方面都有重大的应用实例。下面列出一些主要应用领域^[4]：

（1）模式识别和图像处理。印刷体和手写字符识别、语音识别、签字识别、指纹识别、人体病理分析、目标检测与识别、图像压缩和图像复制等。

（2）控制和优化。化工过程控制、机器人运动控制、家电控制、半导体生产中掺杂控制、石油精炼优化控制和超大规模集成电路布线设计等。

（3）预报和智能信息管理。股票市场预测、地震预报、有价证券管理、借贷风险分析、IC 卡管理和交通管理。

（4）通信。自适应均衡、回波抵消、路由选择和 ATM 网络中的呼叫接纳识别和控制。

（5）空间科学。空间交汇对接控制、导航信息智能管理、飞行器制导和飞行程序优化管理等。

2 神经网络结构及 BP 神经网络

2.1 神经元与网络结构

人工神经网络（artificial neural network, ANN）是模仿生物神经网络功能的一种经验模型。生物神经元受到传入的刺激，其反应又从输出端传到相联的其它神经元，输入和输出之间的变换关系一般是非线性的。神经网络是由若干简单（通常是自适应的）元件及其层次组织，以大规模并行连接方式构造而成的网络，按照生物神经网络类似的方式处理输入的信息。模仿生物神经网络而建立的人工神经网络，对输入信号有功能强大的反应和处理能力^[9]。

神经网络是由大量的处理单元（神经元）互相连接而成的网络。为了模拟大脑的基本特性，在神经科学研究的基础上，提出了神经网络的模型。但是，实际上神经网络并没有完全反映大脑的功能，只是对生物神经网络进行了某种抽象、简化和模拟^[8]。神经网络的信息处理通过神经元的互相作用来实现，知识与信息的存储表现为网络元件互相分布式的物理联系。神经网络的学习和识别取决于各种神经元连接权系数的动态演化过程。

若干神经元连接成网络，其中的一个神经元可以接受多个输入信号，按照一定的规则转换为输出信号。由于神经网络中神经元间复杂的连接关系和各神经元传递信号的非线性方式，输入和输出信号间可以构建出各种各样的关系，因此可以用来作为黑箱模型，表达那些用机理模型还无法精确描述、但输入和输出之间确实有客观的、确定性的或模糊性的规律。因此，人工神经网络作为经验模型的一种，在化工生产、研究和开发中得到了越来越多的用途。

2.1.1 生物神经元

人脑大约由 10^{12} 个神经元组成，神经元互相连接成神经网络。神经元是大脑处理信息的基本单元，以细胞体为主体，由许多向周围延伸的不规则树枝状纤维构成的神经细胞，其形状很像一棵枯树的枝干。它主要由细胞体、树突、轴突和突触(Synapse，又称神经键)组成。如图 2.1 所示。

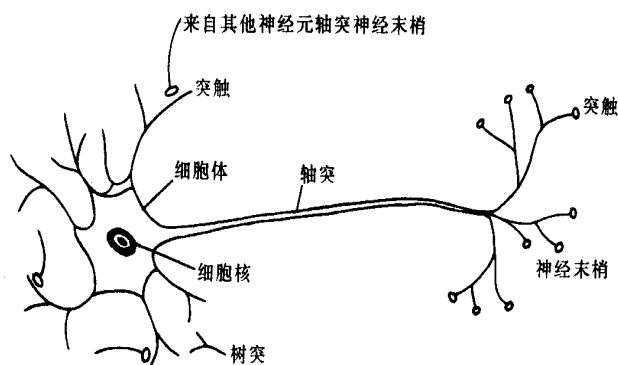


图 2.1 生物神经元

从神经元各组成部分的功能来看，信息的处理与传递主要发生在突触附近。当神经元细胞体通过轴突传到突触前膜的脉冲幅度达到一定强度，即超过其阈值电位后，突触前膜将向突触间隙释放神经传递的化学物质。

2.1.2 人工神经元

归纳一下生物神经元传递信息的过程：生物神经元是一个多输入、单输出单元。常用的人工神经元模型可用图 2.2 模拟。

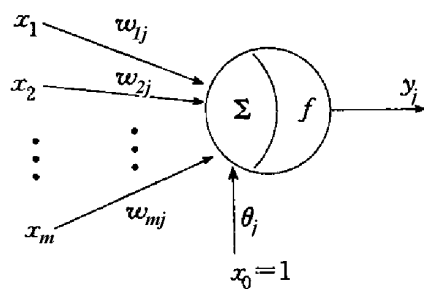


图 2.2 人工神经元（感知器）示意图

当神经元 j 有多个输入 $x_i (i=1, 2, \dots, m)$ 和单个输出 y_j 时，输入和输出的关系可表示为：

$$\begin{cases} s_j = \sum_{i=1}^m w_{ij} x_i - \theta_j \\ y_j = f(s_j) \end{cases} \quad (2.3)$$

其中 θ_j 为阈值， w_{ij} 为从神经元 i 到神经元 j 的连接权重因子， $f(\cdot)$ 为传递函数，或称激励函数。

2.1.3 人工神经网络的构成

神经元的模型确定之后，一个神经网络的特性及能力主要取决于网络的拓扑结构及学习方法。

神经网络连接的几种基本形式^[17]：

1) 前向网络

前向网络结构如图 2.4 所示，网络中的神经元是分层排列的，每个神经元只与前一层的神经元相连接。神经元分层排列，分别组成输入层、中间层（也称为隐含层，可以由若干层组成）和输出层。每一层的神经元只接受来自前一层神经元的输入，后面的层对前面的层没有信号反馈。输入模式经过各层次的顺序传播，最后在输出层上得到输出。感知器网络和 BP 网络均属于前向网络。

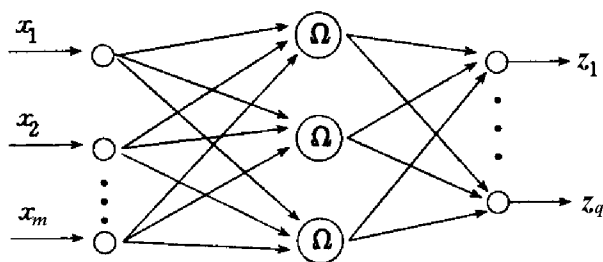


图 2.4 前向网络结构

2) 从输出到输入有反馈的前向网络

其结构如图 2.5 所示，输出层对输入层有信息反馈，这种网络可用于存储某种模式序列，如神经认知机和回归 BP 网络都属于这种类型。

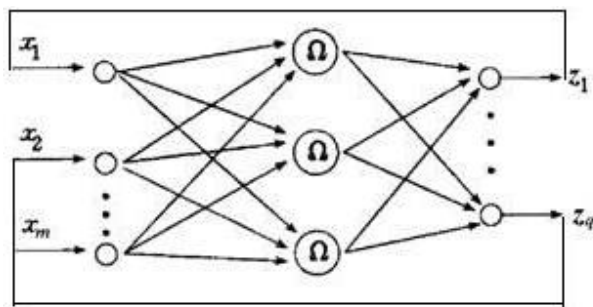


图 2.5 有反馈的前向网络结构

3) 层内互连前向网络

其结构如图 2.6 所示，通过层内神经元的相互结合，可以实现同一层神经元之间的横向抑制或兴奋机制。这样可以限制每层内可以同时动作的神经元素，或者把每层内的神经元分为若干组，让每一组作为一个整体进行运作。例如，可利用横向抑制机理把某层内的具有最大输出的神经元挑选出来，从而抑制其他神经元，使之处于无输出状态。

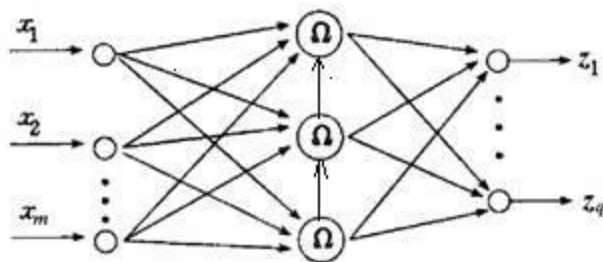


图 2.6 有相互结合的前向网络结构

4) 相互结合型网络

相互结合型网络结构如图 2.7 所示，这种网络在任意两个神经元之间都可能连接。**Hopfield** 网络和 **Boltzmann** 机均属于这种类型。在无反馈的前向网络中，信号一旦通过某神经元，该神经元的处理就结束了。而在相互结合网络中，信号要在神经元之间反复传递，网络处于一种不断变化状态的动态之中。信号从某初始状态开始，经过若干次变化，才会达到某种平衡状态。根据网络的结构和神经元的特性，网络的运行还有可能进入周期振荡或其他如混沌平衡状态。

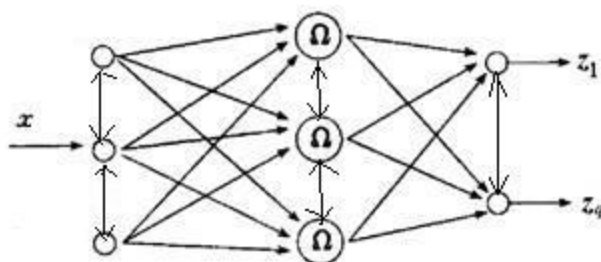


图 2.7 结合型网络结构

综上，可知神经网络有分层网络、层内连接的分层网络、反馈连接的分层网络、互连网络等四种结构，其神经网络模型有感知器网络，线性神经网络，BP 神经网络，径向基函数网络，反馈神经网络等，本文主要学习研究了 BP 神经网络，以及 BP 神经网络在函数逼近和样本含量估计两个实例中的应用分析。

2.2 BP 神经网络及其原理

2.2.1 BP 神经网络定义

BP (Back Propagation) 神经网络是一种神经网络学习算法。其由输入层、中间层、输出层组成的阶层型神经网络，中间层可扩展为多层。相邻层之间各神经元进行全连接，而每层各神经元之间无连接，网络按有教师示教的方式进行学习，当一对学习模式提供给网络后，各神经元获得网络的输入响应产生连接权值 (Weight)。然后按减小希望输出与实际输出误差的方向，从输出层经各中间层逐层修正各连接权，回到输入层。此过程反复交替进行，直至网络的全局误差趋向给定的极小值，即完成学习的过程。

2.2.2 BP 神经网络模型及其基本原理

BP神经网络是误差反向传播神经网络的简称,它由一个输入层,一个或多个隐含层和一个输出层构成,每一次由一定数量的的神经元构成。这些神经元如同人的神经细胞一样是互相关联的。其结构如图2.8所示:

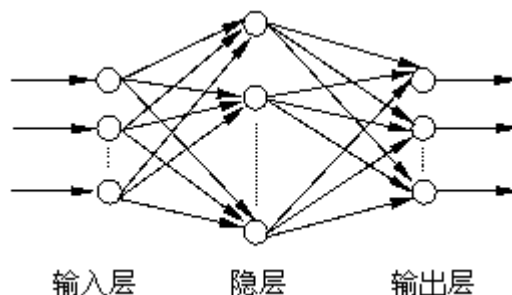


图2.8 BP神经网络模型

生物神经元信号的传递是通过突触进行的一个复杂的电化学等过程,在人工神经网络中是将其简化模拟成一组数字信号通过一定的学习规则而不断变动更新的过程,这组数字储存在神经元之间的连接权重。网络的输入层模拟的是神经系统中的感觉神经元,它接收输入样本信号。输入信号经输入层输入,通过隐含层的复杂计算由输出层输出,输出信号与期望输出相比较,若有误差,再将误差信号反向由输出层通过隐含层处理后向输入层传播。在这个过程中,误差通过梯度下降算法,分摊给各层的所有单元,从而获得各单元的误差信号,以此误差信号为依据修正各单元权值,网络权值因此被重新分布。此过程完成后,输入信号再次由输入层输入网络,重复上述过程。这种信号正向传播与误差反向传播的各层权值调整过程周而复始地进行着,直到网络输出的误差减少到可以接受的程度,或进行到预先设定的学习次数为止。权值不断调整的过程就是网络的学习训练过程。

BP神经网络的信息处理方式具有如下特点:

1) 信息分布存储。人脑存储信息的特点是利用突触效能的变化来调整存储内容,即信息存储在神经元之间的连接强度的分布上,BP神经网络模拟人脑的这一特点,使信息以连接权值的形式分布于整个网络。

2) 信息并行处理。人脑神经元之间传递脉冲信号的速度远低于冯·诺依曼计算机的工作速度,但是在很多问题上却可以做出快速的判断、决策和处理,这是由于人脑是一个大规模并行与串行组合的处理系统。BP神经网络的基本结构模仿人脑,具有并行处理的特征,大大提高了网络功能。

3) 具有容错性。生物神经系统部分不严重损伤并不影响整体功能,BP神经网络也具有这种特性,网络的高度连接意味着少量的误差可能不会产生严重的后果,部分神经元的损伤不破坏整体,它可以自动修正误差。这与现代计算机的脆弱性形成鲜明对比。

4) 具有自学习、自组织、自适应的能力。BP神经网络具有初步的自适应与自组织能力,在学习或训练中改变突触权值以适应环境,可以在使用过程中不断学习完善自己的功能,并且同一网络因学习方式的不同可以具有不同的功能,它甚至具有创新能力,可以发展知识,以至超过设计者原有的知识水平。

2.3 BP神经网络的主要功能

目前，在人工神经网络的实际应用中。绝大部分的神经网络模型都采用 BP 神经网络及其变化形式。它也是前向网络的核心部分，体现了人工神经网络的精华。

BP 网络主要用于以下四方面^[3]。

- (1) 函数逼近：用输入向量和相应的输出向量训练一个网络以逼近一个函数。
- (2) 模式识别：用一个待定的输出向量将它与输入向量联系起来。
- (3) 分类：把输入向量所定义的合适方式进行分类。
- (4) 数据压缩：减少输出向量维数以便传输或存储。

2.4 BP 网络的优点以及局限性

BP 神经网络最主要的优点是具有极强的非线性映射能力^[12]。理论上，对于一个三层和三层以上的 BP 网络，只要隐层神经元数目足够多，该网络就能以任意精度逼近一个非线性函数。其次，BP 神经网络具有对外界刺激和输入信息进行联想记忆的能力。这是因为它采用了分布并行的信息处理方式，对信息的提取必须采用联想的方式，才能将相关神经元全部调动起来。BP 神经网络通过预先存储信息和学习机制进行自适应训练，可以从不完整的信息和噪声干扰中恢复原始的完整信息。这种能力使其在图像复原、语言处理、模式识别等方面具有重要应用。再次，BP 神经网络对外界输入样本有很强的识别与分类能力。由于它具有强大的非线性处理能力，因此可以较好地进行非线性分类，解决了神经网络发展史上的非线性分类难题。另外，BP 神经网络具有优化计算能力。BP 神经网络本质上是一个非线性优化问题，它可以在已知的约束条件下，寻找一组参数组合，使该组合确定的目标函数达到最小。不过，其优化计算存在局部极小问题，必须通过改进完善^[13]。

由于 BP 网络训练中稳定性要求学习效率很小，所以梯度下降法使得训练很慢。动量法因为学习率的提高通常比单纯的梯度下降法要快一些，但在实际应用中还是速度不够，这两种方法通常只应用于递增训练。

多层神经网络可以应用于线性系统和非线性系统中，对于任意函数模拟逼近。当然，感知器和线性神经网络能够解决这类网络问题。但是，虽然理论上是可行的，但实际上 BP 网络并不一定总能求解。

对于非线性系统，选择合适的学习率是一个重要的问题。在线性网络中，学习率过大会导致训练过程不稳定。相反，学习率过小又会造成训练时间过长。和线性网络不同，对于非线性多层网络很难选择很好的学习率。对那些快速训练算法，缺省参数值基本上都是最有效的设置^[10]。

非线性网络的误差面比线性网络的误差面复杂得多，问题在于多层网络中非线性传递函数有多个局部最优解。寻优的过程与初始点的选择关系很大，初始点如果更靠近局部最优点，而不是全局最优点，就不会得到正确的结果，这也是多层网络无法得到最优解的一个原因。为了解决这个问题，在实际训练过程中，应重复选取多个初始点进行训练，以保证训练结果的全局最优性^[11]。

网络隐层神经元的数目也对网络有一定的影响。神经元数目太少会造成网络的不适性，而神经元数目太多又会引起网络的过适性。

3 BP 神经网络在实例中的应用

快速发展的 Matlab 软件为神经网络理论的实现提供了一种便利的仿真手段。Matlab 神经网络工具箱的出现，更加拓宽了神经网络的应用空间。神经网络工具箱将很多原本需要手动计算的工作交给计算机，一方面提高了工作效率，另一方面，还提高了计算的准确度和精度，减轻了工程人员的负担。

神经网络工具箱是在 MATLAB 环境下开发出来的许多工具箱之一。它以人工神经网络理论为基础，利用 MATLAB 编程语言构造出许多典型神经网络的框架和相关的函数。这些工具箱函数主要为两大部分。一部分函数特别针对某一种类型的神经网络的，如感知器的创建函数、BP 网络的训练函数等。而另外一部分函数则是通用的，几乎可以用于所有类型的神经网络，如神经网络仿真函数、初始化函数和训练函数等^[15]。这些函数的 MATLAB 实现，使得设计者对所选定网络进行计算过程，转变为对函数的调用和参数的选择，这样一来，网络设计人员可以根据自己的需要去调用工具箱中有关的设计和训练程序，从烦琐的编程中解脱出来，集中精力解决其他问题，从而提高了工作效率。

3.1 基于 MATLAB 的 BP 神经网络工具箱函数

最新版本的神经网络工具箱几乎涵盖了所有的神经网络的基本常用模型，如感知器和 BP 网络等。对于各种不同的网络模型，神经网络工具箱集成了多种学习算法，为用户提供了极大的方便^[16]。Matlab R2007 神经网络工具箱中包含了许多用于 BP 网络分析与设计的函数，BP 网络的常用函数如表 3.1 所示。

表 3.1 BP 网络的常用函数表

函数类型	函数名称	函数用途
前向网络创建函数	newcf	创建级联前向网络
	Newff	创建前向 BP 网络
传递函数	logsig	S 型的对数函数
	tansig	S 型的正切函数
	purelin	纯线性函数
学习函数	learngd	基于梯度下降法的学习函数
	learngdm	梯度下降动量学习函数
性能函数	mse	均方误差函数
	msereg	均方误差规范化函数
显示函数	plotperf	绘制网络的性能
	plotes	绘制一个单独神经元的误差曲面
	plotep	绘制权值和阈值在误差曲面上的位置
	errsurf	计算单个神经元的误差曲面

3.1.1 BP 网络创建函数

1) newff

该函数用于创建一个 BP 网络。调用格式为：

`net=newff`

`net=newff(PR, [S1 S2.. SN1], {TF1 TF2.. TFN1}, BTF, BLF, PF)`

其中，

`net=newff` 用于在对话框中创建一个 BP 网络。

`net` 为创建的新 BP 神经网络；

`PR` 为网络输入向量取值范围的矩阵；

`[S1 S2...SN1]` 表示网络隐含层和输出层神经元的个数；

`{TF1 TF2...TFN1}` 表示网络隐含层和输出层的传输函数，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` 函数用于创建级联前向 BP 网络，`newfftd` 函数用于创建一个存在输入延迟的前向网络。

3.1.2 神经元上的传递函数

传递函数是 BP 网络的重要组成部分。传递函数又称为激活函数，必须是连续可微的。BP 网络经常采用 S 型的对数或正切函数和线性函数。

1) `logsig`

该传递函数为 S 型的对数函数。调用格式为：

`A=logsig (N)`

`info=logsig (code)`

其中，

`N`：Q 个 S 维的输入列向量；

`A`：函数返回值，位于区间 (0, 1) 中；

2) `tansig`

该函数为双曲正切 S 型传递函数。调用格式为：

`A=tansig (N)`

`info=tansig (code)`

其中，

`N`：Q 个 S 维的输入列向量；

`A`：函数返回值，位于区间 (-1, 1) 之间。

3) `purelin`

该函数为线性传递函数。调用格式为：

`A=purelin (N)`

`info=purelin (code)`

其中，

N: Q 个 S 维的输入列向量；

A: 函数返回值，A=N。

3.1.3 BP 网络学习函数

1) learngd

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效率，来计算权值或阈值的变化率。调用格式为：

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

```
[db, ls]=learngd(b, ones(1, Q), Z, N, A, T, E, gW, gA, D, LP, LS)
```

```
info=learngd(code)
```

2) 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。

3.1.4 BP 网络训练函数

1) train

神经网络训练函数，调用其他训练函数，对网络进行训练。该函数的调用格式为：

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

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

2) traingd 函数为梯度下降 BP 算法函数。traingdm 函数为梯度下降动量 BP 算法函数。

3.2 BP 网络在函数逼近中的应用

3.2.1 问题的提出

BP 网络由很强的映射能力，主要用于模式识别分类、函数逼近、函数压缩等。下面将通过实例来说明 BP 网络在函数逼近方面的应用。

要求设计一个 BP 网络，逼近以下函数： $g(x)=1+\sin(k\pi/4\cdot x)$ ，实现对该非线性函数的逼近。其中，分别令 $k=1, 2, 4$ 进行仿真，通过调节参数（如隐藏层节点个数等）得出信号的频率与隐层节点之间，隐层节点与函数逼近能力之间的关系。

3.2.2 基于 BP 神经网络逼近函数

步骤 1：假设频率参数 $k=1$ ，绘制要逼近的非线性函数的曲线。函数的曲线如图 3.2 所示

```
k=1;
p=[-1:.05:8];
t=1+sin(k*pi/4*p);
plot(p, t, '-');
title('要逼近的非线性函数');
xlabel('时间');
ylabel('非线性函数');
```

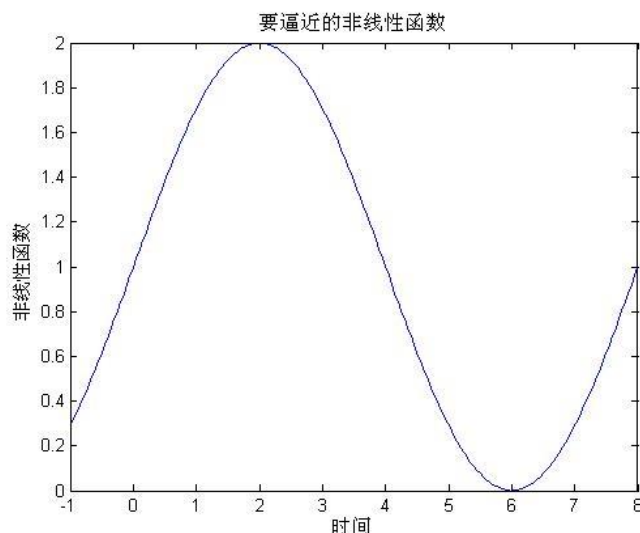


图 3.2 要逼近的非线性函数曲线

步骤 2：网络的建立

应用 `newff()` 函数建立 BP 网络结构。隐层神经元数目 n 可以改变，暂设为 $n=3$ ，输出层有一个神经元。选择隐层和输出层神经元传递函数分别为 `tansig` 函数和 `purelin` 函数，网络训练的算法采用 Levenberg - Marquardt 算法 `trainlm`。

```
n=3;
net = newff(minmax(p), [n, 1], {'tansig' 'purelin'}, 'trainlm');
对于初始网络，可以应用 sim() 函数观察网络输出。
y1=sim(net, p);
figure;
plot(p, t, '- ', p, y1, ':');
title('未训练网络的输出结果');
xlabel('时间');
ylabel('仿真输出--原函数');
```

同时绘制网络输出曲线，并与原函数相比较，结果如图 3.3 所示。

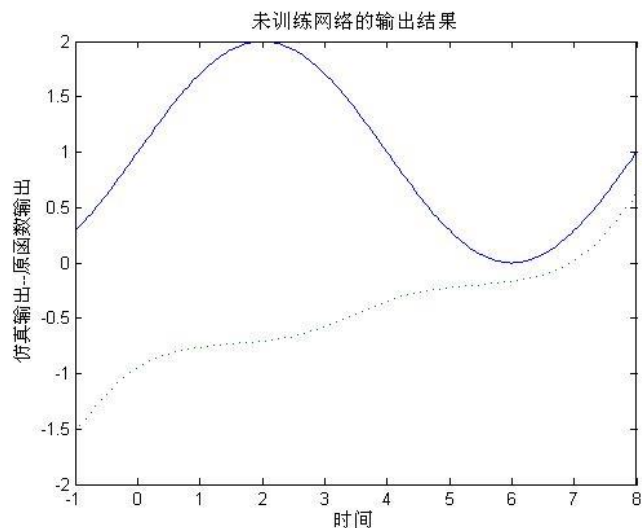


图 3.3 未训练网络的输出结果

其中 “——” 代表要逼近的非线性函数曲线；

“.....” 代表未经训练的函数曲线；

因为使用 `newff()` 函数建立函数网络时，权值和阈值的初始化是随机的，所以网络输出结构很差，根本达不到函数逼近的目的，每次运行的结果也有时不同。

步骤 3：网络训练

应用 `train()` 函数对网络进行训练之前，需要预先设置网络训练参数。将训练时间设置为 50，训练精度设置为 0.01，其余参数使用缺省值。训练后得到的误差变化过程如图 3.4 所示。

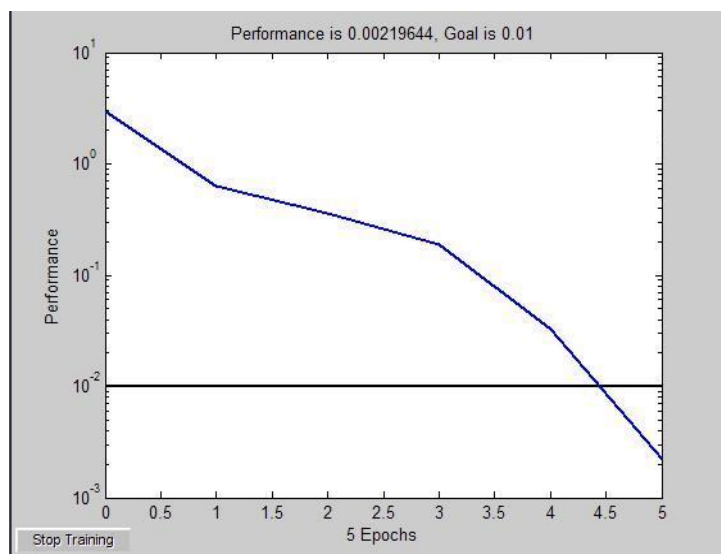


图 3.4 训练过程

`net.trainParam.epochs=50;` (网络训练时间设置为 50)

`net.trainParam.goal=0.01;` (网络训练精度设置为 0.01)

`net=train(net, p, t);` (开始训练网络)

TRAINLM-calcjx, Epoch 0/50, MSE 9.27774/0.01, Gradient 13.3122/1e-010

TRAINLM-calcjx, Epoch 3/50, MSE 0.00127047/0.01, Gradient 0.0337555/1e-010

TRAINLM, Performance goal met.

从以上结果可以看出，网络训练速度很快，经过一次循环迭送过程就达到了要求的精度 0.01。

步骤 4： 网络测试

对于训练好的网络进行仿真：

`y2=sim(net, p);`

`figure;`

`plot(p, t, '-', p, y1, ':', p, y2, '--')`

`title('训练后网络的输出结果');`

`xlabel('时间');`

`ylabel('仿真输出');`

绘制网络输出曲线，并与原始非线性函数曲线以及未训练网络的输出结果曲线相比较，比较出来的结果如图 3.5 所示。

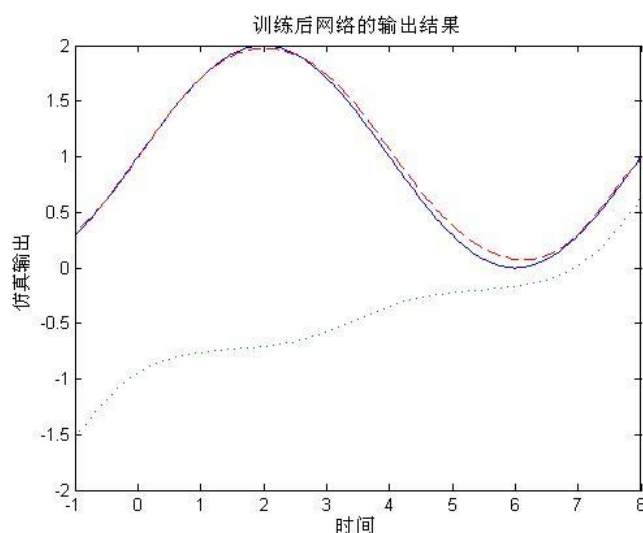


图 3.5 训练后网络的输出结果

其中 “——” 代表要逼近的非线性函数曲线；

“.....” 代表未经训练的函数曲线；

“——” 代表经过训练的函数曲线；

从图中可以看出，得到的曲线和原始的非线性函数曲线很接近。这说明经过训练后，BP 网络对非线性函数的逼近效果比较好。

3.2.3 不同频率下的逼近效果

改变非线性函数的频率和 BP 函数隐层神经元的数目，对于函数逼近的效果有一定的影响。网络非线性程度越高，对于 BP 网络的要求越高，则相同的网络逼近效果要差一些；隐层神经元的数目对于网络逼近效果也有一定影响，一般来说隐层神经元数目越多，则 BP 网络逼近非线性函数的能力越强。

下面通过改变频率参数和非线性函数的隐层神经元数目来加以比较证明。

(1) 频率参数设为 $k=2$ ，当隐层神经元数目分别取 $n=3$ 、 $n=6$ 时，得到了训练后的网络输出结果如图 3.6，3.7 所示。

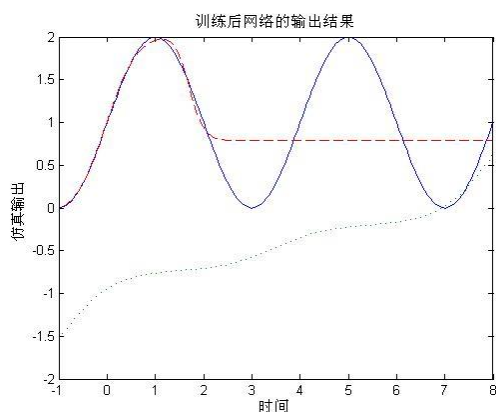


图 3.6 当 $n=3$ 时训练后网络的输出结果

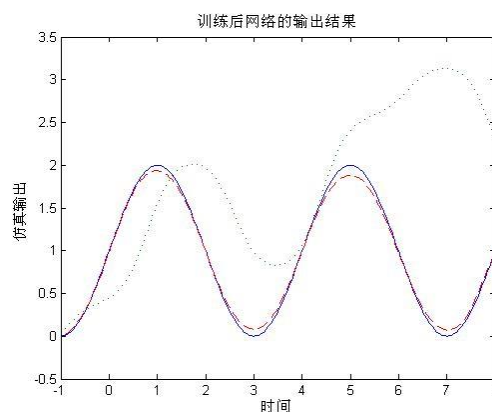


图 3.7 当 $n=6$ 时训练后网络的输出结果

其中 “——” 代表要逼近的非线性函数曲线；

“.....” 代表未经训练的函数曲线；

“——” 代表经过训练的函数曲线；

(2) 频率参数设为 $k=4$ ，当隐层神经元数目分别取 $n=6$ 、 $n=8$ 时，得到了训练后的网络输出结果如图 3.8，3.9 所示。

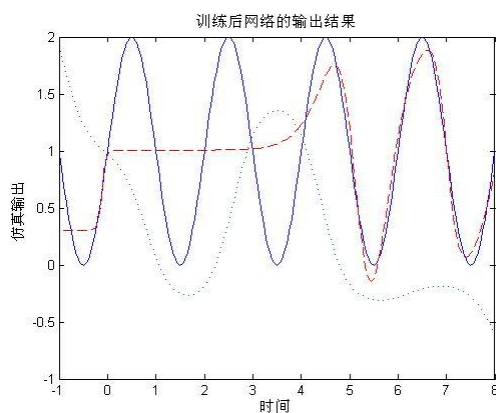


图 3.8 当 $n=6$ 时训练后网络的输出结果

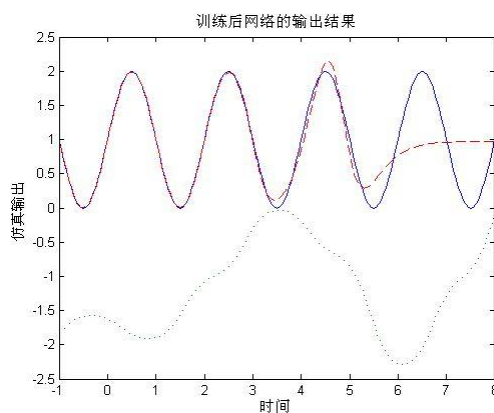


图 3.9 当 $n=8$ 时训练后网络的输出结果

其中 “——” 代表要逼近的非线性函数曲线；

“.....” 代表未经训练的函数曲线;

“——” 代表经过训练的函数曲线;

(2) 频率参数设为 $k=8$, 当隐层神经元数目分别取 $n=10$ 、 $n=15$ 时, 得到了训练后的网络输出结果如图 3.10, 3.11 所示。

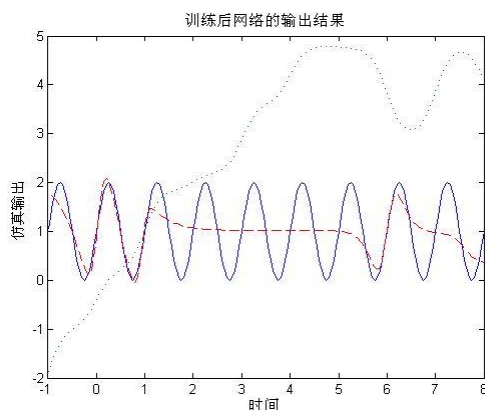


图 3.10 当 $n=10$ 时训练后网络的输出结果

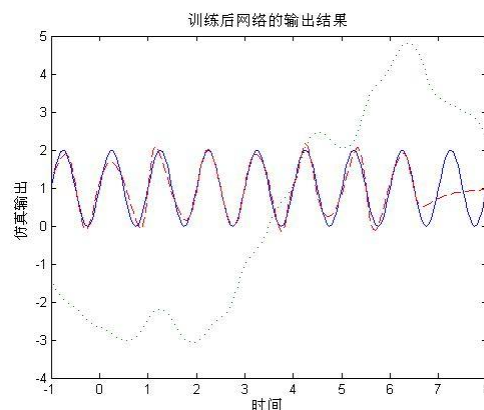


图 3.11 当 $n=15$ 时训练后网络的输出结果

其中 “——” 代表要逼近的非线性函数曲线;

“.....” 代表未经训练的函数曲线;

“——” 代表经过训练的函数曲线;

3.2.4 讨论

通过上述仿真结果可知, 当 $k=1$, $n=3$ 时; $k=2$, $n=6$ 时; $k=4$, $n=8$ 时; $k=8$, $n=15$ 时, BP 神经网络分别对函数取得了较好的逼近效果。由此可见, n 取不同的值对函数逼近的效果有很大的影响。改变 BP 网络隐层神经元的数目, 可以改变 BP 神经网络对于函数的逼近效果。隐层神经元数目越多, 则 BP 网络逼近非线性函数的能力越强。

3.3 BP 网络在样本含量估计中的应用

3.3.1 问题的提出

这是一个神经网络在医学中应用的例子。拟设计一台仪器, 通过对血液样本进行光谱分析来测试血清中胆固醇水平。共采集了 264 位病人的血液样本, 对其光谱分析共发现 21 种光谱波长。对这些病人, 通过血清分离, 同样也测量了 hdl、ldl、vldl 胆固醇水平。

(1) 将数据导入 Matlab 工作空间, 进行主要成分的分析。数据文件为 choles_all.mat。

load choles_all %matlab 中有一个 choles_all.mat 文件它包含了本问题需要的原始数据

[pn, meanp, stdp, tn, meant, stdt]=prestd(p, t); % prestd 函数对样本数据作归一化处理

[ptrans, sransMat]=prepca(pn, 0.001); % 利用 prepca 函数对归一化后的样本数据进行主元分析

这里剔除了一些数据，只保留了所占 99.9% 的主要成分数据。下面检验一下转换后数据矩阵的大小。

```
[R, Q]=size(ptrans)
```

```
R =
```

```
4
```

```
Q =
```

```
264
```

从中可以看出通过主要成分分析，输入数据从 21 组减少到 4 组，由此可见原始数据有着很大的冗余度。

(2) 将这些数据分成几个部分，分别用于训练、验证和测试。将数据的 1/4 用于验证、1/4 用于测试，其余的用于训练网络。采用等间隔的方式在原始数据中抽取出这些数据。

```
iitst=2:4:Q;
```

```
iival=4:4:Q;
```

```
iitr=[1:4:Q 3:4:Q];
```

```
val.P=ptrans(:, iival); val.T=tn(:, iival);
```

```
test.P=ptrans(:, iitst); test.T=tn(:, iitst);
```

```
ptr=ptrans(:, iitr); ttr=tn(:, iitr);
```

3.3.2 基于 BP 神经网络对胆固醇含量估计

步骤一：网络建立

一切准备工作做好后，开始建立网络。在本例中，建立一个两层网络，隐层的传递函数采用 `tan-sigmoid`，输出层采用线性传递函数。隐层神经元初步设计为五个，因为需要得到三个目标，网络输出层设计为三个神经元。

```
net=newff(minmax(ptr), [5 3], {'tansig' 'purelin'}, 'trainlm');
```

步骤二：网络训练

进行网络训练，其训练过程如图 3.12 所示。

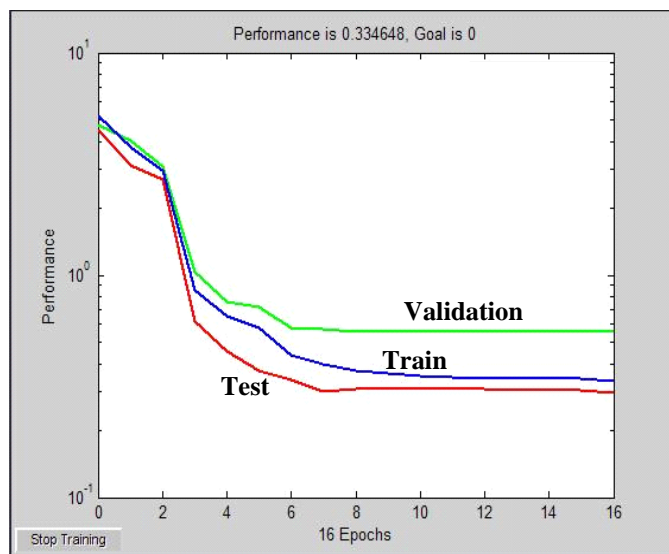


图 3.12 训练过程

```
net.trainParam.show=5;
[net,tr]=train(net,ptr,ttr,[],[],val,test);
```

```
TRAINLM-calcjx, Epoch 0/100, MSE 5.23521/0, Gradient 5.13228/1e-010
TRAINLM-calcjx, Epoch 5/100, MSE 0.578625/0, Gradient 1.62833/1e-010
TRAINLM-calcjx, Epoch 10/100, MSE 0.352352/0, Gradient 0.505629/1e-010
TRAINLM-calcjx, Epoch 15/100, MSE 0.340796/0, Gradient 0.404806/1e-010
TRAINLM-calcjx, Epoch 16/100, MSE 0.334648/0, Gradient 0.513826/1e-010
TRAINLM, Validation stop.
```

可以看出，经过 20 次训练后，由于验证误差的增加，训练停止。将训练误差，验证误差和测试误差绘制在一副图中，如图 3.13 所示，这样就可以更加直观地观察训练过程。

```
plot(tr.epoch, tr.perf, '- ', tr.epoch, tr.vperf, ': ', tr.epoch, tr.tperf, '-. ');
legend(' Training', ' Validation', ' Test', -1);
ylabel(' 平方差 ');
xlabel(' 时间 ')
```

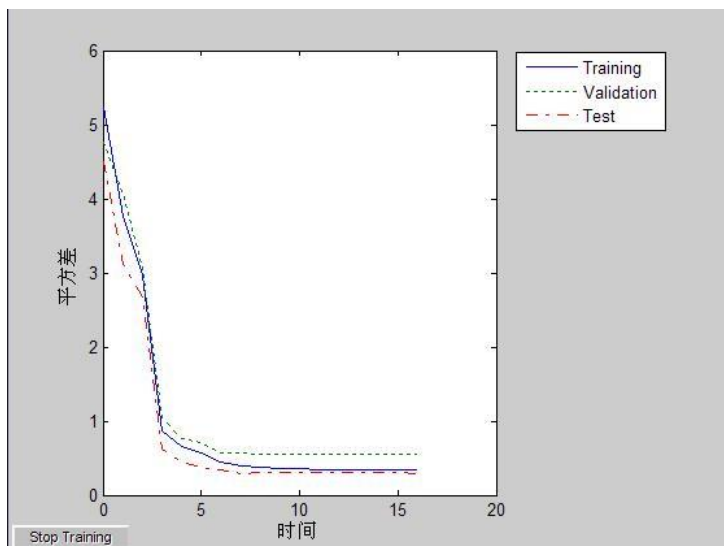
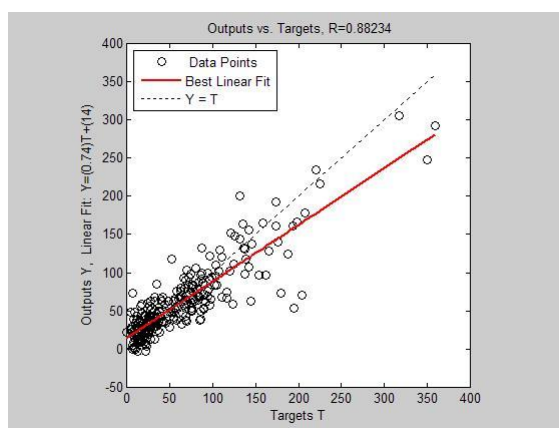


图 3.13 训练误差、验证误差、测试误差曲线

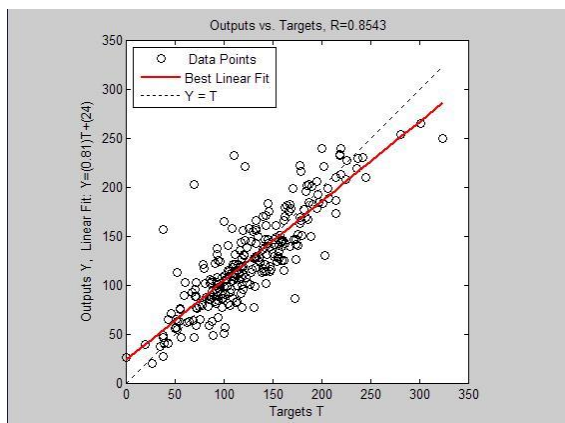
步骤三：输出结果

最后，对网络响应进行分析，将所有数据都放在整个数据集中，包括训练数据、验证数据和测试数据，然后网络输出和相应的期望输出向量进行线性回归分析。线性回归之前需要对网络输出进行反规范转换，因为有三组输出值，所以应进行三次线性回归。其结果如图 3.14 所示。

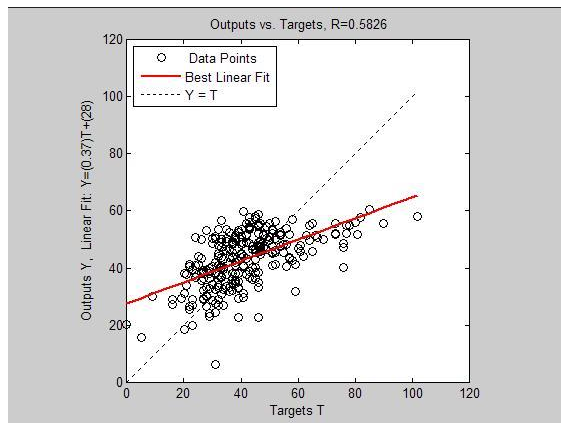
```
an=sim(net,ptrans);
a=poststd(an,meant,stdt);
for i=1:3
figure(i)
[m(i),b(i),r(i)]=postreg(a(i,:),t(i,:)); %postreg()将三组输出进行线性回归分析
End
```



(a)



(b)



(c)

图 3.14 线性回归结果

(a) hd1 线性回归; (b) 1d1 线性回归; (c) vhd1 线性回归

3.3.3 不同条件下的输出结果

对 BP 网络型进行改进, 将隐层神经元数目加为 15 个, 观察其输出结果其训练过程如图 3.16 所示。三种误差曲线如图 3.17 所示, 改进后的线性回归结果如图 3.15 所示。

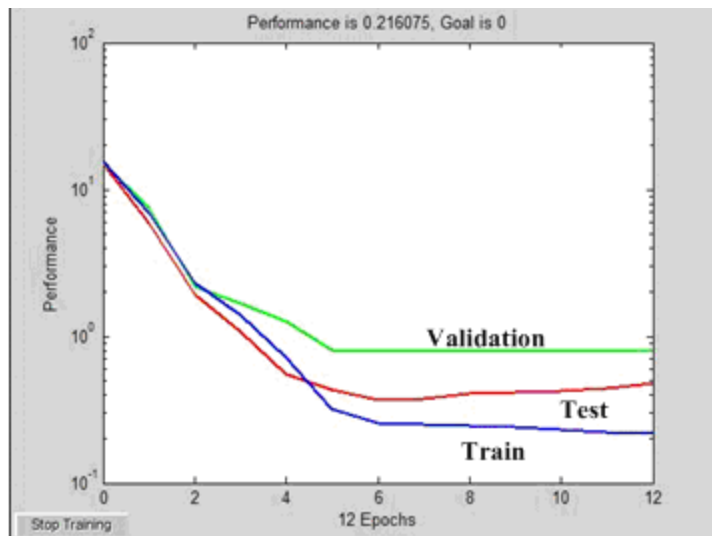


图 3.16 训练过程

```
net=newff(minmax(ptr),[20 3],{'tansig' 'purelin'},'trainlm');
[net,tr]=train(net,ptr,ttr,[],[],val,test);
```

TRAINLM-calcjx, Epoch 0/100, MSE 15.4753/0, Gradient 16.3115/1e-010

TRAINLM-calcjx, Epoch 12/100, MSE 0.216075/0, Gradient 0.204225/1e-010

TRAINLM, Validation stop.

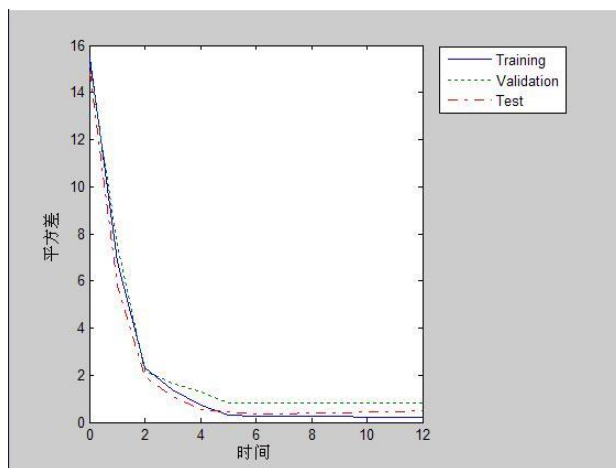
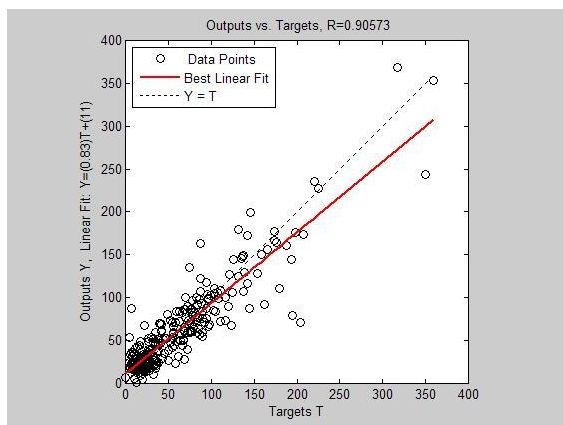


图 3.17 三种误差曲线

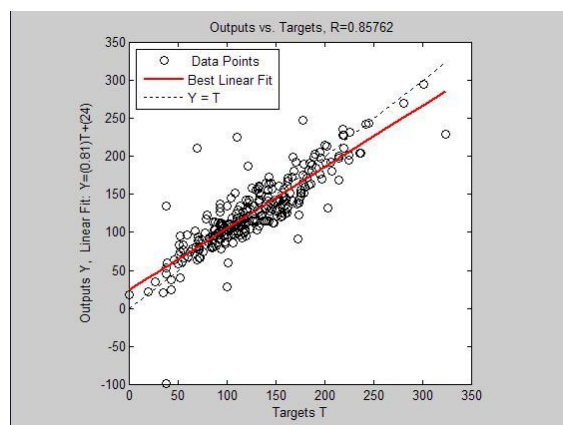
```

plot(tr.epoch,tr.perf,'-',tr.epoch,tr.vperf,'-',tr.epoch,tr.tperf,'-')
legend('Training','Validation','Test',-1);
ylabel('平方差');
xlabel('时间')
plot(tr.epoch,tr.perf,'-',tr.epoch,tr.vperf,'-',tr.epoch,tr.tperf,'-')
an=sim(net,ptrans);
a=poststd(an,meant,stdt);
for i=1:3
figure(i)
[m(i),b(i),r(i)]=postreg(a(i,:),t(i,:));
end
[m(1),b(1),r(1)]=postreg(a(1,:),t(1,:));
    
```

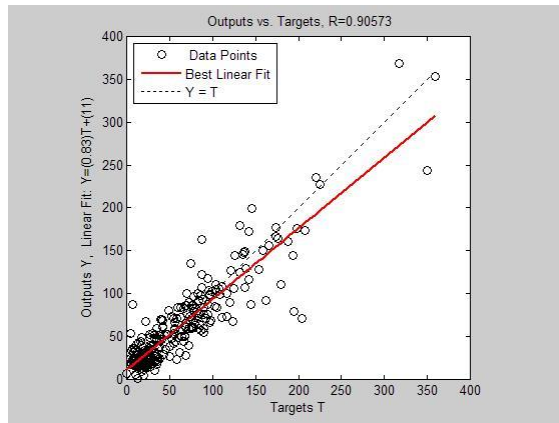
同样将所有数据都放在整个数据集中，包括训练数据、验证数据和测试数据，然后网络输出和相应的期望输出向量进行线性回归分析。最后线性回归的输出结果如图 3.15 所示。



(a)



(b)



(c)

图 3.15 线性回归结果

(a) hdl 线性回归; (b) ld1 线性回归; (c) vhd1 线性回归

3.3.4 讨论

综上两种情况比较下，其中当隐层神经元为五个时，前面两个输出期望值的跟踪较好，相应的 R 值几乎达到了 0.9，而第三个输出则吻合得不是很理想，作进一步研究，在当隐层数据为 15 时，较第一种情况，它的输出更加接近于期望值。比较可知，当隐层数目越多，则测试得到的样本水平越接近于期望值。

4 结束语

本文首先总结了神经网络的研究目的和意义，介绍了神经网络的研究背景和现状，分析了目前神经网络研究中存在的问题，然后描述了 BP 神经网络算法的实现以及 BP 神经网络的工作原理，给出了 BP 网络的局限性。在以 BP 神经网络为基础的前提下，分析研究了 BP 神经网络在函数逼近和样本含量估计两个实例中的应用。以及分析了结论，即信号的频率越高，则隐层节点越多，隐单元个数越多，逼近能力越强。和隐层数目越多，测试得到的样本的水平越接近于期望值。

本文虽然总结分析了 BP 神经网络算法的实现，给出了实例分析，但是还有很多的不足。本文所总结的 BP 神经网络和目前研究的现状都还不够全面，经过程序调试的图形有可能都还存在很多细节上的问题，而图形曲线所实现效果都还不够好，以及结果分析不够全面，正确，缺乏科学性等，这些都还是需加强提高的，本文的完成不代表就对这门学科研究的结束，还有很多知识，理论，研究成果需要不断学习，

近几年的不断发展，神经网络更是取得了非常广泛的应用，和令人瞩目的发展，在很多方面都发挥了其独特的作用，特别是在人工智能、自动控制、计算机科学、信息处理、机器人、模式识别等众多方面的应用实例，给人们带来了应用上到思考，和解决方法的研究。但是神经网络的研究最近几年还没有达到非常热门的阶段，这还需有很多热爱神经网络和研究神经网络人员的不断研究和创新，在科技高度发达

的现在，我们有理由期待，也有理由相信。我想在不久的将来神经网络会应用到更多更广的方面，人们的生活会更加便捷。

参考文献:

- [1] 郑君里,杨行峻.《人工神经网络》. 北京: 高等教育出版社,1992. 5:15-30
- [2] 郝中华.《B P神经网络的非线性思想》. 洛阳师范学院学报2008. 3(4)
- [3] 巨军让,卓戎.《B P神经网络在 Matlab 中的方便实现》.新疆石油学院学报. 2008. 2(1)
- [4] 蒋宗礼.《人工神经网络导论》. 高等教育出版社, 2001. 5:15-90
- [5] 闻新、周露、王丹力、熊晓英.《MATLAB 神经网络应用设计. 科学出版社》, 2001. 5:10-50
- [6] 葛哲学、孙志强编著 .《神经网络与 matlab2007 实现》. 北京: 电子工业出版社, 2007. 9:1-5
- [7] 董长虹 编著 .《Matlab 神经网络与应用》. 北京: 国防工业出版社, 2007. 9:1-13
- [8] 胡守仁, 等.《神经网络导论》.长沙: 国防科技大学出版社, 1993:23-45
- [9] 张玲, 张钊.《人工神经网络理及应用》. 浙江: 浙江科技大学出版社, 1997. 5:20-62
- [10] Neural Network Tolbox User' s, The MathWorks. Inc,2003:16-27
- [11] Matlab user' s Guide. The MathWorks. Inc,2003:23-27
- [12] Neural Network Toolbox. Mathworks,2007:1-13
- [13] Philip D. Wasserman, Neural Computing: Theory and Practice, Van Nostrand Reinhold, 1989.
- [14] 李学桥.《神经网络工程应用》. 重庆: 重庆大学出版社, 1995. 6:20-73
- [15] 楼顺天, 等.《基于 Matlab 的系统分析与设计---神经网络》. 西安: 西安电子科技大学出版社, 2000. 8:23-40
- [16] 周开利, 康耀红 编著.《神经网络模型及其 MATLAB 仿真程序设计》. 2006:10-43
- [17] 魏海坤.《神经网络结构设计的理论与方法》. 国防工业出版社, 2005. 9:20-30

BP Neural Network And Application based on MATLAB

Jiang Liangliang

School of Binjiang, Nanjing University of Information Science and Technology, Nanjing 210044

ABSTRACT:

First, the research purpose and significance of neural network is expound in this article. Commentary studies current situation at the problem home and abroad. Leads to the existing problems, and then have analysed algorithmic basal principle of neural networks, Give algorithmic concrete of classics neural networks out the realization method. Summing up the characteristics of neural network algorithm. Neural network algorithm is given the basic processes. The arithmetic of BP neural network is realized in Matlab software. The algorithm applies of BP neural networks to the function approximation problem and compute the swatch content. And analysis of relevant parameters on the results of algorithm. Finally, The BP neural network algorithm is Outlook.

Key word: neural network ; BP neural network; function approximation

