

机电工程师继续教育丛书

人工神经网络技术及应用

陈祥光 裴旭东 编著

33



中国电力出版社

www.cepp.com.cn

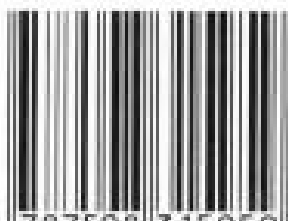
责任编辑: 龚广杰 王卫华

CEPP



- 现代制造企业信息化技术
- 模糊控制技术
- 人工神经网络技术及应用
- 现场总线技术
- 可编程序控制器应用技术
- 传感器与测控技术
- 电力电子与运动控制技术

ISBN 7-5083-1595-2



9 787508 315959 >

ISBN 7-5083-1595-2

定价: 14.00 元

CHINA ELECTRIC POWER PRESS

—— 机电工程师继续教育丛书 ——

人工神经网络技术及应用

陈祥光 裴旭东 编著



中国电力出版社
www.cepp.com.cn

内 容 提 要

本书为机电工程师继续教育丛书之一。本书以工程应用为背景,系统地阐述常用人工神经网络的基本原理、学习算法及分析方法,全书共分六章。第一章简述人工神经元的发展及一些领域的应用状况;第二章介绍感知器网络结构、算法及其应用;第三章讨论BP网络结构、算法改进及其应用实例;第四章介绍RBF网络结构、算法及其应用;第五章介绍Hopfield反馈网络的学习规则、稳定性及其应用;第六章介绍自组织人工神经网络的结构、基本学习算法及其应用。本书结合实际应用,适用性强。本书精选了一些工程实例,对读者掌握人工神经网络的理论和应用技术很有帮助。

本书可作为工程技术人员、科研人员及大专院校高年级学生的参考书。

图书在版编目(CIP)数据

人工神经网络技术及应用/陈祥光,裴旭东编著.北京:中国电力出版社,2003

(机电工程师继续教育丛书)

ISBN 7-5083-1595-2

I. 人… II. ①陈… ②裴… III. 人工神经元网络
IV. TP183

中国版本图书馆CIP数据核字(2003)第052993号

中国电力出版社出版、发行

(北京三里河路6号 100044 <http://www.cepp.com.cn>)

利森达印刷厂印刷

各地新华书店经售

*

2003年9月第一版 2003年9月北京第一次印刷

787毫米×1092毫米 16开本 8.5印张 189千字

印数0001—3000册 定价14.00元

版 权 专 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)



编委会名单

名誉主任 丁舜年

主 任 沈烈初

副主任 周鹤良 苏竹荆

委 员 (按姓氏笔画为序)

万遇良 马 阳 王 忱 王赞基

齐智平 刘玉琛 吴石增 肖 兰

杨昌焜 陈 瑜 陈祥光 张林昌

赵长德 郝广发 夏德海 廉小亲

主 编 万遇良

副主编 王 忱



序 言

中国电工技术学会电气工程师进修学院组织编写的《机电工程师继续教育丛书》第一套(6册)是以机电一体化技术为核心内容的,于1998年出版。这套丛书出版后深受广大读者欢迎,多次印刷,总印数超过一万套。此套丛书除供广大科技人员自学外,还曾被国家人事部列入《百千万人才工程》高级研修班计划,并由中国电工技术学会以及新疆、武汉等自治区、市有关部门作为系统培训科技人员的教材。据不完全统计,全国参加培训班的科技人员近万人。

信息化带动工业化、工业化促进信息化是新兴工业化道路重要特征之一,而智能化技术又是电工技术的重要发展方向。为了配合我国经济发展和工程建设的需求,根据国家“十五”规划的要求,电气工程师进修学院现编辑出版第二套丛书。该套丛书突出信息化技术,以高新技术改造提升制造业、推动电工行业技术进步与可持续发展的关键技术和共性技术。这套丛书共7册,包括了《传感器与测控技术》,《电力电子与运动控制技术》,《可编程序控制器技术》和《现场总线技术》,正在发展的智能控制技术的两个分支——《模糊控制技术》和《人工神经网络技术及应用》,以及涉及技术和管理的《现代制造企业信息化技术》。

该套丛书和第一套丛书一样,力求达到以下几点要求:

(1) 力争先进。本套丛书的内容跟踪国内外信息化技术的发展与进步,反映其发展现状和趋势。

(2) 力求实用。本套丛书主要适用于具有大专以上专业水平的在职科技人员的继续教育和知识更新,也可用作高等院校学生和相关专业研究生的参考用书。丛书内容理论联系实际,能够指导读者在需要时运用到实际工作中去。

(3) 内容简明。每种书集中阐述一个问题,简明扼要。

(4) 学习方式灵活。既可用于自学,也可用于面授。每种书除包括基本内容外,还包括一些习题,以及必要的参考书目,以帮助读者理解掌握和深入钻研。

编写出版这套丛书是一个新的尝试,不可避免地会存在问题和缺点,热切希望广大读者给予支持,更欢迎给予批评指正。

沈维初

2003年6月



前 言

人工神经网络是基于模仿人类大脑的结构和功能而构成的一种信息处理系统，具有很多与人类智能相似的特点。人工神经网络是一门高度综合的交叉学科，目前已经渗透到各个领域，尤其在模式识别、知识处理、传感技术、控制工程、电力工程、化学工程、环境工程、生物工程以及机器人研制等方面均有成功的应用，因此，其应用前景非常广泛。

本书是“机电工程师继续教育丛书”之一，本书汲取了国内、外文献和著作的精华部分，结合编者多年的教学、科研实践经验编写的。

本书系统地阐述人工神经网络的基本原理、网络结构和学习算法，着重介绍神经网络的分析方法和应用技术。本书包括了目前常用的人工神经网络。在结构安排上，即侧重于目前最常用的 BP 网络，又较系统地介绍了 RBF 网络、Hopfield 反馈网络及自组织人工神经网络。在内容处理上，对于基本概念阐述力求深入浅出、清晰易懂，突出网络结构、学习算法和工程应用，注重理论与实际应用相结合，便于读者自学。书中精选了一些工程实例和科研成果，有助于读者掌握人工神经网络的理论和应用技术。

本书在编写过程中，北京电气工程师进修学院的万遇良教授、王忱教授给予了大力支持和帮助，在此深表谢意。

限于作者水平，书中疏漏和错误在所难免，恳请读者指评指正。

编者

2003 年 4 月



《机电工程师继续教育丛书》

目 录

序言

前言

第一章 概述	1
第一节 人工神经网络的发展简史	1
第二节 生物神经元系统	2
第三节 人工神经网络模型	3
第四节 人工神经网络的应用领域	6
第二章 感知器	9
第一节 单层感知器网络	9
第二节 多层感知器网络	11
第三节 感知器的学习算法	14
第四节 感知器应用举例	15
第五节 本章小结	17
第三章 BP 网络	19
第一节 BP 神经元模型	19
第二节 BP 网络结构	21
第三节 BP 学习算法	22
第四节 BP 算法的改进	28
第五节 BP 网络的应用	31
第六节 本章小结	68
第四章 RBF 网络	70
第一节 RBF 神经元模型	70
第二节 RBF 网络结构	71
第三节 RBF 学习算法	72
第四节 RBF 网络的应用	74
第五节 本章小结	84
第五章 Hopfield 反馈网络	85
第一节 离散型 Hopfield 网络	85

第二节	离散型 Hopfield 网络的应用	97
第三节	连续型 Hopfield 网络	102
第四节	连续型 Hopfield 网络的应用	107
第五节	本章小结	110
第六章	自组织人工神经网络	111
第一节	竞争学习的基本原理	111
第二节	几种基本学习算法	112
第三节	自组织特征映射网络	115
第四节	自适应共振理论网络	117
第五节	自组织网络应用举例	121
第六节	本章小结	126
参考文献	127

第一章

概述

第一节 人工神经网络的发展简史

人脑是宇宙中已知的最复杂、最完善和最有效的信息处理系统，是生物进化的最高级产物，是目前任何机器无法比拟的智能系统。因此，科学家们从仿制人脑神经系统的结构和功能出发，研究人类智能活动和认识现象，尽管迄今为止，大脑的奥秘还在继续研究中，但从初级的模型中已取得很大的成果。

1943年，美国心理学家麦卡洛克（W. McCulloch）和数学家皮茨（W. Pitts）根据生物神经元的基本特性，发表了M-P神经元模型，开创了人工神经元研究的新纪元，M-P神经元模型如图1-1所示。其数学表达式为

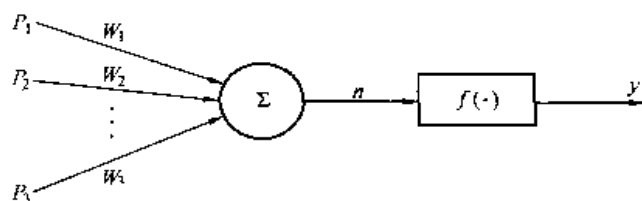


图 1-1 M-P 神经元模型

$$y_i = f\left(\sum_{j=1}^m w_{ij}P_j - \theta_i\right) \quad (1-1)$$

式中 y_i ——输出；

P_j ——第 j 个输入；

θ_i —— i 个神经元的阈值；

$f(\cdot)$ ——激活传递函数；

w_{ij} ——连接权值。

M-P 神经元模型首次用简单的数学模型模仿出生物系统神经元的活动功能，并通过数学关系式表达了神经元相互之间的连接关系。

1957年，美国计算机学家罗森布拉特（F. Rosenblatt）提出了著名的感知器（perceptron）模型，此模型类似于 M-P 模型，但连接权值 w_{ij} 可变的，因而神经网络具有学习功能，即在一定的输入值或（样本）情况下，通过搜索方式改变连接权值以便获得一组适当的连接权值，即经过训练可以达到对一定的输入向量模式进行分类和识别的目的。

1969年，美国麻省理工学院人工智能专家明斯基（M. Minsky）与帕伯特（S. Papert）在其合著的《感知器》一书中指出：单层感知器只能进行现行分类，不能解决异或问题，

也不能解决非线性问题。由于当时的神经网络技术确实存在这一问题。这结论的提出使得当时许多神经网络研究人员感到前途渺茫,以使人工神经网络在随后的10年左右处于萧条状态。

1982年,美国物理学家霍普菲尔德(J. Hopfield)提出了HNN全互连网络模型 $y_i = \text{sgn}(\sum_{j=1}^N w_{ij} \rho_j - \theta_i)$ 以及能量函数 $E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} y_i \rho_j + \sum_{i=1}^N \theta_i y_i$, 利用搜索算法以及能量函数最小化,适当确定 w_{ij} 等,可求解组合优化问题。此外,霍普菲尔德网络模型可以用电子模拟线路来实现,这种神经网络所执行的运算在本质上不同于布尔代数运算,从而兴起了对新一代电子神经计算机的研究热潮。

1986年,鲁默哈特(D. Rumelhart)与麦勒兰德(J. McClelland)发展了BP网络算法(1974年由美国哈佛大学P. Werbos在其博士论文中,提出的误差反传算法——BP网络),在多层神经网络模型的基础上提出了多层神经网络模型的反向传播学习算法,解决了多层神经网络的学习问题,解决了明斯基和帕伯特所提出的问题,使神经网络的研究迅速发展起来。

1987年6月在美国加州举行了第一届神经网络国际会议,来自世界各地的1000多名学者参加了大会,并成立了国际神经网络学会,掀起了用人工神经网络来模拟人类智能的热潮。

1990年3月在北京召开我国首届神经网络学术大会。1991年在南京成立中国神经网络学会。从此,我国的神经网络理论研究与应用进入一个崭新的发展阶段。

第二节 生物神经元系统

神经元(即神经细胞)是脑的基本组成单位,从人脑的结构来看,它由大量的(约 $10^{10} \sim 10^{12}$ 个)神经细胞组合而成。这些细胞相互连接,每个细胞完成某种基本功能,如兴奋和抑制,它们并行工作,整体上完成复杂思维活动和信息处理。大多数神经元具有某些结构上的共同特征,通常可以将其分为细胞体、树突和轴突三个区,如图1-2所示。

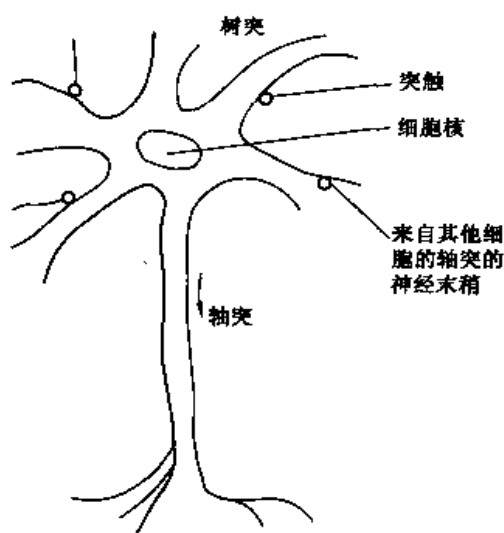


图 1-2 生物神经元结构

细胞体由细胞核、细胞质和细胞膜等组成;树突是精致的管状延伸物,是细胞体向外伸出的许多较短的分支,围绕细胞体形成灌木丛状,它们的作用是接受来自四面八方传人的神经冲击信息,相当于细胞的“输入端”,信息流从树突出发,经过细胞体,然后由轴突传出。轴突是由细胞体向外伸出的最长的一条分支,形成一条通路,信号能经过此通路从细胞体长距离地传送到脑神经系统的其他部分,其相当于细胞的“输出端”。

细胞体由细胞核、细胞质和细胞膜等组成;树突是精致的管状延伸物,是细胞体向外伸出的许多较短的分支,围绕细胞体形成灌木丛状,它们的作用是接受来自四面八方传人的神经冲击信息,相当于细胞的“输入端”,信息流从树突出发,经过细胞体,然后由轴突传出。轴突是由细胞体向外伸出的最长的一条分支,形成一条通路,信号能经过此通路从细胞体长距离地传送到脑神经系统的其他部分,其相当于细胞的“输出端”。

人脑功能是依靠信息流经由神经元组成的复杂

网络来完成的，信息是通过神经元系统中的接合部（即突触）由一个细胞传递至另一个细胞。如图 1-3 所示，突触是两个细胞之间连接的接口，一个典型的神经元可能有 1000 ~ 10000 个突触，能接收来自大约 10000 个其他神经元的信号。突触有两种类型：即兴奋型和抑制型。当输入脉冲使细胞膜电位升高，超过动作电位的阈值时，产生正的突触后电位的称为兴奋型突触；当输入脉冲使细胞膜电位下降至动作电位的阈值时，产生负的突触后电位的称为抑制型突触。

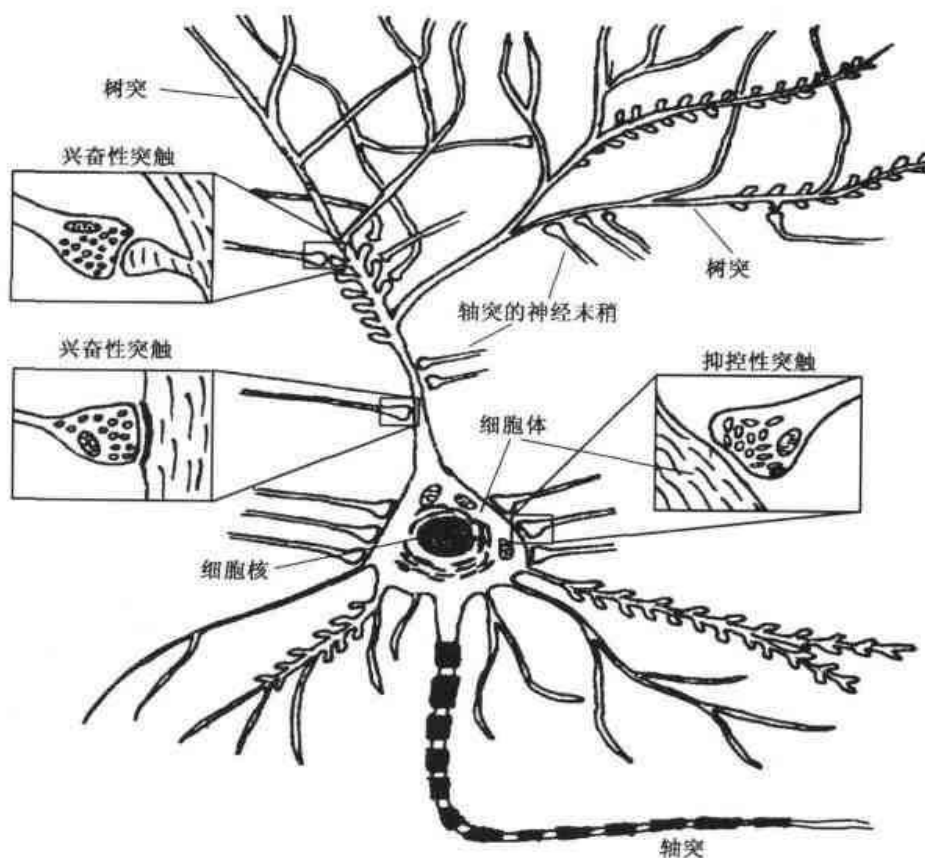


图 1-3 生物突触图解

由于轴突的末端分出许多末梢，它们与其他神经元的树突（和细胞体、轴突）构成一种称为突触的结合部，树突在突触接收信号后，将它传递给细胞体，信号在那里积累，激起神经元兴奋或抑制，从而决定神经元的状态。

第三节 人工神经网络模型

一、人工神经元模型

人工神经网络（Artificial Neural Networks）是采用物理可实现的系统来模仿人脑神经细胞的结构和功能。它反映了生物神经系统的基本特征，是对生物神经系统的某种抽象、简化与模拟。神经网络的基本要素是人工神经元，也就是说人工神经元是神经网络的基本处理单元，它只模拟了生物神经元的三个基本功能：①对每个输入信号进行处理，以确定其

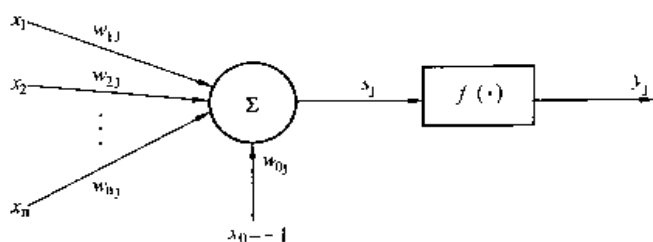


图 1-4 处理单元示意图

强度 (权值), 如神经元中突触的可变强度; ② 确定所有输入信号的组合 (加权和); ③ 确定其输出 (转移特性)。

典型的人工神经元模型如图 1-4 所示。

处理单元的内部阈值或门限值为 θ_j , 若用 $x_0 = -1$ 的固定偏置输入表示, 其连接强度取 $w_{0j} = \theta_j$ 。于是, 该神经元模型的输入输出关系为

$$s_j = \sum_{i=1}^n w_{ji} x_i - \theta_j = \sum_{i=1}^n w_{ji} x_i \quad (x_0 = \theta_j, w_{j0} = -1) \quad (1-2)$$

$$y_j = f(s_j)$$

式中 w_{ji} —— 连接权系数;

$f(\cdot)$ —— 输出激活函数。

如果用向量表示, 则

$$\left. \begin{aligned} X &= (x_0, x_1, x_2, \dots, x_i, \dots, x_n)^T \\ W_j &= (w_{0j}, w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj})^T \end{aligned} \right\} \quad (1-3)$$

神经元模型的输出向量可表示为

$$y = f(W \times X + \theta_j) = f\left(\sum_{i=1}^n w_{ij} x_i - \theta_j\right) \quad (1-4)$$

从式 (1-4) 可以看出, 阈值也被看作是一个输入分量, 也就是阈值也是一个权值, 在此用固定常数来表示。在网络的设计中, 偏差起着重要的作用, 它使得激活函数的图形可以左右移动而增加了解决问题的可能性。

二、处理单元的激活转移函数

激活转移函数 (Activation transfer function) 是一个神经元的重要组成部分, 转移函数 $f(\cdot)$ 也称为激励函数, 它描述了生物神经元的转移特性, 激活函数的基本作用是:

- (1) 控制输入对输出的激活作用。
- (2) 对输入、输出进行函数转换。
- (3) 将可能无限域的输入变换成指定的有限范围内的输出。

常用的激活函数有以下几种:

1. 阈值型函数

这种激活函数将任意输入转化为 ± 1 或 $(0, 1)$ 两种状态输出, 有时称为硬限幅函数, 如图 1-5 (a) 所示, 其表达式为

$$y_j = f(s_j) = \begin{cases} 1 & s_j \geq 0 \\ -1 & s_j < 0 \end{cases} \quad (1-5)$$

或

$$y_j = f(s_j) = \begin{cases} 1 & s_j \geq 0 \\ 0 & s_j < 0 \end{cases} \quad (1-6)$$

其主要特征是不可微、阶跃形，常应用于感知器模型、M-P 模型及 Hopfield 模型。

2. 线性型函数

线性型函数可将输入转化为任意值输出，即将输入原封不动地输出，而不像阈值型函数的输出只是两种状态，其输入输出关系如图 1-5 (b) 所示，线性函数表达式为

$$y_j = f(s_j) = s_j \quad (1-7)$$

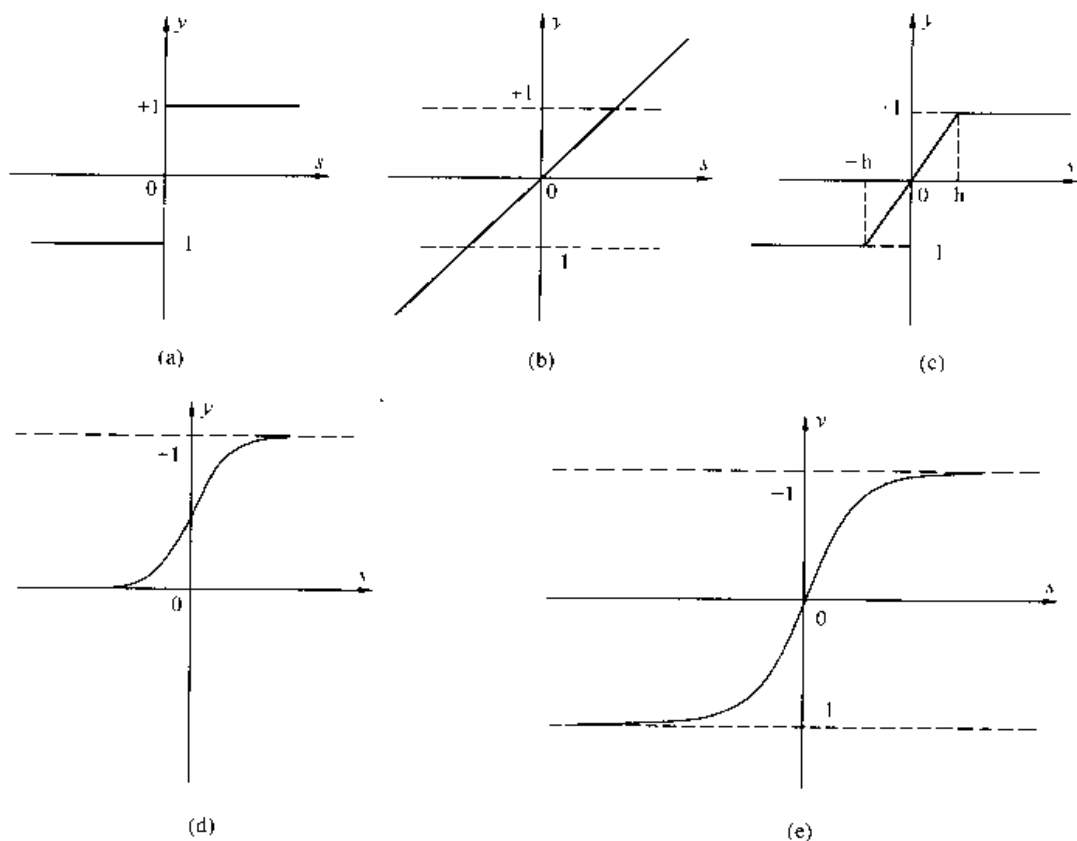


图 1-5 激活转移函数

(a) 阈值型函数；(b) 线性型函数；(c) 饱和型函数；

(d) S 型函数；(e) 双曲正切函数

3. 分段线性型（饱和型）函数

实质上，饱和型函数是阈值型函数和线性型函数的综合，如图 1-5 (c) 所示，其表达式为

$$y_j = f(s_j) = \begin{cases} 1 & s \geq h \\ s_j/h & |s| < h \\ -1(\text{或 } 0) & s \leq -h \end{cases} \quad (1-8)$$

其主要特征是不可微、阶跃型，常用于细胞神经网络，如模式识别、文字识别或噪声控制等。

4. S 型函数

S 型激活函数将任意输入值压缩到 (0, 1) 的范围内，常用的 S 型函数有对数函数，

即

$$y_j = f(s_j) = \frac{1}{1 + \exp(-s_j)} \quad (1-9)$$

或双曲正切函数

$$y_j = f(s_j) = \frac{1 - e^{-s_j}}{1 + e^{-s_j}} \quad (1-10)$$

如图 1-5 (d)、(e) 所示, 其主要特征是可微、阶跃型, 常用于 B-P 模型或 Fukushima (认识机/神经认识机) 模型。

第四节 人工神经网络的应用领域

人工神经网络能模拟人脑的并行信息处理方式, 具有惊人的自学习、思维推理、判断和记忆的功能。目前人工神经网络主要应用于解决下述几类问题:

1. 模式信息处理和模式识别

模式信息处理就是对模式信息进行特征提取、聚类分析、边缘检测、信息增强、噪声抑制、数据压缩以及各种变换等; 模式识别就是将所研究课题的特征类属映射成“类别号”, 以实现对客体特定类别的识别。这方面的主要应用有: 图形、符号、手写体及语音识别, 雷达及声纳等目标识别, 药物构效关系等化学模式信息辨识, 机器人视觉、听觉, 各种最近相邻模式聚类及识别分类等。

2. 最优化问题计算

人工神经网络的大部分模型是非线性动态系统, 若将所计算问题的目标函数与网络某种能量函数对应起来, 网络动态能量函数朝极小值方向移动的过程则可视作优化问题的求解过程, 其稳态点则是优化问题的局部或全局最优动态过程解, 这方面的主要应用有: 任务分配、货物调度、路径选择、组合编码、系统规划、交通管理等问题的最优化求解。

下面介绍其中两个领域的一些主要应用情况。

一、人工神经网络在化学工程中的应用

1. 化工过程故障诊断

把人工神经网络应用到一个流态化催化裂化的单元进行故障诊断, 其正确率可达 94% ~ 98%。基于人工神经网络采用连续输入变量 (仪表观测值), 对 3 个连续搅拌釜反应器串联操作和庚烷催化为甲苯反应器进行故障诊断, 取 12 个输入时正确率可达 100%。对于由一个连续搅拌釜反应器、一个活塞流反应器以及换热和分离等装置组成的比较复杂的化工生产过程, 运用人工神经网络构成故障诊断模型, 其分辨正确率达到 85%。通过研究反应器—分离器稳态和非稳态的故障诊断, 采用原始时间序列和时间的平均移动值作为人工神经网络的输入, 设“窗口宽度”为 4 个采样时间, 其故障诊断的正确率为 96%。

2. 化工过程控制

1986 年 Rumelhart 第一次将 ANN 用于控制界以后, 在短短数年间出现的 ANN 在控制系统中的应用有: 发酵过程的预测控制、生物工艺学的系统建模、精馏塔的系统辨识、自

适应控制、内模控制、最优化控制、专家控制、PID 参数的整定等。人工神经网络用于控制有两种方法：一种用来构造模型，主要利用对象的先验信息，经过误差校正反馈，修正网络权值，最终得到具有因果关系的函数，实现状态估计，由此进行推断控制；另一种直接充当控制器，如像 PID 控制器那样进行实时控制。

3. 化工过程建模

在化工过程中，通常遇到实际生产中的问题，如：①反应机理的复杂性和大量物性数据的缺乏；②工艺流程的复杂性及前后流程具有严重的反馈和关联耦合；③众多的随机因素。利用传统的基于数学模型的方法很难解决。但采用 ANN 方法在处理复杂的系统建模问题上恰恰表现了它的优越性。ANN 不需要预先对模型的形式和参数加以限制，网络只根据训练样本输入、输出数据来自动寻找相互关系，并给出过程对象具体的数学表达。如沧州化肥厂“八五”科技攻关项目，在合成氨和尿素装置的优化控制和调度系统开发中，将多层复杂前馈型网络、多层复杂反馈型网络及多层混合型网络用于合成塔等关键设备的建模问题。

4. 化工过程优化

ANN 提供了一种新的优化问题的求解方法，即 ANN 优化技术。由于 ANN 并行计算的性质，其计算量不会因为问题维数增加而发生指数“爆炸”，因此被广泛应用。对于换热网络这样的组合设计问题，可利用 Hopfield 网络，在求解实际问题时，首先找出问题的表达式，使神经元输出与问题的解相对应；构造神经网络能量函数，使其最优值与问题的最小值相对应；由能量函数设计出 Hopfield 网络，使网络运行后达到稳定状态，即为最优解。在优化催化剂配方的基础上，利用基于 BP 算法的多层前向神经网络对其进行工艺条件的辅助优化设计。优化设计的结果是：丙烷转化率为 85.2%，丙烯腈选择性 69.2%，丙烯腈的单收达到 58.9%。其结果表明，利用神经网络的良好性能来辅助催化剂设计是十分有效的方法。

二、人工神经网络在电力系统中的应用

1. 故障诊断

变压器的故障诊断是 ANN 在电力系统应用研究中较早提出的一个课题。当变压器内部出现异常时，都会在绝缘油中产生异常气体，将反映变压器绝缘油的特征元素以及油温、油压、各种噪声和绕组直流电阻、绝缘电阻介质损等信号汇集成一个样本集，对 ANN 进行学习训练，网络投入实际运行后，即能对各种可能出现的异常故障作出正确的诊断。应用 6 个 ANN 组成模块化结构的变压器故障诊断系统，正确率可达 90% 以上。基于一种联想记忆神经网络，通过网络的联想能力快速准确地进行电机早期故障的双向诊断，有效地处理各种模式并存的故障诊断问题，使诊断系统具有良好的鲁棒性。利用 ANN 判断绝缘破坏前的电晕放电波形和一般噪声波形的区别，经过学习对故障波形判断的准确率在 99% 以上。针对目前大型电站测点传感装置损坏率高，导致有些分析软件失效的现状，应用人工神经网络原理，设计了一个自适应 BP 网络模型，以某一实际机组为例，对其参数进行了仿真计算，绝大多数数据的相对误差在 1.5% 以内，可以满足工程实际的需要。

2. 智能控制

应用 4 层 BP 网络对发电机运行方式和系统干扰进行精确在线识别,并以此为基础设计了一种最优励磁调节器模型,这种调节器比固定点线性励磁方式具有更强的稳定性和较好的动态品质。采用改进型神经网络由两个子神经网络综合构成:一个是应用递推最小二乘算法的线性神经网络;一个是应用动态 BP 算法的动态递归网络。对基于相控整流直流传动系统的自适应控制实现策略的进行仿真,结果表明该自适应控制器性能好于传统的 PID 控制器。基于 ANN 自适应励磁控制、快控汽门控制和动态电阻制动的综合控制器,使三者达到最佳协调。基于 ANN 进行电力系统的实时切换负荷控制,在西北电网模拟某线路故障失去了 700MW 电源时,稳定控制切换负荷误差小于 0.5MW 的比例达到 81.5%,最大误差在 2~3MW 之间的仅为 1%,显示了较好的控制特性。

3. 继电保护

基于 ANN 的方向高频保护,经仿真测试表明,所实现的高频保护不仅在各种复杂运行方式和故障条件下能正确地识别各种故障,而且在整个时域都能保证有准确的识别能力,可满足作为超高压线路主保护的要求。应用 ANN 构成的距离保护模型,这种距离保护模式完全抛弃了阻抗继电器动作特性的概念,经仿真结果表明,对于线路全长的 85% 内的故障,动作时间小于 26ms,对于线路全长的 70% 以内的故障,动作时间小于 16ms。基于小波变换和 ANN 的保护模型,将其输入特征量经过小波变换,并选用 3 个三层的 BP 网络可实现对故障种类的判断、对故障性质的分析和故障的定位。使用 3 层前向网络构成变压器保护模型,取变压器二次侧的电流和其他故障特征量进行综合判断,经大量样本训练后,可准确判断变压器的励磁涌流和各种故障。

4. 优化运算

基于连续 Hopfield 人工神经网络对设备布置问题进行优化求解计算,可得到较快的计算速度和稳定的解。把 Hopfield 网络应用于梯级水电站实现实时优化调度,克服了梯级水电站经济调度约束条件复杂、计算量大、优化问题具有非线性特性等缺陷。针对排放口最优规划算法中存在非线性目标函数、约束条件过于复杂等难点,基于 Hopfield 网络的排放口最优规划算法,通过实际应用表明该算法具有较好的应用效果。将状态观测器设计问题转化为约束非线性规划问题,利用 Hopfield 网络在线计算观测器的增益矩阵和输入矩阵,能够保证观测器输入信号的平均幅值接近最小,避免了重复计算,可达到在线优化计算的要求。针对常规 Hopfield 网络对可变对象适应性学习的困难,建立了一动态 Hopfield 网络及其学习算法,使该网络可以识别不同大小的字符,从而克服了只能识别同一号大小字符的局限性。

第二章

感知器

感知器 (perceptron) 是由美国学者 F. Rosenblatt 于 1957 年提出的, 它是一个具有单层计算神经元的神经网络, 并由线性阈值单元组成。原始的 perceptron 算法只有一个输出节点, 它相当于单个神经元。当它用于两类模式的分类时, 相当于在高维样本空间中, 用一个超平面将两类样本分开。F. Rosenblatt 已证明, 如果两类模式是线性可分的 (指存在一个超平面将它们分开), 则算法一定收敛。感知器特别适用于简单的模式分类问题。

第一节 单层感知器网络

为了理解多层感知器的功能, 在此首先介绍单层感知器, 其网络结构图如图 2-1 所示, 图中 $X = (x_1, x_2, \dots, x_n)^T$ 是输入特征向量, w_{ji} 是 x_i 到 y_j 的连接权, 输出量 $y_j (j = 1, 2, \dots, m)$ 是按照不同特征的分类结果。由于按不同特征的分类是互相独立的, 为了方便可以取出其中一个神经元来讨论, 如图 2-2 所示, 其输入到输出的变换关系为

$$y_j = f(s_j) = \begin{cases} 1 & s_j \geq 0 \\ -1 & s_j < 0 \end{cases} \quad (2-1)$$

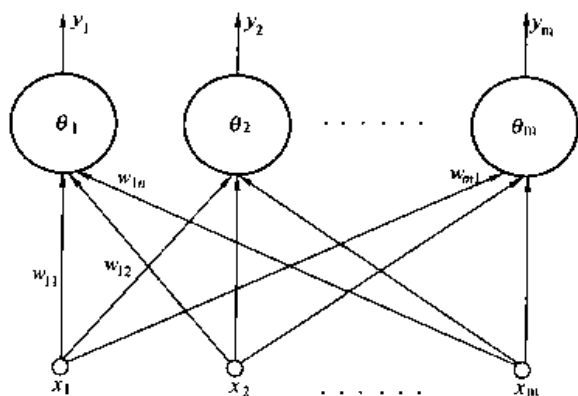


图 2-1 单层感知器网络

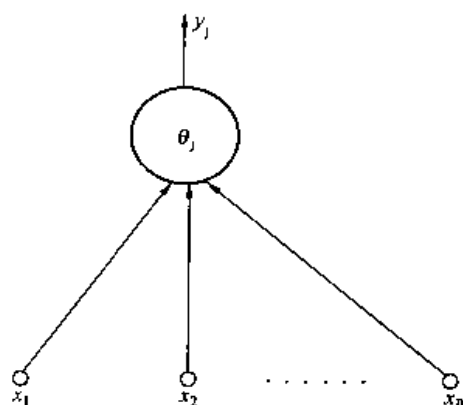


图 2-2 单个神经元的感知器

如果观察向量 $X = (x_1, x_2, \dots, x_n)^T$ 有 k 个样本 (即 $X^p, p = 1, 2, \dots, k$), 当将这些样本分别输入到神经元中, 在一定的连接权 w_{ji} 和阈值 θ_j 下, 神经元的输出值有两种可能, 即

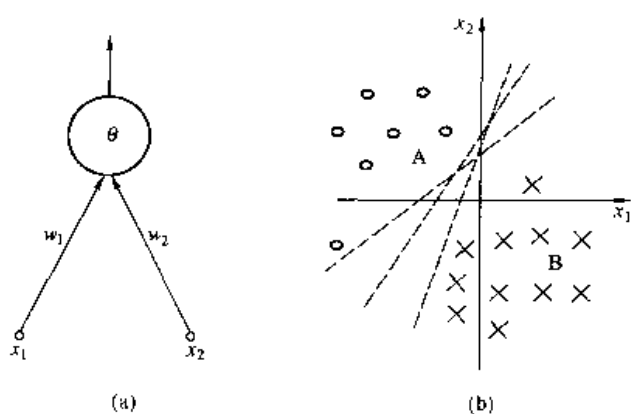


图 2-3 两维输入感知器的线性可分的几何解释
(a) 两维输入感知器网络; (b) 线性可分示意图

+1 或 -1, 这就是说, 该神经元将输入模式分成两类, 若将 $X^p (p = 1, 2, \dots, k)$ 看成是 n 维空间的 k 个点, 则该感知器将该 k 个点分成了两类, 它们分别属于 n 维空间中的两个不同部分。

为了形象地说明这种分类的特点, 下面以二维输入向量为例, 如图 2-3 所示。

设图中“o”和“x”表示具有不同特征的两类向量, 将其表示为

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad W_j = \begin{bmatrix} w_{j1} \\ w_{j2} \end{bmatrix}$$

于是神经元的模型表达式可改写为

$$s_j = w_{j1}x_1 + w_{j2}x_2 - \theta_j$$

$$y_j = f(s_j) = \text{sgn}(s_j) = \begin{cases} +1 & s_j \geq 0 \\ -1 & s_j < 0 \end{cases} \quad (2-2)$$

两类的分界线为 $s_j = 0$, 即

$$w_{j1}x_1 + w_{j2}x_2 - \theta_j = 0$$

$$x_2 = -\frac{w_{j1}}{w_{j2}}x_1 + \frac{1}{w_{j2}}\theta_j \quad (2-3)$$

由式 (2-3) 可知, 分界线为一直线, 此时的样本集是线性可分的。由此可说明, 只要输入样本满足线性可分性, 则可找到合适的连接权 w_j 和阈值 θ_j , 实现对样本的正确分类。图 2-3 中 A 类样本用“o”表示, B 类样本用“x”表示。由于这两类样本满足线性可分性, 分界线为一直线, 如图 2-3 中虚线所示。当输入样本 X 位于分界线的上方, 属于 A 类, 则 $y_j = 1$; 当 X 位于分界线的下方, 属于 B 类, 则 $y_j = -1$ 。可根据神经元的输出值对输入样本进行分类。由于这两类样本间有一段距离, 因此, 分界线可以不止一条, 每条分界线对应一组 w_j 和 θ_j 值, 这意味着 w_j 和 θ_j 的解不是唯一的。

对于一般的模型表达式 (2-1), 若输入有 k 个样本 $X^p, p = 1, 2, \dots, k$, 且 $X \in R^n$, 则它们可用 n 维输入空间的 k 个向量来表示, 只要样本集满足线性可分性, 即其分界线为

$$s_j = \sum_{i=1}^n w_{ji}x_i - \theta_j = 0$$

或

$$w_{j1}x_1 + w_{j2}x_2 + \dots + w_{jn}x_n = \theta_j \quad (2-4)$$

当输入样本维数不同时, 其分界线分别为一直线 ($n = 2$ 时), 平面 ($n = 3$ 时) 或一般情况的 ($n - 1$) 维超平面。

第二节 多层感知器网络

单层的感知器模型只能用于线性可分的模式分类问题，如果数据不是线性可分的，如图 2-4 所示。

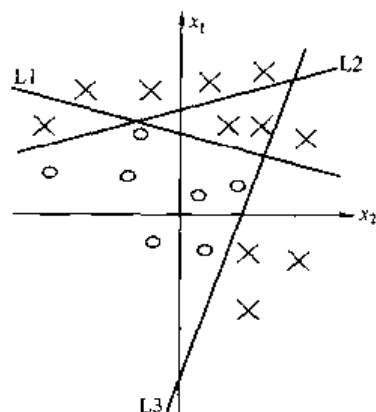


图 2-4 数据不是线性可分的

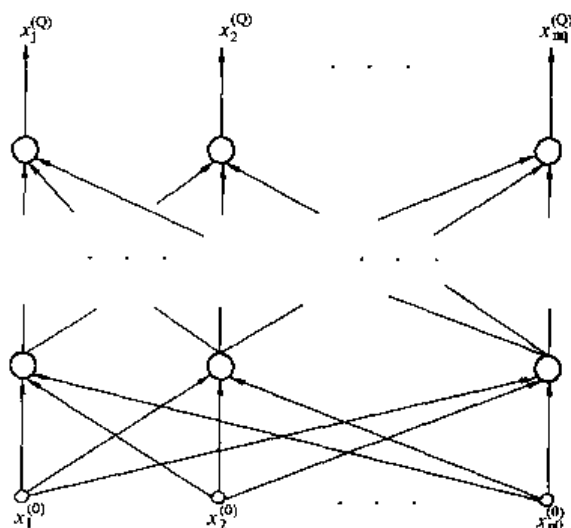


图 2-5 多层感知器网络

由于图 2-4 所示数据不是线性可分的，因此就找不到合适的权值将数据完全分开，但是，如果能找到 3 条直线 L_1 、 L_2 、 L_3 相与，就可以将两类数据完全分开了，多层感知器网络可以完成这一功能。如图 2-5 所示的多层感知器网络。

其中第 0 层为输入层，有 n_0 个神经元；第 Q 层为输出层，有 n_Q 个输出，中间层为隐层，该多层感知器网络的输入输出变换关系为

$$\left. \begin{aligned} s_i^{(q)} &= \sum_{j=0}^{n_{q-1}} w_{ij}^{(q)} x_j^{(q-1)} \quad [x_0^{(q-1)} = \theta_i^{(q)}, w_{i0}^{(q)} = -1] \\ x_i^{(q)} &= f(s_i^{(q)}) = \begin{cases} +1 & s_i^{(q)} \geq 0 \\ -1 & s_i^{(q)} < 0 \end{cases} \\ i &= 1, 2, \dots, n_q; j = 1, 2, \dots, n_{q-1}; q = 1, 2, \dots, Q \end{aligned} \right\} \quad (2-5)$$

此时，每一层相当于一个单层感知器网络，如对于第 q 层，它形成一个 n_{q-1} 维的超平面，它对于该层的输入模式进行线性分类，由于多层网络的组合，最终可实现对输入模式的较复杂的分类。

一、二层感知器

多层感知器与单层感知器在结构上的区别在于前者含有隐层。含有一个隐层的二层感知器可正确地完成对任何可分凸集分类。所谓的“可分凸集合”是指设被区分的客体有 m 类，记为 c_1, c_2, \dots, c_m ，与每类 c_j 相对应的输入向量在输入空间构成一个集合 R_i 。若所有

m 个集合 R_1, R_2, \dots, R_m 均为凸集合, 且互不相交, 则称之为可分凸集合, 如图 2-6 所示。

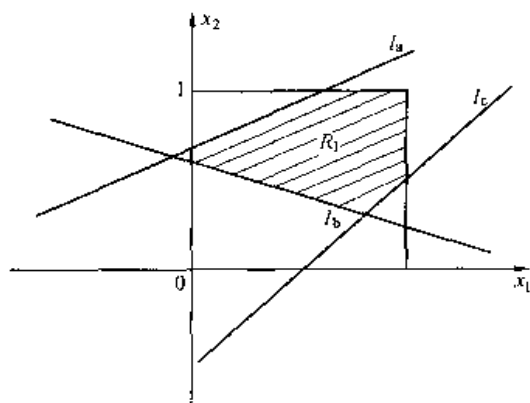


图 2-6 对二维输入空间凸集合 R_1 的分类示例

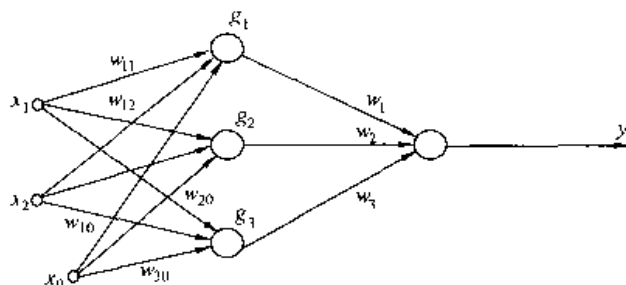


图 2-7 二层感知器网络示例

假设输入向量为二维的, 可分为两类: 一类输入向量全在凸集合 R_1 内; 另一类全在 R_1 外。可以证明, 采用图 2-7 所示的二层感知器网络, 则可完成对它们的正确分类。

在图 2-7 所示的网络中, 输入向量 $X = (x_1, x_2)^T$ 为二维的, w_{10}, w_{20}, w_{30} 分别表示隐层三个神经元的阈值。该网络的模型表达式为

$$\text{第一层(隐层)} \quad g_i = \text{sgn}\left(\sum_{j=1}^3 w_{ij}x_j\right) \quad i = 1, 2, 3, \text{ 而 } x_0 = 1 \quad (2-6)$$

$$\text{第二层(输出层)} \quad y = \text{sgn}\left(\sum_{j=1}^3 w_j g_j - \theta\right) \quad (2-7)$$

网络的分类算法分两层进行: 先完成第一层的运算; 然后以第一层输出量作为输入信号, 完成第二层的运算。第一层起线性分类作用; 第二层起“与”逻辑运算, 实现可分凸集合的分类作用。于是, 可根据网络的输出值, 来判断输入向量的类别。

$$\left. \begin{array}{l} \text{当 } X \in R_1 \text{ 时, } y = 1 \\ \text{当 } X \notin R_1 \text{ 时, } y = -1 \end{array} \right\} \quad (2-8)$$

第一层有 3 个神经元, 起线性分类作用, 它们的分界线分别为 3 根直线 l_a 、 l_b 和 l_c , 输出为 g_1 的神经元对直线 l_a 两侧的输入向量进行分类, 设 l_a 的 R_1 一侧集合为 R_{a1} , 另一侧的集合为 R_{a2} , 则此神经元的运行算法为

$$\left\{ \begin{array}{l} \text{当 } X \in R_{a1} \text{ 时, } g_1 = 1 \\ \text{当 } X \in R_{a2} \text{ 时, } g_1 = -1 \end{array} \right.$$

同理, 输出为 g_2 和 g_3 的另两个神经元, 则分别对直线 l_b 和 l_c 两侧的集合进行分类。设它们在 R_1 一侧的集合分别为 R_{b1} 和 R_{c1} , 在另一侧的集合分别为 R_{b2} 和 R_{c2} , 则该两个神经元的运行算法分别是

$$\left\{ \begin{array}{l} \text{当 } X \in R_{b1} \text{ 时, } g_2 = 1 \\ \text{当 } X \in R_{b2} \text{ 时, } g_2 = -1 \end{array} \right.$$

和

$$\left\{ \begin{array}{l} \text{当 } X \in R_{c1} \text{ 时, } g_3 = 1 \\ \text{当 } X \in R_{c2} \text{ 时, } g_3 = -1 \end{array} \right.$$

按照上述算法进行分类, 当 $X \in R_1$ 时, 第一层的 3 个神经元的输出值均为 1, 即 $g_1 = g_2 = g_3 = 1$; 当 $X \notin R_1$ 时, g_1 、 g_2 和 g_3 中至少有一个为 -1。若令式 (2-7) 中 $w_1 = w_2 = w_3 = 1, \theta = 2$, 则式 (2-8) 成立, 这说明该网络能完成对可分凸集合的分类作用。

二、三层感知器

具有两个隐层的三层感知器, 可以证明, 它可完成对任何非交集进行分类。现以图 2-8 所示的二维输入空间的非凸集合 R 为例, 采用三层感知器网络, 对集合 R 进行准确地分类。其网络的分类功能表现如下

$$\begin{cases} \text{当 } X \in R \text{ 时, } y = 1 \\ \text{当 } X \notin R \text{ 时, } y = -1 \end{cases}$$

对于图 2-8 所示的非凸集合 R 中的两个区域 R_1 和 R_2 可通过直线 l_{A1} 、 l_{A2} 、 l_{A3} 和 l_{B1} 、 l_{B2} 、 l_{B3} 来进行分类。

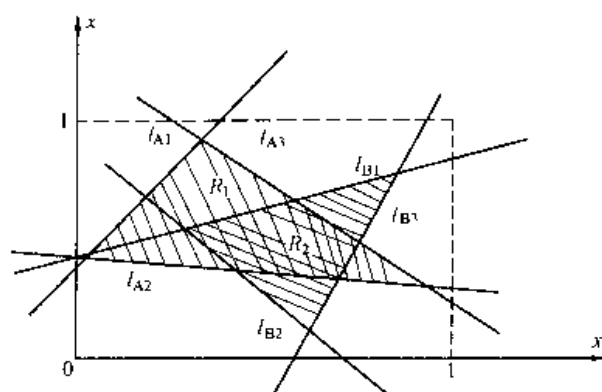


图 2-8 二维输入空间非凸集合 ($R_1, R_2 \in R$) 的分类示例

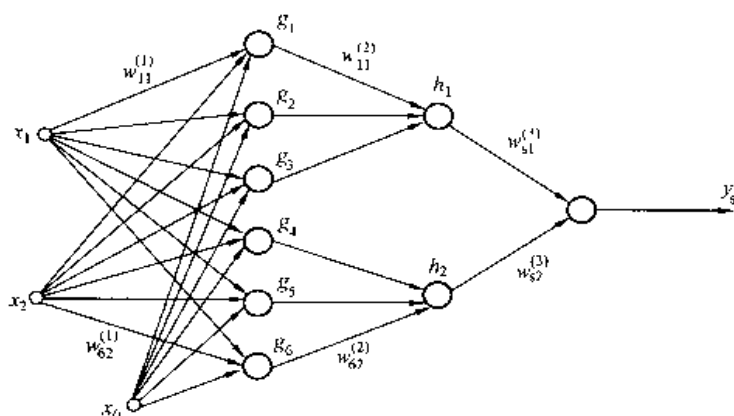


图 2-9 三层感知器网络示例

图 2-9 所示感知器网络, 其输入向量 $X = (x_1, x_2)^T$ 为二维的。 $X_0 = 1$, 因此 w_{i0} ($i = 1, 2, \dots, 6$) 表示的是第一隐层各神经元的阈值, 网络的模型表达式为

第一隐层

$$g_i = \text{sgn} \left[\sum_{j=1}^2 w_{ij}^{(1)} x_j \right], \quad i = 1, 2, \dots, 6, \quad x_0 \equiv 1 \quad (2-9)$$

第二隐层

$$h_j = \text{sgn} \left[\sum_{i=1}^6 w_{ji}^{(2)} g_i - \theta_j^{(2)} \right], \quad j = 1, 2 \quad (2-10)$$

第三层 (输出层)

$$y_s = \text{sgn} \left[\sum_{j=1}^2 w_{sj}^{(3)} h_j - \theta_s^{(3)} \right] \quad (2-11)$$

网络的分类法分三层进行：先完成第一层的运算；然后以第一层的输出量作为输入信号，完成第二层的运算；最后以第二层的输出量作为输入信号，完成第三层的运算。第一层起线性分类作用；第二层起“与”逻辑运算，实现非交凸集合的分类作用；第三层起“或”逻辑运算，实现非交集合的分类作用。

图 2-9 所示网络的第一层有 6 个神经元，分别完成对直线 l_{A1} 、 l_{A2} 、 l_{A3} 和 l_{B1} 、 l_{B2} 、 l_{B3} 两侧的集合进行线性分类。设置权系数使得当输入量 X 位于直线 l_{A1} 、 l_{A2} 、 l_{A3} 所围成的凸集合 R_1 内部一侧时， $g_1 = g_2 = g_3 = 1$ ，其他情况时 g_1 、 g_2 和 g_3 中至少有一个输出为 -1 ；当 X 位于直线 l_{B1} 、 l_{B2} 、 l_{B3} 所围成的凸集合 R_2 内部一侧时， $g_4 = g_5 = g_6 = 1$ ，其他情况中 g_4 、 g_5 和 g_6 至少有一个输出为 -1 。

第二层 2 个神经元，分别对第一层的神经元输出信号进行逻辑“与”运算，可设置权系数和阈值使得

$$\begin{cases} \text{当 } X \in R_1 \text{ 时, } h_1 = 1 \\ \text{当 } X \notin R_1 \text{ 时, } h_1 = -1 \end{cases} \quad \text{和} \quad \begin{cases} \text{当 } X \in R_2 \text{ 时, } h_2 = 1 \\ \text{当 } X \notin R_2 \text{ 时, } h_2 = -1 \end{cases}$$

从而完成对凸集合的分类作用。

第三层的神经元对第二层的输出信号进行逻辑“或”运算。可设置权系数和阈值如令 $w_{s1}^{(3)} = w_{s2}^{(3)} = 1, \theta_s^{(3)} = -1$ ，使得：当 $X \in R$ 时， h_1 与 h_2 中至少有一个为 1。于是由式 (2-11) 可得 $y_s = 1$ ；当 $X \notin R$ 时， h_1 与 h_2 均为 -1 ，于是 $y_s = -1$ ，从而实现对非凸集合 R 的分类功能。

第三节 感知器的学习算法

当输入样本集是线性可分类时，网络的连接权和阈值，可通过神经网络学习（训练）来确定。在此采用的学习算法为有导师的学习，学习算法的基本思想是：给网络提供一组训练样本对 (x^p, T_j^p) ，其中 x^p 为第 p 个输入样本向量， T_j^p 为对应的输出目标值（即教师信号），将输入样本 x^p 作用到网络上，计算相应的网络输出值 y_j^p ，若 $y_j^p = T_j^p$ 则保持原连接权（包括阈值）不变；若 $y_j^p \neq T_j^p$ 则按 Hebb 规则调整各连接权值，调整量 Δw_{ji} 与该连接权所对应的输入和输出量的乘积成正比例，即该算法的调整关系式为

$$w_{ji}(n_0 + 1) = w_{ji}(n_0) + \Delta w_{ji} \quad (2-12)$$

$$\Delta w_{ji} = \begin{cases} 2\eta T_j^p x_i^p & y_j^p \neq T_j^p \\ 0 & y_j^p = T_j^p \end{cases}$$

$$y_j^p = f(s_j^p) = f\left(\sum_{i=0}^n w_{ji} x_i^p\right), x_0^p = -1, w_{j0} = \theta_j$$

若将式 (2-12) 中的系数 2 和 0，用 $1 - y_j^p T_j^p$ 表示，当 $y_j^p \neq T_j^p$ 时， y_j^p 与 T_j^p 异号， y_j^p 与 T_j^p 同号，于是 $y_j^p T_j^p = 1$ ，则 $1 - y_j^p T_j^p = 0$ ，这样，式 (2-12) 可改写为



0786501

$$\begin{aligned}
 \Delta w_{ji} &= \eta(1 - y_j^p T_j^p) T_j^p x_i^p = \eta(T_j^p - y_j^p T_j^p T_j^p) x_i^p \\
 &= \eta(T_j^p - y_j^p) x_i^p = \eta \delta_j^p x_i^p \\
 \delta_j^p &= T_j^p - y_j^p
 \end{aligned}
 \quad (2-13)$$

式中 δ_j^p ——第 p 个样本时网络输出的误差；

η ——学习速率系数，其值为； $0 < \eta < 1$ ；

T_j^p ——离散值 $\{+1, -1\}$ ；

x^p ——离散值或连续值，归一化为 $(-1, 1)$ 范围内。

为此，网络的学习律为

$$w_{ji}(n_0 + 1) = w_{ji}(n_0) + \eta \delta_j^p x_i^p \quad (2-14)$$

网络的学习步骤如下：

(1) 设置初始连接权值（包括阈值），赋给 $w_i(0)$ 各一个较小的随机非零值。

(2) 在训练样本对中，任选一对 $(x^p - T_j^p)$ ，在输入向量 $X^p = (x_1^p, x_2^p, \dots, x_n^p)^T$ 的作用下计算神经网络的输出值，即

$$\begin{aligned}
 y_j^p &= f(s_j^p) = f\left(\sum_{i=0}^n w_{ji} x_i^p\right) \\
 &= \text{sgn}(s_j^p) = \begin{cases} 1 & s_j^p \geq 0 \\ -1 & s_j^p < 0 \end{cases}
 \end{aligned}
 \quad (2-15)$$

(3) 计算对应的输出误差值

$$\delta_j^p = T_j^p - y_j^p \quad (2-16)$$

(4) 调整各连接权值

$$w_{ji}(n_0 + 1) = w_{ji}(n_0) + \eta \delta_j^p x_i^p \quad (2-17)$$

(5) 重复步骤 (2) ~ (4)，直到所有误差均为零止。

可以证明，如果输入向量满足线性可分性、可分为两类时，无论输入向量的维数为何值，采用硬限幅函数的单神经元，按步骤 (4) 的学习算法，且训练样本足够多时，只要选取合适（ η 选取太小，学习太慢； η 太大，学习过程易产生震荡），权向量将收敛到给出正确分类结果的对应值。

第四节 感知器应用举例

【例 2-1】 简单分类问题。

本例采用单一感知器神经元（如图 2-10 所示）来解决一个简单的分类问题，将 4 个输入的向量分为两类，其中 2 个向量对应的目标值为 1，另 2 个向量对应的目标值为 0。

输入矩阵

$$X = \begin{bmatrix} -0.5 & -0.5 & 0.3 & 0.0 \\ -0.5 & +0.5 & -0.5 & 1.0 \end{bmatrix}$$

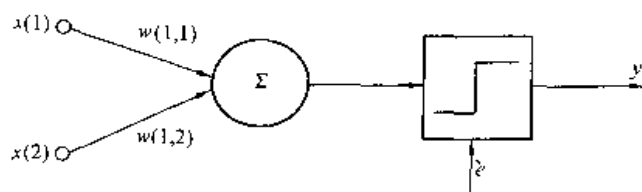


图 2-10 单一感知器神经元模型

目标分类向量

$$T = [+1.0 \quad +1.0 \quad 0.0 \quad 0.0]$$

输入向量可以用图 2-11 来描述。在图中，对应于目标值 0 的输入向量，用符号“0”表示，对应于目标值 1 的输入向量用符号“+”来表示。

对于这个相对简单的模式分类问题，

采用单层感知器就可解决，其分类结果如图 2-12 所示。

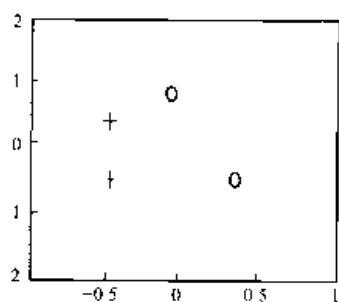


图 2-11 输入向量与目标向量关系图

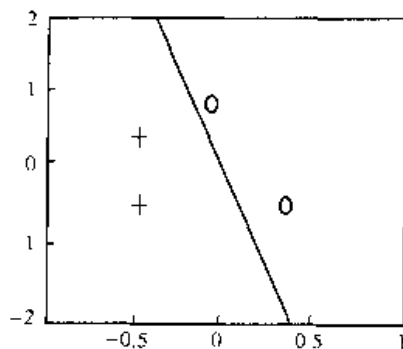


图 2-12 感知器分类结果图

【例 2-2】 多个神经元分类。

若将 [例 2-1] 的输入向量和目标向量增加为 10 组的二元矩阵，即输入矩阵为

$$X = \begin{bmatrix} 0.1 & 0.7 & 0.8 & 0.8 & 1.0 & 0.3 & 0.0 & -0.3 & -0.5 & -1.5 \\ 1.2 & 1.8 & 1.6 & 0.6 & 0.8 & 0.5 & 0.2 & 0.8 & -1.5 & -1.3 \end{bmatrix}$$

所对应的 10 组二元目标向量为

$$T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

由于增加了向量数组，必然增加了解决问题的复杂程度，这个问题用一般的方法来解决是相当困难和费时的，网络的规模和训练时间都要增加。

本例采用的网络具有两个输入，且有两个神经元，网络结构如图 2-13 所示。

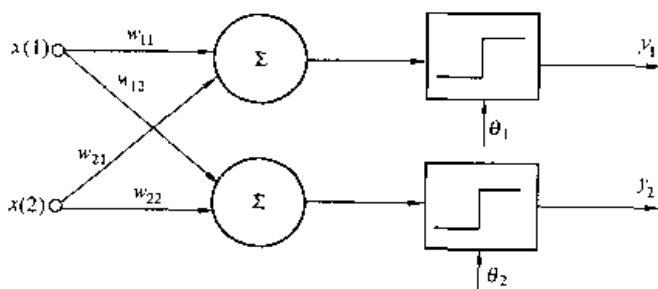


图 2-13 具有两个神经元的感知器网络结构图

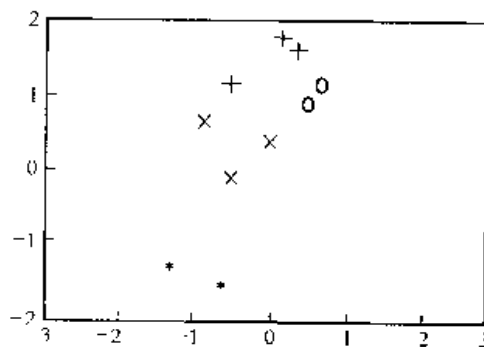


图 2-14 输入向量目标向量对应关系图

图 2-14 给出了输入向量与不同的目标向量所对应的不同关系, 不同的目标向量分别用 4 种符号表示, 00: '0'; 01: '*'; 10: '+'; 11: '+', 如图 2-14 所示。

经过训练调整神经元的权向量, 网络将输入向量分成期望的四类, 如图 2-15 所示。

【例 2-3】 线性不可分输入向量。

单层感知器神经网络的局限性, 那就是输入向量必须线性可分, 而对于线性不可分输入向量, 单层感知器神经网络则是无能为力。本例揭示了单层感知器神经网络的这一固有缺陷, 定义输入向量为

$$X = \begin{bmatrix} -0.5 & -0.5 & +0.3 & -0.1 & -0.8 \\ -0.5 & +0.5 & -0.5 & +1.0 & +0.0 \end{bmatrix}$$

对应的目标分类向量

$$T = [1 \quad 1 \quad 0 \quad 0 \quad 0]$$

输入向量可以用图 2-16 来表示, 采用单层感知器的分类结果如图 2-17 所示。

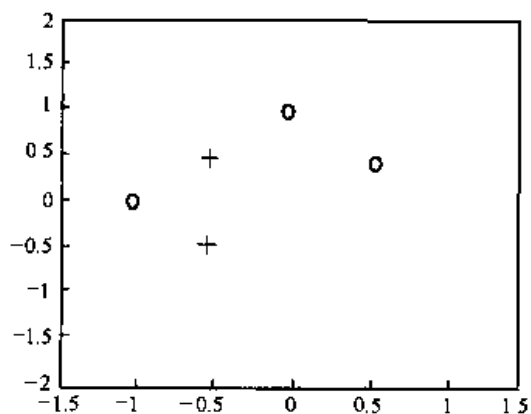


图 2-16 输入向量与目标向量关系图

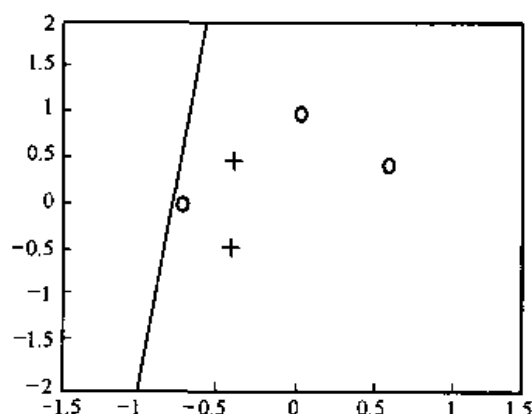


图 2-17 单层感知器分类结果

第五节 本章小结

(1) 感知器是早期的人工神经网络, 单层感知器是一个具有一层神经元, 采用阈值激活函数的前向网络。

(2) 感知器的基本功能是将输入向量转化为 1 或 0 的输出, 因此感知器通过输入向量的 m 维空间分成若干区域而具有达到将输入向量进行分类的能力。

(3) 感知器的学习规则属于梯度下降法; 如果解存在, 该算法有限次的循环迭代后可以收敛到正确的目标向量。

(4) 由于感知器自身结构的限制, 在解决具体问题时, 存在以下局限性:

1) 由于感知器的激活函数是阈值函数, 输出向量只能取 0 或 1, 所以只能用它来解决简单的分类问题。

2) 感知器仅能够线性地将输入向量进行分类。

3) 感知器对异常数据比较敏感, 如输入向量中有一个数比其他数大或小的很多时, 网络训练所需时间将很长, 即导致较慢的收敛速度。

(5) 单层感知器不能解决简单的“异或”问题。当采用两层网络结构, 并在隐含层中采用两个神经元, 即可解决“异或”问题。若采用三层感知器 (即两个隐含层和一个输出层) 可实现对任意形状的非交输入向量集合进行正确的分类。

第三章

BP 网络

1969 年, 人工智能创始人之一 M. Minsky 和 S. papert 在合著的《感知器》一书中指出: 单层感知器只能进行线性分类, 对线性不可分的输入模式, 如简单的“异或”逻辑运算, 单层感知器也无能为力, 解决的方法是要训练出具有隐含层的多层神经网络, 并要找到一种有效地修正权向量的学习算法。这一结论曾使许多人对神经网络的研究失去信心。虽然采用线性阈值单元组成的前馈多层网络, 具有很强的分类功能, 但由于阈值函数不可微, 又没有一种有效的学习算法, 从而限制了该网络的应用。Rumelhart, McClelland 和他们的同事们于 1982 年成立了一个 PDP 小组, 研究并行分布信息处理方法, 探索人类认知的微结构, 在 1985 年他们发展了多层前馈网络的学习算法——Back Propagation Training Algorithm (简称 B-P 算法), 使多层前馈网络的应用与研究取得了重大突破, 将 B-P 算法应用于前馈网络, 通常称之为 BP 网络。

第一节 BP 神经元模型

图 3-1 给出了一个基本的 BP 神经元, 它有 n 各输入, 每个输入都通过一个适当的权值 $w_i, i = 1, 2, \dots, n_0$, 连接到神经元节点上, 其中输出函数 $y = f(W \times X, \theta), a = \sum_{i=1}^{n_0} w_i x_i$, a 作为激活函数 f 的输入。激活函数的另一个输入是神经元的阈值 θ 。

BP 网络中隐层激活函数通常采用 S 型对数函数, 如图 3-2 所示。

BP 网络的隐层激活函数也可以采用 S 型正切函数, 如图 3-3 所示。

BP 网络的隐层或输出层激活函数也可以采用线性函数, 如图 3-4 所示。

图 3-1 中的权值和输入矩阵可写成

$$W = [w_1, w_2, \dots, w_{n0}]$$

$$X = [x_1, x_2, \dots, x_{n0}]^T$$

神经元模型的输出向量可表示为

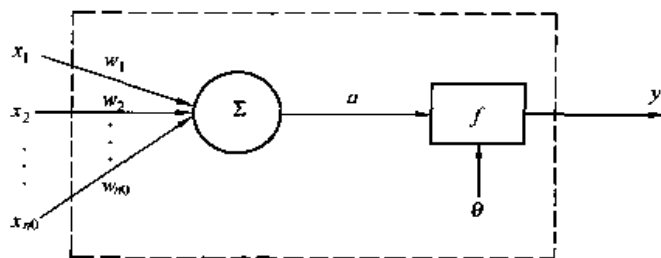


图 3-1 BP 神经元

$$y = f(W \times X + \theta) = f\left(\sum_{i=1}^{n_0} w_i x_i + \theta\right) \quad (3-1)$$

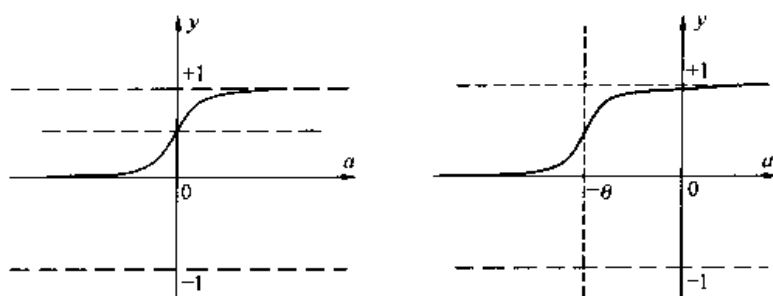


图 3-2 S 型对数激活函数

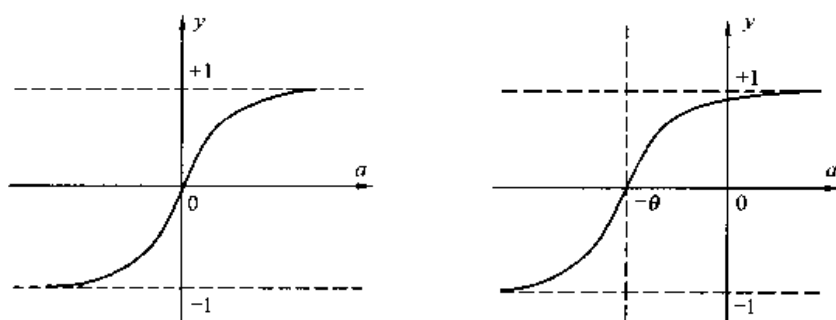


图 3-3 S 型正切激活函数

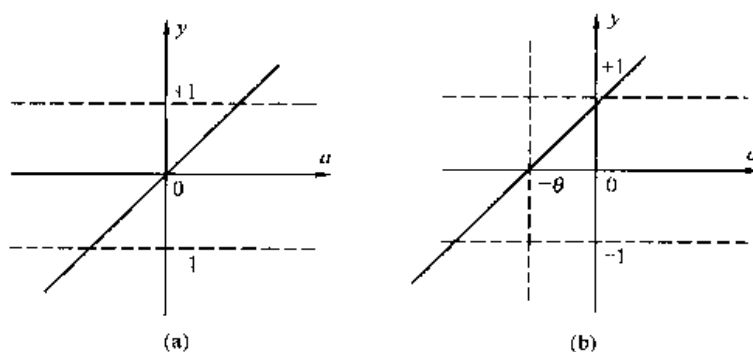


图 3-4 线性激活函数

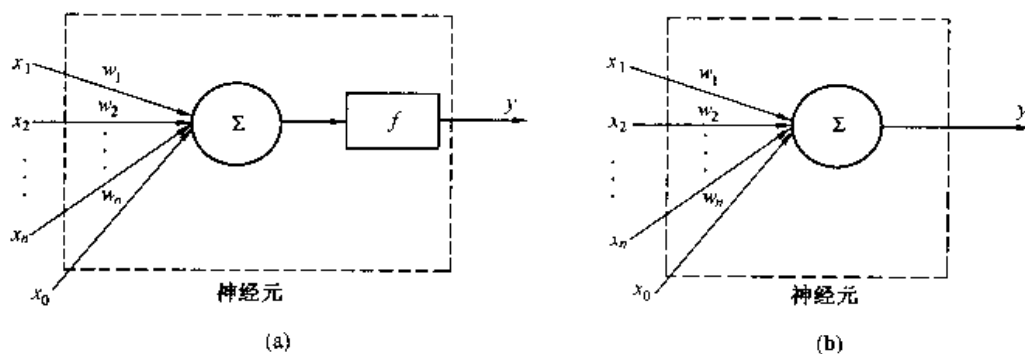


图 3-5 BP 网络的单个神经元模型

(a) 激活函数单独表示；(b) 激活函数隐含在神经元模型中

由图 3-1 中激活函数的另一个输入分量为阈值（偏差），在网络的设计中（阈值）偏差起着重要的作用，它使得激活函数的图形可以左右移动而增加了解决问题的可能性。因此，可也将阈值作为一个权值，将其输入设为固定常数 1（即令 $x_0 = 1$ ）。为了在网络设计中方便起见，图 3-1 可改为图 3-5（a）或图 3-5（b）。

第二节 BP 网络结构

一、单隐层 BP 网络

BP 网络通常有一个或多个隐层，隐层中的神经元均采用 S 型激活函数，输出层神经元采用线性传递函数，如果 BP 网络的最后一层采用 S 型激活函数，那么整个网络的输出就被限制在一个较小的范围内；如果 BP 网络的最后一层采用线性激活函数，那么整个网络的输出可以取任意值。图 3-6 描述了一个具有一个隐层的 BP 网络，图 3-6（a）网络通常简化为 3-6（b）网络。

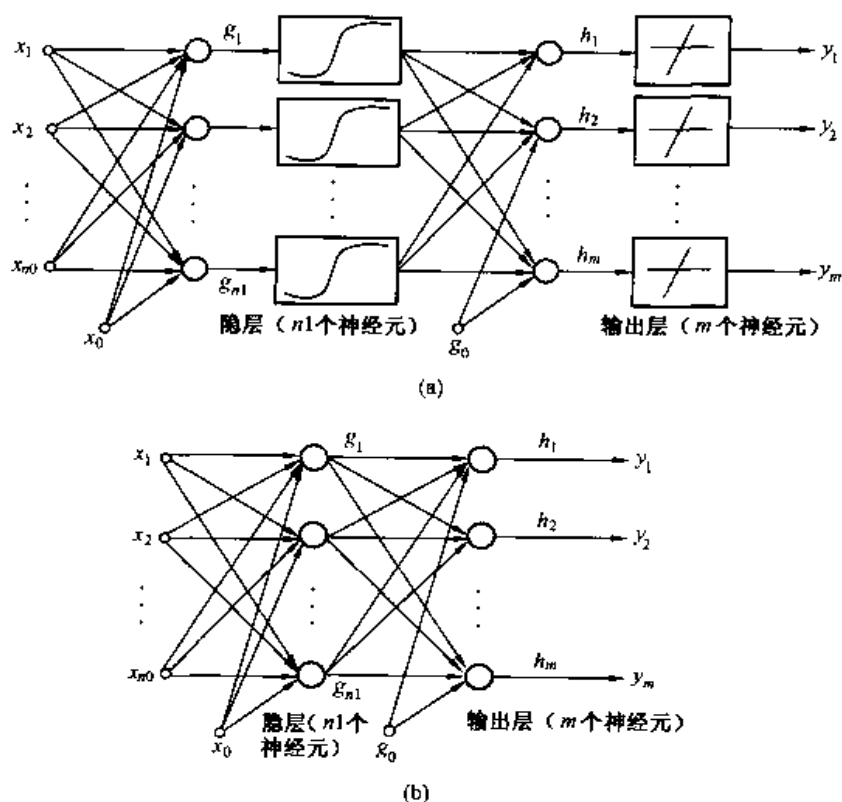


图 3-6 具有一个隐层和一个输出层的 BP 网络

二、双隐层 BP 网络

具有两个隐层和一个输出层的 BP 网络如图 3-7 所示。

设输入向量为 $n+1$ 维，其中 x_0 为第一隐层中激活函数的阈值，将其增广到输入量中，作为一个分量 x_0 ，即 $x \in R^{n+1}$, $X = (x_0, x_1, \dots, x_n)^T$ ，输入量 x_i 可以是离散量 $\{0, 1\}$ 或 $\{-1, 1\}$ ，也可以是连续量；图 3-7 网络含有两个隐层，设第一隐层有 $n1$ 个神经元，

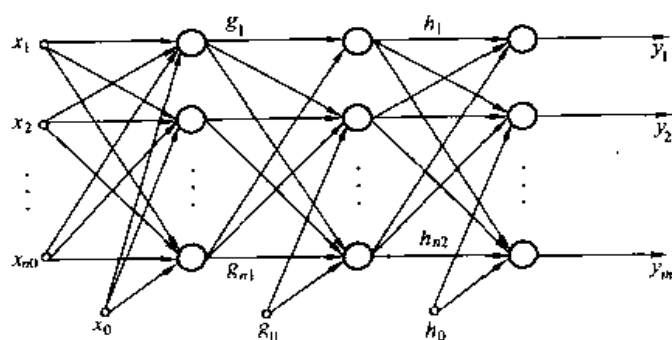


图 3-7 具有两个隐层的 BP 网络

元的输出为

$$\begin{cases} g_j = f\left(\sum_{i=1}^{n_0} w_{ji}x_i - \theta_j\right) & j = 1, 2, \dots, n_1 \\ h_k = f\left(\sum_{j=1}^{n_1} w_{kj}g_j - \theta_k\right) & k = 1, 2, \dots, n_2 \\ y_l = f\left(\sum_{k=1}^{n_2} w_{lk}h_k - \theta_l\right) & l = 1, 2, \dots, m \end{cases} \quad (3-2)$$

为了方便讨论，将神经元的阈值 θ 视为连接权来处理，即令 $\theta_j = w_{j0}$, $\theta_k = w_{k0}$, $\theta_l = w_{l0}$, $x_0 = g_0 = h_0 = -1$ ，于是式 (3-2) 可改写为如下形式

$$\begin{cases} g_j = f\left(\sum_{i=0}^{n_0} w_{ji}x_i\right) \\ h_k = f\left(\sum_{j=0}^{n_1} w_{kj}g_j\right) \\ y_l = f\left(\sum_{k=0}^{n_2} w_{lk}h_k\right) \end{cases} \quad (3-3)$$

第三节 BP 学习算法

BP 算法是一种有导师的学习算法，这种算法通常是应用最速下降法。图 3-8 描述了多层前向网络中的一部分，其中有两种信号在传送：

(1) 工作信号（用实线表示），它是施加输入信号后向前传播直到在输出端产生实际输出的信号，它是输入和权值的函数。

(2) 误差信号（用虚线表示），实际输出与应有输出之间的差值即为误差，它由输出端开始逐层向后传播。

由图 3-8 可知，网络的信号传送是由两部分组成，即正向传播与反向传播，在正向传播阶段，学习样本送入输入层，经隐层逐层运算后，传至输出层，每一层神经元的状态只

影响下一层神经元的状态,如果在输出层没有得到期望的输出结果,即网络的实际输出与期望输出之间存在误差,则计算输出层的误差变化值,然后输入误差反向传播阶段,这是误差信号沿着原来的连接从输出层返回至输入层,并逐层调整连接权值,以使误差达到最小。由于输出层对产生误差的调整,首先必须通过改变隐层与输出层之间的连接

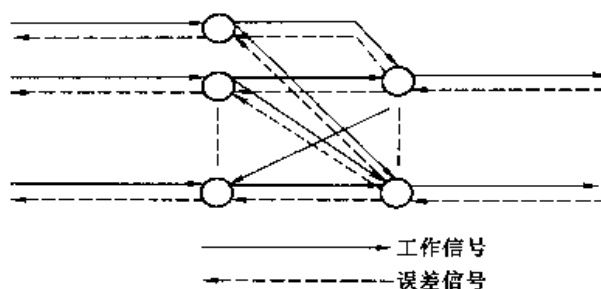


图 3-8 向前工作信号与反向误差信号示意图

误差,也须调整前一隐层(或输入层)与它之间的连接权值,如此进行下去,这就是 BP 网络学习过程的一个重要特点。

一、BP 网络权值的调整原理

以图 3-7 所示的多层前馈网络为例,进行 BP 算法的推导。设输入的学习样本有 α 个: $x^1, x^2, \dots, x^\alpha$, 对应的教师信号(即输出的期望值)为 $t^1, t^2, \dots, t^\alpha$ 。在此用这 α 个样本对 $(x^p, t^p), p = 1, 2, \dots, \alpha$ 对网络进行训练。当第 p 个样本对输入到网络后,由式(3-3)可得正向传播阶段经各层运算后,网络的实际输出值 $y_l^p, l = 1, 2, \dots, m$ 。将其与期望值相比较,可在第 p 个输入模式后输出方差为

$$E_p = \frac{1}{2} \sum_{l=1}^m (t_l^p - y_l^p)^2 \quad (3-4)$$

若输入所有 α 个样本对经正向传递运算后,则网络的总误差为

$$E_\Sigma = \sum_{p=1}^{\alpha} E_p = \frac{1}{2} \sum_{p=1}^{\alpha} \sum_{l=1}^m (t_l^p - y_l^p)^2 \quad (3-5)$$

应用最速下降法,反向调整各层连接权值,使误差达到最小。设 W_{uv} 为网络任意两个神经元之间的连接权,沿负梯度方向调整连接权 W_{uv} 的修正量为

$$\Delta W_{uv} = -\eta \frac{\partial E_\Sigma}{\partial W_{uv}} = -\sum_{p=1}^{\alpha} \eta \frac{\partial E_p}{\partial W_{uv}} \quad (3-6)$$

设 n 为迭代次数,应用梯度法可得网络各层连接权的迭代关系式为

$$\begin{cases} W_{lk}(n+1) = W_{lk}(n) - \eta \frac{\partial E_\Sigma}{\partial W_{lk}} \\ W_{kj}(n+1) = W_{kj}(n) - \eta \frac{\partial E_\Sigma}{\partial W_{kj}} \\ W_{ji}(n+1) = W_{ji}(n) - \eta \frac{\partial E_\Sigma}{\partial W_{ji}} \end{cases} \quad (3-7)$$

式(3-7)中等号右边的第二项,为各层连接权的修正量。

下面从输出层开始,讨论逐层反向传播的调整规律。

(1) 第二隐层至输出层的连接权修正量为

$$\Delta W_{lk} = -\eta \frac{\partial E_{\Sigma}}{\partial W_{lk}} \quad (3-8)$$

令式 (3-8) 中

$$\begin{aligned} \frac{\partial E_{\Sigma}}{\partial W_{lk}} &= \sum_{p=1}^a \frac{\partial E_p}{\partial y_l^p} \cdot \frac{\partial y_l^p}{\partial net_l^p} \cdot \frac{\partial net_l^p}{\partial w_{lk}} \\ &= - \sum_{p=1}^a (t_l^p - y_l^p) f'(net_l^p) h_k^p \end{aligned}$$

故式 (3-8) 可写成

$$\Delta W_{lk} = \eta \sum_{p=1}^a f'(net_l^p) h_k^p \cdot \Delta_l^p = \eta \sum_{p=1}^a \delta_l^p h_k^p \quad (3-9)$$

式中, η 为学习速度, 其值通常在 0.01 ~ 1.0 之间, $\Delta_l^p = t_l^p - y_l^p$ 为输出神经元的实际误差; δ_l^p 为在第 p 个输入样本对作用下, 输出神经元 l 的等效误差, 即

$$\delta_l^p = f'(net_l^p)(t_l^p - y_l^p) = f'(net_l^p)\Delta_l^p \quad (3-10)$$

(2) 第一隐层至第二隐层的连接权修正量为

$$\begin{aligned} \Delta W_{kj} &= -\eta \frac{\partial E_{\Sigma}}{\partial W_{kj}} \\ &= \eta \sum_{p=1}^a \sum_{l=1}^m (t_l^p - y_l^p) \cdot \frac{\partial y_l^p}{\partial net_l^p} \cdot \frac{\partial h_k^p}{\partial net_k^p} \cdot \frac{\partial net_k^p}{\partial w_{kj}} \\ &= \eta \sum_{p=1}^a \sum_{l=1}^m (t_l^p - y_l^p) f'(net_l^p) w_{lk} f'(net_k^p) g_j^p \\ &= \eta \sum_{p=1}^a \delta_l^p g_j^p \end{aligned} \quad (3-11)$$

式中, δ_k^p 为在第 p 个输入样本对下, “分摊”给第二隐层神经元 k 的等效误差, 其值为

$$\begin{aligned} \delta_k^p &= \sum_{l=1}^m (t_l^p - y_l^p) f'(net_l^p) w_{lk} f'(net_k^p) \\ &= \sum_{l=1}^m \delta_l^p w_{lk} f'(net_k^p) \\ &= f'(net_k^p) \Delta_k^p \end{aligned} \quad (3-12)$$

其中
$$\Delta_k^p = \sum_{l=1}^m \delta_l^p w_{lk} \quad (3-13)$$

(3) 第一隐层至输入层的连接权修正量为

$$\Delta w_{ji} = -\eta \frac{\partial E_{\Sigma}}{\partial w_{ji}} = \eta \sum_{p=1}^a \delta_j^p x_i^p \quad (3-14)$$

式中, δ_j^p 为在第 p 个输入样本对作用下, “分摊”给第一隐层神经元 j 的等效误差, 其值为

$$\begin{aligned}\delta_j^p &= \sum_{k=0}^{n_2} \delta_k^p w_{kj} f'(net_j^p) \\ &= f'(net_j^p) \Delta_g^p\end{aligned}\quad (3-15)$$

其中

$$\Delta_g^p = \sum_{k=0}^{n_2} \delta_k^p w_{kj} \quad (3-16)$$

根据以上的推导过程, 可进行以下几点讨论:

(1) BP 网络连接权的调整, 按递推关系式进行, 其一般表达式为

$$w_{uv}(n+1) = w_{uv}(n) + \Delta w_{uv} \quad (3-17)$$

式(3-17)中连接权修正量如式(3-9)、式(3-11)及式(3-14)所示, 其一般表达式为

$$\Delta w_{uv} = \eta \sum_{p=1}^a \delta_{out}^p y_{in}^p = \eta \sum_{p=1}^a \delta_{out}^p y_{in}^p \quad (3-18)$$

由式(3-18)可知, BP 网络任意连接权的修正量, 取决于该连接权输出端点的等效误差 δ_{out}^p 和输入端点的实际信号 y_{in}^p 。

(2) 等效误差由输出层向输入层逐层反向传播而产生。

对于输出层神经元的等效误差 δ_l^p , 可按式(3-10)计算, 对于隐层, 由于没有教师信号 (即隐层输出的期望值), 因此其等效误差不能直接计算, 而只能由前一层的等效误差反向传播, “分摊” 而得。如式(3-13)和式(3-16)所示。误差反向传播过程示意图如图 3-9 所示。

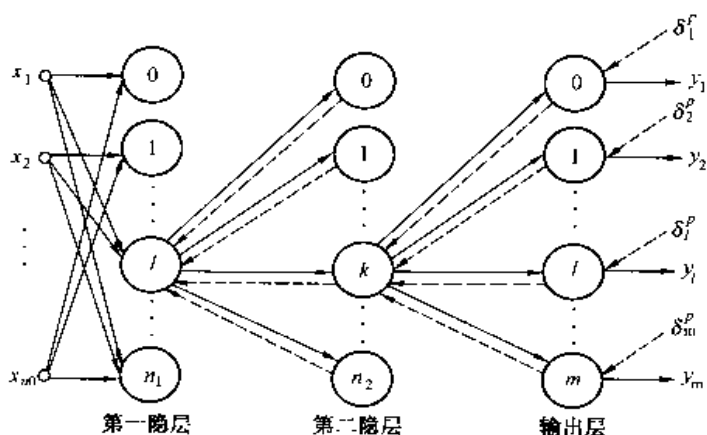


图 3-9 误差反向传播过程示意图

例如, 第二隐层神经元 k 的等效误差 δ_k^p [式(3-12)], 只能根据前一层 (输出层) 的等效误差 δ_l^p 通过连接权反向传播求 Δ_k^p [式(3-13)], 然后再由式(3-12)求 δ_k^p 值。该算法的特点是: 当网络输出存在误差时, 通过输出层神经元的等效误差, 逐层反向传播 “分摊” 给各隐层神经元, 以便按 “分摊” 得到的等效误差值调整对应的连接权值。故称这种算法为误差反向传播学习算法。

(3) 权值修正方式。

连接权值修正的方式有批处理方式与模式处理两种。以上介绍的属于批处理方式, 即网络连接权的修正, 是输入所有学习样本对并计算网络的总误差后, 才按负梯度方向进行调整, 批处理方式可避免逐个样本对作权值修正时可能引起的震荡。而模式处理是每输入一个样本对, 就对网络的所有连接权值修正一次。此时, 迭代的基本关系式与上述的相类

似, 所不同的只是在模式处理时, 可以取消上述式子中所有对模式 p 的求和运算。例如, 连接权修正量 [式 (3-18)] 可按下式调整

$$\Delta w_{uv} = \eta \delta_u y_v = \eta \delta_{out} y_{in} \quad (3-19)$$

二、反传学习算法的实现步骤

(1) 初始化。设置网络各连接权 (包括阈值), 置所有可调参数 (权和阈值) 为均匀分布的较小的随机数, 使每个神经元的净输入值较小, 以保证工作在激活函数变化较灵敏的区域。

(2) 信息正向传递过程的计算。输入学习样本对 $(x^p, t^p), p = 1, 2, \dots, \alpha$ 。对每个输入样本按下式顺序计算各层神经元的输出值, 即

$$\begin{cases} g_j^p = f\left(\sum_{i=0}^{n_0} w_{ji} x_i^p\right) \\ h_k^p = f\left(\sum_{j=0}^{n_1} w_{kj} g_j^p\right) \\ y_l^p = f\left(\sum_{k=0}^{n_2} w_{lk} h_k^p\right) \end{cases} \quad (3-20)$$

(3) 误差反向传播。

从输出层至输入层, 反向依次按式 (3-10)、式 (3-12) 和式 (3-15) 计算各层神经元的等效误差 δ_l^p, δ_k^p 和 δ_j^p 。然后返回步骤 (2), 对其他学习样本对进行正向传播计算和误差反向传播, 一直到所有 α 个学习样本对都进行类似运算为止。

(4) 调整各层的连接权值。按下式修改各层的连接权值, 即

$$\begin{cases} w_{lk}(n+1) = w_{lk}(n) + \eta \sum_{p=1}^{\alpha} \delta_l^p h_k^p \\ w_{kj}(n+1) = w_{kj}(n) + \eta \sum_{p=1}^{\alpha} \delta_k^p g_j^p \\ w_{ji}(n+1) = w_{ji}(n) + \eta \sum_{p=1}^{\alpha} \delta_j^p x_i^p \end{cases} \quad (3-21)$$

(5) 返回步骤 (2)。根据新的连接权值, 进行正向计算。若对每一个学习样本对 (x^p, t^p) 和输出层的每一个神经元 l 均满足精度要求, 即

$$|t_l^p - y_l^p|^2 < \epsilon \quad p = 1, 2, \dots, \alpha; l = 1, 2, \dots, m \quad (3-22)$$

其中 ϵ 为由精度要求给定的某一小数, 则网络的学习完成; 否则重复进行新一轮的正向传递与误差反向传播计算并调整各层的连接权, 直到式 (3-22) 满足为止。

三、反向传播算法的限制与不足

虽然反向传播 (BP) 算法得到广泛的应用, 但它也存在自身的限制与不足, 由于 BP 算法实质上是非线性优化问题的梯度算法, 它存在收敛性问题。即该算法不能保证学习的

结果一定收敛到均方误差的全局最小点,而可能落入局部极小点,使算法不收敛,导致网络陷入错误的工作模式。

由式(3-5)可知,网络的误差 E_S 是一复杂的非线性函数,它不仅与连接权值有关,而且与输入样本和输出期望值有关。就连接权值而言,权值空间是一个高维的复杂空间。

图 3-7 所示网络的权空间维数为

$$n_w = [(n_0 + 1)n_1 + (n_1 + 1)n_2 + (n_2 + 1)l]$$

因此,网络误差 E_S 是在高维空间中具有复杂形状(凹凸不平)的曲面,对这样复杂的误差曲面,沿负梯度方向的调整连接权值,可能出现不同的复杂情况。对于两维权值的误差表面可用图 3-10 来描述。

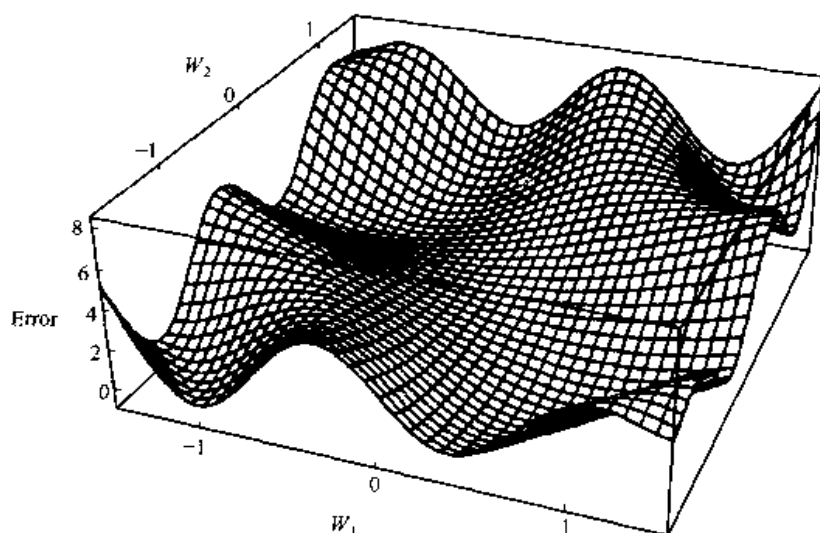


图 3-10 w_1 和 w_2 为变量的误差 E 表面示意图

下面叙述几种在学习过程中可能出现的情况:

(1) 收敛到局部极小点。由于 BP 算法采用的是梯度下降法,训练是从某一起始点沿误差函数的斜面逐渐达到误差的最小值。而网络误差曲面是高维的凹凸不平的复杂曲面,因此,在学习过程中可能陷入某个局部极小点,如图 3-11 (a) 所示。

(2) 在平坦区域内连接权值调整缓慢。误差曲面往往存在一些平坦区,由于激活函数的导数 $f'(\cdot)$ 趋于零,即使误差 $|t_i^p - y_i^p|$ 较大,但梯度已趋于零使得等效误差 δ 以及连接权的修正量 Δw 均趋于零,因而网络连接权的调整过程几乎处于停顿状态。即出现所谓的“网络和麻痹现象”。但只要调整方向正确,则经较长时间的调整后,总可以从平坦区退出,朝全局最小点趋近,如图 3-11 (b) 所示。

(3) 在陡峭的沟壑(峡谷)中振荡。误差曲面通常也存在陡峭的沟壑,这就使网络的权值调整收敛不到极小点上,而在极小点的周围来回振荡。这种情况如图 3-11 (c) 所示。

(4) 跳过较好的极小点。BP 网络在学习过程,也可能在网络权值调整过程中,跳过较好极小点,由于误差曲面的凹处太窄,而梯度值又较大,致使网络权值从一个较好的极小点附近跳到另一个次好极小点区域。这种现象如图 3-11 (d) 所示。

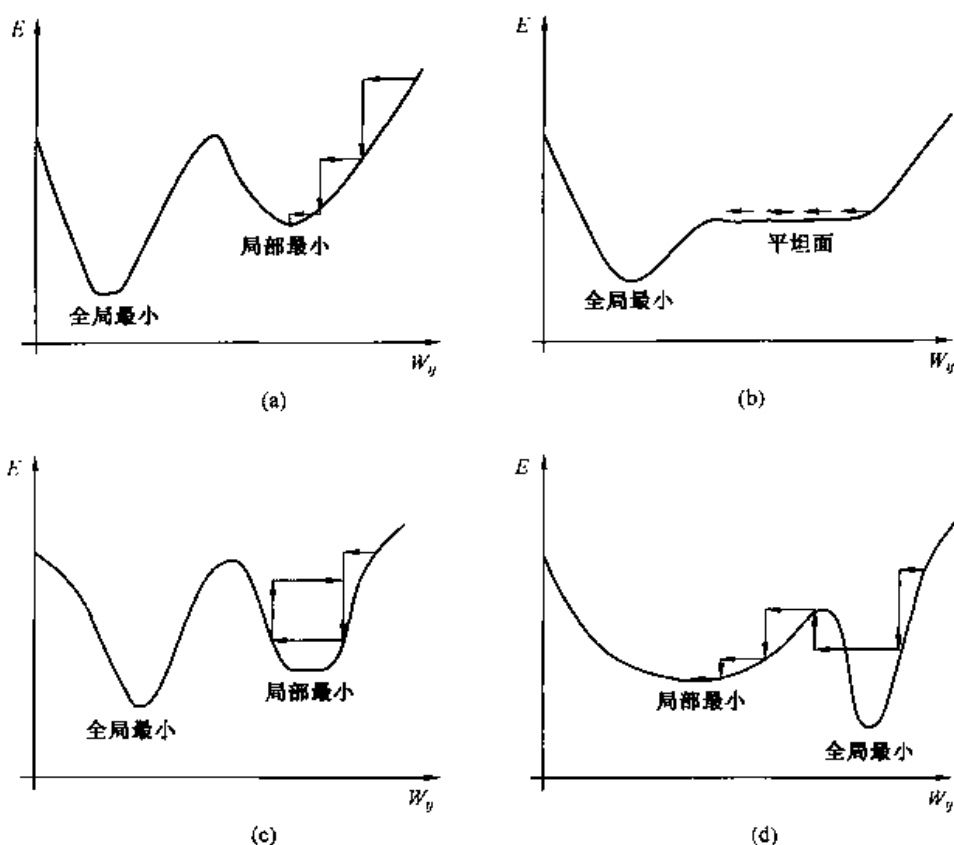


图 3-11 反传算法存在的问题

- (a) 收敛至局部最小点；(b) 在平坦区域内权值调整缓慢；
(c) 在极小点内来回振荡；(d) 跳过较好的极小点

第四节 BP 算法的改进

为了改善 BP 算法的有效性，提高学习速度，避免陷入局部极小点和增强泛化能力，所以近十几年来，许多研究人员对其做了深入的研究，提出了许多改进的方法。下面讨论几种常用改进方法的有关内容。

一、附加动量项

附加动量法使网络在修正其权值时，不仅考虑误差在梯度上的作用，而且考虑在误差曲面上变化趋势的影响，其作用如同一个低通滤波器，它允许网络上的微小变化特性。在没有附加动量的作用下，网络可能陷入浅的局部极小值，而利用附加动量的作用则有可能滑过这些极小值。因此，在引入动量项后，网络连接权的迭代关系具有如下形式

$$\begin{aligned}
 w_{uv}(n+1) &= w_{uv}(n) + \Delta w_{uv}(n+1) \\
 &= w_{uv}(n) + \eta \sum_{p=1}^a \delta_u^p y_v^p + m_c \Delta w_{uv}(n)
 \end{aligned} \quad (3-23)$$

其中动量项 $m_c \Delta w_{uv}(n) = m_c [w_{uv}(n) - w_{uv}(n-1)]$ ， m_c 为动量因子，一般 m_c 取为 0.9 左右。动量项的作用在于，记忆前一时刻连接权的变化方向（即变化量的值）。这样，就

可用较大的学习速率 η ，以提高学习的速度；增加动量项，利用其“惯性效应”来抑制可能产生的振荡，起到平滑的作用，此外，当网络的训练进入了误差曲面底部的平坦区时， δ_u 很小，于是

$$\Delta w_{uv}(n+1) \approx m_c \Delta w_{uv}(n) \approx \Delta w_{uv}(n)$$

由式 (3-23) 可得

$$\Delta w_{uv}(n) \approx \eta \sum_{p=1}^a \delta_u^p y_v^p + m_c \Delta w_{uv}$$

移项整理

$$\Delta w_{uv}(n) \approx \frac{\eta}{1 - m_c} \sum_{p=1}^a \delta_u^p y_v^p \quad (3-24)$$

与基本 BP 算法的连接权修正量相比较，其附加动量项后系数增大为 $\eta/(1 - m_c)$ ，从而有利于使网络连接权加快脱离误差曲面的平坦区。

二、学习速率的调整

由于 BP 网络中的误差曲面是一高维的非线性复杂曲面，因此，对于一个特定的问题，学习速率的确定往往是一件既繁琐又费时的的工作。在网络权值的调整过程中，人们通常希望在学习的初始阶段，步长可选择大一些，以使学习速度加快；当接近最优点时，步长又必须相当小，否则连接权值将产生振荡而难以收敛；当处在误差曲面的平坦区时，步长太小将使迭代次数增多；当处在误差曲面的剧烈变化区域，步长又不宜太大，否则将跳过较好的极小值或全局极小值。因此，合理地选择系数 η 和 m_c 是 BP 学习算法中的一个重要内容，目前还没有通用的学习速率调整公式可应用，通常是针对特定的问题，凭经验和依据实验结果调整这些参数值。学习速率自适应调整的一般规则是：当在连续迭代几步的过程中，新误差都比旧误差值大，学习速率将减小；若新误差小于旧误差时，则增大大学习速率。此方法可以保证网络总是以最大的可接受的学习速率进行训练，当一个较大的学习速率仍能使网络稳定学习，使其误差继续下降，则增加学习速率。一旦学习速率调得过大，而不能保证误差继续减少，则减小学习速率直到使其学习过程稳定为止。

三、改变误差函数的形式

基本 BP 算法中误差函数通常采用式 (3-4) 的形式，当 y_i^p 趋向 1 时， E 趋向一个常数，即处于 E_s 的平坦区，从而造成不能完全训练的麻痹现象。因此，可以采用误差函数的形式。例如 Baum 和 Wilczek 等人 1988 年提出的另一种误差函数为

$$E_s = \frac{1}{2} \sum_p \sum_i \left[(1 + t_i^p) \log \frac{1 + t_i^p}{1 + y_i^p} + (1 - t_i^p) \log \frac{1 - t_i^p}{1 - y_i^p} \right] \quad (3-25)$$

此时神经元的激活函数采用的是双曲正切函数，即

$$f(a) = \tanh(a) = \frac{1 - e^{-2a}}{1 + e^{-2a}} = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (3-26)$$

式中 a ——激活函数的输入。

又因为

$$f^2(a) = \tanh^2(a) = \frac{e^{2a} - 2e^a e^{-a} + e^{-2a}}{(e^a + e^{-a})^2} = \frac{e^{2a} + e^{-2a} - 2}{(e^a + e^{-a})^2} \quad (3-27)$$

那么

$$\begin{aligned} f'(a) = \tanh'(a) &= \frac{(e^a + e^{-a})(e^a + e^{-a}) - (e^a - e^{-a})(e^a - e^{-a})}{(e^a + e^{-a})^2} \\ &= \frac{(e^{2a} + e^{-2a} + 2) - (e^{2a} + e^{-2a} - 2)}{(e^a + e^{-a})^2} = 1 - f^2(a) \end{aligned} \quad (3-28)$$

求误差函数对输出层的变量 y_l^p 的一阶导数并同时考虑关系式, $y_l' = 1 - y_l^2$

$$\begin{aligned} \frac{\partial E}{\partial y_l^p} &= \frac{1}{2} (1 + t_l) \frac{1 + y_l}{1 + t_l} \frac{(1 + t_l)(-y_l)}{(1 + t_l)^2} (1 - y_l^2) \\ &\quad + \frac{1}{2} (1 - t_l) \frac{(1 - y_l)}{(1 - t_l)} \frac{(1 - t_l)y_l}{(1 - y_l)^2} (1 - y_l^2) \\ &= t_l - y_l = \delta_l \end{aligned} \quad (3-29)$$

与基本误差函数的形式 $\delta_l = f'(a)(t_l - y_l)$ 相比较, 其中的 $f(a)$ 项消失了, 这样当 a 增大时, 进入激活函数的平坦区, 即使 $f'(a) \rightarrow 0$, 也不会产生网络训练过程的麻痹现象, 有利于学习速度的加快。由于失去了 $f'(a)$ 对 Δw 的控制作用, 过大的 Δw 又有可能导致网络过调或振荡, 为了解决这一问题, 1989 年 S. Fahlman 提出一种折中的方案, 即

$$\delta_l^p = [f'(a_l^p) + 0.1](t_l^p - y_l^p) \quad (3-30)$$

式 (3-30) 一方面恢复了 $f'(a)$ 的某些影响, 另一方面当 $|a|$ 变大时, 仍能保持 δ_l^p 有一定的大小, 从而避免了麻痹现象的发生。

四、激活函数的选择

BP 网络中神经元的激活函数要求是连续可微、单调上升的有界函数。目前, 已提出有许多不同形式的激活函数, 下面介绍两种常用的改进形式。

1. 采用双极性 S 形函数

基本 BP 算法中神经元的激活函数, 通常采用 S 形函数, 即

$$y = f(a) = \frac{1}{1 + e^{-a}} \quad (3-31)$$

一般对数 S 型激活函数的输出动态范围为 $(0, 1)$, 从网络训练的要求来看, 这不是最佳的。因为从权值调整公式可知, 权值的变化也正比于前一层的输出, 而因其中一半是趋向 0 的一边, 这必然引起权值调整量的减少或不调整, 从而延长了训练时间, 为了解决这个问题, 采用双极性 S 型函数, 即

$$y = f(a) \approx -\frac{1}{2} + \frac{1}{1 + e^{-a}} \quad (3-32)$$

$$y^2 = [f(a)]^2 = \left(-\frac{1}{2} + \frac{1}{1 + e^{-a}}\right)^2 = \frac{1}{4} - \frac{1}{1 + e^{-a}} + \left(\frac{1}{1 + e^{-a}}\right)^2 \quad (3-33)$$

其一阶导数为

$$y' = f'(a) = \frac{-e^{-a}}{(1+e^{-a})^2} = -\frac{(-1+1+e^{-a})}{(1+e^{-a})^2} = \frac{1}{(1+e^{-a})^2} - \frac{1}{1+e^{-a}}$$

$$= \frac{1}{4} - \left[\frac{1}{4} - \frac{1}{1+e^{-a}} + \left(\frac{1}{1+e^{-a}} \right)^2 \right] = \frac{1}{4} - y^2$$

这时神经元输出值的变化范围为 $\pm 1/2$ ，实验表明，采用双极性 S 型函数后，收敛时间平均可减少 30% ~ 50%。

2. 在 S 型函数中加入可调系数

在基本的 S 型函数中加入可调参数，则激活函数的形式变为

$$y = f(a) = \frac{1}{\alpha_0 + \beta_0 e^{-r_0 a}} \quad (3-34)$$

其中 α_0 、 β_0 和 r_0 为可调参数，调节这些参数值可改变非线性的特性，通常在实际应用中，令 $\alpha_0 = \beta_0 = 1$ ，则式 (3-34) 可改写为下列形式

$$y = f(a) = \frac{1}{1 + e^{-r_0 a}} \quad (3-35)$$

当网络训练进入误差曲面的平坦区 ($t_i - y_i \neq 0$ ，而 $f'(a) \rightarrow 0$ 时，可调整 r_0 ，使 $r_0 < 1$ ，而使导数值 $f'(a)$ 增大，尽快地退出平坦区。

待退出平坦区后再恢复 $r_0 = 1$ 。因此，调节 r_0 对加快学习速度，克服“麻痹现象”的产生是比较有效的。其激活函数的输入、输出变化曲线如图 3-12 所示。

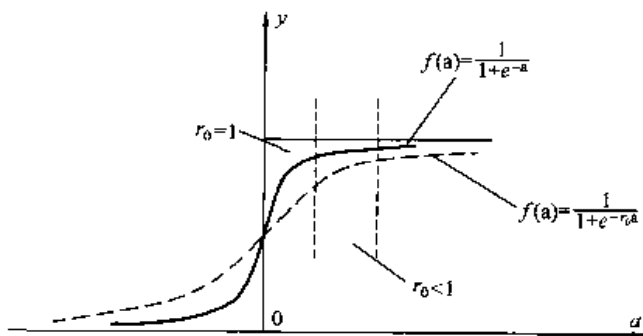


图 3-12 具有可调参数的 S 型函数

第五节 BP 网络的应用

一、BP 网络在预测中的应用

【例 3-1】 BP 网络用于室内温度预测

首先考虑影响温度变化的主要因素来确定 BP 网络的输入参数（即网络的输入节点数目）。根据传热机理，墙体与房间传热过程包括：①室内外温差引起通过墙壁进行的热传导 Q_a ；②由太阳辐射引起的热传导 Q_b ；③由室内人体散热，照明及其他发热设备带来的热量 Q_c ；④由空调输入的热量 Q_d 。

Q_a 与传热系数、传热面积、室内外温差有关；而 Q_b 与节气、气候、建筑物的朝向以及日期、时间有关；因此，对室内温度的影响可以综合为日期、时间、传热系数、室内初始温度 $t(0)$ 等系数。

Q_c 可认为是由辐射和对流组成的，人体的散热与性别、年龄、体重及人数等有关，而设备散热与使用方式、时间及设备的数量等有关。因此，可以认为 Q_c 与主要人数、灯

具功率、设备功率有关。

Q_d 由空调带入的热量与新风量和室内外空气焓值有关；室外空气焓值与地区有关，室内空气焓值可以取一个平均值。

考虑到建筑物的蓄热作用，辐射获得热量对室内温度的影响存在时间滞后，所以模型的输入还应包括反映这些变量过去状态的信息和过去时刻的室内温度。对于某一变量需要考虑过去多长时间的状态，与该变量的物理特性有关。由实验表明，室外温度 t_w 的变化需要考虑前 2 个小时的状态，太阳辐射可考虑前 1 个小时的情况。

依据以上分析，可以确定 BP 网络的输入层节点为 12 个，它们分别对应影响室内温度的 12 个主要参数为：即某一时刻室外温度 $t(k)$, $k = 0, 1, \dots$ ，前 2 个小时室外温度 $t(k-1)$ 、 $t(k-2)$ ，人员密度 P_p ，灯光功率密度 L_p ，设备功率密度 E_p ，新风标准 N_s ，传热系数 k_f ，窗地比 c_p ，墙地比 w_p ，室内容积 v ，室内初始温度 t_0 ；输出层参数有 1 个，即某时刻室内温度；隐层数为 1，隐层节点数为 15 个，选用 $\text{Tanh}(x)$ 函数作为激活函数，其 BP 网络模型结构如图 3-13 所示。

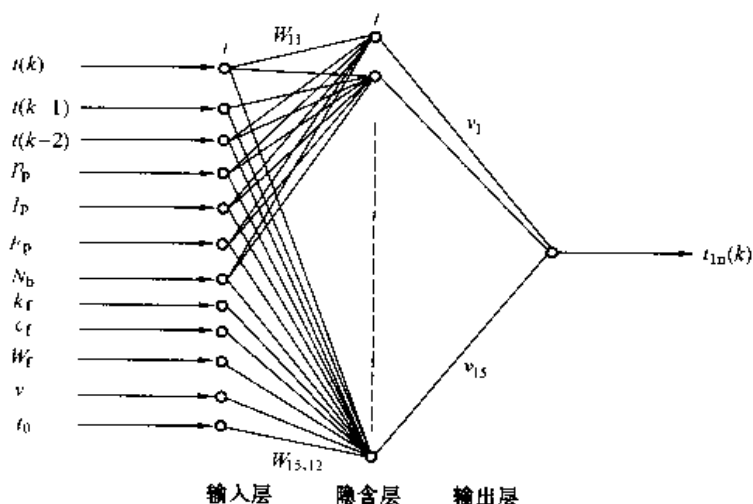


图 3-13 BP 网络模型结构

【例 3-2】 BP 网络用于油气井产能预测

对于油田气井产能通常用无阻流量的大小来描述，因此，只要预测出气井的无阻流量，就相当于知道了气井的产能。对气井无阻流量大小产生直接影响的参数有：地层渗透率、气层有效厚度，流体粘度，排驱半径，气井半径，表皮因子，气层孔隙度，原始地层压力，含气饱和度等因素。经过对这些影响因素的分析研究，筛选出对产能有影响的 5 个主要因素：设 x_1 为地层渗透率， x_2 为气层有效厚度， x_3 为气层孔隙度， x_4 为原始地层压力， x_5 为含气饱和度。此外，设无阻流量为 Q 。

已知影响因素可确定 BP 网络的输入节点为 5 个，输出节点为 1 个，根据经验选择隐含节点为 8 个。因此，建立的预测气井产能的三层神经网络模型为 $5 \times 8 \times 1$ ，其网络结构图如图 3-14 所示。

【例 3-3】 BP 网络用于水电发电量的预测

某省水力发电量是一非平稳过程, 具有趋势性和 12 个月为周期的季节变化规律, 如图 3-15 所示。

设 y_t 为 t 月发电量观测值; G_t 为 t 月发电量观测值的趋势分量; H_t 为 t 月发电量观测值的周期分量; E_t 是均值为零的随机噪声, 包括测量噪声或模型误差, 对于含有趋势性和季节周期变化规律的发电量模型, 可用如下乘积模型描述。

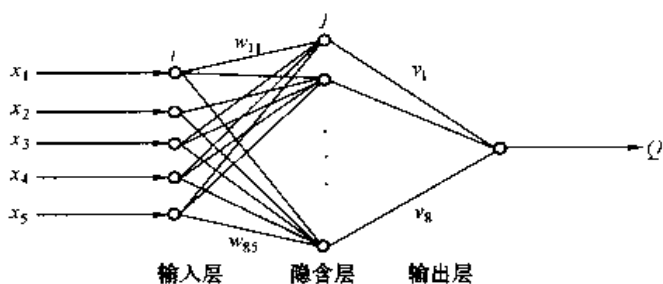


图 3-14 BP 网络模型结构

$$y_t = G_t H_t E_t \quad (3-36)$$

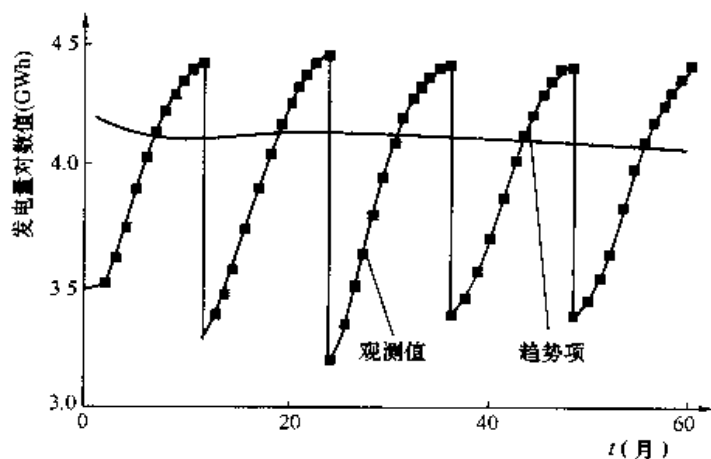


图 3-15 某省水力发电量对数值图形

设月发电量序列为 y_1, y_2, \dots, y_T , 其中 T 为序列长度, 它与观测数据序列年份个数 N 及月发电量周期数 12 的关系为 $T = 12N$ 。用中值移动平均法, 可提取不含季节周期波动的趋势项

$$G_t = \frac{1}{24} \left[\sum_{i=0}^{11} y_{t+i} + \sum_{i=0}^{11} y_{t+12+i} \right] \quad t = 1, 2, \dots, T-12 \quad (3-37)$$

上式表示以 12 为周期, 以 t 为中心的 2 阶对称滑动平均数字滤波, 经数字滤波后, 不再含有周期分量。从观测数据序列中分解出趋势项后, 就可对它进行描述、建模。并根据趋势项数据及其曲线特征, 可取多项式趋势、线性趋势、指数趋势等形式。

某省水力发电量趋势项数据用多项式描述较为合适。根据拟合的多项式与趋势项之间的残差平方和的 F 检验, 确定以 7 阶多项式描述 G_t 为最佳, 即

$$\begin{aligned} \hat{G}_t = & 16705.791018 - 1035750766t + 81.391808t^2 - 1.919231t^3 - 0.025536t^4 \\ & + 0.001816t^5 - 0.000027t^6 + 0.000000143t^7 \end{aligned} \quad (3-38)$$

利用 (3-36) 可分离出带有噪声的季节周期项

$$H_t E_t = y_t / \hat{G}_t \quad (3-39)$$

噪声 E_t 的估计值可从下式得到

$$\hat{E}_t = H_t E_t / \hat{H}_t \quad (3-40)$$

通过对观测数据的反复分析比较和实验, 基于该省水力发电量具有趋势性和周期性的特点, 确定网络结构为: 输入层神经元 24 个, 即趋势分量和周期分量分别为 12 个; 隐层节点数为 12 个; 输出层节点为 1 个, 两层的激活函数都采用非对称的 sigmoid 函数, 网络的目标值是发电量实际观测序列 Y_t 。BP 网络的学习样本为 1995 ~ 2000 年 8 月的发电量数据, 2000 年 9 ~ 11 月的发电量数据作为网络检验使用, 也就是将网络学习训练结果, 对 2000 年 9 月 ~ 11 月的发电量进行预测。其预测结果比较理想。

【例 3-4】 BP 网络用于产品成本预测

首先分析指标体系中的定性指标又有定量指标, 为使各指标在整个系统中具有可比性, 应将定性指标进行定量化处理。对定量指标, 用其衡量单位不同, 级差有大有小, 趋向也不一定一致, 必须利用效应系数对其进行规范化和同趋化处理, 当目标越大成本越低时, 效应系数可由下式计算

$$F_j = (x_j - x_{j\min}) / (x_{j\max} - x_{j\min}) \quad (3-41)$$

当目标越小成本越低时, 效应系数计算公式为

$$F_j = 1 - (x_j - x_{j\min}) / (x_{j\max} - x_{j\min}) \quad (3-42)$$

当目标越偏于中间成本越低时, 效应系数计算公式为

$$F_j = 1 - |x_j - (x_{j\min} + x_{j\max}) / 2| / (x_{j\max} - x_{j\min}) / 2 \quad (3-43)$$

式中, F_j ——目标值为 x_j 的效应系数;

$x_{j\min}$ ——预先确定的第 j 个指标的最小值;

$x_{j\max}$ ——预先确定的第 j 个指标的最大值;

j ——评价指标的数目。

以流程工业中的冶金工业为例, 建立 BP 预测模型, 并将其应用于某无缝钢管厂热轧管分厂的产品成本预测。针对冶金工业在技术上具有分段连续, 有间歇的大批量分解型生产及工艺路线单一固定, 工序间物流一致的特点; 在管理上具有工艺条件制约生产计划, 生产的间歇性要求制定合理的批量和多阶段的生产计划等特点, 结合实际经验, 给出各种影响钢管成本的因素, 设立如下评估指标体系。

x_1 : 工艺复杂程度, 工艺越复杂, 成本越高。

x_2 : 外径 (mm), 外径越偏于中间值, 成本越低; 越向两端, 成本越高。

x_3 : 壁厚 (mm), 壁厚越偏于中间值, 成本越低; 越向两端, 成本越高。

x_4 : 原料等级, 原料等级越高, 成本越高。

x_5 : 钢耗 (T), 钢耗越大, 成本越高。

x_6 : 动力消耗 (元/t), 动力消耗越大, 成本越高。

x_7 : 质量要求, 质量要求越高, 成本越高。

x_8 : 工具消耗 (元/t), 工具消耗越大, 成本越高。

由表 3-1 和表 3-2 可确定网络结构, 即设 BP 网络输入层为 8 个节点, 隐层节点可通过实验式由经验公式来确定, 输出节点为 1 个, 即成本。

表 3-1 学 习 样 本

成本影响因素 钢管品种	工艺复 杂程度	外 径	壁 厚	原料 等级	钢耗 (去头尾)	动力 消耗	质量 要求	工具 消耗	成本 (元/t)
35CrMo	9.8	95	9.7	9	6528	256	9	34.04	6280
45Mn2 汽车半轴套管	10	114	10	9.5	9490	272	9.5	39.27	6600
热轧 200 高压锅炉管	9.5	60	6	8.8	6443	244	8.8	31.86	6000
API 接箍用 J55 热轧管	7	94	8.8	8.5	4069	232	8.4	30.66	5800
API 箍用 N82 热轧管	8	108	14	9.2	6782	261	9.2	36.22	6450
低中压锅炉管	6	77	14	7	3808	214	7.5	30.38	4880
API 油管 (J55)	9	89	6.42	7.5	3848	220	7.8	31.27	5050
API 油管 (N80)	10	73	4.83	9.7	18444	284	9.8	40.26	6700

表 3-2 学习样本输入数据表

实例 编号	评估指标效用系数值								成本 (元/t)
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
1	0.05	0.908	0.97	0.259	0.814	0.4	0.348	0.37	0.628
2	0	0.690	0.938	0.074	0.612	0.171	0.130	0.1	0.66
3	0.125	0.690	0.637	0.333	0.82	0.571	0.435	0.85	0.60
4	0.75	0.920	0.935	0.444	0.982	0.743	0.609	0.972	0.58
5	0.5	0.759	0.425	0.185	0.800	0.329	0.261	0.409	0.645
6	1	0.885	0.425	1	1	1	1	1	0.488
7	0.25	0.977	0.682	0.815	0.997	0.914	0.87	0.91	0.505
8	0	0.839	0.513	0	0	0	0	0	0.67

【例 3-5】 BP 网络在烧结矿指标预测中的应用

基于人工神经网络技术建立一种烧结矿指标预测模型, 根据烧结矿性能指标确定 BP 网络模型结构为 $7 \times 22 \times 2$, 如图 3-16 所示。该网络的输入变量分别设 x_1 为料层高度 (mm), x_2 为烧结机速 (m/min), x_3 为点火温度 ($^{\circ}\text{C}$), x_4 为烧结负压 (Pa), x_5 为混合料温度 ($^{\circ}\text{C}$), x_6 为混合料水分 (%), x_7 为燃料配比 (%), 两个输出 y_1 为烧结矿 Feo 含量 (%) 和 y_2 为烧结矿转鼓强度 (%)。

设隐层神经元的激活函数为

$$y_j = \frac{1}{1 + e^{-\sum_j w_{pj} + \theta_j}} \quad (3-44)$$

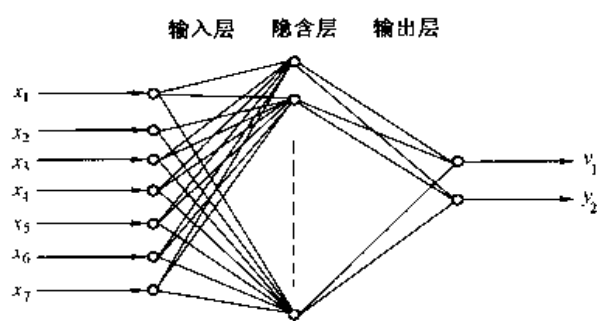


图 3-16 BP 网络结构图

式中 y_j ——当前神经元的输出；
 y_i ——上一层神经元的输出；
 w_{ij} ——从神经元 i 到神经元 j 权值；
 θ_j ——神经元的可调阈值。

对给定的样本组数目训练网络，并预留若干组样本以检验网络的预报推广能力。

【例 3-6】 BP 网络在水库水质预测方面的应用

水作为自然界中的一种可再生性资源，对社会经济的可持续发展起着重要的决定性作用。然而随着现代化工业的快速发展，水质不断受到污染，使这种可再生性资源不断地减少，甚至危及人类的生存和健康。在对水量需求不断增长，并且水量为自然条件所限定而难以扩大时，水质就成为限制水资源利用的重要因素。然而，由于影响资源与环境各因素之间的关系是多维和非线性的，所以采用常规的预测方法往往达不到理想的效果。

人工神经网络 (Artificial Neural Networks, 简称 ANN)，是基于模仿人类大脑的结构和功能而构成的一种信息处理系统，具有很多与人类智能相似的特点，诸如结构与处理的并行性，知识分布存储，很强的容错性，通过学习而具备适应外界环境的能力，模式识别能力和综合推理能力等。近年来在模式识别、知识处理、非线性优化、传感技术、智能控制、生物工程以及机器人研制等方面得到广泛应用。

大伙房水库位于辽宁省抚顺市东部，建于 1958 年，集水面积 5440km²，平均年净流量 1.65 × 10⁸m³，总库容 21.9 × 10⁸m³，水库以防洪灌溉、城市生活及工业用水水源为主，兼有发电、养鱼等综合利用的大型水利枢纽工程。尽管目前大伙房水库水质仍处于国家Ⅱ级标准，但每到夏季却连年发生水华，1975、1977、1979 年在库区曾发生大面积水华，1980、1981 年也发生过局部较短时间的水华，最严重的是 1982 年，水华覆盖全库，持续时间长达 2 个月，数量达 773 × 104/L，其中微囊藻占 90% 以上。浮游植物生长和繁殖除受水体氮磷含量影响外，还受温度及水库水文特征的影响。调查表明：大伙房水库的富营养化现象并不是随时间逐年加剧而是取决于每年径流量大小、雨水冲刷和河流带入库内的营养盐类。现以年降雨量、7~9 月水温、7~8 月入库水量/7~8 月库容、7~8 月出库水量/7~8 月库容和磷酸盐等 5 项因子为输入端，浮游植物生物量和密度为输出端建立人工神经网络模型。用 9 年 (1986~1994) 的 5 组实测输入和 2 组实测输出数据对神经网络进行训练，经过反复学习，得到实测与计算输出相当吻合的结果 (图 3-17)，经过 2 年 (1995~1996) 实测输入因子预测的输出计算结果与实测结果也非常吻合 (表 3-3)。

表 3-3 大伙房水库浮游植物群落变化人工神经网络模型预报结果

	浮游植物密度 (万/L)				浮游植物生物量 (mg/L)			
	实测值	模拟值	绝对误差	相对误差	实测值	模拟值	绝对误差	相对误差
1995	410.95	449.10	38.15	9.2%	12.27	12.53	0.26	2.1%
1996	349.98	349.98	0	0	10.09	10.32	0.27	2.2%

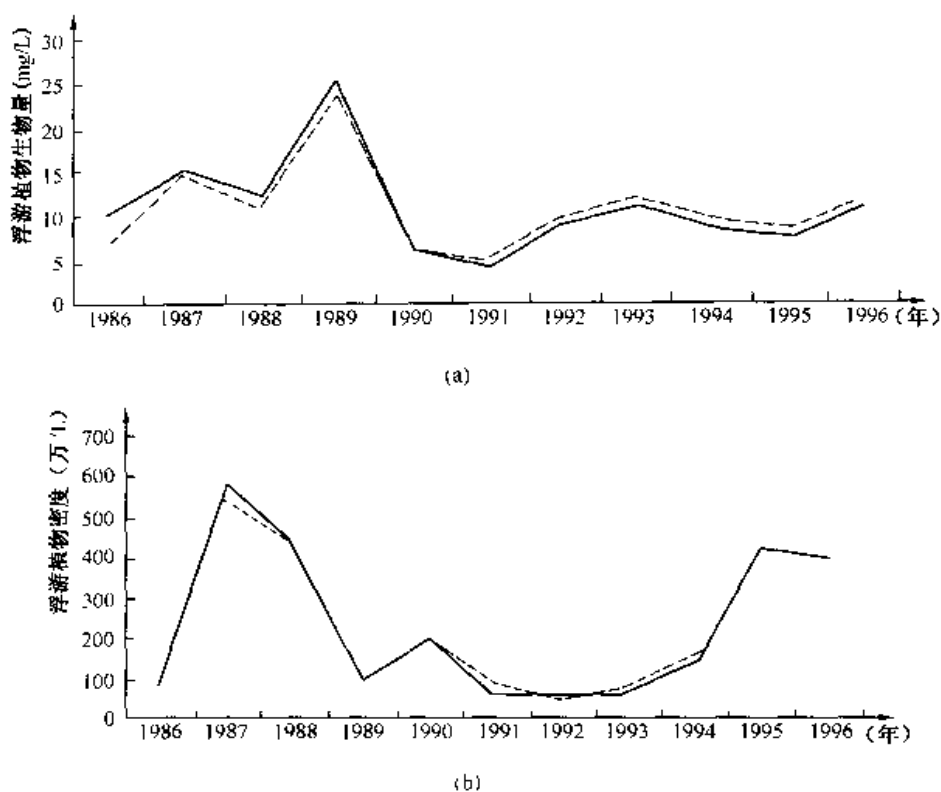


图 3-17 某水库浮游植物动态人工神经网络模型检验
(a) 浮游植物生物量 (mg/L); (b) 浮游植物密度 (万/L)
——模拟值; ----实测值

【例 3-7】 BP 网络在城市用水量预测中的应用

城市供水总体规划和工程规划的基础是用水量的预测, 常规的预测方法有数理统计法 (包括时序法、回归分析法、灰色预测法) 和用水量定额法。考虑到城市用水量受城市人口、规模、经济发展水平、产业结构与技术进步、管理水平等众多因素影响, 具有非线性的特征, 故采用 BP 神经网络的非线性时间序列递推预测方法, 并用于某城市的用水量预测。

基于神经网络预测城市用水量的基本思想: 从预测城市用水量的角度出发, 首先确定神经网络的基本结构, 即输入层节点数为 m , 输出层节点数为 n , 隐层节点数取为 $l = \frac{3}{2} \sqrt{nm}$ 。然后用生成的输入样本 x_k 和输出样本 y_k 对神经网络进行训练, 直到达到一定的误差要求, 即得到可预测城市用水量的神经网络模型。

在预测城市用水量时, 用 $X_{t-m+1} = (x_{t-m+1}, x_{t-m+2}, \dots, x_t)$ 输入训练好的神经网络, 输出为 $Y_{t-m+1} = (y_{t-m+1}, y_{t-m+2}, \dots, y_{t+n-m}) = (x_{t-m+2}, x_{t-m+3}, \dots, x_{t+1})$, 则第 $t+1$ 年城市用水量的预测值为 $Q_{t+1} = Q_1 \times lh \left| \frac{x_{t+1}}{1 - x_{t+1}} \right|$ 递推可预测出 $t+1$ 年之后各年数值。

某城市 1987~1996 年每年用水量见表 3-4, 用 1996 年数据对模型进行校验。确定神

神经网络的结构参数为：输入层节点数 $m = 4$ ，输出层节点数 $n = 3$ ，隐层数 $l = \frac{3}{2}\sqrt{nm} \approx 5$ 。生成的输出样本如表 3-5 所示，取精度 $\varepsilon = 5 \times 10^{-4}$ ，用动量—自适应学习率调整算法对神经网络进行训练，训练后的神经网络预测 1989 ~ 1996 年用水量及相对误差见表 3-6。

表 3-4 某城市 1987 ~ 1995 年历年用水量 百万 t

年份	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996
用水量	236.48	236.64	237.85	240.37	254.52	277.21	314.89	316.22	329.91	341.27

表 3-5 某城市用水量预测神经网络的训练样本

序 号	输 入 样 本					输 出 样 本				
1	0.7311	0.7312	0.7322	0.7343	0.7322	0.7343	0.7458			
2	0.7312	0.7322	0.7343	0.7458	0.7343	0.7458	0.7635			
3	0.7322	0.7343	0.7458	0.7635	0.7458	0.7635	0.7911			
4	0.7343	0.7458	0.7635	0.7911	0.7635	0.7911	0.7922			
5	0.7458	0.7635	0.7911	0.7922	0.7911	0.7922	0.8014			

表 3-6 某城市用水量预测（拟合）结果

年 份	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996
原始值/百万吨	236.48	236.64	237.85	240.37	254.52	277.21	314.89	316.22	329.91	341.27
预测值/百万吨			237.01	240.64	255.41	278.98	312.46	327.09	331.24	348.00
相对误差/%			0.35	0.11	0.35	0.64	0.77	3.4	1.34	1.97

【例 3-8】 采用 BP 网络对中国汽车保有量的预测

由于影响汽车产业的大部分因素是不确定的，因此采用常规的预测方法达不到令人满意的效果。采用人工神经网络具有广泛的适应能力、学习能力及映射能力等，因而在理论上可以逼近任何非线性函数，在多变量非线性系统的建模与预测方面通常可取得满意的结果。但是，基本的 BP 算法实质上是一种简单的快速下降静态寻优算法，在修正 $w(k)$ 时，只是按照 k 时刻的负梯度方向进行修正，而没有考虑到以前积累的经验，即以前时刻的梯度方向，从而常常使学习过程发生振荡，收敛速度缓慢。在实际应用中，有多种改进的 BP 算法，在此介绍其中一种改进算法的应用，即自适应调整学习率的改进算法，其公式为

$$w(k+1) = w(k) + \alpha(k) D(k)$$

$$\alpha(k) = 2^\gamma \alpha(k-1)$$

$$\gamma = \text{sign}[D(k)D(k-1)]$$

当连续两次迭代其梯度方向相同时，表明下降太慢，这时可使步长加倍；当连续两次迭代其梯度方向相反时，表明下降过头，这时可使步长减半。利用自适应调整方法，既可以提高学习速度，又可以改善网络的收敛性。

经实际分析，影响汽车保有量的因素很多，其中主要因素有社会总产值、道路货运

量、道路客运量、人口总数、公路里程。从统计年鉴中查得 1975 ~ 1995 年中国社会总产值、道路货运量、道路客运量、人口总数、公路里程、汽车保有量如表 3-7 所示。

表 3-7 汽车保有量及其主要的影响因素统计表

年 份	汽车保有量 ($\times 10^4/\text{veh}$)	社会总产值 ($\times 10^8/\text{元}$)	公路总里程 ($\times 10^4/\text{km}$)	公路客运量 ($\times 10^4/\text{t}$)	公路货运量 ($\times 10^4/\text{t}$)	人口总数 ($\times 10^4/\text{人}$)
1975	91.71	5379	78.36	101350	72499	92420
1976	106.70	5433	82.34	108718	74256	93717
1977	119.93	6003	85.56	122919	80833	94974
1978	135.84	6846	89.02	149229	85182	96259
1979	155.49	7642	87.58	178618	371036	97542
1980	178.28	8534	88.83	222799	382048	98705
1981	199.14	9075	89.75	261.559	363663	100072
1982	215.75	9966	90.70	300610	379205	101654
1983	232.63	11131	91.51	336965	401413	103008
1984	260.41	13171	92.67	390336	533382	104357
1985	321.12	16582	94.24	476486	538062	105851
1986	361.95	19045	96.28	544259	620113	107507
1987	408.07	23034	98.22	593682	711424	109300
1988	464.38	29807	99.96	650473	732315	111026
1989	511.32	34519	101.43	644508	733781	112704
1990	551.36	38035	102.83	648085	724040	114333
1991	606.11	43803	104.11	682681	733907	115823
1992	691.74	52602	106.26	736485	803242	117790
1993	758.09	61650	108.02	774299	836566	119584
1994	840.25	72255	109.84	814055	871273	121405
1995	931.32	84684	111.66	855852	907420	123253

在构建神经网络模型时，设社会总产值为 x_1 、道路货运量为 x_2 、道路客运量为 x_3 、人口总数为 x_4 、公路里程为 x_5 ，作为 BP 网络的输入，其网络输入层节点数为 5；设汽车保有量 y 作为 BP 网络的输出，经试验认为隐含层节点数为 10 比较合适。为了减少单隐层的节点数，在此采用双隐层，因此，网络层次结构为 5-10—10-1。其结构如图 3-18 所示。

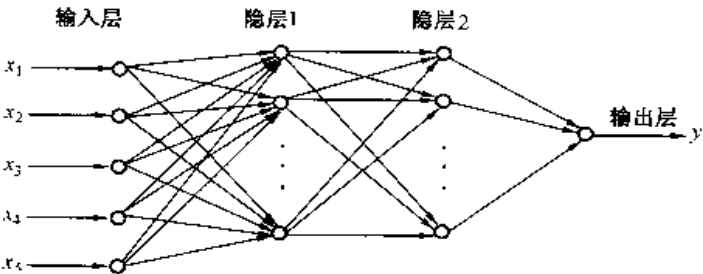


图 3-18 BP 网络结构图

将 1975 ~ 1991 年的数据作为 BP 网络训练数据，1992 ~ 1995 年的数据作为 BP 网络检验数据。为了提高网络的收敛速度和稳定性，对网络输入向量的每个量作归一化处理。其

检验结果如表 3-8 所示, 预测结果也列在表 3-8 最后两行。

表 3-8 BP 网络预测汽车保有量计算结果 ($\times 10^4/\text{veh}$)

年限	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984
实际值	91.71	106.7	119.9	135.8	155.5	178.3	199.1	215.8	232.6	260.4
模型值	90.55	104.8	121.3	134.9	157.8	176.8	198.9	213.1	236.8	259.7
年限	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
实际值	321.1	361.9	408.1	464.4	511.3	551.4	606.1	691.7	758.1	840.3
模型值	319.6	362.5	411.9	459.6	508.6	557.8	610.6	696.4	777.8	868.4
年限	1995									
实际值	931.3									
模型值	966.2									
年限	2000	2005	2010							
预测值	1572.0	2494.4	5474.5							

【例 3-9】 人工神经网络在气相色谱保留值预测上的应用

(一) 引言

人工神经网络模仿人脑处理信息, 具有自学习、自组织、自适应能力, 很强的容错能力, 分布储存与并行处理信息的功能与高度非线性表达能力, 因此在分析化学领域已有多方面的应用。保留值是气相色谱中定性组分的重要参数, 预测气相色谱各溶质的保留值对评价色谱柱的分离能力十分重要。

M. H. Abraham 提出的溶剂化模型, 根据溶质进入溶剂的过程, 将溶质与溶剂间的作用分为具有线性加合性的五种作用, 并用其总和来决定溶质的保留, 其一般的线性自由能关系为

$$\log SP = c + rR_2 + s\pi_2^H + a\Sigma\alpha_2^H + b\Sigma\beta_2^H + l\log L^{16} \quad (3-45)$$

其中 $\log SP$ 为因变量, 表示气液色谱中在给定温度下溶质的保留数据, 可以是保留体积、气液分配系数及相对保留时间。如果因变量是 Kovats 指数 I , 那么溶剂化参数模型的自由能关系则为

$$I = c' + r'R_2 + s'\pi_2^H + a'\Sigma\alpha_2^H + b'\Sigma\beta_2^H + l'\log L^{16} \quad (3-46)$$

式 (3-46) 的自变量为 R_2 、 π_2^H 、 α_2^H 、 β_2^H 、 $\log L^{16}$ 。这 5 个参数是描述溶质溶剂化作用大小的描述性参数, 对不同的溶质, 其值也不同。其中 R_2 为极化参数, 反映了溶质通过 π 或 n 电子对与溶剂作用的能力, π_2^H 为溶质的有效偶极—极化参数, $\Sigma\alpha_2^H$ 为溶质的有效氢键酸性, $\Sigma\beta_2^H$ 为有效的氢键碱性, $\log L^{16}$ 为 25℃ 下溶质在正十六烷上的气液分配系数。而 c' 、 r' 、 s' 、 a' 、 b' 和 l' 为相参数, 衡量了溶剂对溶质保留贡献的大小, 不同的固定相参数不同。 r' 常数定量衡量了溶剂与溶质 π -电子及 n -电子对作用的能力, s' 常数是溶剂参与偶极—偶极作用以及偶极—诱导偶极作用的能力, a' 常数为溶剂作为氢键碱的能力, b' 为溶剂作为氢键酸的能力, l' 常数代表了空穴效应和色散作用的组合, 它提供了衡量固定相 (溶剂) 分离同系物系列邻近组分的能力。 c' 常数最不易确定, 它体现了测定溶剂化

参数时与色谱实验时, 温度及实验条件一致性的变化且与空穴形成有关。为了与其他方法进行比较, 书中还用多元线性回归的方法建立了溶质 Kovats 指数与溶剂参数的数学模型并进行了预测工作, 并与 BP 方法进行对照比较。

下面通过列举 20 种固定相 (其中 17 种是环糊精) 的相参数, 19 种溶质的溶质参数, 以及相应的各溶质的 Kovats 保留指数等数据, 并以此数据为网络的训练样本和检验样本, 通过反向传播算法 (即 BP 算法) 进行样本的学习, 建立了两个网络模型, 并用于 Kovats 指数的预测, 取得了较好的预测效果。

(二) 数学模型的建立

1. 同一溶质, 在不同色谱柱上 Kovats 保留指数的数学模型建立与预测

使用同一种溶质 2-indanol (茛满醇) 在 20 种不同相参数的固定液上的 Kovats 保留指数, 建立了 6 种相参数与 2-indanol (茛满醇) 的 Kovats 保留指数的关系模型。原始数据见表 3-9, 建模时, 首先将原始数据归一化处理, 使其落在 -1 到 +1 之间, 见表 3-10, 再分别使用前向神经网络 (BP) 及多元线性回归方法进行建模及预测研究。

表 3-9 原始数据

数组	色谱柱	c'	r'	s'	a'	b'	l'	I
1	SQU	79.2	37.5	0	0	0	196.2	1149
2	PHA	59.1	85.9	234.8	491.8	101.7	199.8	1408
3	PH-B	261.0	141.8	211.5	472.6	0	151.4	1285
4	PH-C	94.0	99.6	262.7	513.9	188.4	192.4	1451
5	DA-A	28.7	69.0	147.0	403.1	0	208.0	1276
6	DA-B	1.2	109.7	184.3	452.6	0	212.8	1320
7	DA-C	20.7	86.4	193.9	451.3	0	209.1	1331
8	TA-A	107.4	-75.7	321.7	159.0	0	189.3	1270
9	TA-B	91.0	145.3	168.7	408.2	0	193.0	1310
10	TA-G	110.4	170.6	130.2	504.4	0	189.8	1334
11	Hydrodex	27.4	29.8	250.2	388.0	0	210.0	1347
12	Cydodex	24.9	10.0	260.3	372.3	0	210.8	1448
13	CTC-SV	61.4	43.4	220.9	440.9	0	199.5	1312
14	PBC-SV	52.9	43.7	214.6	401.5	0	202.5	1318
15	Lip-A	-17.7	52.7	233.0	384.7	0	222.0	1383
16	Lip-B	80.7	40.6	236.6	418.8	0	196.4	1347
17	Lip-C	112.1	26.7	153.3	260.5	0	188.3	1220
18	Lip-D	37.0	-248.5	623.0	290.9	0	205.6	1422
19	Lip-E	38.6	-190.3	491.4	256.7	0	204.9	1364
20	B-1:4	-19.7	114.7	184.7	441.6	0	218.4	1302

表 3-10

归一化数据

数组	色谱柱	c'	r'	s'	a'	b'	l'	I
1	SQU	-0.295	0.365	-1.0	-1.0	-1.0	0.269	-1.0
2	PHA	-0.439	0.596	-0.246	0.914	0.08	0.371	0.715
3	PH-B	1.0	0.863	-0.321	0.839	-1.0	-1.0	-0.099
4	PH-G	-0.221	0.661	-0.157	1.0	1.0	0.161	1.0
5	DA-A	-0.655	0.515	-0.528	0.569	-1.0	0.603	-0.159
6	DA-B	-0.851	0.338	-0.408	0.761	-1.0	0.739	0.132
7	DA-G	-0.712	0.598	-0.378	0.755	-1.0	0.635	0.205
8	TA-A	-0.094	-0.175	0.033	-0.381	-1.0	0.074	-0.199
9	TA-B	-0.211	0.879	-0.458	0.589	1.0	0.718	0.066
10	TA-G	-0.073	1.0	-0.582	0.963	-1.0	0.088	0.225
11	Hydrodex	-0.664	0.328	-0.197	0.510	-1.0	0.660	0.311
12	Cyclodex	-0.682	0.234	-0.164	0.449	-1.0	0.683	0.980
13	CTC-SV	-0.422	0.393	-0.291	0.716	-1.0	0.363	0.079
14	PBC-SV	-0.482	0.394	-0.311	0.563	-1.0	0.448	0.119
15	Lip-A	-0.986	0.437	-0.252	0.497	-1.0	1.0	0.550
16	Lip-B	-0.285	0.380	-0.240	0.630	-1.0	0.275	0.311
17	Lip-C	-0.061	0.313	-0.508	0.014	-1.0	0.045	-0.530
18	Lip-D	-0.596	-1.0	1.0	0.132	-1.0	0.535	0.808
19	Lip-E	-0.585	-0.722	0.578	-0.001	-1.0	0.516	0.424
20	B-1:4	-1.0	0.733	-0.407	0.719	-1.0	0.898	0.013

(1) 使用不包含预测值 (DA-B, Cyclodex, Lip-E 三种固定相上的 Kovats 指数) 在内的其余 17 组数据建模, 然后对 2-indanol 在 DA-B, Cyclodex, Lip-E 三种固定相上的 Kovats 指数进行预测, 并与实测结果进行比较。结果见表 3-11。

1) 采用多元线性回归方法, 对不包含预测值在内的 17 组数据建模, 其回归方程为

$$\hat{I} = c_0 + c' + r'R_2 + s'\pi_2^H + a'\Sigma\alpha_2^H + b'\Sigma\beta_2^H + l'\log L^{16} \quad (3-47)$$

式中, $c_0 = 0.440$; $c' = 3.411$; $r' = 0.336$; $s' = 0.999$; $a' = 0.310$; $b' = 0.380$; $l' = 3.795$ 。残差平方和

$$Q = \sum_{i=1}^m (I - \hat{I})^2 = 0.103$$

回归平方和

$$U = \sum_{i=1}^m (\hat{I} - \bar{I})^2 = 3.678$$

复相关系数

$$R = \sqrt{\frac{U}{U+Q}} = \sqrt{1 - \frac{Q}{U+Q}} = 0.986$$

使用回归方程 (3-47) 对 2-indanol 在 DA-B、Cyclodex、Lip-E 三种固定相上的 Kovats 指数同时进行 3 点预测, 结果见表 3-11。

2) 采用前向神经网络, 使用同上的 17 组数据建立 BP 模型, 其网络结构为 6-9-1 (即 6 个输入节点, 9 个隐层节点, 1 个输出节点), 如图 3-19 所示。

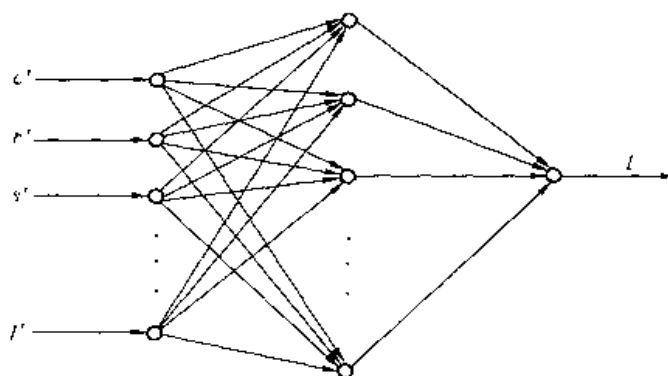


图 3-19 BP 预测网络结构图

将 17 组数据输入, 设学习速率 $l = 0.01$, 经过 12000 次迭代后, 得到网络的权值和阈值。依此网络模型, 对 DA-B、Cyclodex、Lip-E 三种固定相上的 Kovats 指数同时进行 3 点预测, 结果见表 3-11。

表 3-11 采用线性回归和 BP 网络对 DA-B、Cyclodex、Lip-E 三种固定相上的 Kovats 指数预测

溶 质	固 定 相	I (实验值)	线性回归预测 \hat{I}	BP 网络预测 \hat{I}
2-indanol	DA-B	1320	1285.448	1312.336
	Cyclodex	1448	1357.319	1384.816
	Lip-E	1364	1353.952	1364.099

(2) 类似地, 使用不包含预测值 (PH-G, CTC-SV, Lip-C) 在内的另外 17 组数据, 分别使用多元线性回归及 BP 网络建模。然后对 PH-G, CTC-SV, Lip-C 三种固定相上的 Kovats 指数进行 3 点预测。结果见表 3-12。

1) 采用多元线性回归方法, 对不包含预测值在内的 17 组数据建模, 其回归方程为

$$\hat{I} = c_0 + c' + r'R_2 + s'\pi_2^H + a'\sum\alpha_2^H + b'\sum\beta_2^H + l'\log L^{16} \quad (3-48)$$

式中, $c_0 = 0.575$; $c' = 4.641$; $r' = -0.172$; $s' = 0.528$; $a' = 0.555$; $b' = 0.455$; $l' = 5.032$ 。

残差平方和 $Q = 0.354$; 回归平方和 $U = 2.968$; 复相关系数 $R = 0.945$ 。

使用具回归方程 (3-48) 对 PH-G、CTC-SV、Lip-C 三种固定相上的 Kovats 指数同时进行 3 点预测, 结果见表 3-12。

2) 采用前向神经网络, 使用同上的 17 组数据建立 BP 模型, 其网络结构为 6-6-1, 将 17 组数据输入, 设学习速率 $l = 0.01$, 经过 8000 次迭代后, 得到网络的权值和阈值。依此网络模型, 对 PH-G、CTC-SV、Lip-C 三种固定相上的 Kovats 指数同时进行 3 点预测, 结果见表 3-12。

表 3-12 采用线性回归和 BP 网络对 PH-G、CTC-SV、Lip-C 三种固定相上的指数预测

溶 质	固 定 相	I (实验值)	线性回归预测 \hat{I}	BP 网络预测 \hat{I}
2-indanol	PH-G	1451	1477.108	1458.943
	CTC-SV	1312	1324.794	1336.421
	Lip-C	1220	1262.114	1206.078

2. 在同一色谱柱上, 不同物质 Kovats 保留指数的数学模型建立与预测

在同一种固定相 2, 6-二戊基-3-乙酰基- β -环糊精 (DA-B) 上, 以 19 溶质的五个描述性溶质参数作为输入, 以溶质的 Kovats 保留指数作为输出建立模型。原始数据见表 3-13, 将原始实验数据归一化处理, 使其落在 -1 到 +1 之间, 见表 3-14。

表 3-13 原始数据

序号	不同物质	R_2	π_2^H	$\Sigma\alpha_2^H$	$\Sigma\beta_2^H$	$\log L^{16}$	I
1	2-pentanone	0.143	0.68	0	0.51	2.755	713.5
2	1-butanol	0.224	0.42	0.37	0.48	2.601	778.5
3	1-nitropropane	0.242	0.95	0	0.31	2.894	800
4	benzene	0.610	0.52	0	0.14	2.786	742.8
5	pyridine	0.631	0.84	0	0.52	3.022	859.9
6	Limonene	0.488	0.30	0	0.14	4.693	1118
7	exo-2-bromonorbornane	0.718	0.67	0	0.04	4.675	1260
8	β -butyrolactone	0.303	1.21	0	0.46	3.245	957
9	α -methyl- α butyrolactone	0.340	1.23	0	0.51	4.070	1124
10	pantolactone	0.462	1.25	0.48	0.60	4.250	1445
11	2-aminoheptane (TFA)	-0.240	0.47	0.43	0.43	4.520	1194
12	2-aminorbornane (TFA)	0.108	0.72	0.35	0.37	4.975	1369
13	α -phenylethylamine (TFA)	0.356	0.83	0.21	0.85	5.405	1413
14	α -phenylethanol (TFA)	0.354	0.69	0	0.45	4.170	1069
15	2-indanol (TFA)	0.544	0.68	0	0.50	5.375	1320
16	2-amino-1-methoxypropane (TFA)	-0.210	0.13	0.28	0.43	4.380	1066
17	1-amino-2-propanol (di TFA)	-0.430	1.01	0.55	0.67	3.690	1361
18	2-amino-1-propenol (di TFA)	-0.430	0.91	0.93	0.77	3.815	1374
19	tetrahydro-2- (OCH ₂ C \equiv CH) pyran	0.406	0.49	0.15	0.05	4.50	1106

表 3-14 归一化数据

数组	不同物质	R_2	π_2^H	$\Sigma\alpha_2^H$	$\Sigma\beta_2^H$	$\log L^{16}$	I
1	2-pentanone	-0.0017	-0.018	-1.0	0.160	-0.890	-1.0
2	1-butanol	0.139	-0.482	-0.204	0.086	-1	-0.822
3	1-nitropropane	0.171	0.464	-1.0	-0.333	-0.791	-0.763
4	benzene	0.812	-0.304	-1.0	-0.753	-0.868	-0.920
5	pyridine	0.848	0.268	-1.0	0.185	-0.699	-0.600
6	Limonene	0.599	-0.696	-1.0	-0.753	0.492	0.106
7	exo-2-bromonorbornane	1.0	-0.036	-1.0	-1.0	0.479	0.494
8	β -butyrolactone	0.277	0.929	-1.0	0.037	-0.541	-0.334
9	α -methyl- α butyrolactone	0.341	0.964	-1.0	0.160	0.478	0.122
10	pantolactone	0.554	1.0	0.032	0.383	0.176	1.0

续表

数组	不同物质	R_2	π_2^H	$\Sigma \alpha_2^H$	$\Sigma \beta_2^H$	$\log L^{16}$	I
11	2-aminoheptane (TFA)	-0.669	-0.393	-0.075	-0.037	0.369	0.314
12	2-aminobornane (TFA)	-0.063	0.054	-0.247	-0.185	0.693	0.792
13	α -phenylethylamine (TFA)	0.369	0.250	-0.548	1.0	1.0	0.913
14	α -phenylethanol (TFA)	0.366	0.0	-1.0	0.012	0.119	0.028
15	2-indanol (TFA)	0.714	-0.018	-1.0	0.136	0.979	0.658
16	2-amino-1-methoxypropane (TFA)	-0.617	-1.0	-0.398	-0.037	0.269	0.036
17	1-amino-2-propanol (di TFA)	-1.0	0.571	0.183	0.555	-0.223	0.770
18	2-amino-1-propapenol (diTFA)	-1.0	0.393	1.0	0.802	-0.134	0.806
19	tetrahydro-2- (OCH ₂ C≡CH) pyran	0.456	-0.357	-0.677	-0.975	0.354	0.073

(1) 使用不包含预测值 [β -butyrolactone (丁酸内酯)、2-indanol (TFA) (茚满醇) 的 Kovats 保留指数] 在内的 17 组数据分别用多元线性回归及 BP 网络建模, 然后对 β -butyrolactone 和 2-indanol (TFA) 的 Kovats 保留指数同时进行 2 点预测, 并与实测结果进行比较。结果见表 3-15。

1) 采用多元线性回归方法, 对不包含预测值在内的 17 组数据建模, 其回归方程为

$$\hat{I} = c_0 + r'R_2 + s'\pi_2^H + a'\Sigma\alpha_2^H + b'\Sigma\beta_2^H + l'\log L^{16} \quad (3-49)$$

式中, $c_0 = 0.403$; $r' = 0.118$; $s' = 0.327$; $a' = 0.675$; $b' = -0.064$; $l' = 0.798$ 。

残差平方和 $Q = 0.414$; 回归平方和 $U = 7.042$; 复相关系数 $R = 0.972$ 。

通过回归方程 (3-49) 对 β -butyrolactone、2-indanol (TFA) 的保留指数同时进行 2 点预测, 结果如表 3-15 所示。

2) 采用前向神经网络, 使用同上的 17 组数据建立 BP 模型, 网络结构为 5-6-1 (即 5 个输入节点, 6 个隐层节点, 1 个输出节点)。将 17 组数据输入, 设学习速率 $l = 0.01$, 经过 5000 次迭代后, 得到网络的权值和阈值。依此网络模型, 对 β -butyrolactone、2-indanol (TFA) 的保留指数同时进行 2 点预测, 结果见表 3-15。

表 3-15 采用线性回归和 BP 网络对 β -butyrolactone、2-indanol (TFA) 的保留指数预测

固定相	溶质	I (实验值)	线性回归预测 \hat{I}	BP 网络预测 \hat{I}
DA-B	β -butyrolactone (β -丁酸内酯)	957	944.068	963.197
	1-indanol (TFA) (茚满醇)	1320	1290.983	1337.286

(2) 类似地, 使用不包含预测值 [Pantolactone (戊酸内酯)、2-amino-1-methoxypropane (2-氨基-1-甲氧基丙烷)] 在内的另外 17 组数据, 分别使用多元线性回归方法及 BP 网络建模。然后对 Pantolactone、2-amino-1-methoxypropane 的保留指数同时进行 2 点预测。结果见表 3-16。

1) 采用多元线性回归方法, 对不包含预测值在内的 17 组数据建模, 其回归方程为

$$\hat{I} = c_0 + r'R_2 + s'\pi_2^H + a'\Sigma\alpha_2^H + b'\Sigma\beta_2^H + l'\log L^{16} \quad (3-50)$$

式中: $c_0 = 0.380$; $r' = 0.078$; $s' = 0.305$; $a' = 0.617$; $b' = -0.051$; $l' = 0.807$ 。

残差平方和 $Q = 0.413$; 回归平方和 $U = 6.655$; 复相关系数 $R = 0.970$ 。

使用回归方程 (3-50) 对 Pantolactone、2-amino-1-methoxypropane 的保留指数同时进行 2 点预测, 结果如表 3-16 所示。

2) 采用前向神经网络, 使用同上的 17 组数据建立 BP 模型, 网络结构为 5-6-1。将 17 组数据输入, 设学习速率 $l = 0.01$, 经过 8000 次迭代后, 得到网络的权值和阈值。依此网络模型, 对 Pantolactone、2-amino-1-methoxypropane 的保留指数同时进行 2 点预测, 结果见表 3-16。

表 3-16 采用线性回归和 BP 网络对 Pantolactone、2-amino-1-methoxypropane 的保留指数预测

固 定 相	溶 质	I (实验值)	线性回归预测 \hat{I}	BP 网络预测 \hat{I}
DA-B	Pantolactone 戊酸内酯	1445	1397.599	1540.936
	2-amino-1-methoxypropane (TFA) (2-氨基-1 甲氧基丙烷)	1066	1079.352	1056.098

(三) 结论

本例采用多元线性回归和 BP 网络两种方法分别建立了溶质 Kovats 指数与固定相的参数及其与溶质的溶剂化参数的数学模型, 并用所建模型对实验值进行多点预测, 结果如表 3-11 ~ 表 3-12、3-15 ~ 表 3-16 所示, 实验结果表明: 使用人工神经网络可获得较理想的预测结果, 这说明了溶剂化参数模型建立的合理性, 同时也说明了人工神经网络可用于色谱分析保留值的预测, 从上述研究结果可以看出, 人工神经网络是预测色谱保留值十分有用的方法, 使用溶剂化参数模型与人工神经网络相结合, 对色谱分析及其他方面的应用研究具有较好的实际应用价值。

【例 3-10】 提高 BP 网络泛化能力的一种方法

(一) 引言

目前具有代表性的网络模型有感知器、多层映射 BP 网络、RBF 网络、双向联想记忆 (BAM)、Hopfield 模型等。但最常用的仍是 BP 网络, BP 网络不仅被广泛用于化学工程领域, 在分析化学领域也有许多应用。然而, 人们在应用 BP 网络作为预测模型时, 应考虑其不足之处或如何提高网络的预测精度。本例在应用 BP 网络时, 首先将其与 LR (基于最小二乘法的线性回归) 模型进行比较, 认为在有些情况下, LR 模型预测的结果优于 BP 网络, 但对于目标值突变或所讨论问题的非线性度较强时, 其预测结果偏差较大。经过训练后的 BP 网络对目标样本值有较好的拟合, 而对预测值, 往往偏差较大, 其主要原因是网络存在过拟合现象。如何提高 BP 网络的泛化能力, 改善其预测精度, 这是 BP 网络在实际应用中急待解决的主要问题之一。

(二) 预测模型的建立

1. BP 网络模型结构

在此设 BP 网络结构为 $x-6-1$ (即 x 个输入节点, 6 个隐层节点, 1 个输出节点), x 的大小由自变量的个数确定, 隐层中的神经元均采用 tan-sigmoid 型传递函数, 输出层的神经元采用线性传递函数, 如图 3-20 所示。

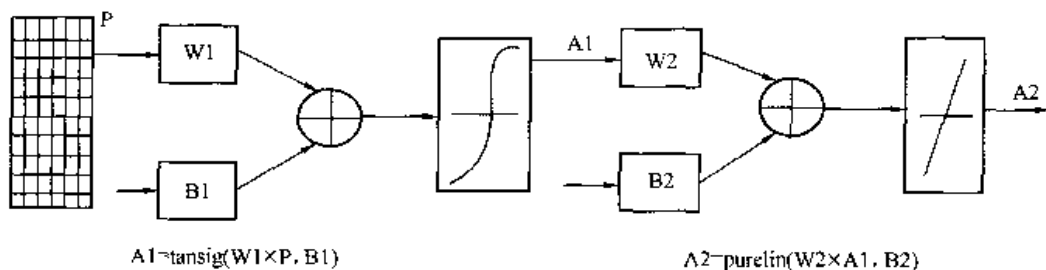


图 3-20 BP 网络结构图

2. 多元线性回归模型

基于最小二乘法的多元线性回归 (简称 LR) 模型如下所示

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n \quad (3-51)$$

式中 b_0, b_1, \cdots, b_n ——回归系数。

由最小二乘法求出式 (3-51) 的各回归系是否可信, 须进行方差分析, 方差分析就是要计算如下几个统计量:

(1) 残差平方和

$$Q = \sum_{i=1}^m (Y_i - \hat{Y}_i)^2 \quad (3-52)$$

Q 值的大小反映了所有观测点距回归曲线的残差平方和, 根据最小二乘原则, 这个量是在所有类似的曲线中与观测点距离的平方和最小的一个, 它是除了 X_i 对 Y 的线性影响之外的一切因素 (包括 X_i 对 Y 的非线性影响及测量误差等) 对 Y 变差的作用。Y 的这部分变差不能通过选择 X_i 取值而加以改变, 因此是不可控部分。

(2) 回归平方和

$$U = \sum_{i=1}^m (\hat{Y}_i - \bar{Y})^2 \quad (3-53)$$

U 值的大小反映了 Y 总的变差中由于 X_i 和 Y 线性关系而引起的可变化部分, U 愈大 (或 Q 愈小) 则表示 Y 与这些自变量的线性关系愈密切, 回归的规律性愈强, 回归出来的结果可信度愈高。Y 的这部分变差是可以通过控制 X_i 的取值而避免的, 因此属于可控部分。

(3) 剩余标准差

$$S = \sqrt{\frac{Q}{m - n - 1}} \quad (3-54)$$

S 的大小可以看作是排除了 X_i 对 Y 的线性影响之后 (或者说当 X_i 取固定值时) 衡量 Y 值随机波动的大小的一个估计值。

(4) 相关系数

$$R = \sqrt{\frac{U}{U + Q}} = \sqrt{\frac{U}{A_y}} \quad (3-55)$$

式 (3-55) 中的残差平方和 Q 与回归平方和 U 相加之和, 称为 Y 的离差平方和, 记作 A_y 。 R^2 实质上是 Y 的回归平方和 U 与 Y 的离差平方和 A_y 的比值, 这个比值大小表示

了 Y 与 X_1, X_2, \dots, X_n 的线性关系的密切程度。显然 $0 \leq R \leq 1$, R 越接近于 1, 表明 Y 与 X_1, X_2, \dots, X_n 的线性关系越强。

3. 采用模拟辅助样本的 BP 网络

(1) 模拟辅助样本的产生。将用于训练网络的样本数据, 用最小二乘法建立多元线性回归模型, 然后用随机搜索法 (简称 IJ 法) 产生模拟辅助样本。设目标函数为

$$y = f(x_1, x_2, \dots, x_n) = b_0 + b_1 x_1 + \dots + b_n x_n \quad (3-56)$$

满足约束条件

$$\left. \begin{array}{l} -1 \leq y \leq +1 \\ -1 \leq x_i \leq +1 \end{array} \right\} i = 1, 2, \dots, n \quad (3-57)$$

随机搜索法的迭代步骤可参考相关文献。在满足约束条件下, 从产生的随机数据中挑选 3~5 组数据作为模拟辅助样本, 数据组的多少可依据研究对象的自变量数目、线性关系的强弱来确定。在实际应用中, 待预测值往往是未知的或不能实时测定的, 因而训练样本不可能包含问题的全部模式。将模拟辅助样本加入到 BP 网络样本训练集中, 以提高网络的预报能力。

(2) 网络结构。引入辅助样本的 BP 网络结构与基本 BP 网络相同, 如图 3-20 所示。区别之处是输入向量不同, 即基本 BP 网络的输入数据为 Q_1 组, 而引入辅助样本的 BP 网络的输入数据为 Q_2 组, $Q_2 = Q_1 + Q_a$, Q_a 组数据即为引入的辅助样本。

(三) 预测实例及结果分析

首先使用上述三种方法, 即多元线性回归模型、BP 网络及采用模拟样本的 BP 网络对表 3-9、表 3-13 中的数据建立模型并进行预测如下:

举例 1 2-茚满醇在 20 种不同参数固定液上的保留值

使用同一种溶质 2-indanol (茚满醇) 在 20 种不同相参数的固定液上的保留指数, 建立了 6 种相参数与 2-indanol (茚满醇) 的保留指数的关系模型, 即 6 个输入与 1 个输出的关系。原始数据参见表 3-9, 建模时, 通常将原始数据归一化处理, 使其落在 -1 到 +1 之间, 见表 3-10。

每次顺序留下 1 组数据作为预测, 取其余 19 组数据训练网络, 设网络为 6-6-1 结构, 学习速率为 0.01, 迭代最大次数为 8000。其结果见表 3-17。同样地, 采用多元线性回归及采用辅助样本数据的 BP 网络对 2-indanol (茚满醇) 在各个固定相上的保留指数的预测结果也列于表 3-17 (设线性回归预测保留值为 \hat{I}_{LR} 、基本 BP 网络预测保留值为 \hat{I}_{BP} 、引入模拟样本预测保留值为 \hat{I}_{IBP}), 三种方法的预测误差平方和如下:

$$\text{采用多元线性回归方法预测误差平方和为 } e_{LR} = \sum_{i=1}^{20} (I - \hat{I}_{LR})^2 = 1.0053$$

$$\text{采用 BP 网络的预测误差平方和为 } e_{BP} = \sum_{i=1}^{20} (I - \hat{I}_{BP})^2 = 2.6103$$

$$\text{基于 BP 网络采用模拟样本数据的预测误差平方和为 } e_{IBP} = \sum_{i=1}^{20} (I - \hat{I}_{IBP})^2 = 0.7877$$

比较结果可知, 基于 BP 网络采用模拟样本数据的误差平方和最小, BP 网络的误差平方和最大, 多元线性回归的误差平方和居中。

表 3-17

2-萜满醇在 20 种不同参数固定液上的保留值预测

数 组	固 定 相	I (实验值)	线性回归预测 \hat{I}_{LR}	BP 网络预测 \hat{I}_{BP}	加入辅助样本的 BP 网络预测 \hat{I}_{IBP}
1	SQU	-1.0	-1.4547	-0.3522	-0.8402
2	PH-A	0.715	0.6154	0.5925	0.5661
3	PH-B	-0.099	-0.3133	-0.5359	0.1557
4	PH-G	1.0	1.1856	1.1887	0.7749
5	DA-A	-0.159	0.0615	-0.0974	-0.0219
6	DA-B	0.132	-0.1202	0.0097	0.1147
7	DA-G	0.205	0.1175	-0.0041	0.0194
8	TA-A	-0.199	-0.1046	-0.4402	-0.2162
9	TA-B	0.066	-0.0470	0.6529	0.0308
10	TA-G	0.225	0.2670	0.2885	0.1574
11	Hydrodex	0.311	0.5031	0.8214	0.461
12	Cydodex	0.980	0.3737	0.4054	0.4112
13	CTC-SV	0.079	0.1406	0.2399	0.1661
14	PBC-SV	0.119	0.1969	0.2010	0.1667
15	Lip-A	0.550	0.7393	0.8266	0.4709
16	Lip-B	0.311	0.3023	0.1405	0.1645
17	Lip-C	-0.530	-0.2615	-0.7313	-0.3427
18	Lip-D	0.808	0.8442	0.4522	0.5639
19	Lip-E	0.424	0.4122	1.1857	0.6668
20	B-1:4	0.013	0.2158	0.0753	-0.028

举例 2 19 种不同物质在同一固定液上的保留值

在同一种固定相 2, 6-二戊基-3-乙酰基- β -环糊精 (DA-B) 上, 以各个溶质的五个描述性溶质参数作为输入, 以溶质的保留指数作为输出建立模型。原始数据参见表 3-13, 将原始实验数据归一化处理, 使其落在 -1 到 +1 之间, 见表 3-14。

每次顺序留下 1 组数据作为预测, 取其余 18 组数据训练网络, 设网络为 5-6-1 结构, 学习速率为 0.01, 迭代最大次数为 8000。其结果如表 3-18 所示。类似地, 使用 LR 和使用模拟样本数据的 BP 网络的预测结果也列于表 3-18。三种方法预测的误差平方和如下:

采用多元线性回归方法的预测误差平方和为 $e_{LR} = \sum_{i=1}^{19} (I - \hat{I}_{LR})^2 = 0.9380$

采用 BP 网络的预测误差平方和为 $e_{BP} = \sum_{i=1}^{19} (I - \hat{I}_{BP})^2 = 1.9553$

基于 BP 网络采用模拟样本数据的预测误差平方和为 $e_{IBP} = \sum_{i=1}^{19} (I - \hat{I}_{IBP})^2 = 0.6148$

比较结果可知, 基于 BP 网络采用模拟样本数据预测的误差平方和最小, BP 网络最大, LR 模型居中。

表 3-18

在同一种固定相 (DA-B) 上, 不同物质保留值的预测

数 组	溶 质	实验值 I	线性回归预测 \hat{I}_{LR}	BP 网络预测 \hat{I}_{BP}	加入辅助样本的 BP 网络预测 \hat{I}_{IBP}
1	2-pentanone	-1.0	-0.9902	-0.9718	-0.9528
2	1-butanol	-0.822	-0.5872	-0.2886	-0.4035
3	1-nitropropane	-0.763	-0.6865	-0.8989	-0.9139
4	benzene	-0.920	-0.9218	-1.1210	-0.7967
5	pyridine	-0.600	-0.6627	-0.5383	-0.5637
6	Limonene	0.106	-0.0054	-0.2288	0.1088
7	exo-2-bromonorbornane	0.494	0.1898	0.3726	0.3435
8	β -butyrolactone	-0.334	-0.3708	-0.3294	-0.2763
9	α -methyl- α butyrolactone	0.122	0.6426	0.7957	0.5558
10	pantolactone	1.0	0.8865	0.3457	1.1200
11	2-aminoheptane (TFA)	0.314	0.4796	0.3845	0.3877
12	2-aminorbornane (TFA)	0.792	0.8224	0.5639	0.6151
13	α -phenylethylamine (TFA)	0.913	0.9183	1.4054	0.9225
14	α -phenylethanol (TFA)	-0.028	-0.1337	-0.0572	-0.0671
15	2-indanol (TFA)	0.658	0.5777	0.6574	0.5657
16	2-amino-1-methoxypropane (TFA)	-0.036	-0.0530	-0.1135	-0.1962
17	1-amino-2-propanol (di TFA)	0.770	0.1898	0.3937	0.5703
18	2-amino-1-propaenol (di TFA)	0.806	1.0370	0.5779	1.0375
19	tetrahydro-2- (OCH ₂ C \equiv CH) pyran	0.073	0.2886	0.3869	0.1329

举例 3 脂肪胺的色谱保留值。

脂肪胺的色谱保留值及相关参数见表 3-19。以特征变量 (R_D , $I \times 10^{-3}$, 1x , $\lg p$) 为自变量, 以保留值 [$\lg (V_g^T/\text{cm}^3)$] 为因变量。每次顺序留下 1 组数据作为预测, 取其余 11 组数据训练网络, 设网络为 4-6-1 结构, 学习速率为 0.01, 迭代最大次数为 8000。其结果如表 3-20 所示 [在此设保留值 $I = \lg (V_g^T/\text{cm}^3)$]。同样地, 使用 LR 和使用模拟样本数据的 BP 网络的预测结果也列于表 3-20。三种方法预测的误差平方和如下:

采用多元线性回归方法的预测误差平方和为 $e_{LR} = \sum_{i=1}^{12} (I - \hat{I}_{LR})^2 = 0.0758$

采用 BP 网络的预测误差平方和为 $e_{BP} = \sum_{i=1}^{12} (I - \hat{I}_{BP})^2 = 0.0562$

基于 BP 网络采用模拟样本数据的预测误差平方和为 $e_{IBP} = \sum_{i=1}^{12} (I - \hat{I}_{IBP})^2 = 0.0440$

比较结果可知, 基于 BP 网络采用模拟样本数据预测的误差平方和最小, LR 模型最大, BP 网络居中。

表 3-19

脂肪酸的色谱保留值及相关参数

序号	胺	$\lg(V_R'/\text{cm}^3)$	R_D	$I \times 10^{-3}$	I_x	$\lg p$
1	乙胺	1.25	14.80	12.37	1.414	-0.0191
2	叔丁胺	1.20	19.54	15.72	1.732	0.2699
3	正丙胺	1.34	19.46	13.61	1.914	0.4425
4	另丁基胺	1.29	24.07	16.96	2.270	0.7310
5	异丁胺	1.34	24.30	14.85	2.270	0.7914
6	叔丁胺	1.30	28.30	20.30	2.561	0.8873
7	正丁胺	1.59	24.16	14.07	2.414	0.9041
8	甲基正丁胺	1.48	28.80	17.42	2.770	1.1931
9	胺基戊胺	1.35	28.56	18.20	2.808	1.1931
10	异戊胺	1.70	28.38	14.53	2.770	1.2533
11	正戊胺	1.80	28.78	14.24	2.914	1.3657
12	正己胺	2.03	33.42	14.31	3.414	1.8273

表 3-20

脂肪酸的色谱保留值预测

数 组	胺	(实验值) $\lg(V_R'/\text{cm}^3)$	线性回归预测 \hat{I}_{LR}	BP 网络预测 \hat{I}_{BP}	加入辅助样本的 BP 网络预测 \hat{I}_{IBP}
1	乙胺	1.25	1.3027	1.2664	1.2530
2	叔丁胺	1.20	1.0632	1.3344	1.2823
3	正丙胺	1.34	1.3589	1.2592	1.3921
4	另丁基胺	1.29	1.2462	1.3189	1.2799
5	异丁胺	1.34	1.4945	1.3511	1.3473
6	叔丁胺	1.30	1.2578	1.2825	1.2723
7	正丁胺	1.59	1.5437	1.5519	1.4546
8	甲基正丁胺	1.48	1.4604	1.3849	1.4308
9	胺基戊胺	1.35	1.4966	1.4139	1.4117
10	异戊胺	1.70	1.6983	1.7102	1.7695
11	正戊胺	1.80	1.7539	1.9213	1.7676
12	正己胺	2.03	2.0154	2.0565	1.9705

结果分析

(1) 研究对象自变量与变量相关性的变化影响模型的预测精度。举例 1 研究对象的相关系数由缺第 1 组至缺第 20 组时的变化范围为 0.936 ~ 0.983, 变化的中心在 0.95 左右, 即举例 1 中数据的相关性较小; 举例 2 研究对象的相关系数由缺第 1 组至缺第 19 组时的变化范围为 0.968 ~ 0.985, 变化的中心在 0.97 左右, 举例 2 数据的相关性较大; 举例 3 研究对象的相关系数由缺第 1 组至缺第 11 组时的变化范围为 0.970 ~ 0.996, 变化的中心在 0.984 左右, 举例 3 中数据相关性最大。由以上预测结果可知, 随着从举例 1 到举例 3 相关系数 R 值的增大, BP 网络的预测精度相应提高。然而, 有些研究对象的相关系数 R 值

可能小于 0.9, 这样网络的预测精度将降低, 即研究对象自变量与因变量相关性对模型的预测精度有影响。

(2) 有时线性回归模型的预测优于基本 BP 网络。从举例 1、举例 2 可以看出, LR 模型在大部分点上的预测均优于基本 BP 网络, 但随着相关系数 R 值的增大, BP 网络的预测精度相应提高, 特别对于目标样本突变点的预测, 基本 BP 网络优于 LR 模型, 如举例 3 所示。因此, 对于相关性较弱的研究对象, 采用 LR 模型的预测效果往往优于基本 BP 网络; 而对于相关性较强的研究对象, 采用基本 BP 网络能满足预测要求。

(3) 引入模拟辅助样本明显提高网络预测精度。由上述预测结果可知, 引入模拟辅助样本的 BP 网络预测精度不仅高于基本 BP 网络, 而且也优于 LR 模型, 其根本原因在于模拟辅助样本是基于 LR 模型通过优化方法产生的, 它基本上可以代替残缺的 (或待预测的) 样本。模拟辅助样本与训练样本集的结合, 基本上能包含所研究问题的全部模式, 以使网络的泛化 (generalization) 能力有了明显提高。因此, 引入模拟辅助样本是提高网络预测精度的重要方法之一。

(四) 小结

本例在综合 LR 模型和 BP 网络模型优点的基础上, 提出增添模拟辅助样本改善网络泛化能力的思想, 基于 LR 模型运用 LJ 最优化方法产生模拟辅助样本。通过对表 3-9、表 3-13、表 3-19 的数据预测表明: 采用添加模拟辅助样本训练的网络不仅保留 BP 网络的基本特点, 而且提高了网络的稳定性及预测精度, 其预测结果优于原有 BP 网络模型和 LR 模型。对于残缺样本问题的研究, 引入模拟辅助样本, 无疑是一种可靠而又有效的方法。

【例 3-11】 采用 BP 网络对石油馏分临界性质建立数学模型。

石油馏分的临界性质是石油基础物性的重要组成部分, 常用来关联和计算石油馏分其他有关物理化学性质。下面以某油田原油的直馏宽窄馏分为例, 给出了在不同方法下对石油馏分的临界温度和临界压力与物性参数 (标准密度和平均沸点) 的关系进行建模。

(一) 原始数据

表 3-21 临界温度和临界压力与物性参数

序 号	标准密度 ρ (g/cm^3)	平均沸点 T_m (K)	临界温度 T_c (K)	临界压力 P_c (MPa)	序 号	标准密度 ρ (g/cm^3)	平均沸点 T_m (K)	临界温度 T_c (K)	临界压力 P_c (MPa)
1	0.7139	361.25	549.82	3.346	11	0.8149	591.95	760.66	1.551
2	0.7403	387.15	568.74	3.106	12	0.8274	591.75	771.28	1.567
3	0.7579	410.95	591.35	2.875	13	0.8401	624.15	808.62	1.427
4	0.7722	433.75	612.91	2.634	14	0.8452	647.15	820.06	1.401
5	0.7861	459.15	644.05	2.510	15	0.7273	378.15	552.17	3.385
6	0.7989	483.65	672.83	2.256	16	0.7869	457.45	638.62	2.483
7	0.8151	503.15	700.82	2.060	17	0.8075	541.95	726.95	1.826
8	0.8167	520.55	716.73	1.954	18	0.8115	536.25	720.41	1.892
9	0.8162	541.15	726.97	1.823	19	0.7221	369.85	547.69	3.203
10	0.8169	565.85	746.26	1.635	20	0.7375	392.65	576.12	2.872

续表

序 号	标准密度 ρ (g/cm ³)	平均沸点 T_m (K)	临界温度 T_c (K)	临界压力 P_c (MPa)	序 号	标准密度 ρ (g/cm ³)	平均沸点 T_m (K)	临界温度 T_c (K)	临界压力 P_c (MPa)
21	0.7670	439.35	620.87	2.514	47	0.7549	397.90	578.15	3.107
22	0.7810	465.15	642.70	2.235	48	0.7725	420.09	603.97	2.967
23	0.7957	488.25	673.93	2.205	49	0.7865	442.68	628.98	2.768
24	0.8069	511.95	698.58	1.936	50	0.8088	464.29	655.31	2.591
25	0.8140	537.85	710.73	1.744	51	0.8257	486.08	679.68	2.338
26	0.7125	363.25	543.27	3.344	52	0.8423	507.71	703.59	2.300
27	0.7799	460.65	644.58	2.503	53	0.8684	535.61	735.06	2.120
28	0.8079	523.25	715.32	1.862	54	0.8702	572.67	764.71	2.054
29	0.8216	587.25	747.89	1.687	55	0.8797	596.24	773.63	2.008
30	0.7302	386.75	564.57	2.942	56	0.8952	619.07	796.56	1.899
31	0.7487	411.55	590.17	2.724	57	0.9030	645.98	823.37	1.796
32	0.7640	435.35	610.38	2.568	58	0.9096	672.54	853.09	1.708
33	0.7793	459.95	656.33	2.305	59	0.9164	688.84	869.27	1.533
34	0.7900	484.75	686.65	2.132	60	0.9272	712.13	898.95	1.465
35	0.8066	500.55	705.77	2.094	61	0.9381	732.82	925.59	1.366
36	0.8089	517.35	715.20	1.880	62	0.9400	754.44	956.49	1.276
37	0.8070	535.55	724.15	1.861	63	0.7303	384.35	561.89	3.467
38	0.8100	552.85	733.51	1.739	64	0.7395	379.59	553.21	3.462
39	0.8104	569.15	743.53	1.567	65	0.7986	451.63	634.35	2.737
40	0.8076	576.95	750.68	1.562	66	0.8666	568.18	757.24	2.291
41	0.8275	606.75	776.21	1.592	67	0.9086	690.03	867.56	1.527
42	0.8323	652.35	820.36	1.349	68	0.9115	667.03	840.36	1.707
43	0.8361	683.95	837.45	1.131	69	0.9216	697.95	875.38	1.549
44	0.8005	508.45	681.16	2.019	70	0.9302	728.12	918.61	1.466
45	0.7043	360.05	527.76	3.557	71	0.9376	774.39	978.81	1.428
46	0.7341	375.78	551.13	3.371	72				

(二) 建立两个输入、单个输出——温度模型

BP神经网络结构采用三层结构：2输入节点、20个隐节点、1个输出节点，如图3-21所示。

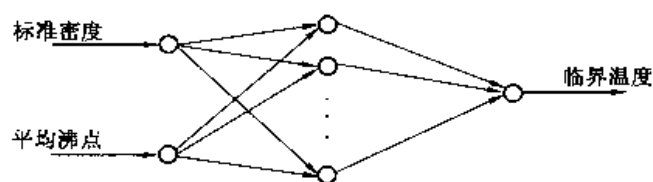


图 3-21 双输入、单输出（临界温度）BP网络

隐含层传递函数采用双曲正切函数；输出层传递函数采用S型函数。采用自适应变步长学习算法训练网络。

(三) 建立两个输入、单个输出——压力模型

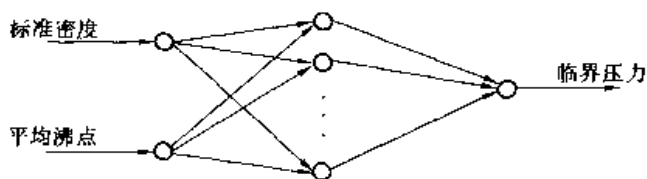


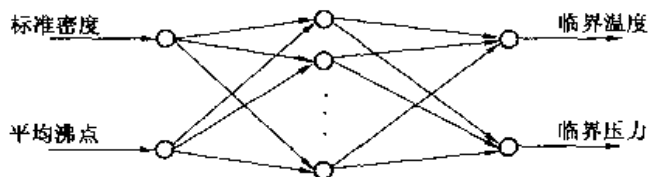
图 3-22 双输入、单输出（临界压力）BP 网络

BP 神经网络结构采用三层结构：2 输入节点、20 个隐节点、1 个输出节点，如图 3-22 所示。

隐含层传递函数采用双曲正切函数；输出层传递函数采用 S 型函数。采用自适应变步长学习算法训练网络。

（四）建立两个输入、两个输出——温度、压力模型

BP 神经网络结构采用三层结构：2 输入节点、10 个隐节点、2 个输出节点，如图 3-23 所示。



隐含层传递函数采用双曲正切函数；输出层传递函数采用 S 型函数。采用自适应变步长学习算法训练网络。

（五）网络模型拟合效果

表 3-22 双输入与单输出、双输入与双输出的拟合结果

序号	温度、压力模型		温度模型误差	压力模型误差	序号	温度、压力模型		温度模型误差	压力模型误差
	温度误差	压力误差				温度误差	压力误差		
1	9.2627	-0.0820	7.8307	-0.0833	21	0.8241	-0.0283	2.0497	-0.0228
2	3.8270	-0.0709	3.8608	-0.0800	22	-7.6472	-0.0611	-6.5679	-0.0594
3	1.6450	-0.0495	2.1384	-0.0581	23	-3.5728	0.0560	-3.2909	0.0558
4	-2.7841	-0.0498	-2.2782	-0.0552	24	-3.3174	-0.0695	-3.9483	-0.0708
5	-1.7569	0.0763	-1.3886	0.0746	25	-13.4981	-0.1026	-14.7250	-0.1078
6	-1.2888	0.0205	-1.4107	0.0215	26	1.5955	-0.0563	0.3015	-0.0531
7	4.1954	-0.1084	2.9100	-0.1034	27	-0.8878	0.1542	0.1123	0.1555
8	5.1824	-0.0624	3.4985	-0.0635	28	3.9087	-0.0488	3.3262	-0.0517
9	-0.2336	-0.0182	-1.7454	-0.0242	29	-13.1160	0.1041	-13.4468	0.1000
10	1.2218	-0.0381	0.5991	-0.0441	30	1.8334	-0.1857	2.1349	-0.1744
11	-2.3740	0.0521	-0.4643	0.0547	31	1.8162	-0.1136	2.9670	-0.1027
12	5.9664	-0.0385	4.4821	-0.0460	32	-4.8270	-0.0092	-3.5393	-0.0019
13	16.4787	-0.0993	13.9223	-0.1029	33	11.7557	-0.0477	12.7852	-0.0461
14	6.9701	-0.0425	4.3628	-0.0350	34	14.2406	-0.0008	15.0782	-0.0006
15	-3.1334	0.1386	-3.4073	0.1404	35	13.7875	-0.0239	13.1138	-0.0228
16	-5.6700	0.0170	-5.4266	0.0149	36	8.2809	-0.0936	7.4566	-0.0955
17	1.2656	0.0723	1.4279	0.0685	37	3.4054	0.0622	3.3752	0.0585
18	-2.0033	0.0560	-2.8459	0.0514	38	-0.7558	0.0414	-0.5525	0.0373
19	-0.3840	-0.1399	-1.2157	-0.1416	39	-2.4245	-0.0261	-1.1944	-0.0278
20	7.0211	-0.2023	7.5273	-0.1970	40	-0.1741	0.0417	2.4727	0.0437

续表

序号	温度、压力模型		温度模型误差	压力模型误差	序号	温度、压力模型		温度模型误差	压力模型误差
	温度误差	压力误差				温度误差	压力误差		
41	-0.0602	0.0694	-0.4851	0.0684	57	1.1762	0.0173	0.9582	0.0055
42	4.9357	0.0176	6.2170	0.0399	58	2.1337	0.0469	0.5715	0.0545
43	-11.4030	-0.0880	-10.3647	-0.0551	59	-1.5934	-0.0660	-2.4211	-0.0479
44	-15.9349	0.0416	-15.7649	0.0400	60	-1.8054	-0.0469	-0.7102	-0.0250
45	-10.2619	0.1360	-11.7858	0.1449	61	-1.4858	-0.0720	1.9178	-0.0596
46	-3.4578	0.0624	-4.0179	0.0494	62	5.8037	-0.0605	7.5562	-0.0809
47	1.3036	0.0007	1.2630	-0.0181	63	1.0900	0.3006	1.2156	0.3071
48	2.2348	0.0655	2.0610	0.0502	64	-5.2734	0.1809	-5.7170	0.1641
49	0.2893	0.0887	0.0363	0.0829	65	-7.1148	0.0815	-7.7800	0.0848
50	-2.3423	0.0226	-3.0399	0.0366	66	1.5991	0.2588	1.0208	0.2331
51	-4.2567	-0.0984	-4.6603	-0.0709	67	-3.2951	-0.0338	-6.9925	-0.0234
52	-3.5032	-0.0335	-2.8347	-0.0055	68	-4.8558	0.0075	-4.8085	0.0153
53	1.7340	-0.1413	8.0810	-0.1260	69	-7.2622	-0.0198	-6.9634	0.0029
54	5.4670	0.0286	5.4086	0.0003	70	-1.7141	0.0287	-1.0263	0.0381
55	-3.5455	0.0686	-4.2166	0.0310	71	10.0813	0.1862	9.8206	0.1303
56	-0.4333	0.0087	1.0817	-0.0207	72				

(六) 均方差分析

依据表 3-22 可知:

- (1) 两个输入、两个输出模型的临界温度、临界压力均方差分别为 6.1169、0.0929。
- (2) 两个输入、单个输出模型的临界温度均方差分别为 6.1591。
- (3) 两个输入、单个输出模型的临界压力均方差分别为 0.088846。

二、BP 网络在控制系统中的应用

【例 3-12】 BP 网络在连续搅拌反应器 (CSTR) pH 值的预测与控制。

(一) 研究对象

为了利用人工神经网络预测 CSTR 中 pH 值, 然后对其进行控制, 首先应研究 CSTR 中的动力响应。CSTR 系统如图 3-24 所示。

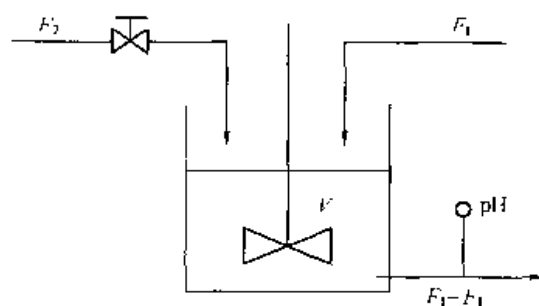


图 3-24 CSTR 中的 pH 值

CSTR 有两个输入流, 一个包含氢氧化钠, 另一个包含醋酸。当 Na^+ 和所有醋酸根 ($\text{HAC}^- + \text{AC}^-$) 达到物质平衡时, 可有下列等式。

总体醋酸根平衡式

$$F_1 C_1 - (F_1 + F_2) = V d\xi/dt \quad (3-58)$$

钠离子平衡式

$$F_2 C_2 - (F_1 + F_2) = V d\xi/dt \quad (3-59)$$

$$\xi = [\text{Na}^+]$$

$$\xi = [\text{HAC}] + [\text{AC}^-]$$

由于电中性的要求，合并 HAC 和 H_2O 的平衡关系得出

$$\xi + [\text{H}^+] = K_w / [\text{H}^+] + K_a [\text{HAC}] / [\text{H}^+] \quad (3-60)$$

CSTR 参数见表 3-23，通过在 F_2 中稳态数据中加入 2% 的伪随机二进制信号（简称 PRBS），可得到一个训练数据库。用于训练的 F_2 和 pH 响应如图 3-25 所示。在 pH 模型中，式 (3-58) 和式 (3-59) 为线性微分方程。式 (3-60) 是非线性稳态代数方程。因而模型具有高阶非线性稳态性质和实质的线性动力学特点。

表 3-23 连续搅拌反应器的有关参数

CSTR 所用的参数	数 值	CSTR 所用的参数	数 值
箱体体积 V	100/l	稳态 pH 值	9
HAC 流速	81l/min	F_1 中的 HAC 浓度	0.3178mol/l
NaOH 的稳态流速	515l/min	F_2 中的 NaOH 浓度	0.05mol/l

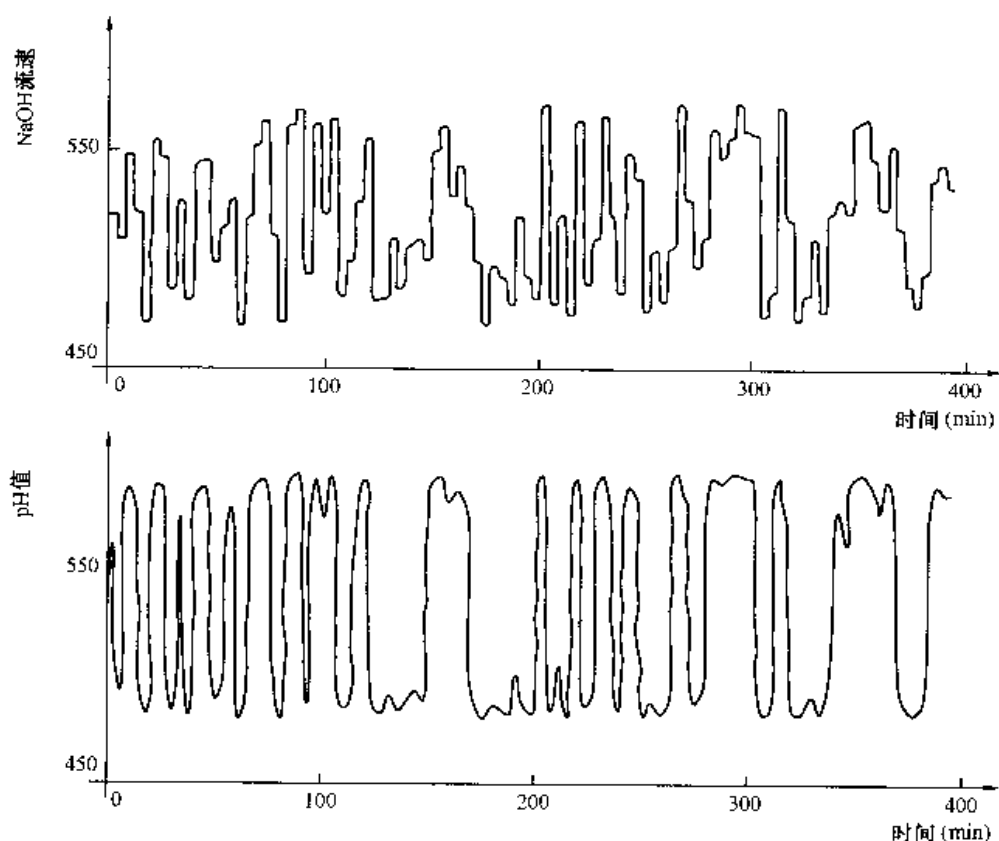


图 3-25 基于 PRBS 的 CSTR 系统测试信号

（二）BP 网络的预测

在此采用 BP 网络进行预测，首先用图 3-25 所示的数据对网络进行训练，然后由训练好的网络对实验数据进行预测，其预测结果如图 3-26 所示。网络对运行系统数据进行预测，其预测结果如图 3-27 所示。

（三）BP 网络的应用

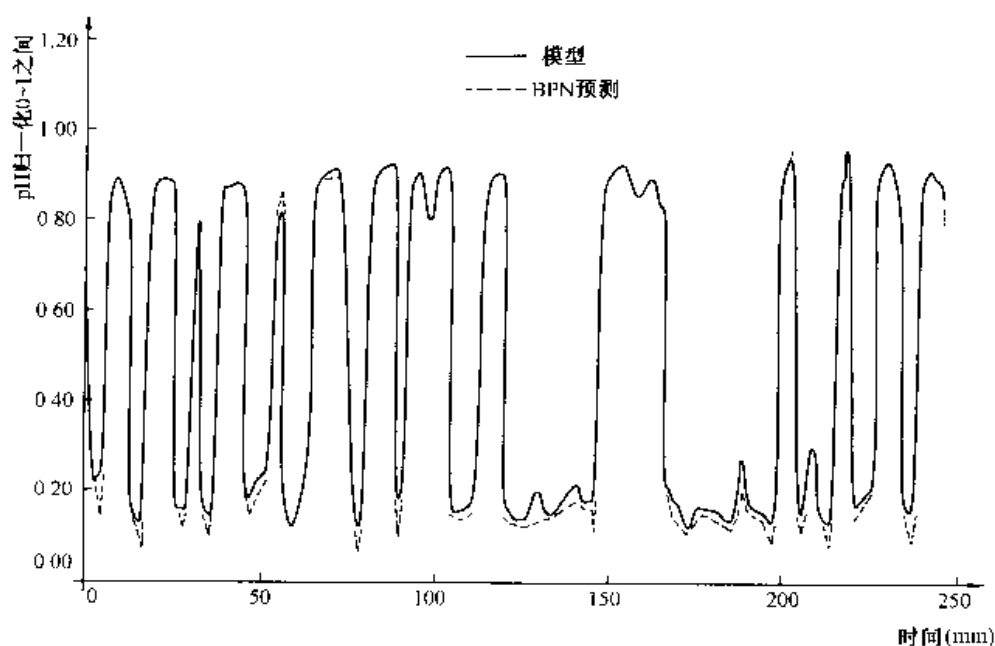


图 3-26 训练数据的预测

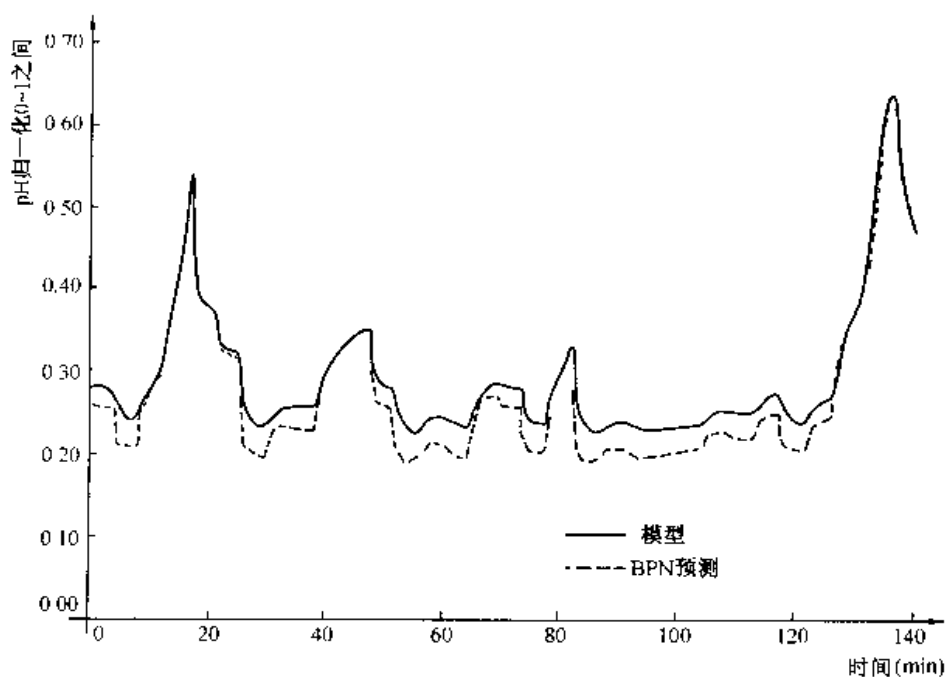


图 3-27 系统运行过程数据的预测

一旦得到训练好的 BP 网络模型，可直接用于过程控制。图 3-28 表示了如何使用神经网络模型框图构造预测控制系统。图 3-28 中的 $y(t)$ 为当前的 pH 值， $y_d(t)$ 为希望的 pH 值，从原理上讲，控制系统的优化器是朝着使性能指标最小的方向移动，例如使未来 pH 值的期望值同预测值之间的平方差和逐渐减小。

预测控制，又称基于模型的控制，图 3-28 所示的神经网络预测控制系统，其中神经网络预测器建立了非线性被控对象的预测模型，并可在线学习修正。利用此预测模型，即可由当前的控制输入 $u(t)$ ，预报出被控系统在将来一段时间范围内的输出值。图 3-28 中

的非线性优化器实际上是一个优化算法, 因此, 也可利用本书介绍的动态反馈网络, 来实现这一算法, 并进一步构成动态网络预测器。

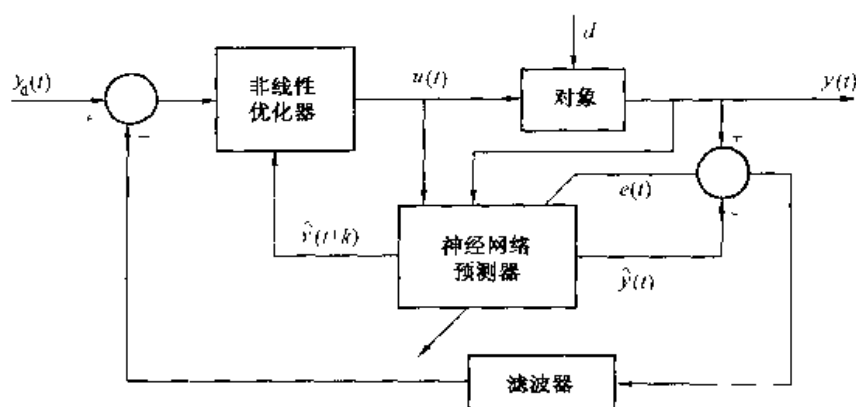


图 3-28 基于神经网络的预测控制

【例 3-13】 改进 BP 反传算法在控制系统中的应用

人工神经网络是一种大规模、并行、分布式信息处理系统, 它具有很强的自适应和自学习功能, 基于人工神经网络的控制方法为智能控制开辟了一条崭新的途径。神经网络在控制中应用最广泛的学习算法是误差反向传播 (error backpropagation) 算法 (简称 BP 算法)。本文中建立了学习速率与系统参数之间的定量关系, 在线调整学习速率, 以增强 BP 算法的自适应和自学习功能, 仿真实验结果表明该方法是有用的。

(一) 基本 BP 算法的改进

1. 基于神经网络的智能控制系统结构

神经网络智能控制系统, 通常要求两个神经网络, 一个用于系统辨识 (NNM 网络); 另一个用作控制器 (NNC 网络), 其系统结构如图 3-29 所示。

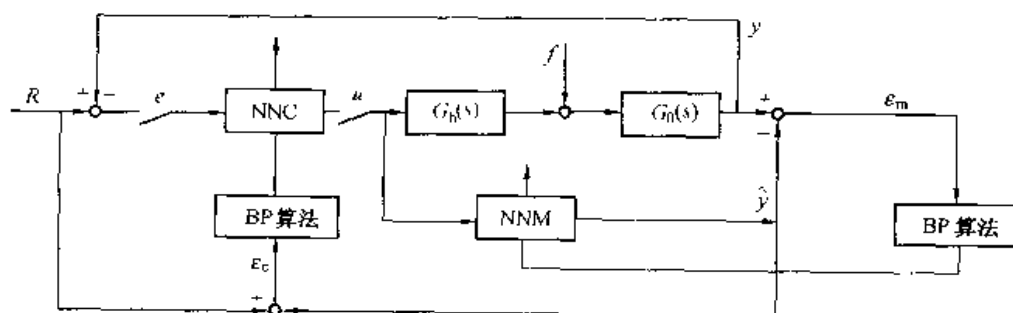


图 3-29 神经网络控制系统结构图

由于系统的控制方案不同, 所辨识的模型结构也不同, 所选择的控制规律不同, 则参数的调整方式也不同。

2. 基本的 BP 算法

神经网络不仅有输入层和输出层, 而且还有一层或多层隐层, 对于网络的输入信号, 首先向前传播到隐层节点 (或神经元), 经过作用函数后, 再把隐层节点的输出信号传播到输出层, 最后获得输出结果。BP 反传算法的学习过程是由正向传播和反向传播组成,

在正向传播过程中,输入信息从输入层经隐单元层逐层处理,最后传向输出层,每一层神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出,则转入反向传播,将误差信号沿着原来的连接道路返回,通过修改各层神经元的权值和阈值,使得误差信号最小。其目标函数为

$$E = \frac{1}{2} \sum_{k=1}^N (y_k - \hat{y}_k)^2 \quad (3-61)$$

式中 y_k ——样本输出;

\hat{y}_k ——网络输出。

设网络各节点的激发函数为 S 型函数 $f(x) = 1/[1 + \exp(-x)]$, 有 N 个外部样本 (x_k, y_k) ($k=1, 2, \dots, N$), 在各样本输入 x_k 的作用下, 逐层计算各节点的输出, 直至求出网络的输出 $O_k^m = y_k^m$, $m=1, 2, \dots, M$, M 为网络的总层数。(为叙述方便, 设网络只有一个输出) 如果网络输出不能达到期望值 (即样本输出值), 则按下式调整网络权值

$$W_{ij}^m \Big|_{new} = W_{ij}^m \Big|_{old} - l(\partial E / \partial W_{ij}^m) \quad (3-62)$$

式中 l ——学习速率;

$W_{ij}^m \Big|_{new}$ ——新网络权值;

$W_{ij}^m \Big|_{old}$ ——旧网络权值。

3. 改进的 BP 算法学习速率的基本思想

由式 (3-62) 可知, 学习速率 l 的选取对网络权值的调整有影响, 当学习速率太大时, 会引起系统响应出现振荡甚至发散; 而学习速率太小时, 将减慢搜索速度。对于受控对象的模型参数, 系统设定值发生大范围变化或系统出现较大扰动情况, 应根据参数变化对系统影响的研究, 建立学习速率与它们之间的定量关系, 以改进 BP 算法。在变参数、强扰动的系统中, 可在线实时调整学习速率, 使控制系统具有较强的适应性和鲁棒性, 从而改善系统的调节品质。调整学习速率的基本思想是:

(1) 当受控系统模型的纯滞后时间 t_0 增大时, 使调节作用落后于被调参数的变化, 所以容易引起超调, 产生振荡。因此, 相应减小学习速率, 减缓调节过程, 可提高系统的稳定性。

(2) 对于二阶加纯滞后对象, 时间常数 t_1 (或 t_2) 逐渐增大, 这相当于对象的容量滞后更显著, 致使调节作用落后于被调参数的变化, 所以容易引起超调, 产生振荡。相应减小学习速率, 减缓调节过程, 可提高系统的稳定性。

(3) 增益大的调节对象, 调节起来比较灵敏, 但稳定性较差。绝大多数化工对象的静态特性不是线性的, 当增益改变时, 适当调整学习速率, 可改善系统的调节品质。

(4) 当系统出现外部扰动 (或改变设定值) 时, 应根据系统输出的变化趋势相应调整学习速率, 以缩短调节时间, 减小超调量, 提高系统的跟踪能力。

(二) 仿真研究

1. 对象模型辨识

设受控对象模型为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k_0 e^{-t_0 s}}{(1 + t_1 s)(1 + t_2 s)} \quad (3-63)$$

式(3-63)通过Z变换写成差分方程形式

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + k_0 [b_1 u(n-m_1-1) + b_2 u(n-m_1-2)] \quad (3-64)$$

式中 T ——采样周期;

t_1, t_2 ——时间常数;

k_0 ——模型增益;

t_0 ——纯滞后时间;

n ——采样序号, $n = 1, 2, \dots$ 。

$$f_0 = t_2 - t_1$$

$$m_1 = \text{int}(t_0/T)$$

$$q_1 = T/t_1$$

$$q_2 = T/t_2$$

$$c_1 = e^{-q_1}$$

$$c_2 = e^{-q_2}$$

$$a_1 = c_1 + c_2$$

$$a_2 = c_1 c_2$$

$$b_1 = 1 + (t_1 c_1 - t_2 c_2) / f_0$$

$$b_2 = a_2 - (t_1 c_2 - t_2 c_1) / f_0$$

在此, 设纯滞后时间 t_0 是采样周期 T 的整数倍, 即 $t_0 = m_1 T$, 由式(3-64)可知网络的输入节点数为4, 即输入向量为 $p = [y(n-1) y(n-2) u(n-m_1-1) u(n-m_1-2)]$, 设隐层节点数为5, 一个输出节点, 此时受控对象模型网络NNM为4-5-1结构, 如图3-30所示。

2. 控制器参数调整网络

在此, 采用具有PID结构的NNC控制网络, 如图3-31所示。设常规的PID位置型算式的差分方程为

$$u(n) = k_p \left\{ e(n) + \frac{T}{T_i} \sum_{j=0}^n e(j) + \frac{T_d}{T} [e(n) - e(n-1)] \right\} \quad (3-65)$$

$$e(n) = R - y(n)$$

式中 $e(n)$ ——系统偏差;

R ——系统设定值;

$y(n)$ ——系统当前输出;

T ——采样周期;

n ——采样序号, $n = 1, 2, \dots$;

T_i ——积分时间;

T_d ——微分时间。

写成脉冲传递函数的型式

$$\frac{U(z)}{E(z)} = \left[k_p + k_i \frac{1}{1 - z^{-1}} + k_d (1 - z^{-1}) \right] \quad (3-66)$$

$$k_i = \frac{k_p T}{T_i}$$

$$k_d = \frac{T_d}{T}$$

式中 k_p ——比例增益;

k_i ——积分增益;

k_d ——微分增益。

设 $w_{c1} = k_p$, $w_{c2} = k_i$, $w_{c3} = k_d$, w_{c4} (为一放大器增益)。本例采用了 3-1-1 三层线性网络, 其输入层有 3 个节点, 如图 3-30 所示。采用 BP 反传学习算法调整控制器网络权值 (w_{c1} , w_{c2} , w_{c3} , w_{c4}), 使受控对象输出 y 与给定值 R 之间的误差最小。

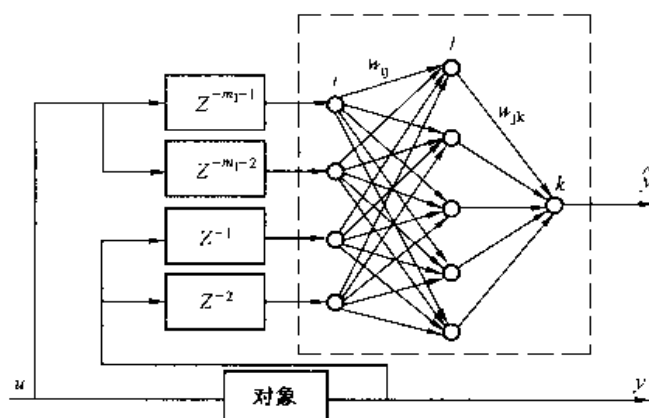


图 3-30 受控对象模型网络结构

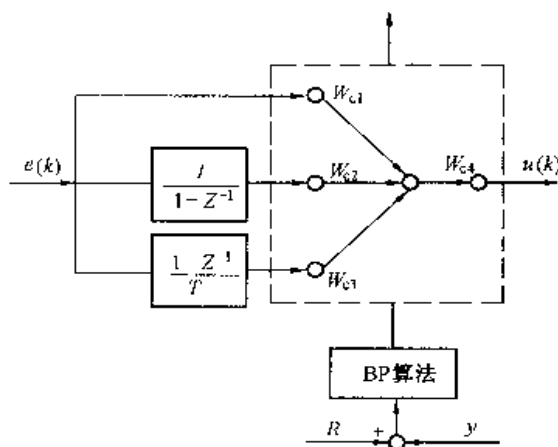


图 3-31 调节器网络结构

3. 受控对象模型纯滞后时间与学习速率的关系

受控对象传递函数如式 (3-63) 所示, 其中模型增益 $k_0 = 1$, 时间常数 $t_1 = 1s$, $t_2 = 10s$, 设定值 $R = 1.0$, 采样周期 $T = 0.5s$, 学习速率 $l = 0.1$, 当纯滞后时间 t_0 从 $0.5s$ 变化至 1.5 、 $2.5s$ 时, 系统输出发散 (图略)。学习速率为

$$l = l_0 / (1 + c_0 t_0) \quad (3-67)$$

式中 l_0 ——初始学习速率;

c_0 ——调整系数。

此时, 当 t_0 按上述方式变化时, 系统趋于稳定, 其结果如图 3-32 所示。书中采用美国 MATLAB 软件进行编程、调试。Learnbp 为反传学习算法函数, 在编程中可直接调用。

4. 受控对象模型时间常数与学习速率的关系

受控对象传递函数如式 (3-63) 所示, 其中模型增益 $k_0 = 1$, 纯滞后时间 $t_0 = 0.5\text{s}$, 时间常数 $t_1 = 1\text{s}$, 设定值 $R = 1.0$, 采样周期 $T = 0.5\text{s}$, 学习速率 $l = 0.1$, 当时间常数 t_2 从 10s 变化至 30s 时, 系统输出等幅振荡 (图略)。学习速率为

$$l = l_0 t_{20} / c_0 t_2 \quad (3-68)$$

式中 l_0 ——初始学习速率;

t_{20} ——受控对象的初始时间常数 (在此, 设 $t_{20} = 10\text{s}$);

c_0 ——调整系数。

当 t_2 按上述方式变化时, 其结果如图 3-33 所示。

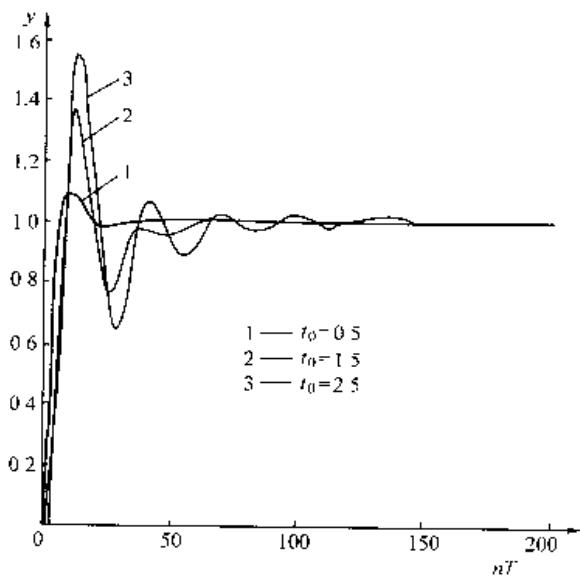


图 3-32 纯滞后时间变化时改变学习速率的调节特性

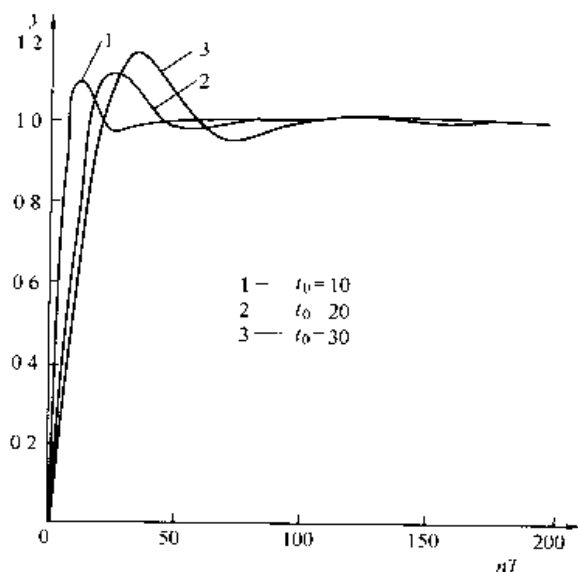


图 3-33 时间常数变化时改变学习速率的调节特性

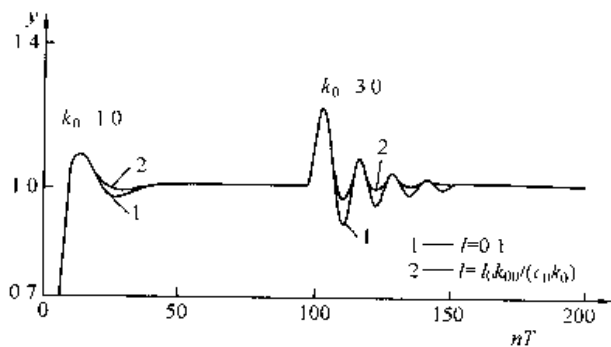


图 3-34 模型增益变化时的调节特性

5. 受控对象模型增益与学习速率的关系

受控对象传递函数见式 (3-63), 其中纯滞后时间 $t_0 = 0.5\text{s}$, 时间常数 $t_1 = 1\text{s}$, $t_2 = 10\text{s}$, 设定值 $R = 1.0$, 采样周期 $T = 0.5\text{s}$, 学习速率 $l = 0.1$, 当采样点数 $n < 100$ 时, 模型增益 $k_0 = 1.0$, $n \geq 100$ 时, $k_0 = 3.0$, 其结果如图 3-34 曲线 1 所示。学习速率为

$$l = l_0 k_{00} / c_0 k_0 \quad (3-69)$$

式中 l_0 ——初始学习速率;

k_{00} ——受控对象初始增益 (在此, 设 $k_{00} = 1.0$);

c_0 ——调整系数。当 k_0 按上述方式变化时, 其结果如图 3-34 曲线 2 所示。

6. 系统扰动幅值与学习速率的关系

受控对象传递函数如式 (3-63) 所示, 其中模型增益 $k_0 = 0.6$, 纯滞后时间 $t_0 = 0.5\text{s}$, 时间常数 $t_1 = 1\text{s}$, $t_2 = 10\text{s}$, 设定值 $R = 1.0$, 采样周期 $T = 0.5\text{s}$, 学习速率 $l = 0.1$, 当采样点数 $n \geq 100$ 时, 扰动幅值 $f = 0.8$, 其结果如图 3-35 曲线 1 所示。学习速率为

$$l = l_0 y_0 / c_0 y_f \quad (3-70)$$

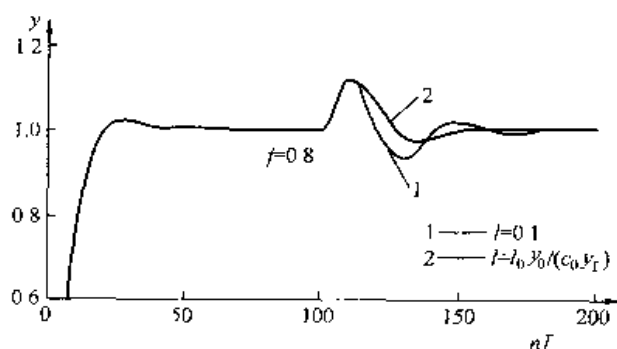


图 3-35 存在扰动时的调节特性

式中 l_0 ——初始学习速率;

y_0 ——系统输出稳态值;

y_f ——干扰出现时的系统输出值;

c_0 ——调整系数。结果如图 3-35 曲线 2 所示。

7. 系统设定值与学习速率的关系

受控对象传递函数如式 (3-63) 所示, 其中模型增益 $k_0 = 1$, 纯滞后时间 $t_0 = 0.5\text{s}$, 时间常数 $t_1 = 1\text{s}$, $t_2 = 10\text{s}$, 采样周期 $T = 0.5\text{s}$, 学习速率 $l = 0.1$, 设定值 R 从 1.0 变化至 1.5, 再变至 1.0 时, 其结果如图 3-36 曲线 1 所示。学习速率为

$$l = l_0 c_0 (1.0 - |R_0 - R|) \quad (3-71)$$

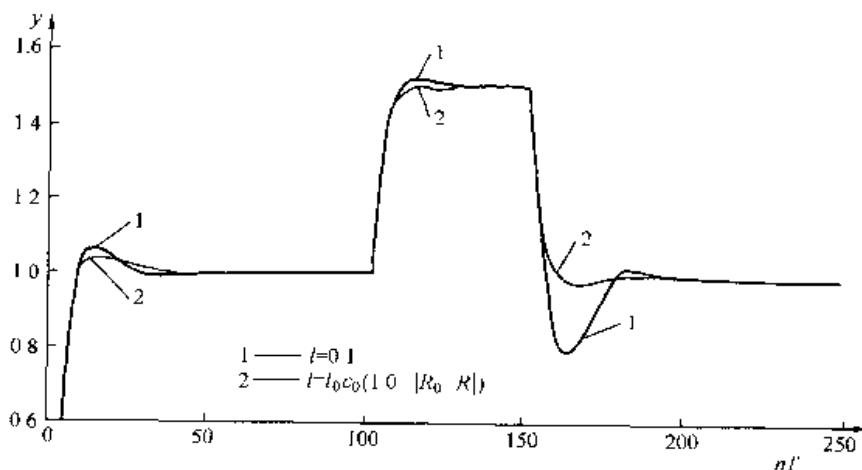


图 3-36 改变设定值时的系统调节特性

式中 l_0 ——初始学习速率;

R_0 ——系统初始设定值;

c_0 ——调整系数。

结果如图 3-36 曲线 2 所示。实验证明本文所选的初始学习速率相对于初始参数是较适合的。

(三) 小结

在过程控制中,对于受控对象模型参数、系统设定值随时间发生大范围变化以及系统出现较大扰动情况,通过建立BP算法的学习速率与它们之间的定量关系,以在线调整学习速率,可提高控制系统的自适应性和鲁棒性,改善系统的调节品质。

【例 3-14】 BP 网络系统辨识模型结构的应用

系统辨识是神经网络在控制系统中应用的一个重要方面,也是实现智能控制的一个基本内容。然而,基于神经网络的系统辨识与建模方法还有一些问题有待进一步研究解决。作者就控制系统受控对象模型阶次与神经网络模型结构的选择问题进行探讨,确立了辨识模型基本结构的研究方案,通过仿真实验,证明该方法是有用的。

(一) 辨识模型的基本结构

1. 问题的提出

神经网络智能控制系统,通常要求有两个神经网络:一个用于系统辨识(NNM网络),另一个用作控制器(NNC网络),其系统结构如图3-37所示。

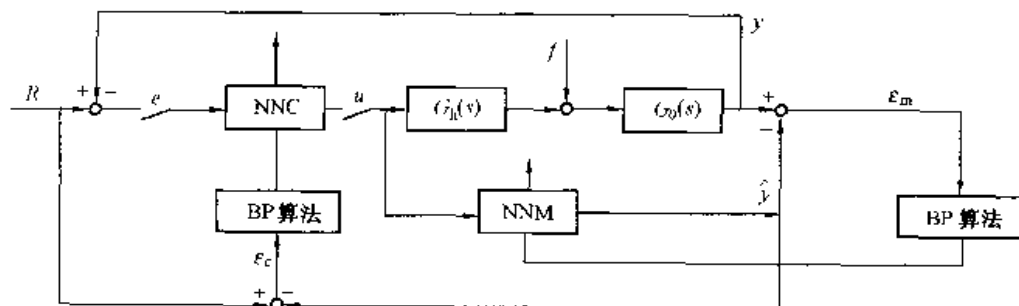


图 3-37 神经网络控制系统结构图

图 3-37 中, R 为系统设定值, u 为控制器网络输出, $G_h(s)$ 为零阶保持器的传递函数, y 为系统输出, \hat{y} 为辨识网络输出, $G_0(s)$ 为受控对象的传递函数。

上述系统首先要求辨识受控对象的模型,由此通过误差 ($R-y$) 反传来调整控制器网络(NNC)的权值,因此受控对象模型精度对控制器的影响极大,然而模型结构的设计与辨识模型的精度密切相关。

2. 辨识网络基本结构的设计思想

与传统的辨识方法相似,采用神经网络进行系统辨识时,也存在一些影响辨识精度的可调参数,如:学习步长、辨识网络的输入层节点数、隐层节点数等。因此,辨识模型结构的设计和辨识参数的调整往往是一个繁琐而费时的过程,工作人员需经过反复实验和探索,才能获得满意的结果。能否设计一个广义的、适用的辨识网络基本结构,尽可能减少辨识参数的反复、交叉调整,以达到预期的辨识精度,这是本文的基本思想。经过仿真实验研究,基于神经网络辨识模型的基本结构依照下列几点设计原则:

(1) 根据系统动态方程的输入、输出关系确定辨识网络的输入层节点数目即 $x-4-1$ 结构, $x=1, 2, \dots$ 。

(2) 对于二阶以上的高阶系统,可按二阶加纯滞后模型确定辨识网络的输入层节点数目,即 $4-4-1$ 结构。

(3) 当要求较高的辨识精度时,可适当增加隐层的节点(神经元)数目,即 $4-x-1$ 结构, $x=1, 2, 3, 4\cdots$ 。

3. 网络训练与在线辨识

图 3-37 所示控制系统 NNM 辨识网络的任务在于采用人工神经网络来逼近系统动态数学模型,辨识网络的输入为

$$p = [y(n-1)y(n-2)\cdots y(n-q_1)u(n-1)u(n-2)\cdots u(n-q_2)] \quad (3-72)$$

p 为 $(q_1+q_2) \times 1$ 维向量,网络的输出为 $y(n)$ 。在对受控对象进行辨识时,首先对 NNM 网络初始化,然后以 $y(n)$ 作为指导信号, $\hat{y}(n)$ 为估计信号,用 $y(n)$ 和 $\hat{y}(n)$ 的误差平方和作为指标,对网络权值进行学习训练,网络的训练与辨识步骤可归纳为以下几点:

(1) 定义一 $M \times N$ 矩阵

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \cdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MN} \end{bmatrix} \quad (3-73)$$

由矩阵 P 可知网络具有 N 个输入节点和 M 组输入向量

$$P_{iN} = [p_{i1} \cdots p_{iN}] \quad i=1, 2, \cdots M$$

$$N = q_1 + q_2$$

(2) 定义一个 $M \times 1$ 维的列向量存放目标样本数据 $T = [T_1 \cdots T_N]$, 应使网络的学习函数通过这些数据点。

(3) 设计一个由 R 个输入的输入层、 S 个双曲正切神经元组成的隐层和一个线性神经元组成的输出层构成的 BP 网络。

(4) 根据输入矩阵 P 、隐层神经元数目 S 及各层的激活函数确定各层的初始权值和阈值,即 $W_1, b_1, W_2, b_2\cdots$ 。

(5) 设定迭代次数 J , 误差平方和 $error = \frac{1}{2} \sum [y(n) - \hat{y}(n)]^2$, 以及学习速率 l 。

(6) 开始训练网络,直至 BP 网络输出 \hat{y} 与目标值 T 的误差平方和满足要求为止。如果到达规定的迭代次数 J 时,误差平方和 $error$ 仍不满足要求,可重复训练网络或适当增加隐层神经元数目 S 。

(7) 网络训练成功后,将训练好的权值和阈值赋予辨识网络作为初值,使辨识网络并入系统,实现自适应控制。

(二) 仿真研究

1. 二阶加纯滞后网络模型的训练与辨识

设受控对象模型为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k_0 e^{-t_0 s}}{(t_1 s + 1)(t_2 s + 1)} \quad (3-74)$$

通过 Z 变换写成差分方程形式

$$y(n) = a_1 y(n-1) - a_2 y(n-2) + k_0 [b_1 u(n-m_1-1) + b_2 u(n-m_1-2)] \quad (3-75)$$

式中 T ——采样周期；

t_1, t_2 ——时间常数；

k_0 ——模型增益；

t_0 ——纯滞后时间；

n ——采样序号, $n = 1, 2, \dots$ 。

$$f_0 = t_2 - t_1$$

$$m_1 = \text{int}(t_0/T)$$

$$q_1 = T/t_1$$

$$q_2 = T/t_2$$

$$c_1 = e^{-q_1}$$

$$c_2 = e^{-q_2}$$

$$a_1 = c_1 + c_2$$

$$a_2 = c_1 c_2$$

$$b_1 = 1 + (t_1 c_1 - t_2 c_2) / f_0$$

$$b_2 = a_2 - (t_1 c_2 - t_2 c_1) / f_0$$

注：在此设纯滞后时间 t_0 是采样周期 T 的整数倍，即 $t_0 = m_1 T$ ，由式 (3-75) 可知网络的输入节点数为 4，即输入向量为 $p = [y(n-1) \ y(n-2) \ u(n-m_1-1) \ u(n-m_1-2)]$ ，设隐层节点数为 4，一个输出节点，此时受控对象模型网络 NNM 为 4-4-1 结构。

当 $k_0 = 1.0$, $t_1 = 1.0$, $t_2 = 5.0$, $R = 1.0$, $T = 0.5$, 学习速率 $l = 0.1$ 时，从系统中采集 40 组样本数，设误差平方和 $error = 0.01$ ，对网络进行训练，结果如图 3-38 所示。训练结束后，将权值和阈值作为辨识网络的初值，图 3-37 所示系统对阶跃输入的响应如图 3-39 所示。当采样点数 n 大于 50 时，在受控对象的输入端加入 $-0.2 \sim +0.2$ 的随机干扰，调节过程如图 3-39 所示。

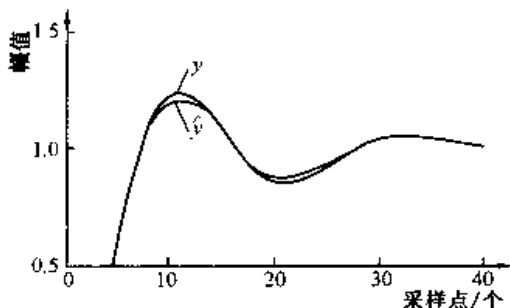


图 3-38 二阶加纯滞后受控对象网络训练

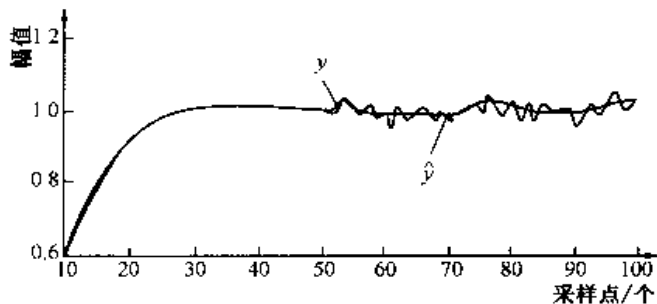


图 3-39 二阶加纯滞后受控对象的实时控制

2. 高阶受控对象网络模型的训练与辨识

下面以 8 阶系统为例, 其传递函数为

$$G(s) = \frac{35s^7 + 1086s^6 + 13258s^5 + 82402s^4 + 278376s^3 + 511812s^2 + 482964s + 194480}{20(s^8 + 33s^7 + 437s^6 + 3017s^5 + 11870s^4 + 27470s^3 + 37492s^2 + 28880s + 9600)} \quad (3-76)$$

对应的脉冲传递函数为

$$G(z) = \frac{0.05(7.0788z^{-1} - 20.9138z^{-2} + 25.0039z^{-3} - 15.486z^{-4} + 5.2957z^{-5} - 0.9853z^{-6} + 0.0907z^{-7} - 0.0032z^{-8})}{1 - 3.6322z^{-1} + 5.5279z^{-2} - 4.5543z^{-3} + 2.194z^{-4} - 0.6242z^{-5} + 0.1008z^{-6} - 0.0083z^{-7} + 0.00031z^{-8}} \quad (3-77)$$

写成相应的差分方程

$$\begin{aligned} y(n) = & 3.6322y(n-1) - 5.5279y(n-2) + 4.5543y(n-3) - 2.1940y(n-4) \\ & + 0.6242y(n-5) - 0.1008y(n-6) + 0.0083y(n-7) - 0.0003y(n-8) \\ & + 0.05[7.0788u(n-1) - 20.9138u(n-2) + 25.0039u(n-3) \\ & - 15.4846u(n-4) + 5.2957u(n-5) - 0.9853u(n-6) + 0.0907u(n-7) \\ & - 0.0032u(n-8)] \end{aligned} \quad (3-78)$$

由式 (3-78) 可知, NNM 网络的输入节点数应为 16, 但是, 根据模型等价最优化原理, 式 (3-78) 所示的高阶模型可用二阶模型来近似, 即设网络的节点数为 4, 输入向量 $p = [y(n-1) y(n-2) u(n-1) u(n-2)]$, 设隐层节点数为 4, 一个输出节点, 此时受控对象模型网络 NNM 为 4-4-1 结构。

设 $R=1.0$, $T=0.25$, 学习速率 $l=0.1$, 误差平方和 $error=0.01$, 从系统中采集 25 组样本数, 对网络进行训练, 结果如图 3-40 所示。训练结束后, 将权值和阈值作为辨识网络的初值, 图 3-37 所示系统对阶跃输入的响应如图 3-41 所示。当采样点数 n 大于 30 时, 在受控对象的输入端加入 0.1 的阶跃干扰, 调节过程如图 3-41 所示。

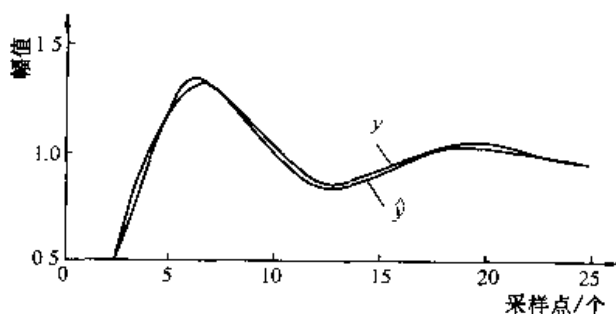


图 3-40 高阶受控对象网络训练

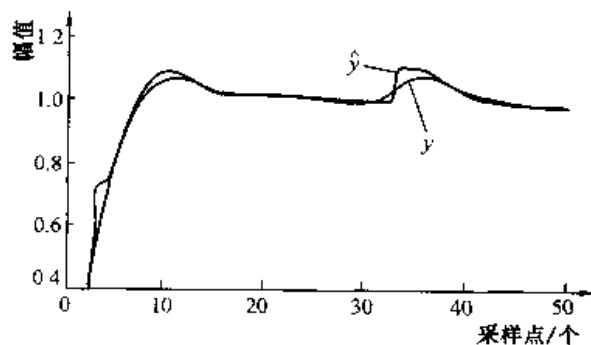


图 3-41 高阶受控对象的实时控制

3. 慢时变、非线性受控对象网络模型的训练与辨识

对象的动态模型为

$$y(n) = 3(1 - 0.8e^{-0.05n}) \frac{y(n-1)}{1 + y^2(n-1)} + u(n-1) \quad (3-79)$$

由式 (3-79) 可知, NNM 网络的输入节点数应为 2, 即输入向量 $p = [y(n-1) \ u(n-1)]$, 设隐层节点数为 4, 一个输出节点, 此时受控对象模型网络 NNM 为 2-4-1 结构。

设 $R=1.0$, $T=0.25$, 学习速率 $l=0.1$, 误差平方和 $error=0.01$, 从系统中采集 40 组样本数, 对网络进行训练, 结果如图 3-42 所示。训练结束后, 将权值和阈值作为辨识网络的初值, 图 3-37 所示系统对阶跃输入的响应如图 3-43 所示。

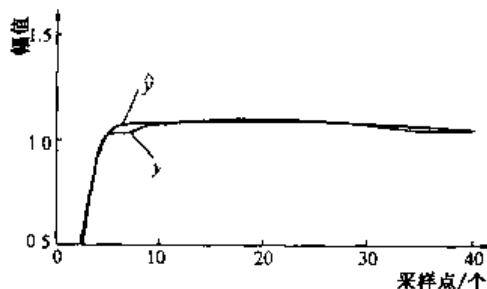


图 3-42 慢时变非线性受控对象网络训练

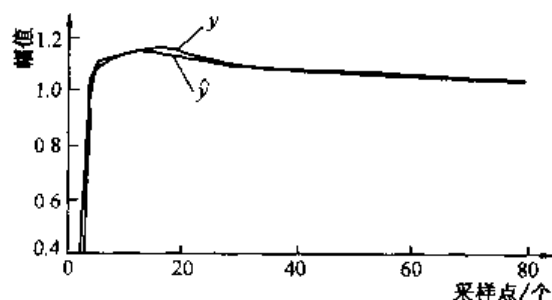


图 3-43 慢时变非线性受控对象的实时控制

(三) 小结

通过仿真实验表明, 本文提出的基于神经网络系统辨识模型基本结构具有其一般性和适用性, 主要特点归纳如下:

- (1) 对于二阶以上的高阶系统, 可按二阶加纯滞后模型设计网络的基本结构。
- (2) 对于一类非线性、时变及不确定系统, 可根据其动态方程的输入、输出关系设计网络的基本结构。
- (3) 适当增加隐层节点 (神经元) 数目, 可提高辨识精度。

第六节 本章小结

- (1) BP 网络与感知器网络的主要区别在于采用不同的激活函数。
- (2) 由于 BP 网络采用连续可微、单调上升的有界函数作为神经元的激活函数, 所以 BP 网络可用于函数逼近、模式分类、预测及控制等工作。
- (3) BP 算法的主要缺点是易于收敛到局部极小点, 而且在误差曲面的平坦区内, 收敛缓慢。
- (4) 附加动量项可使反向传播减少网络权值的调整陷入低谷的可能性, 并有助于减少训练时间。
- (5) 太大的学习速率导致学习的不稳定, 太小值又导致极长的训练时间, 应用自适应学习速率, 可以减少网络的训练时间, 达到较好的学习效果。
- (6) 网络的输入节点数和输出参量的数目根据具体的应用问题来确定。而网络中的隐层数和隐层中的神经元个数, 往往是通过经验或实验来选择。
- (7) BP 网络中单个神经元可采用图 3-44 进行模拟分析。

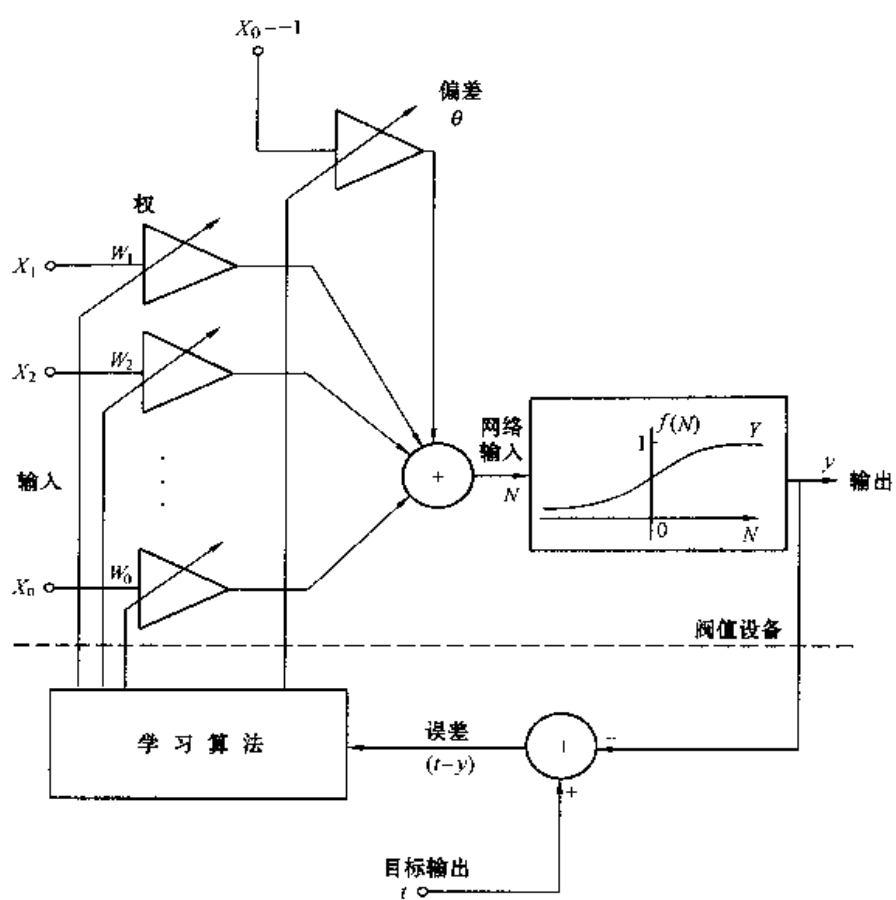


图 3-44 单个神经元学习过程模拟示意图

第四章

RBF 网络

近年来,随着智能技术的发展,神经网络理论已得到了广泛的应用,其中前馈网络和反馈网络是两种典型的网络模型,从学习的观点看,前馈网络(包括BP网络、RBF网络等)是一种较强的学习系统,具有复杂的非线性处理能力。实现映射和函数逼近是前馈(向)网络的共同特点,径向基函数网络(RBF网络)具有较强的输入、输出映射功能,并且理论证明在前向网络中RBF网络是完成映射功能的最优网络。因此,RBF网络以其简单的结构、快速的训练过程和良好的推广能力等诸多优点已在许多应用领域取得了巨大的成功,特别是在模式分类和函数逼近方面。

第一节 RBF 神经元模型

图4-1给出了一个基本的RBF神经元,它有 n_0 个输入,每个输入都通过一个适当的权值 w_i ($i=1, 2, \dots, n_0$) 连接到神经元节点上。与前面提到的神经网络的表达方式有所不同,其网络的输出为权值向量 w 与输入向量 x 之间的向量距离乘以阈值 θ ,即 $y = \text{radbas} [\text{dis}(w, x) \times \theta]$ 。

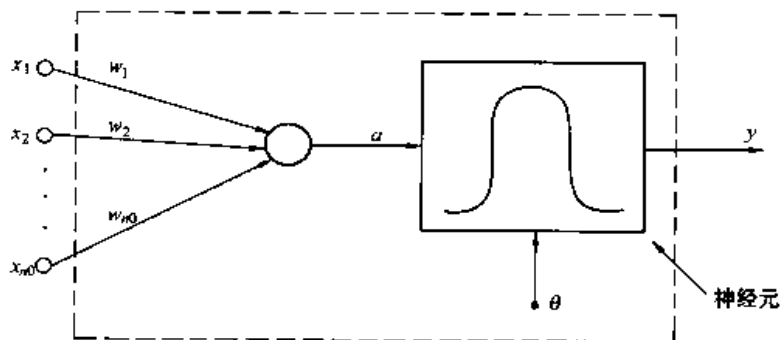


图4-1 具有 n_0 个输入的径向基神经元模型

径向基函数如图4-2所示,从图4-2(a)可以看出,当设 $\theta=1$,径向基函数的输入为0时,输出为极大值1。从图4-2(b)可以看出,径向基函数的输出随 w 和 x 之间的距离减小而增大,因此,当输入 x 同权值 w 完全相同时,径向基函数的输出为1,这时,径向基函数起到了信号检测器的作用。阈值 θ 可调节RBF神经元的敏感程度。例如,如果一个神经元的阈值取为 $\theta=0.1$,那么当向量距离为8.326 ($0.836/\theta$)时,神经元的输出为0.5。

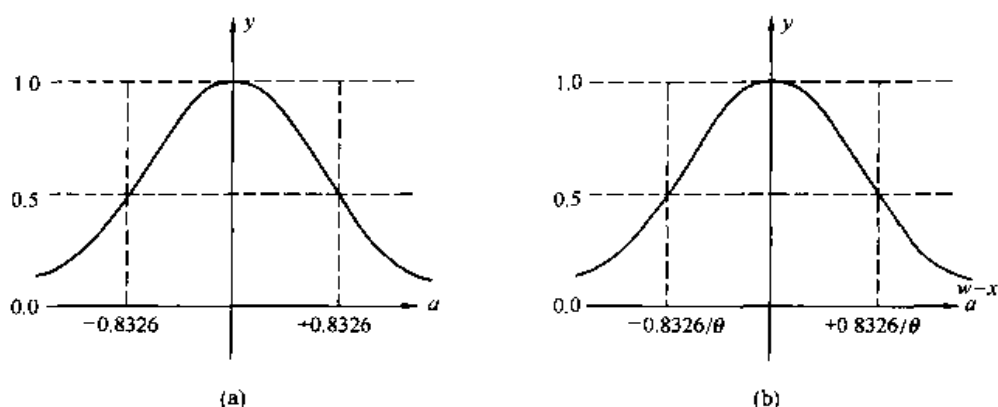


图 4-2 径向激活函数示意图

(a) $Y = \text{radbas}(a)$; (b) $Y = \text{radbas}[\text{dist}(w, x) \times \theta]$

第二节 RBF 网络结构

径向基函数 (RBF) 网络结构由两层组成, 即隐层和输出层, 同层神经元之间没有连接, 相邻两层神经元完全连接, 输入个节点的作用是将 n_0 个输入分布给隐含层的各神经元。输入的数目等效于所研究问题的独立变量数目, 隐层节点选取基函数作为激活函数, 隐层神经元通过径向基函数——高斯函数对输入产生非线性映射。输出层神经元对隐含层的输出进行线性加权组合。即输出层的节点是线性组合器, 设第 j 个输出节点的输出如图 4-3 所示。

一、高斯函数表达式

RBF 网络的基函数可有以下几种选择:

(1) 高斯函数 $\phi(x, \sigma) = \exp[-(x - c_i)^2 / \sigma_i^2]$ 图 4-3 具有一个隐层和一个输出层的 RBF 网络

(2) 细板条样函数 $\phi(x, \sigma) = (x - c_i)^2 \log(x - c_k)$

(3) 平方根函数 $\phi(x, \sigma) = [(x - c_i)^2 + \sigma^2]^{1/2}$

(4) 逆平方根函数 $\phi(x, \sigma) = [(x - c_i)^2 + \rho^2]^{-\alpha} \quad \alpha > 0, i = 1, 2, \dots, n_1$

RBF 网络对激活函数的选择要求并不苛刻, 仅需保证该函数是一个偶多项式即可, 通常采用的是高斯函数。

隐含层中神经元的高斯表达式

$$\beta_j = \exp(-\|x - c_j\|^2 / \sigma_j^2) \quad j = 1, 2, \dots, n_1 \quad (4-1)$$

式中 $\|\cdot\|$ ——欧式范数;

β_j ——隐层第 j 个单元的输出;

x ——输入量;

c_j ——隐层第 j 个单元基函数的中心，也可以看成该单元的权向量；

σ_j^2 ——第 j 个隐节点的归一化参数，其决定该中心点对应的基函数的作用范围。

隐层节点输出在 0 到 1 之间，输入与中心的距离越近，隐节点的响应就越大。RBF 网络的输出为其隐含层节点的线性组合，即

$$y_k = \sum_{j=1}^n w_{jk} \beta_j \quad j = 1, 2, \dots, n, \quad k = 1, 2, \dots, m \quad (4-2)$$

二、归一化参数 σ^2 的确定

归一化参数 σ^2 是一个系数参数，影响每个高斯函数响应面的“坡度”。图 4-4 描绘了在 σ 取值不同的情况下，中心为原点 $(0, 0)$ 的高斯函数对 $x \in [-2, 2]$, $y \in [-2, 2]$ 正方形区域内的输入产生不同的响应 β ，如图 4-4 所示，随着 σ 值由小变大，高斯函数响应面从陡峭向平坦变化，响应面越陡峭，网络对样本与高斯函数中心的区分越灵敏。但是， σ 过小，网络对噪声太敏感，易失真；而 σ 过大，会使网络丧失区分和拟合的能力，因此 RBF 网络需要选择合适的 σ 值。

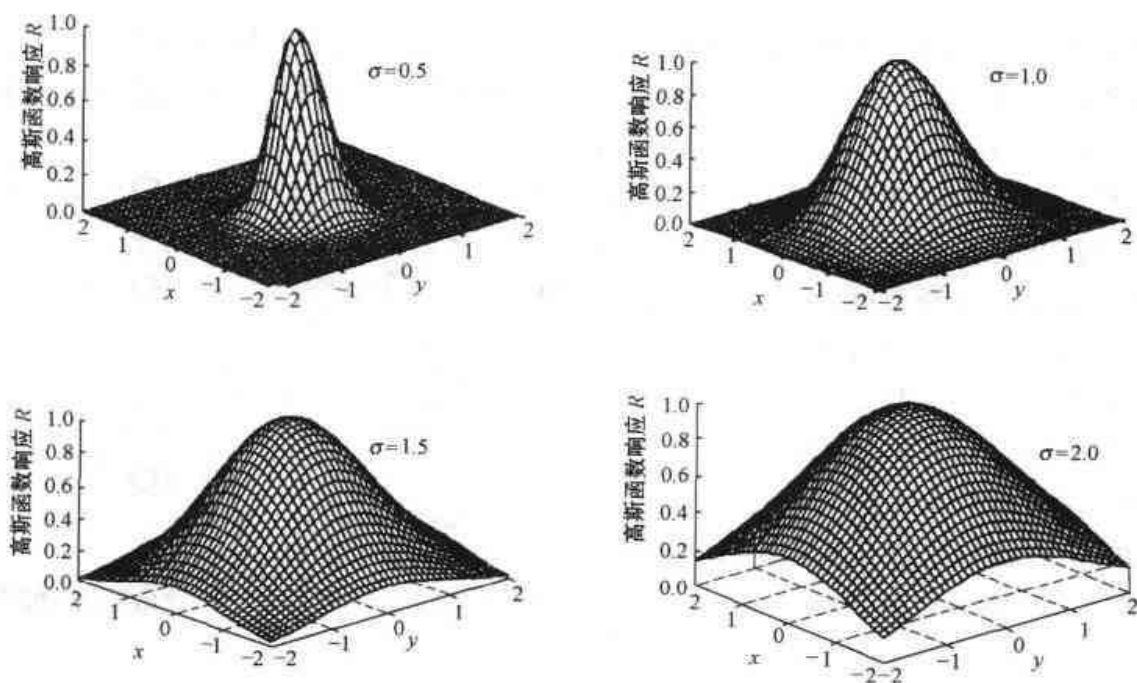


图 4-4 高斯函数响应面随 σ 变化的三维图

第三节 RBF 学习算法

由于可以把 RBF 网络看成对未知函数 $F(x)$ 的逼近器，一般任何函数都可表示成一组基函数的加权和，这相当于选择各隐单元的作用函数构成一组基函数用于近似 $F(x)$ ，即

$$F(x) \approx a_1 \phi_1(x) + a_2 \phi_2(x) + \dots + a_n \phi_n(x) \quad (4-3)$$

$$x = [x_1, x_2, \dots, x_n]^T$$

式中, $|\phi_n(x)|$ 是正交的。

由模式识别理论可知, 在低维空间非线性可分的问题总可映射到一个高维空间, 使其在高维空间中为线性可分。在 RBF 网络中, 输入到隐层的映射为非线性 (因为隐单元的作用函数是非线性函数), 而隐层到输出则是线性的。可把输出单元部分看作一个单层感知器, 这样, 只要合理选择隐单元数 (高维空间的维数) 及其作用函数, 就可以把原来问题映射为一个线性可分问题, 最后可用一个线性单元来解决。

RBF 网络有两个可调参数, 即中心位置 c_j 和方差 σ_j^2 (或称函数的宽度参数 σ), 此时, 整个网络的可调参数有 3 组, 即各基函数的中心位置、方差和输出单元的权值。一种较好地确定 c_j 和 w 的方法是用聚类方法实时调整中心, 并同时调整中心和权值。聚类方法就是把样本聚成几类, 以类中心为各 RBF 函数的中心, 常用的方法有 k 均值法和自组织法, 下面介绍 k 均值聚类法。

1. 中心调整算法

中心调整算法以聚类最小距离为指标, 将输入数据集分解为 k 类, 给出 k 个中心, 其计算步骤如下:

(1) 随机选择初始中心 $c_j(0)$, $1 \leq j \leq n_1$, 设置初始学习速率 $\alpha(0)$ 。

(2) 计算第 k 步的最小距离

$$l_j(k) = \|x(k) - c_j(k-1)\| \quad 1 \leq j \leq n_1 \quad (4-4)$$

(3) 求最小距离的节点 q

$$q = \arg [\min l_j(k)] \quad 1 \leq j \leq n_1 \quad (4-5)$$

(4) 更新中心

$$\begin{cases} c_j(k) = c_j(k-1) & 1 \leq j \leq n_1 \quad j \neq q \\ c_q(k) = c_q(k-1) + \alpha(k) [x(k) - c_q(k-1)] \end{cases} \quad (4-6)$$

(5) 重新计算第 q 节点的距离

$$l_q(k) = \|x(k) - c_q(k-1)\| \quad (4-7)$$

(6) 修正学习率

$$\alpha(k+1) = \frac{\alpha(k)}{1 + \text{int}[k/n_1]^{1/2}} \quad (4-8)$$

式中, $\text{int}[\]$ 表示取整。

(7) 令 $k = k+1$, 返回步骤 (2)。

2. 权值更新算法

输出层的每一个节点或线性组合器是一个权值估计, 可用最小二乘法或其他方法求取。将权值看作状态向量, 即

$$w_j(k) = [w_{1j}(k), w_{2j}(k), \dots, w_{n_1j}(k)]^T \quad 1 \leq j \leq n_1 \quad (4-9)$$

设在第 k 步时, 中间层的输出向量为

$$\phi(k) = [\phi_1(k), \dots, \phi_{n_1}(k)]^T = [\phi_1(l_1(k)), \sigma, \dots, \phi_{n_1}(l_{n_1}(k)), \sigma]^T \quad (4-10)$$

第 k 步第 j 个估计输出为

$$\hat{y}_j(k) = \sum w_{ij} \phi_i[l_i(k), \sigma] \quad (4-11)$$

设实际输出为 $y_j(k)$, 则误差为

$$\epsilon_j(k) = y_j(k) - \hat{y}_j(k) \quad (4-12)$$

根据递推最小二乘法, 权值(状态参数)的更新如下

$$w_j(k+1) = w_j(k) + p(k) \phi(k) \epsilon_j(k) \quad (4-13)$$

式(4-13)中有

$$p(k) = \frac{1}{\lambda(k)} \left[p(k) - \frac{p(k-1) \phi(k) \phi^T(k) p(k-1)}{\lambda(k) + \phi^T(k) p(k-1) \phi(k)} \right] \quad (4-14)$$

式中 p ——误差方差阵;

λ ——遗忘因子。

以上的计算过程实际是有指导的学习, y 是指导数据, 通过对权值的训练后估计输出逼近实际输出。

k 均值聚类算法和权值更新算法是两个独立的学习算法, 与 BP 网络算法相比, 由于参数调整是线性的, 可获得较快的收敛速度, 这一特点非常适合用于系统的实时辨识与控制。

第四节 RBF 网络的应用

径向基函数(RBF)网络已被广泛应用于模式识别、函数逼近、信号处理、系统建模和控制等领域, 下面介绍几个典型的应用实例。

【例 4-1】 用 RBF 网络解 XOR 问题。

选择隐层激活函数为高斯函数, 网络如图 4-5 所示, 即

$$G(\|x - c_i\|) = \exp(-\|x - c_i\|^2 / \sigma^2) \quad i = 1, 2 \quad (4-15)$$

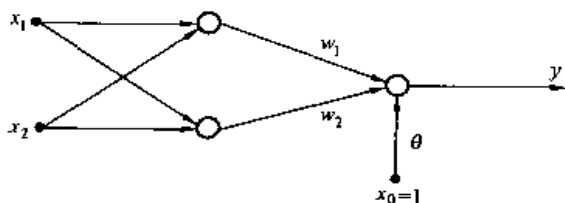


图 4-5 RBF 网络用于解 XOR 问题

式中 x ——输入向量;

c_i ——中心, $i = 1, 2$;

σ ——宽度, 为一正数。

由于问题的对称性, 隐层点到输出节点的权值选为相同(即 $w_1 = w_2$), 输入到输出层的阈值为 1, 其输出可表示为

$$\hat{y}(x) = \sum_{i=1}^2 w G(\|x - c_i\|) + \theta \quad (4-16)$$

c_i 为中心, $c_1 = [1, 1]^T$, $c_2 = [0, 0]^T$, σ 取为 $1/\sqrt{2}$ 。

用 g_{ji} 表示各相应输入下隐节点的输出。

$$g_{ji} = G(\|x_j - c_j\|) \quad j = 1, 2, 3, 4 \quad i = 1, 2 \quad (4-17)$$

希望的输出值为已知, 即

$$y = [1, 0, 1, 0]^T$$

将 W, θ 作为待求参数并表示为

$$W = [w_1, w_2, \theta]^T$$

则有以下关系

$$GW = y \quad (4-18)$$

其中

$$G = \begin{bmatrix} g_{11} & g_{12} & 1 \\ g_{21} & g_{22} & 1 \\ g_{31} & g_{32} & 1 \\ g_{41} & g_{42} & 1 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.1353 & 1 \\ 0.3678 & 0.3678 & 1 \\ 0.1353 & 1.0000 & 1 \\ 0.3678 & 0.3678 & 1 \end{bmatrix}$$

给定 4 对输入数据, 求 3 个待求参数, 可用最小二乘求伪逆

$$W = G^+ y = (G^T G)^{-1} G^T y \quad (4-19)$$

把 G 值及 y 值代入, 得

$$G^+ = \begin{bmatrix} 1.656 & -1.158 & 0.628 & -1.158 \\ 0.628 & -1.158 & 1.656 & -1.158 \\ -0.846 & 1.301 & -0.846 & 1.301 \end{bmatrix}$$

$$W = \begin{bmatrix} 2.284 \\ 2.284 \\ -1.692 \end{bmatrix}$$

检查结果见表 4-1。

表 4-1

检 查 结 果

数 据 序 号	输入 x_i	希望输出 y_i	实际输出 \hat{y}_i
1	(1, 1)	1	+ 0.901
2	(0, 1)	0	- 0.01
3	(0, 0)	1	+ 0.901
4	(1, 0)	0	- 0.01

【例 4-2】 RBF 网络用于油田累计产量系统的预测。

1. RBF 网络模型

RBF 网络结构如图 4-6 所示, 网络输入节点为 N , 隐层节点为 M , 输出节点为 1。网络输入输出之间可以认为是一种映射关系 $y(x): R^N \rightarrow R^1$ (N 为输入节点数), 即

$$y(x) = \sum_{i=1}^M w_i \phi(x, c_i) = \|\phi(x - c_i)\|$$

$$i = 1, 2, \dots, M \quad (4-20)$$

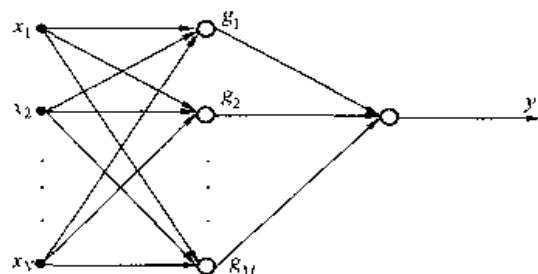


图 4-6 RBF 网络结构

式中, $\phi(x, c_i)$ 为 RBF, 是一个非线性函数, 在此取常用的高斯函数; M 为隐含层节点数, $c_i \in R^M$ 为径向基函数的中心, ϕ 对于 c_i 径向对称。

2. RBF 网络学习算法

在 RBF 神经网络中, 输出层和隐含层所完成的任务是不同的, 因而它们学习的策略也不相同。输出层是对线性权值进行调整, 采用的是线性优化策略, 学习速度较快; 而隐含层是对激活函数 (高斯函数) 的参数 (中心和均方差) 进行调整, 采用非线性优化策略, 因而学习速度较慢。下面概述 RBF 网络的学习算法。

RBF 中心的选择:

(1) 应用 K-mains 聚类算法选择 RBF 中心。

1) 初始化聚类中心 c_i ($i = 1, 2, \dots, M$), 一般是从输入样本 x_i ($i = 1, 2, \dots, N$) 中随机选取 M 个样本作为初始中心, M 一般由经验来取。

2) 把输入样本按最临近 (即与 c_i 的欧几里德范数最小) 原则归类分组, 即把 c_i ($i = 1, 2, \dots, N$) 分配给与它距离最近的中心 c_i 所在聚类集合 ϕ_i ($i = 1, 2, \dots, M$)。

3) 计算聚类集合 ϕ_i ($i = 1, 2, \dots, M$) 中的样本均值 \bar{c}_i , 得出新的中心 $c_i = \bar{c}_i$ ($i = 1, 2, \dots, M$), 按上述步骤计算, 直到中心分布不再变化为止, 否则, 转到 2), 继续计算。

4) 计算均方差 σ , $\sigma = d_m / \sqrt{2M}$, d_m 为所选中心之间的最大距离, M 为隐层单元个数。

(2) 计算 $\phi(x, c_i) = \|\phi(x - c_i)\|$ 。

权值的调整:

1) 以小的随机数初始化权值矩阵。

2) 由公式 $\hat{y} = F(x) = \sum_{i=1}^M w_i \phi(x, c_i)$ 计算网络的输出值。

3) 计算绝对误差 $\epsilon = y - \hat{y}$, y 为期望输出, \hat{y} 为实际输出。

4) 修正权值 $\Delta w_i(k+1) = \eta \epsilon \phi(x, c_i)$ 。

3. 油田累计产量系统的建模和预测

设 Q_i 是某油田区块第 i 年的累积产油量, RBF 网络的输入、输出关系如下 ($i = 1, 2, \dots, N; j = 1, 2, \dots, n$)

$$\begin{aligned} Q_i &= \sum_{j=1}^n w_j \phi_{ij} \\ d^T &= (d_1, d_2, \dots, d_n) \\ \phi_{ij} &= \|\phi(x_i - c_j)\| \\ d &= \phi_y w \\ \phi^T d &= \phi^T \phi w \\ w &= (\phi^T \phi)^{-1} \phi^T d \end{aligned}$$

式中 w_i ——第 i 个隐层单元输出的连接权值;

ϕ_{ij} ——第 i 个隐层节点变换函数。

在已有现场的数据, 通过 RBF 网络模型, 对油田累积产量进行了预测, 取得较满意的结果。

【例 4-3】 RBF 用于微合金钢流变应力的预测。

1. RBF 网络结构

设 RBF 网络为两层网络结构, 即一层隐层和一层输出层。其结构如图 4-7 所示。

输入层不处理信息仅传播输入信号到隐层, 隐层节点选用高斯函数, 输出节点是简单的线性函数。高斯函数表达式如下:

$$\beta_j = \exp(-\|x - c_j\|^2 / \sigma_j^2) \quad j = 1, 2, \dots, M \quad (4-21)$$

$$x = [x_1, x_2, \dots, x_N]^T$$

式中 β_j ——隐层第 j 个单元的输出;

x ——输入向量;

c_j ——隐层第 j 个单元基函数的中心, 也可看成该单元的权向量;

σ_j^2 ——第 j 隐节点的归一化参数, 其决定该中心点对应的基函数的作用范围。

网络的输出为

$$y = \sum_{j=1}^M w_{ij} \beta_j(x) \quad i = 1, 2, \dots, N \quad (4-22)$$

2. RBF 网络模型的建立与预测

由于输入参量之一变形速度输入数据点相对较少, 空间分布稀疏, 相邻两变形速度间的间隔较大, 不利于训练时隐层中心的选择, 故考虑将其先进行预处理, 取其对数函数, 通过相应的 R 度变换缩小其相对间隔。变形速度 ϵ 的归一化方法如下式所示

$$\epsilon' = \frac{(25 + \log \epsilon) - 0.95(25 + \log \epsilon_{\max})}{1.05(25 + \log \epsilon_{\max}) - 0.95(25 + \log \epsilon_{\min})} \quad (4-23)$$

式中 ϵ ——原始数据;

ϵ' ——是 ϵ 归一化的值。

神经网络要求所有的输入在 $(0, 1)$ 之间, 因此对输入进行归一化处理, 采用的归一化方法如下式所示

$$z' = \frac{z - 0.95z_{\min}}{1.05z_{\max} - 0.95z_{\min}} \quad (4-24)$$

本例中采用两层 RBF 网络, 隐层节点数与输入节点相同, 训练样本数为 700, 测试样本数为 140, 测试样本的应变速率 ϵ 为 $60s^{-1}$, 通过调节 σ (宽度) 可以看出, 当 σ 太大时, 预报曲线在实验曲线的下面, 如果 σ 过小则预报曲线在实验曲线的上面, 在 σ 的某一点上两条曲线几乎重合, 则可以确定网络的 σ 参数的范围。由此可提高预测精度, 获得满意的预报结果。

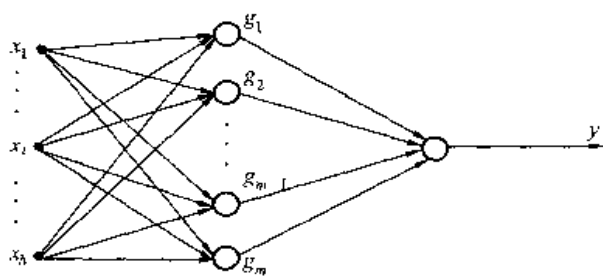


图 4-7 RBF 网络结构

【例 4-4】 基于 RBF 网络实现软测量模型。

在工业过程控制中, 常常存在一些重要的变量难以测量的问题, 为了解决这个问题,

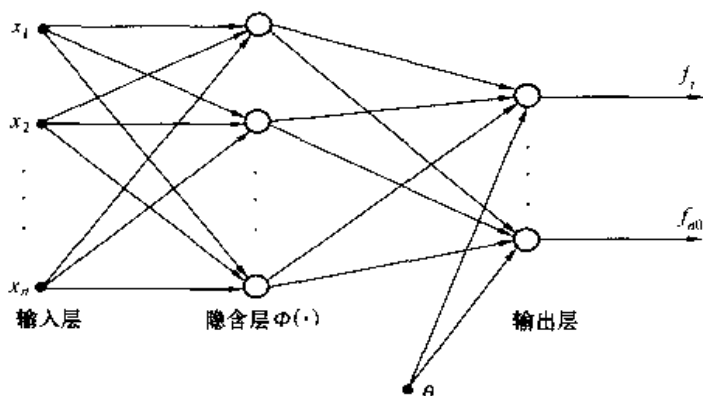


图 4-8 RBF 网络结构

考虑到投资少, 测量速度快, 采用间接测量的方法, 即通过采集工业过程中容易检测到的二次测量变量 (辅助变量), 如温度、压力流量等, 建立过程的数学模型, 由此来运算 (推断) 得到不易测量的过程变量。这种推断估计器的设计就是软测量技术的应用, 基于神经网络的软测量模型在化工过程已得到应用, 下面介绍这种软测量技术。

1. RBF 网络结构

RBF 网络结构如图 4-8 所示。如果前向网络学习集合为 $s = \{ (x_i, c_i) \in R^n \times R \mid i = 1, 2, \dots, N \}$, $\phi(0, \phi)$ 是未知函数, ϕ 是待学习的权值, 那么神经网络学习过程就是寻找 $\phi(0, \phi)$ 及 ϕ 使下述能量函数取最小值

$$H(\phi) = \sum [c_i - \phi(x_i, \phi)]^2 + \lambda \|p\phi\|^2 \quad (4-25)$$

式中, $\|\cdot\|$ 是函数空间上的 L^2 范数, 约束算子 $p(\cdot)$, λ 表示约束的严格程度。 $\phi(\cdot)$ 取径向基函数。

2. 用 RBF 网络模型实现软测量

如化工过程某一精馏塔的例子。精馏塔输入输出可以用如下方程表示

$$\begin{cases} y_B = f_1(p, \Delta T_{935}, \Delta T_{936}) \\ y_D = f_2(p, \Delta T_{935}, \Delta T_{936}) \end{cases} \quad (4-26)$$

式中 y_B, y_D ——分别是精馏塔精丙烯产品中 c_3^+ 含量, 丙烷中 c_3^0 含量;

P ——压力;

$\Delta T_{935}, \Delta T_{936}$ ——塔的温差。

根据现场采集的数据, 利用 RBF 网络对精馏塔的输入输出情况进行学习, 由 RBF 网络模型输出与精馏塔输出的比较曲线可以看出, RBF 网络模型在一定程度上学习到了精馏塔的输入输出状况, 因此该 RBF 网络可以作为精馏塔系统的软测量工具, 同时也可作为精馏塔系统控制器的组成部分。

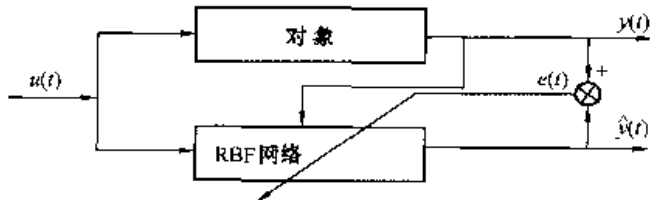


图 4-9 RBF 网络辨识的系统结构

【例 4-5】 RBF 网络用于系统辨识。

RBF 网络用于辨识对象的结构如图 4-9 所示。

设未知的待辨识的对象的动态模型结构为

$$y(t) = 0.7 \sin [0.5y(t-1) + 0.3y(t-2) + 0.6u(t-1) + 0.2u(t-2)] + 0.7 \sin [0.4u(t-1)] \quad (4-27)$$

其中输入 u 取 $[-1, 1]$ 之间的随机数, 初始值 $y(-1) = 0.01$, $y(0) = 0.1$, 采用 RBF 网络算法来辨识这一系统, 网络的输入

$$x(k) = [x_1(k), \dots, x_4(k)]^T$$

$$\text{即为 } x(k) = [y(t-1), y(t-2), u(t-1), u(t-2)]^T$$

中间层 (即隐层) 节点取为 20, 激活函数选细板条样函数, 初始中心为 $[-1, +1]$ 的随机数。输出节点只有一个, 中心的调整采用递推 k 均值聚类算法, 权值估计采用卡尔曼滤波算法, 通过仿真计算结果表明, RBF 算法速度比 BP 算法提高几十倍, 对系统模型的逼近误差在 ± 0.002 范围以内。

【例 4-6】 RBF 网络用于气相色谱保留值的预测。

人工神经网络 (Artificial Neural Networks) 是一种模仿人脑的组织结构、工作方式、思维能力并具有大量连接的并行分布式处理器, 它具有通过学习获取知识并能解决问题能力, 并将获取的知识分布储存在连接权中。不同的神经网络模型是从不同角度对生物神经网络进行不同层次的描述和模拟。目前具有代表性的网络模型有感知器、多层映射 BP 网络、RBF 网络、双向联想记忆 (BAM)、Hopfield 模型等。本例采用 BP 网络、改进 BP 算法网络及 RBF 网络分别对气相色谱各溶质保留值进行预测研究, 并对预测结果进行分析比较, 指出 RBF 网络预测不理想的根本原因。

M. H. Abraham 提出的溶剂化模型, 根据溶质进入溶剂的过程, 将溶质与溶剂间的作用分为具有线性加合性的五种作用, 并用其总和来决定溶质的保留, 其一般的线性自由能关系为

$$\log SP = c + rR_2 + s\pi_2^H + a\Sigma\alpha_2^H + b\Sigma\beta_2^H + l\log L^{16} \quad (4-28)$$

其中 $\log SP$ 为因变量, 表示气液色谱中在给定温度下溶质的保留数据, 可以是保留体积、气液分配系数及相对保留时间。如果因变量是 Kovats 指数 I , 那么溶剂化参数模型的自由能关系则为

$$I = c' + r'R_2 + s'\pi_2^H + a'\Sigma\alpha_2^H + b'\Sigma\beta_2^H + l'\log L^{16} \quad (4-29)$$

上述方程的自变量为 R_2 、 π_2^H 、 α_2^H 、 β_2^H 、 $\log L^{16}$ 。这 5 个参数是描述溶质溶剂化作用大小的描述性参数, 对不同的溶质, 其值也不同。其中 R_2 为极化参数, 反映了溶质通过 π 或 n 电子对与溶剂作用的能力, π_2^H 为溶质的有效偶极-极化参数, α_2^H 为溶质的有效氢键给体参数, β_2^H 为有效的氢键受体参数, $\log L^{16}$ 为 25℃ 下溶质在正十六烷上的气液分配系数。而 c' 、 r' 、 s' 、 a' 、 b' 和 l' 为相参数, 衡量了溶剂对溶质保留贡献的大小, 不同的固定相相参数不同。 r' 常数定量衡量了溶剂与溶质 π -电子及 n -电子对作用的能力, s' 常数是溶剂参与偶极-偶极作用及偶极-诱导偶极作用的能力, a' 常数为溶剂作为氢键碱的能力, b' 为溶剂作为氢键酸的能力, l' 常数代表了空穴效应和色散作用的组合, 它提供了衡量固定相 (溶剂) 分离同系物系列邻近组分的能力。 c' 常数最不易确定, 它体现了测定溶剂化参数时与色谱实验时, 温度及实验条件一致性的变化且与空穴形成有关。

本例采用 20 种固定相（其中 17 种是环糊精）的相参数，19 种溶质的溶质参数，以及相应的各溶质的 Kovats 保留指数等数据，以此数据为网络的训练样本和检验样本，建立了 BP 网络和 RBF 网络模型，并用于 Kovats 指数的预测，对预测效果进行分析比较。

使用同一种溶质 2-indanol（茚满醇）在 20 种不同相参数的固定液上的 Kovats 保留指数，建立了六种相参数与 2-indanol（茚满醇）的 Kovats 保留指数的关系模型，即六个输入与一个输出的关系。原始数据见表 4-2，建模时，将原始数据归一化处理，使其落在 -1 到 +1 之间。

表 4-2 原始数据

序号	固定相	c'	r'	s'	a'	b'	l'	I
1	SQC	79.2	37.5	0	0	0	196.2	1149
2	PHA	59.1	85.9	234.8	491.8	101.7	199.8	1408
3	PH-B	261.0	141.8	211.5	472.6	0	151.4	1285
4	PH-C	94.0	99.6	262.7	513.9	188.4	192.4	1451
5	DA-A	28.7	69.0	147.0	403.1	0	208.0	1276
6	DA-B	1.2	109.7	184.3	452.6	0	212.8	1320
7	DA-G	20.7	86.4	193.9	451.3	0	209.1	1331
8	TA-A	107.4	-75.7	321.7	159.0	0	189.3	1270
9	TA-B	91.0	145.3	168.7	408.2	0	193.0	1310
10	TA-G	110.4	170.6	130.2	504.4	0	189.8	1334
11	Hydrodex	27.4	29.8	250.2	388.0	0	210.0	1347
12	Cydodex	24.9	10.0	260.3	372.3	0	210.8	1448
13	CTC-SV	61.4	43.4	220.9	440.9	0	199.5	1312
14	PBC-SV	52.9	43.7	214.6	401.5	0	202.5	1318
15	Lip-A	-17.7	52.7	233.0	384.7	0	222.0	1383
16	Lip-B	80.7	40.6	236.6	418.8	0	196.4	1347
17	Lip-C	112.1	26.7	153.3	260.5	0	188.3	1220
18	Lip-D	37.0	-248.5	623.0	290.9	0	205.6	1422
19	Lip-E	38.6	-190.3	491.4	256.7	0	204.9	1364
20	B-1:4	-19.7	114.7	184.7	441.6	0	218.4	1302

（一）网络模型结构

在此设 BP 网络结构为 6-6-1（即 6 个输入节点，6 个隐层节点，1 个输出节点），隐层中的神经元均采用 tan-sigmoid 型传递函数，输出层的神经元采用线性传递函数，如图 4-10 所示。

RBF 网络结构为 6- x -1（即 6 个输入节点， x 个隐层节点，1 个输出节点），隐层中的神经元采用 RBF 型传递函数，隐层节点数 x 是在网络训练过程中，依据误差指标由软件自动确定的。输出层的神经元采用线性传递函数，如图 4-11 所示。

基于以上的网络结构，使用表 4-2 的数据，每次采用 19 组数据训练网络，留下 1 组数

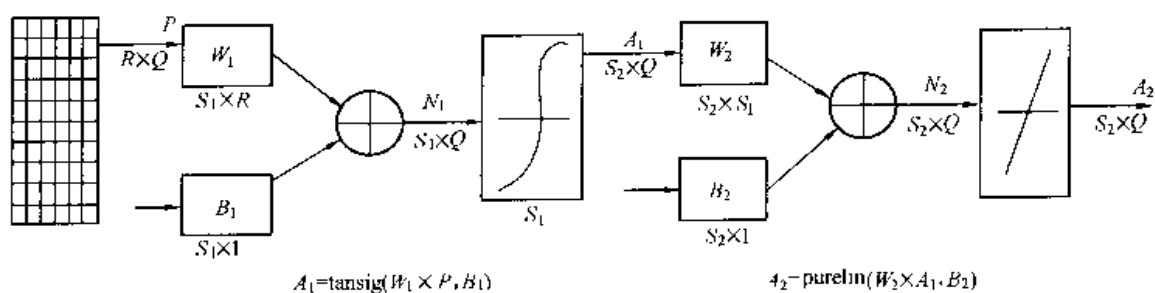


图 4-10 BP 网络结构图

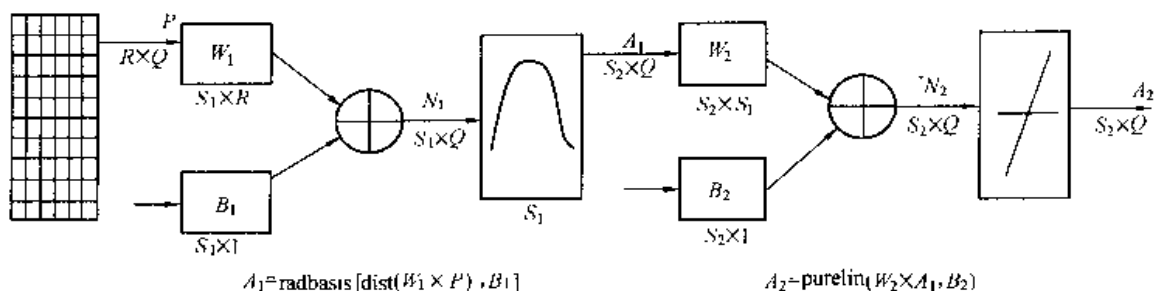


图 4-11 RBF 网络结构图

据作为预测，分别对网络进行 20 次训练和预测，即采用逐点验证的方法，研究两种网络结构预测 2-indanol（茛满醇）在不同固定液上的 Kovats 保留指数的能力，其预测结果如表 4-3 所示。

表 4-3 (逐点预测值)

序 号	固 定 相	原始值 I	BP 网络预测 \hat{I}_{BP}	改进 BP 网络预测 \hat{I}_{BPk}	RBF 网络预测 \hat{I}_{RBF}
1	SQU	1149	1246.818	1536.164	- 27299.445
2	PH-A	1408	1389.467	1400.702	1012.194
3	PH-B	1285	1219.079	1618.217	- 29648.899
4	PH-C	1451	1479.494	1452.435	- 860.916
5	DA-A	1276	1285.293	1284.311	1353.484
6	DA-B	1320	1301.465	1304.409	1387.852
7	DA-G	1331	1299.381	1237.607	1201.759
8	TA-A	1270	1233.530	1429.090	881.383
9	TA-B	1310	1398.588	1340.574	250933.2
10	TA-G	1334	1343.564	1362.816	1393.348
11	Hydrodex	1347	1424.031	1434.707	1422.219
12	Cydodex	1448	1361.215	1357.093	1323.118
13	CTC-SV	1312	1336.225	1346.312	1331.710
14	PBC-SV	1318	1330.351	1311.506	1308.199
15	Lip-A	1383	1424.817	1318.739	1255.817
16	Lip-B	1347	1321.216	1308.260	1342.522
17	Lip-C	1220	1189.574	1134.474	1502.869
18	Lip-D	1422	1368.282	1226.523	151.373
19	Lip-E	1364	1479.041	1548.697	2086.665
20	B-1:4	1302	1311.370	1309.785	1100.197

(二) 预测结果分析

下面以第 8 组数据为例, 分别讨论 3 种方法所建立网络及预测 2-indanol (茚满醇) 在不同固定液上的 Kovats 保留指数的结果。

1. 基本 BP 算法的训练结果

将样本集中 (如表 4-2 所示) 的 19 组数据 (除第 8 组数据外) 作为图 4-10 网络的输入, 网络结构为 6-6-1, 设学习速率为 0.01, 经过 8000 次训练 (或误差平方和小于 0.01) 后, 得到的训练网络输出值与实测值如图 4-12 所示, 图中的 \bigcirc ——表示实测点, 由图可见网络输出只穿过部分实测点, 有些网络输出未穿过实测点, 即采用 BP 算法的网络训练结果不理想, 网络输出与实测值的偏差较大。

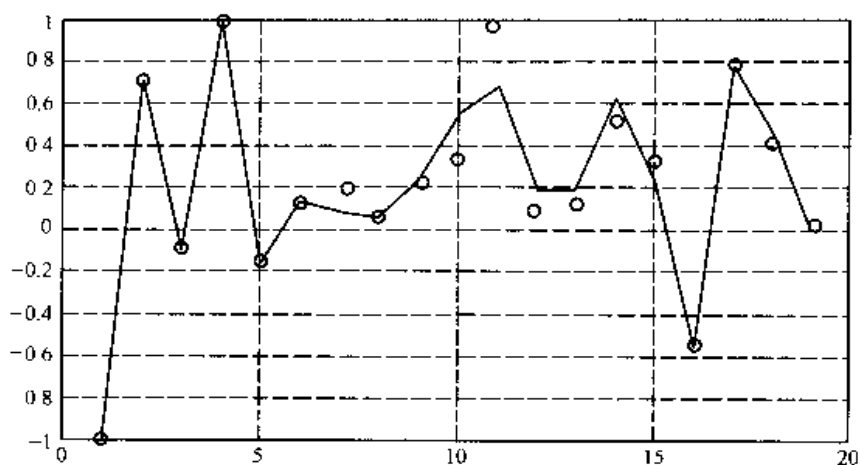


图 4-12 BP 网络对样本集 (除第 8 组外) 的训练过程

2. 改进 BP 算法的训练结果

本文采用的改进 BP 算法是动量-学习速率自调整算法, 将样本集中 (见表 4-2) 的 19 组数据 (除第 8 组数据外) 作为图 4-10 网络的输入, 网络结构为 6-6-1, 设学习速率为 0.01, 经过 8000 次训练 (或误差平方和小于 0.01) 后, 得到的训练网络的输出值与实测值如图 4-13 所示, 图中的 \bigcirc ——表示实测点, 由图可见, 除个别点外, 网络输出基本穿过实测点, 即采用改进 BP 算法的训练结果较理想, 网络输出值与实测值的偏差较小。

3. RBF 网络的训练结果

如表 4-2 所示, 将样本集中 (如表 4-2 所示) 的 19 组数据 (除第 8 组数据外) 作为图 4-11 网络的输入, 网络结构为 6- x -1, 在误差平方和小于 0.0001 时, 停止训练, 得到的训练网络的输出值与实测值如图 4-14 所示, 图中的 \bigcirc ——表示实测点, 由图可见网络输出穿过全部实测点, 即采用 RBF 网络训练的结果较好, 偏差几乎为零。

通过以上网络训练结果可以看出, RBF 网络训练结果令人满意, 训练后的网络输出值与全部实测值重合, 误差几乎为零, 但是, 从表 4-3 可看出, 对于未参与训练的第 8 组样本, 其预测值误差太大, 即预测值为 881.38, 实测值为 1270, 预测的相对误差为 30.6%。对于采用改进 BP 算法 (BPX) 的训练结果也比较理想, 但预测结果误差仍较大。而采用基本 BP 算法的网络训练结果不理想, 即有些网络输出值与实测值偏差较大, 但是, 其预

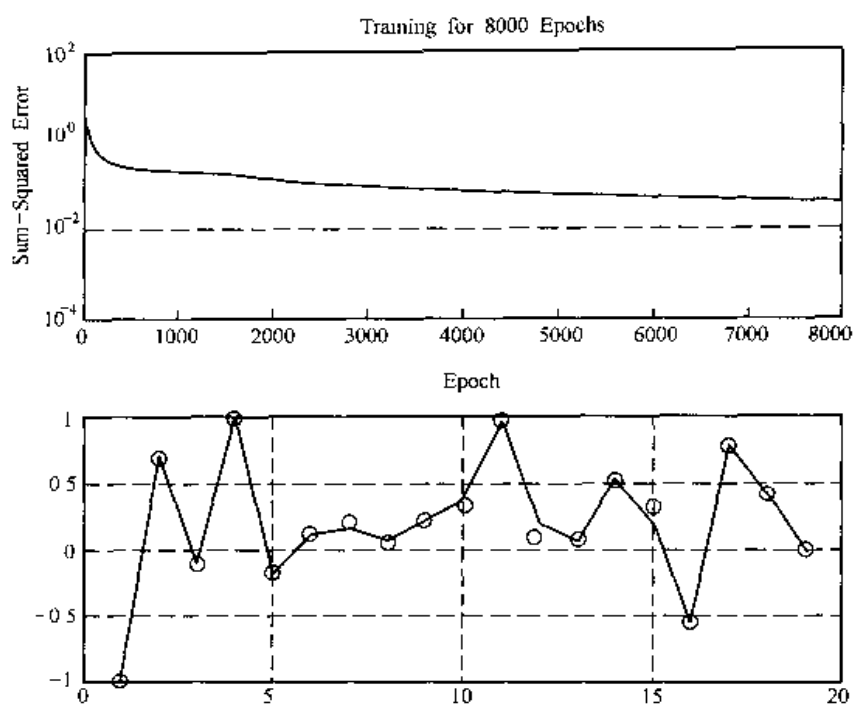


图 4-13 BPX 网络对样本集（除第 8 组外）的训练过程

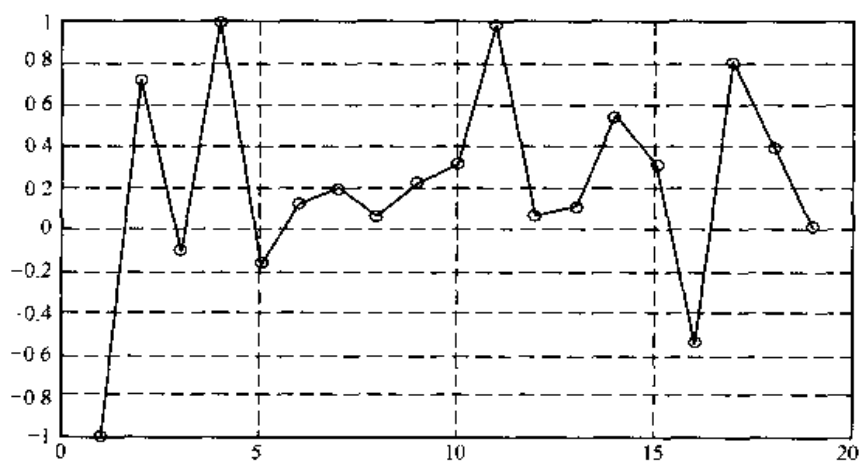


图 4-14 RBF 网络对样本集（除第 8 组外）的训练过程

测结果相对较好。为什么 RBF 网络和改进 BP 算法网络得不到满意的预测结果，在此可通过图 4-15 曲线拟合示意图进一步理解，图 4-15 (a) 中的网络输出值虽然不能穿过每一个实测点，但其预测值与实测值偏差相对较小；而图 4-15 (b) 中的网络输出值能精确地通过全部实测点，但其预测值却远离实测点，这种现象就是过拟合，或称该网络泛化能力差。

影响网络泛化 (generalization) 能力的因素大致可归纳为：训练样本的数目、神经网络的结构以及所研究对象本身的复杂程度。通过网络训练的经验总结如下：

(1) 训练样本的数目。一般要求为 $M \geq 3N$ ， M 为样本数， N 为自变量数目。

(2) 研究对象的内在特性。对于输出样本相对变化比较缓慢的问题，网络训练收敛较快，耗机时间较短。

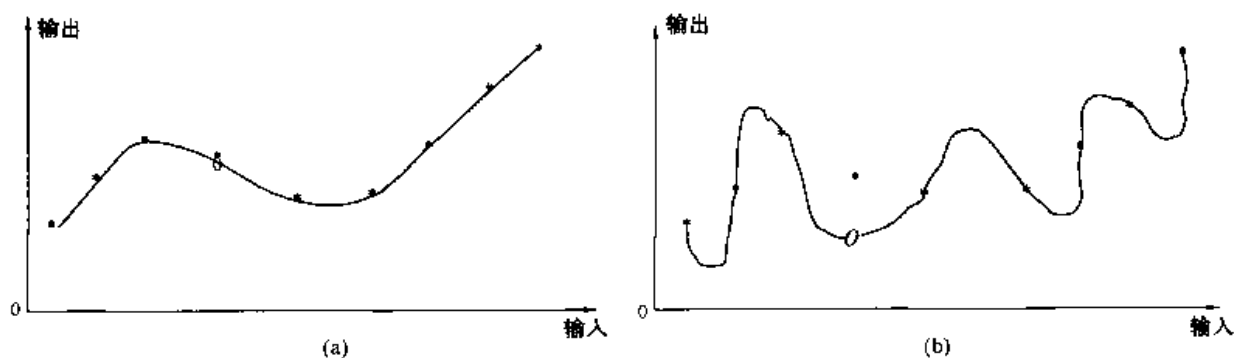


图 4-15 人工神经网络曲线拟合示意图

(a) 较理想的拟合 (泛化能力强); (b) 过拟合 (泛化能力差)

* 一表示实测数据; O—表示预测数据

——待预测的实测数据; ——网络输出值

(3) 网络结构。当样本数目、研究对象的内在特性已经确定时, 网络结构的确定或选择对提高预测精度是十分重要的因素。

(三) 小结

应用人工神经网络模型对气相色谱保留值进行预测研究。基于相关文献中的数据, 建立了反映溶质的保留指数与溶质的溶剂化参数之间关系的数学模型, 分别采用 BP 网络、改进 BP 算法网络及径向基函数 (RBF) 网络对溶质的保留指数进行了预测, 并对预测结果进行了分析比较。说明不同网络用于保留指数的拟合与预测, 其结果差别较大。

第五节 本章小结

(1) 在径向基函数网络里, 非线性映射由并非单调增减的高斯函数的线性组合得到, 而不像 BP 网络那样是单调增减的 Sigmoid 函数。由隐层作用函数产生的表面, 形似一系列山丘, 这些山丘的高度可由第二层 (即输出层) 神经元的标量权调节。如果提供足够数目的隐层神经元, 则通过选择合适的中心、归一化参数和输出权, 就可以很好地逼近所要描述的非线性函数。

(2) RBF 网络的学习过程分为两个阶段: 第一阶段, 根据所有的输入样本决定隐层各节点的高斯函数的中心值 c_j 和标准化常数 (宽度) σ_j ; 第二阶段, 在确定完隐层的参数后, 根据样本, 利用最小二乘原则, 求出输出层的权值 w_j 。

(3) 径向基函数网络学习算法主要有以下几种: ①Mooky 与 Darken 算法; ②局部训练算法; ③正交优选算法; ④聚类和 Givens 变换联合迭代算法。比较这几种算法, 第三、四种方法比较适合 RBF 网络的迭代学习, 而第一、二两种方法都不适合实际应用。虽然第三、四两种方法具有一定的优点, 但在实际应用中仍存在以下一些需要进一步研究的工作: ①网络基函数的中心值确定; ②输出权值计算中的数值变异问题; ③网络的在线校正与数据采集的配合; ④网络综合性能的提高; ⑤过拟合问题。

第五章

Hopfield 反馈网络

1982 年, 美国加州理工学院生物物理学家 J. Hopfield 提出了模拟人脑联想记忆功能的神经网络模型——反馈型神经网络 (即通常称之为 Hopfield 网络)。他提出离散型反馈网络, 并将“能量函数”的概念引入到反馈网络中, 给出了网络的稳定判据。这对人工神经网络的发展产生了巨大影响。1984 年, J. Hopfield 又提出了连续型反馈网络, 采用电子线路 (放大器、电阻、电容等) 模拟网络中神经元的动态方程, 为新一代电子神经计算机的研制奠定了基础。

前馈网络与反馈网络是人工神经网络中两种最基本的网络模型。前馈网络研究的是网络的输出与输入之间的映射关系。由于输出与输入之间无反馈联系, 所以给网络的分析带来了许多方便。而对于反馈网络, 研究的是一个复杂的动力学系统, 该系统通过神经元的状态变迁, 最终稳定于某一状态, 获得联想存储或神经计算的结果。由于网络的输出不仅与当前的输入有关而且与先前的输出 (或输入) 有关, 因此, 描述神经元和神经网络的数学模型只能是动态方程 (微分方程或差分方程)。

反馈网络的模型有多种, 本章主要讨论由 Hopfield 提出的反馈网络。根据激活函数选取不同, Hopfield 网络又可分为离散型 Hopfield 网络和连续型 Hopfield 网络, 离散型 Hopfield 网络的激活函数为二值型, 其输入、输出为 $\{0, 1\}$ 的反馈网络, 主要应用于联想记忆; 连续性 Hopfield 网络的激活函数的输入与输出之间的关系为连续可微、单调上升函数, 主要用于优化计算。但两种网络的基本结构相同。

第一节 离散型 Hopfield 网络

一、离散型 Hopfield 网络结构

Hopfield 网络的基本结构如图 5-1 所示, 图中 N_1, N_2, \dots, N_n 表示网络的 n 个神经元, 各神经元的激活函数是一个二值型的阈值函数, 即 $\{-1, +1\}$ 或 $\{0, 1\}$, 如图 5-2 所示。

设网络的输入向量 $x = (x_1, x_2, \dots, x_n)^T \in \{-1, +1\}^n$; u_j ($j = 1, 2, \dots, n$) 为神经元 j 的输入状态, v_j 为神经元 j 的输出状态, 各状态变量 $v_j(t)$ 通常是指神经元在时刻 t 的输出量, $t \in \{0, 1, 2, \dots\}$ 为离散的时间变量。网络的输入输出关系为

神经元 j 的输入加权和

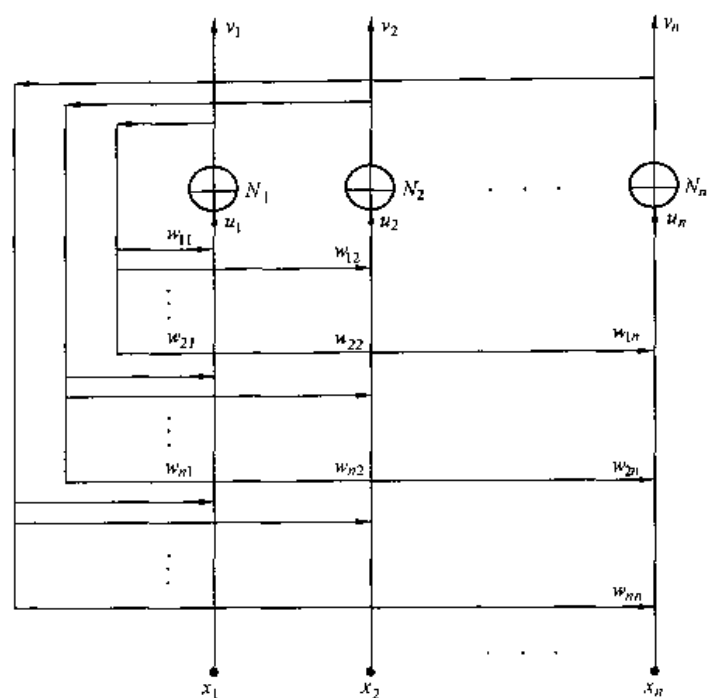


图 5-1 Hopfield 网络结构示意图

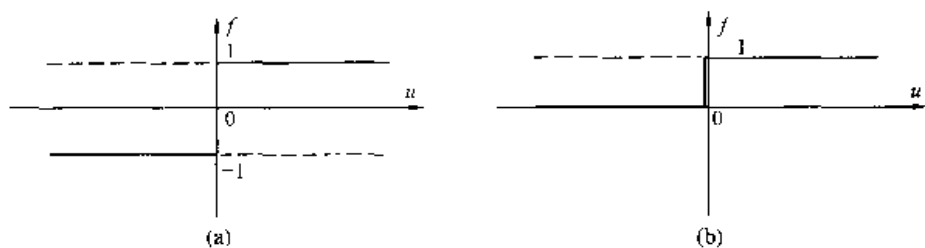


图 5-2 离散型 Hopfield 网络中的激活函数
(a) $(-1, 1)$ 型激活函数; (b) $(0, 1)$ 型激活函数

$$s_j(t) = \sum_{i=1}^n w_{ij}v_i - \theta_j \quad i = 1, 2, \dots, n; j = 1, 2, \dots, n \quad (5-1)$$

神经元 j 的当前时刻 t 的输入状态

$$u_j(t) = s_j(t) \quad j = 1, 2, \dots, n \quad (5-2)$$

神经元下一时刻 $t+1$ 的输出状态

$$V_j(t+1) = f[u_j(t)] = \text{sgn}\left[\sum_{i=1}^n W_{ij}V_i(t) - \theta_j\right] \quad i = 1, 2, \dots, n; j = 1, 2, \dots, n \quad (5-3)$$

上式中 θ_j 为神经元 j 的阈值, 为了分析简便, 令 $\theta_1 = \theta_2 = \dots = \theta_n = 0$; $\text{sgn}(\cdot)$ 为符号函数, 其取值为

$$\text{sgn}[u_j(t)] = \begin{cases} +1 & u_j(t) \geq 0 \\ -1 & u_j(t) < 0 \end{cases} \quad j = 1, 2, \dots, n \quad (5-4)$$

若符号函数 $\text{sgn}(\cdot)$ 的取值为 $\{0, 1\}$, 则式 (5-3) 可以写为

$$V_j(t+1) = \frac{1}{2} \{1 + \operatorname{sgn}[\sum_{i=1}^n W_{ij} V_i(t) - \theta_j]\} \quad (5-5)$$

当网络经过适当训练后, 连接权矩阵 $w = (w_{ji})$ 已经确定, 可以认为网络处于等待工作状态。若给定网络上的初始输入为 x , 则网络各神经元就处于特定的初始状态, 由此可得出当前时刻网络的输出状态。通过网络的反馈作用, 可得到下一时刻网络的输入信号; 再由这个新的输入信号作用于网络, 可得到下一时刻网络的输出状态, 将该输出反馈到输入端, 又形成新的输入信号, 如此不断地循环下去。如果网络是稳定的, 那么, 经过多次反馈运行, 网络达到稳态, 即由输出端可得到网络的稳态输出。

网络的运行方程为

$$\begin{cases} V_j(t+1) = \operatorname{sgn}[\sum_{i=1}^n W_{ij} V_i(t) - \theta_j] \\ V_j(0) = x_j \end{cases} \quad (5-6)$$

若达到 t 时刻后, 网络状态不再改变, 已收敛至稳定点, 即有

$$V(t+1) = V(t) \quad (5-7)$$

这时由输出端可得到网络的稳定输出

$$y = V(t) \quad (5-8)$$

反馈网络有两种基本的工作方式: 串行异步和并行同步方式。

1. 串行异步方式

在任一时刻 t , 随机或确定性地对网络中的某一神经元的状态进行更新, 而其余神经元的状况保持不变。其串行异步工作方式的调整算法为

$$\begin{cases} V_j(t+1) = \operatorname{sgn}[u_j(t)] \\ \quad = \operatorname{sgn}[\sum_{i=1}^n W_{ij} V_i(t) - \theta_j] \\ V_i(t) = V_i(t+1) \quad i \in (1, 2, \dots, n) \text{ 但 } i \neq j \end{cases} \quad (5-9)$$

如果按确定的顺序来更新每一神经元的状态, 则称为顺序更新; 若按随机方式来选择更新神经元, 则称为随机更新。

2. 并行同步方式

在任一时刻 t , 网络中部分神经元 (比如同一层的部分神经元) 的状态同时更新, 或网络中全部神经元的状态同时进行更新, 均称为同步工作方式。其调整算法为

$$\begin{aligned} V_j(t+1) &= \operatorname{sgn}[u_j(t)] \\ &= \operatorname{sgn}[\sum_{i=1}^n W_{ij} V_i(t) - \theta_j] \quad j = 1, 2, \dots, n \end{aligned} \quad (5-10)$$

二、离散型 Hopfield 网络的稳定性

离散 Hopfield 网络实质上是一个离散的非线性动力学系统。如果系统是稳定的, 则它可以从任一初始状态收敛到一个稳定状态; 若系统是不稳定的, 由于网络的输出只有 $\{-1, +1\}$ 或 $\{0, +1\}$ 两种状态, 因而系统不可能出现无限发散, 只可能收敛到极限环或出现限幅的自持振荡。如果将反馈网络稳定的平衡状态作为一种记忆, 那么当网络由任

一初始状态向稳态的转化过程,实质上是一种寻找记忆的过程,网络所具有的稳定平衡点是实现联想记忆的基础,初始可以认为是给定样本的部分信息,网络改变的过程可认为是从部分信息找到全部信息,实现联想记忆的过程。

1. 吸引子和吸引域

如果从网络的初始状态出发,经过一定的时间 t ($t > 0$) 后,到达稳定状态,并保持不变,则称该网络是稳定的,其稳定状态的表达式为

$$V(t + t_0 + \Delta t) = V(t_0 + t) \quad \Delta t > 0 \quad (5-11)$$

若取 $t_0 = 0$, $\Delta t = 1$, 则上式可写为

$$V(t + 1) = V(t) \quad (5-12)$$

或

$$\begin{aligned} V_j(t + 1) &= V_j(t) = \text{sgn}[u_j(t)] \\ &= \text{sgn}\left[\sum_{i=1}^n W_{ij}V_i(t) - \theta_j\right] \\ &= V_j(t) \quad j = 1, 2, \dots, n \end{aligned} \quad (5-13)$$

于是有

$$V_j(t)S_j(t) > 0 \quad j = 1, 2, \dots, n \quad (5-14)$$

式 (5-14) 表明,若 $V(t)$ 为网络的稳定状态,则其各神经元的加权输入量与输出量是同号的,因而其乘积大于零。故式 (5-14) 可作为网络稳定状态的条件表达式。

为了简化分析,令阈值 $\theta_j = 0$, 由 (5-1) 式稳定状态表达式 (5-13) 可表示为

$$\begin{aligned} V(t + 1) &= \text{sgn}[S_j(t)V(t)] \\ &= \text{sgn}[W \cdot V(t)] = V(t) \end{aligned} \quad (5-15)$$

若存在某一有限时刻 t , 从 t 以后的网络状态不再发生变化,即满足

$$V(t + \Delta t) = V(t) \quad \Delta t > 0 \quad (5-16)$$

则称该网络是稳定时,处于稳定时的网络状态叫做稳定状态,也称为“吸引子”,对于非线性系统来说,不同的初始值,可能达到不同的稳定点,因此,在一个反馈网络,存在很多稳定点,而我们希望网络记忆的样本皆为网络的稳定状态;除此之外的稳定点不是网络设计所需求的解(即稳定状态),所以称这些稳定点为伪稳定点。

为了实现良好的联想记忆功能,当网络输入存在缺损样本或输入信号受到干扰时,网络具有自动联想的能力,能由部分信息恢复全部信息,并吸引到样本向量所对应的吸引子上。

为了能够实现正确的联想记忆,对于每个吸引子应该有一定的吸引范围,这个吸引范围便称为吸引域。下面给出两个定义:

(1) 若 $V^{(a)}$ 是吸引子,对于异步方式,若存在一个调整次序可以从 V 演变到 $V^{(a)}$, 则称 V 弱吸引到 $V^{(a)}$;若对于任意调整次序都可以从 V 演变到 $V^{(a)}$, 则称 V 强吸引到 $V^{(a)}$ 。

(2) 对所有 $V \in R[V^{(a)}]$ 均有 V 弱(强)吸引到 $V^{(a)}$, 则称 $R[V^{(a)}]$ 为 $V^{(a)}$ 的弱(强)吸引域。

对于并行同步方式, 由于无调整次序问题, 所以相应的吸引域也无强弱之分。而对于串行异步方式, 对同一个状态, 若采用不同的调整次序, 有可能弱吸引到不同的吸引子。

2. Hamming 距离

为了说明各向量之间的差异程度, 需引入某种度量的标准。通常使用信息与编码理论中广泛采用的海明距离 d_H 来作为度量的标准。海明距离定义为两个向量中不相同的元素的个数。即 $d_H(V^l \text{ 与 } V^k) = V^l \text{ 与 } V^k \text{ 中对应元素不相同的总个数}$ 。

对于元素取二值 $V_j \in \{-1, +1\}$, 则有

$$d_H(V^l, V^k) = \frac{1}{2} \sum_{j=1}^n |V_j^l - V_j^k| \quad (5-17)$$

由上式可知, 若 $V^l = V^k$, 则 $d_H(V^l, V^k) = 0$; 若两个 n 维向量各元素均相反, 则 $d_H(V^l, V^k) = n$ 。

3. 能量函数

李雅普诺夫函数是动力学系统理论中的一个重要工具, Hopfield 将李雅普诺夫稳定性第二方法的一种推广应用, 也是对神经网络发展的一个重要贡献。Hopfield 定义反馈网络的能量函数为

$$E = -\frac{1}{2} \sum_i \sum_j W_{ij} V_i V_j + \sum_i \theta_i V_i \quad (5-18)$$

由于 $V_i, V_j \in \{-1, +1\}$, W_{ij} 与 θ_i 有界, $i, j = 1, 2, \dots, n$, 故能量函数 E 是有界的。即

$$|E| \leq \frac{1}{2} \sum_i \sum_j |W_{ij}| |V_i| |V_j| + \sum_i |\theta_i| |V_i| = \frac{1}{2} \sum_{ij} |W_{ij}| + \sum_i |\theta_i| \quad (5-19)$$

Hopfield 能量函数的物理意义是: 在网络系统渐近稳定点的吸引域内, 离吸引点越远的状态, 所具有的能量越大, 由于在状态演变过程中, 能量函数 E 是单调下降的, 这样, 保证状态的运动方向能从远离吸引点处, 不断地趋于吸引点, 直至到达稳定点, 此时能量达到极小点而不再改变 (即 $\Delta E = 0$)。那么网络就处于稳定状态。

4. 稳定性结论

结论 1: 对于离散 Hopfield 网络, 若按异步方式调整状态, 且连接权矩阵 W 为对称阵, 即满足 $W_{ij} = W_{ji}$, $W_{ii} = 0$, $i, j = 1, 2, \dots, n$, 则从任意状态出发, Hopfield 网络最终将收敛到一个稳定状态。

证明: 当网络工作在串行异步方式时, 设在 t 时刻只有第 I 个神经元调整状态, 其状态可能为

$$\begin{aligned} \Delta V_j &= V_j(t+1) - V_j(t) \\ &= \begin{cases} 0 & V_j(t+1) = V_j(t) \\ 2 & V_j(t+1) = 1, V_j(t) = -1 \\ -2 & V_j(t+1) = -1, V_j(t) = 1 \end{cases} \end{aligned} \quad (5-20)$$

于是由式 (5-18) 可得

$$\begin{aligned}\Delta E &= E(t+1) - E(t) \\ &= -\frac{1}{2} \sum_{j=1}^n W_{ij} V_j(t) \Delta V_i - \frac{1}{2} \sum_{j=1}^n W_{ji} V_j(t) \Delta V_i + \theta_i \Delta V_i\end{aligned}\quad (5-21)$$

由于网络对称, $W_{ij} = W_{ji}$, $W_{ii} = 0$, 故上式可写为

$$\Delta E = - \left[\sum_{j=1}^n W_{ij} V_j(t) - \theta_i \right] \Delta V_i = - S_i(t) \Delta V_i \quad (5-22)$$

由于

$$\begin{cases} V_i(t+1) = \text{sgn}[S_i(t)] \\ \Delta V_i = V_i(t+1) - V_i(t) \end{cases} \quad (5-23)$$

从式 (5-22) 可看出

当 $S_i(t) \geq 0$ 时, $\Delta V_i \geq 0$, 则 $\Delta E \leq 0$

当 $S_i(t) < 0$ 时, $\Delta V_i \leq 0$, 则 $\Delta E \leq 0$

因此, 在何种状态下均有 $\Delta E \leq 0$, 由于 $E(t)$ 是有界函数, 所以 $E(k)$ 将收敛到一常数, 从而保证了网络的稳定性。

结论 2: 对于离散 Hopfield 网络, 若按同步方式调整状态, 且连接权矩阵 W 为非负定对称阵, 即满足 $W_{ij} = W_{ji}$, $W_{ii} > 0$, ($i, j = 1, 2, \dots, n$), 则对于任意初态, 网络都最终收敛到一个稳定状态。

由于网络按同步方式工作时, 必须满足权矩阵 W 为非负定阵 (即 $W \geq 0$), 网络从任意初态出发最终都将收敛到一个稳定状态; 当权矩阵不满足非负定条件时, 网络将不能保证同步演变过程的收敛性, 有可能出现自持振荡即极限环。

因此, 异步工作方式比同步方式有更好的稳定性能, 一般情况下, 常采用异步工作方式, 而异步工作方式的主要缺点是失去神经网络并行处理的优点。

三、离散型 Hopfield 网络的基本学习规则

Hopfield 网络的学习目的是: 对具有 p 个不同输入样本组 x^1, x^2, \dots, x^p , $x^i \in \{-1, +1\}^n$, 希望通过调节计算有限的权值矩阵 W , 使得当每一组输入样本 x^i ($i = 1, 2, \dots, n$), 作为系统的初始值, 经过网络的工作运行后, 网络的状态能够收敛到各自输入样本向量本身。

在离散型 Hopfield 网络学习过程中, 通常采用 Hebb 学习规则, 该规则是一种无指导的死记式学习算法, 其基本原理是: 当网络中神经元输入输出节点的状态相同 (即同时兴奋或抑制) 时, 从第 j 个到第 i 个神经元之间的连接强度增强, 否则减弱。下面介绍 Hebb 学习规则。

1. 对一个模式的学习

为了分析简便, 首先考虑网络对一个模式的学习, 这时网络需要存储的记忆样本只有 1 个 (设为 x^1), 它将成为网络的稳定状态, 并具有最大的吸引域, 或称具有最大的“纠错能力”。将记忆样本 x^1 输入到网络, 并作为网络的初始状态, 经过网络的工作运行后, 它将成为网络的稳定状态, 设阈值 $\theta_j = 0$, 由式 (5-10) 有

$$x'_j = \text{sgn}\left(\sum_{i=1}^n W_{ij}x'_i\right) \quad j = 1, 2, \dots, n \quad (5-24)$$

对于第 j 个神经元, 由 Hebb 学习规则可得网络的权值与输入向量的关系式为

$$W'_{ij} = \alpha x'_j x'_i \quad (5-25)$$

式中, $i, j = 1, 2, \dots, n$, 比例系数 $\alpha > 0$ 为一常数, 一般取 $\alpha = 1$ 或 $\alpha = 1/n$ (n 为样本向量的维数)。

当网络只有一个输入模式时, 它不仅是网络的吸引子, 而且可以证明此时的网络具有相当于 $n/2$ 的最大纠错能力。假设输入网络的观察向量为 x^* , 它与记忆样本 x' 的 Hamming 距离设为 $d_H(x^*, x') = k$, 则网络的输出为

$$V_j = \text{sgn}(S_j) \quad j = 1, 2, \dots, n \quad (5-26)$$

其中第 j 个神经元的加权输入为

$$S_j = \sum_{i=1}^n W_{ij}x_i^* \quad (5-27)$$

将式 (5-25) 代入式 (5-27) 可得

$$S_j = \sum_{i=1}^n \alpha x'_j x'_i x_i^* \quad (5-28)$$

常用的 Hamming 距离来度量两个模式 x^* 和 x' 之间距离, 定义为 $d_H(x^*, x')$ 为

$$d_H(x^*, x') = \frac{1}{2} \sum_i (1 - x_i^* x'_i) \quad x \in \{-1, +1\} \quad (5-29)$$

若设 $d_H(x^*, x') = k$, 则式 (5-29) 可导出

$$\sum_{i=1}^n x'_i x_i^* = n - 2k \quad (5-30)$$

因此, 由式 (5-28) 可得

$$\begin{aligned} S_j &= \sum_{i=1}^n \alpha x'_j x'_i x_i^* \\ &= \alpha x'_j \sum_{i=1}^n x'_i x_i^* \\ &= \alpha x'_j (n - 2k) = \alpha x'_j [(n - k) - k] \end{aligned} \quad (5-31)$$

上式中 $(n - k)$ 为 x^* 与 x' 中元素相同的各元素乘积之和; $(-k)$ 为 x^* 与 x' 中元素不同的各元素乘积之和, 故式 (5-26) 可改写成

$$V_j = \text{sgn}(S_j) = \text{sgn}[\alpha x'_j (n - 2k)] \quad (5-32)$$

为使网络的输入观察向量能收敛到吸引子 x' 上, 根据稳定条件应满足

$$x'_j = \text{sgn}[\alpha x'_j (n - 2k)] \quad (5-33)$$

因 $\alpha > 0$, 由上式可得

$$n - 2k > 0 \text{ 或 } k < n/2 \quad (5-34)$$

从上式 (5-34) 可看出, 当 Hopfield 网络只有一个记忆样本时, 网络的最大纠错能力可达 $n/2$ Hamming 距离, 即当输入 50% 的信息时, 网络可联想出全部信息。

2. 对多个模式的学习

当 Hopfield 网络只输入一个样本, 能使网络收敛到自己。但对于同一权向量 W , 网络不仅要能够使一组输入状态收敛到其稳定值, 而且能够同时记住多个稳态值, 即同一个网络权向量必须能够记住多组输入样本, 使其同时收敛到不同的对应稳态值。因此, 按照 Hebb 学习规则, 当 k 由 1 增加到 2, 直至 p 时, 即当网络需要学习的记忆样本有 p 个 x^1, x^2, \dots, x^p 时, 则在原有设计权值的基础上, 在式 (5-25) 中增加一个新量 $x_i^k x_j^k$, $k = 2, \dots, p$, 所以对网络所有输入样本记忆权值的设计公式为

$$W_{ij} = \alpha \sum_{k=1}^p x_i^k x_j^k \quad (5-35)$$

式 (5-35) 为式 (5-25) 的推广, 当系数 $\alpha = 1$, 根据 Hebb 规则, 称式 (5-35) 为 p 个学习模式 ($x^1, x^2, x^3, \dots, x^p$) 的外积和。由于在 Hopfield 网络由 $W_{ij} = W_{ji}$, 并在实际应用中通常取 $W_{ii} = 0$, 因为不带自环的 Hopfield 网络的稳定性易于保证。对于不带自环的网络, Hebb 学习算式 (5-35) 可改写为

$$W_{ij} = \begin{cases} \alpha \sum_{k=1}^p x_i^k x_j^k & i \neq j \\ 0 & i = j \end{cases} \quad (5-36)$$

用向量矩阵表示为

$$W = \alpha \sum_{k=1}^p [x^k (x^k)^T - I] \quad (5-37)$$

当 $\alpha = 1$ 时有

$$W = \sum_{k=1}^p x^k (x^k)^T - PI \quad (5-38)$$

式中 I ——单位对角矩阵。

由式 (5-37) 和式 (5-38) 所形成的网络权值矩阵为零对角阵。

当网络学习过程一旦结束, 连接权矩阵就形成, 就把 p 个模式记忆在网络之中, 此时的网络便可进入工作状态。在输入观察向量 (或未知向量) X 后, 网络按运行方程 (5-6) 进行状态演变, 直至收敛到稳定状态。此时网络的稳态输出即是联想的结果。

四、影响记忆容量的因素

从 Hebb 学习规则可以看出, 当网络只记忆一个模式时, 该模式就被准确地记忆在网络中, 且具有最大的纠错能力, 可达到 $n/2$ 的 Hamming 距离, 但当需要记忆的模式增多时, 就会产生各模式间的相互影响, 其主要表现为以下两方面:

1. 权值移动

按照 Hebb 学习规则, 网络对输入模式 x^1, x^2, \dots, x^p 的学习记忆是逐个实现的, 比如, 对无自环网络的权矩阵的学习记忆过程对应的编程为

```

W = 0
For k = 1, p
    W = W + α [xk (xk)T - I]
end
    
```

当 $k=1$ 时, 有

$$W_{ij} = x_j^1 x_i^1 \quad (i \neq j)$$

$$V_j^1 = \operatorname{sgn}\left(\sum_{i=1}^n W_{ij} x_i^1\right) = \operatorname{sgn}\left(\sum_{i=1}^n x_i^1 x_j^1 x_i^1\right) = x_j^1 \quad (5-39)$$

此时, 网络准确地记住模式 x^1 , 当 $k=2$ 时, 为了记忆样本 x^2 , 连接权值需在原有基础上移动。所求出新的 W 为 $W_{ij} = x_j^1 x_i^1 + x_j^2 x_i^2$, 而对样本 x^1 来说, 网络的输出为

$$V_j^1 = \operatorname{sgn}\left(\sum_{i=1}^n W_{ij} x_i^1\right) = \operatorname{sgn}\left[x_j^1 + \sum_{i=1}^n (x_i^2 x_j^2 x_i^1)\right] \quad (5-40)$$

在此设 $\alpha=1$, 可见, 对于适用于模式 x^1 的式 (5-24) 可能不再对所有 $j \in \{1, 2, \dots, n\}$ 都成立, 有可能部分地遗忘了以前已记忆住的模式。

对于模式 x^2 来说, 由于权矩阵已学习了 x^1 , 其初始值不再是零, 因而式 (5-35) 也不可能对所有的 $j \in \{1, 2, \dots, n\}$ 都成立, 这就是说, 难以保证网络对模式 x^2 一定能准确地记忆。

随着学习模式数 k 的增加, 初始权值移动现象将进一步发生, 在学习了第 p 个模式 x^p 后, 其初始权值已是经过 $p-1$ 次移动。因此可能存在这样的问题: 经过学习训练的网络有可能部分地遗忘了先前已学习过的样本, 致使网络无法保证能够记住已学习过的全部模式。依据动力学观点, 当 k 较小, 即输入学习的模式较小时, 可以使输入学习的模式成为网络的吸引子。随着 k 值的增大, 可能不仅难以保证后来的学习模式成为吸引子, 而且将使原先已记住的吸引子的吸引域变小, 甚至使网络对已学习过的模式发生演化或遗忘。这种现象被称为“网络疲劳”。对于一定规模的网络 (即节点数 n 一定时), 存在能学习记忆的模式数的上限 P_{\max} , 这即所谓的“网络容量”。当学习的模式数 $k > P_{\max}$ 时, 网络不但不能准确记忆以后输入的模式, 而且对先前输入模式的记忆也将逐渐淡忘。

2. 交叉干扰

从式 (5-40) 可知, “网络疲劳”的主要原因在于交叉干扰项的存在。假设网络的权矩阵已经建立, 若输入向量为 x^l , $l \in \{1, 2, \dots, P\}$, 希望其成为网络的吸引子, 由式 (5-15) 有

$$x^l = \operatorname{sgn}(Wx^l) \quad (5-41)$$

其中网络的加权输入和向量为

$$Wx^l = \alpha \left[\sum_{k=1}^p (x^k (x^k)^T - I) \right] x^l \quad (5-42)$$

设 $\alpha = 1/(n-1)$, 由于 $x^l \in \{-1, +1\}^n$, 于是 $(x^l)^T x^l = n$, 由式 (5-42) 可得

$$Wx^l = x^l + \frac{1}{n-1} \sum_{\substack{k=1 \\ k \neq l}}^p [x^k (x^k)^T - I] x^l \quad (5-43)$$

从式 (5-43) 可以看出, 上式中右边的第一项为期望记忆的模式, 即是网络与 x^l 相应的吸引子; 而上式中右边的第二项则是当网络学习多个样本时。所产生的对式 (5-43) 右边第一项的干扰, 故称之为交叉干扰项。因此, 网络对于所学习模式集中某一模式能否正确地记忆和联想, 完全取决于式 (5-43) 右边第一项与第二项的大小及其符号关系。下面讨

论式 (5-43) 中右边第二项对网络工作过程的影响。

首先令

$$C_j^l = x_j^l \left/ \frac{1}{n} \sum_{\substack{i=1 \\ i \neq j}}^n \sum_{\substack{k=1 \\ k \neq l}}^p x_j^k x_i^k x_i^l \right. \quad (5-44)$$

(1) 当 $c_j^l \geq 0$ 时, 交叉干扰项与 c_j^l 符号一致, 因此, 能够在网络的输出端得到正确的结果, 即 $v_j^l = x_j^l$ 。

(2) 当 $c_j^l \leq -1$ 时, 交叉干扰项与 x_j^l 符号相反, 对正确记忆 x_j^l 产生干扰, 但其干扰幅度较小, 而不至于使状态 $\text{sgn}(s_j^l)$ 或 $\text{sgn}(Wx_j^l)$ 的符号翻转, 输出仍然正确, 可得到 $V_j^l = x_j^l$ 。

(3) $-1 < c_j^l < 0$ 时, 交叉干扰项与 x_j^l 不仅符号相反, 而且其值幅度大于正确的输出信号值 x_j^l , 从而造成状态 $\text{sgn}(Wx_j^l)$ 的符号翻转, 导致 Hopfield 网络的第 j 个神经元输出错误信息, 即 $V_j^l = -x_j^l$ 。

五、离散 Hopfield 网络用于联想记忆

由于 Hopfield 网络是利用稳定状态来对信息进行记忆的, 是利用从初态到稳定状态的演变过程来实现对信息的联想。在前面介绍的人工神经网络中, 如对模式进行识别时, 是在网络已存储的“知识”中寻求与该输入匹配最好的存储“知识”作为其解, 这是一种完成回忆与联想的过程。下面介绍离散型 Hopfield 网络用于联想记忆的基本方法。

1. 网络权值的学习

在 Hopfield 网络中, 所谓的联想是指通过学习训练使网络在稳定时将一组模式 (x^1, x^2, \dots, x^p) 存储在网络中 (以网络的连接权值来表现), 使这组模式 (x^1, x^2, \dots, x^p) 成为网络的 p 个吸引子。以后任意给定一个待测试的模式 x^l 作为网络的输入, 使网络处于运行状态。若 x^l 为模式组 (x^1, x^2, \dots, x^p) 中的一个, 设 x^l 为 x^1 , 则在稳定时, 稳定在吸引子 x^1 上; 如果 x^l 与模式组 (x^1, x^2, \dots, x^p) 中的一个邻近, 则网络稳定在最近的一个吸引子上, 这样过程称为自联想。通常采用 Hamming 距离来度量两个模式 x^i 和 x^j 之间距离, 其定义如式 (5-29), 下面的问题是讨论如何训练网络权值, 以使样本模式成为网络的稳定吸引子。通常采用 Hebb 学习规则。设有 p 个样本模式 (x^1, x^2, \dots, x^p) , 其中, $x^k = (x_1^k, x_2^k, \dots, x_n^k)$, $x^k \in \{-1, +1\}^n$, 可确定两个神经元 i, j 之间的连接权的关系为

$$W_{ij} = \begin{cases} \alpha \sum_{k=1}^p x_i^k x_j^k & i \neq j \\ 0 & i = j \end{cases} \quad (5-45)$$

写成向量形式, 权值矩阵为

$$W = \alpha \sum_{k=1}^p [(x^k)(x^k)^T - I] \quad (5-46)$$

式中 I ——单位阵。

按此规则设计的权值, 满足 $W_{ij} = W_{ji}$, $W_{ii} = 0$ 。系数 α 通常取值为 $\alpha \approx 1$ 或 $\alpha = 1/n$ 。

对样本模式按式 (5-36) 确定权值后, 就把 p 个模式存储记忆在网络中, 这样就可以用确定好的权值对此后输入的待测试的模式进行联想。

2. 联想记忆的基本算法步骤

下面给出一个采用 Hebb 规则, 按异步更新的离散型 Hopfield 网络的算法步骤:

(1) 初始化权值 (学习阶段), 置 $W = [0]$ 。

(2) 将 p 个样本模式 x^1, x^2, \dots, x^p 输入网络, 确定网络的权值

$$W_{ij} = \begin{cases} \alpha \sum_{k=1}^p x_i^k x_j^k & i \neq j \\ 0 & i = j, 1 \leq i, j \leq n \end{cases}$$

(3) 初始化未知输入模式 x^l (运行阶段)

$$x_j(0) = x_j^l \quad 1 \leq j \leq n$$

式中 x_j ——输入模式的第 j 个元素, $x_j \in \{-1, +1\}$ 。

(4) 迭代直至收敛

$$x_j(t+1) = \operatorname{sgn}\left[\sum_{i=1}^n W_{ij}x_i(t)\right] \quad 1 \leq j \leq n$$

式中 $x_j(t)$ ——时刻 t 时神经元 j 的输出状态。

(5) 稳态输出。

当 $x_j(t+1) = x_j(t)$, $1 \leq j \leq n$ 时, 其网络的稳态输出表明与未知输入模式最好的匹配。

3. 网络的联想特性

在采用 Hebb 规则设计权值时, 满足 $W_{ij} = W_{ji}$, $W_{ii} = 0$, 由前面稳定性的讨论可知, 在异步方式工作时, 网络是稳定的; 如果网络的权矩阵是正定的, 那么在同步方式工作时, 网络也是稳定的。但是, 当输入一个未知模式, 网络稳定在哪一点? 能否稳定到要求记忆的点上? 这是关心的主要问题。下面介绍几条性质, 可以用来判断网络的稳定点和网络的容量。

首先定义正交性, 定义: 设有 p 个样本模式 x^k , $k = 1, 2, \dots, p$, 若任意两个样本模式向量 x^i, x^j , $i, j \in k$, $i \neq j$, $i, j = 1, 2, \dots, n$, 有 $(x^i)^T x^j = 0$, 则称样本是两两正交的。

当 $x \in \{-1, +1\}^n$, 如两个 n 维样本向量中有 $n/2$ 个分量是相同的, 另 $n/2$ 个分量是相反的, 则这两个向量满足正交条件。这时, 其 Hamming 距离等于 $n/2$ 。

性质 1: 设网络各神经元的阈值为零, 若 x^l 是稳定点, 则 $-x^l$ 也是稳定点, 但 Wx^l 中有零元素时例外。

证明: 若 x^l 为稳定状态, 由式 (5-10) 则有

$$x^l = \operatorname{sgn}(Wx^l) \quad (5-47)$$

若对网络输入 $-x$, 则网络的状态演变为

$$\operatorname{sgn}[W(-x^l)] = \operatorname{sgn}(-Wx^l) = -\operatorname{sgn}(Wx^l) = -x^l \quad (5-48)$$

因此, $-x^l$ 也是网络的稳定点。

性质2: 对于无自环网络, 若有两个模式 x^i, x^j 的 Hamming 距离 $d_H(x^i, x^j) = 1$ 或 $n-1$, 则 x^i, x^j 不可能同时成为网络的稳定状态。

证明: 设 $x^i = x^1, x^j = x^2$, 若 $d_H(x^i, x^j) = 1$, 这表明 x^1 与 x^2 只有一个元素不同, 设 $x_1^1 \neq x_1^2$, 而其余的元素均相同, $x_i^1 = x_i^2, i = 2, 3, \dots, n$ 。由式 (5-47) 可知, 若 x^1 为稳定状态, 则网络第一个神经元的输出应为

$$x_1^1 = \operatorname{sgn}\left(\sum_{i=2}^n W_{i1} x_i^1\right) \quad (5-49)$$

对于网络输入 x^2 , 若 x^2 也为稳定状态, 则第一个神经元的输出应为

$$x_1^2 = \operatorname{sgn}\left(\sum_{i=2}^n W_{i1} x_i^2\right) \quad (5-50)$$

式 (5-49) 和式 (5-50) 的右端相同, 于是导出其左端也应相等, 即 $x_1^1 = x_1^2$ 。这与假设条件矛盾。故 x^1 和 x^2 不可能同时成为网络的稳定状态。

当 $d_H(x^i, x^j) = n-1$ 时, 也设 $x^i = x^1, x^j = x^2$, 若 $d_H(x^1, x^2) = n-1$, 则 $d_H(x^1, -x^2) = 1$ 。由性质1可知, 若 x^2 是网络的稳定状态, 则 $-x^2$ 也是网络的稳定状态, 但由上面已推出 x^1 与 $-x^2$ 不可能同时成为稳定状态, 由此可导出 x^1 与 x^2 也不可能同时成为网络的稳定状态。

性质3: 设 p 个模式向量 x^1, x^2, \dots, x^p 满足

$$a_n \leq d_H(x^i, x^j) \leq (1-\alpha)n \quad i \neq j \quad 0 < \alpha < \frac{1}{2} \quad (5-51)$$

若满足条件

$$p \leq 1 + \frac{1}{1-2\alpha} \quad (5-52)$$

当网络的节点数 n 足够大时, 则 x^1, x^2, \dots, x^p 为网络的稳定状态。

证明: 设 $d_H(x^i, x^j) = d_{ij}$, 于是有

$$\begin{aligned} (x^i)^T x^j &= (x^i \text{ 与 } x^j \text{ 中相同元素之乘积和}) - (x^i \text{ 与 } x^j \text{ 中不同元素之乘积和}) \\ &= (n - d_{ij}) - d_{ij} = n - 2d_{ij} \\ |(x^i)^T x^j| &= |n - 2d_{ij}| \leq n - 2an = (1 - 2\alpha)n \end{aligned} \quad (5-53)$$

按照 Hebb 学习规则, 取 $\alpha = 1$, 则有

$$Wx = \sum_{k=1}^p [x^k (x^k)^T - I] x$$

设输入向量 $x = x^l, l \in \{1, 2, \dots, p\}$, 于是

$$Wx^l = (n-p)x^l + \sum_{\substack{k=1 \\ k \neq l}}^p x^k (x^k)^T x^l \quad (5-54)$$

上式右边第二项为交叉干扰项, 由式 (5-53) 可知, 该干扰为 n 维列向量, 其第 i 个元素的大小为

$$\left| \sum_{\substack{k=1 \\ k \neq l}}^p x_i^k (x^k)^T x^l \right| \leq \sum_{\substack{k=1 \\ k \neq l}}^p |x_i^k (x^k)^T x^l| \leq (p-1)(1-2\alpha)n \quad (5-55)$$

分析式 (5-54) 可以看出, 如果交叉干扰项的符号与第一项相反, 则当干扰项第 i 个元素的绝对值大于第一项对应元素的值时, 由符号函数性质可知, 这将导致网络第 i 个神经元输出错误信息。因此, 为了使模式 x^l 成为网络的稳定状态, 其条件是式 (5-54) 的第一项应起主导作用, 即

$$(n - p) > (p - 1)(1 - 2\alpha)n$$

或

$$p < \frac{2n(1 - \alpha)}{n(1 - 2\alpha) + 1} \quad (5-56)$$

当 n 足够大时, 上式可化简为

$$p < 1 + \frac{1}{1 - 2\alpha} \quad (5-57)$$

当 p 满足式 (5-57) 时, 对于 p 个模式向量, 有

$$\text{sgn}(WX^l) = x^l \quad l \in \{1, 2, \dots, p\} \quad (5-58)$$

即 p 个模式均为网络的稳定状态。由性质 3 可知, 对于输入的模式向量, 通常不一定都可成为网络的吸引子, 但当满足一定的条件时, 则可同时成为网络的稳定状态。

性质 4: 当 $\alpha = 1/2$ 时, 性质 3 中的 p 个模式向量 x^1, x^2, \dots, x^p 为正交向量, 且它们都是网络的稳定状态。

证明: 当 $\alpha = 1/2$ 时, 由性质 1 可知 $d_H(x^i, x^j) = \pi/2$, 又根据正交性质, $(x^i)^T x^j = 0, i \neq j$ 时, 这就说明模式向量 x^1, x^2, \dots, x^p 满足相互正交的条件, 又由式 (5-54) 可得, 当网络输入模式向量 x^l 时, 对应的交叉干扰项

$$\sum_{\substack{k=1 \\ k \neq l}}^p x^k (x^k)^T x^l = 0$$

因此, 对所有模式向量 $x^l, l \in \{1, 2, \dots, p\}$, 均满足式 (5-58), 即 p 个输入模式向量均是网络的稳定状态。

性质 5: 当模式向量为正交向量时, 网络可稳定存储的模式数等于网络的神经元数, 皆为 n 。

证明: 若网络的神经元数为 n , 相应地网络的输入、输出状态均为 n 维向量, 对于 n 维的向量空间, 其最大的正交向量子集向量的个数为 n 。由性质 4 (即令 $p = n$), 可证网络的储存模式也为 n 。

一个网络通过学习能记忆的样本模式数称为这一网络的记忆容量, 用 Hebb 规则训练的 Hopfield 网络, 其记忆容量是有限的。通过仿真已经证明, 如果随机产生样本模式, 并且其数目 p 与网络节点数目 n 满足 $p < 0.15n$, 则当输入一个待测模式时, 联想出错的可能性较小。例如, 要存储 10 个模式, 则要求有约 70 个节点, 5000 个连接权值。

第二节 离散型 Hopfield 网络的应用

【例 5-1】对于 $n = 5$ 的离散 Hopfield 网络, 要求记忆的样本为 $x^1 = (1, 1, 1, 1, 1)^T, x^2 = (1, -1, -1, 1, -1)^T, x^3 = (-1, 1, -1, -1, -1)^T$ 。

解：样本并不满足正交条件，由 Hebb 规则训练得到权矩阵为

$$W = [x^1 \quad x^2 \quad x^3] \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ (x^3)^T \end{bmatrix} - 3I = \begin{bmatrix} 0 & -1 & 1 & 3 & 1 \\ -1 & 0 & 1 & -1 & 1 \\ 1 & 1 & 0 & 1 & 3 \\ 3 & -1 & 1 & 0 & 1 \\ 1 & 1 & 3 & 1 & 0 \end{bmatrix}$$

按同步方式工作，计算可得：

(1) 当输入 x^1 , x^2 和 x^3 时，设 $t=1$ ，则

$$\begin{aligned} x^1(1) &= \operatorname{sgn}[Wx^1(0)] = x^1 \\ x^2(1) &= \operatorname{sgn}[Wx^2(0)] = x^2 \\ x^3(1) &= \operatorname{sgn}[Wx^3(0)] = x^3 \end{aligned}$$

即 x^1 , x^2 , x^3 为网络的稳定点。

(2) 当输入 $x^4 = (-1, 1, 1, -1, 1)^T$ 时，设 $t=1$ ，则

$$x^4(1) = \operatorname{sgn}[Wx^4(0)] = x^4$$

由于以 x^4 为稳定点，但 $x^4 = -x^2$ ，不是系统要求记忆的样本，所以称 x^4 为伪稳定点。

(3) 当输入 $x^5 = (1, -1, 1, 1, 1)^T$ 时，设 $t=1$ ，则

$$x^5(1) = \operatorname{sgn}[Wx^5(0)] = x^1$$

即以 Hamming 距离最小收敛于 x^1 ，在模式识别时， x^5 也可以看作带噪声的 x^1 模式。

(4) 当输入 $x^6 = (1, -1, 1, 1, 1)^T$ 时，有

$$x^6(t) \neq x^6(t+1), x^6(t+1) \neq x^6(t-1), t \in \{1, 2, \dots, N\}$$

此时，称网络收敛于周期为 2 的极限环。

【例 5-2】 对于 $n=3$ 的离散 Hopfield 网络，有三个要求记忆的样本为

$x^1 = (1, 1, -1)^T$, $x^2 = (-1, -1, 1)^T$, $x^3 = (1, -1, 1)^T$ ，讨论系统的稳定状态。

解：根据 Hebb 学习规则，取 $\alpha=1$ ，于是可得网络的连接权矩阵为

$$W = \sum_{k=1}^3 [x^k(x^k)^T - I] = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -3 \\ -1 & -3 & 0 \end{bmatrix}$$

验证得

$$\operatorname{sgn}(Wx^1) = x^1, \operatorname{sgn}(Wx^2) = x^2$$

即模式 x^1 与 x^2 均为网络的稳定状态。

而 $d_H(x^3, x^2) = 1$ ，有性质 2 可知，既然 x^2 为稳定状态，那么模式 x^3 不可能成为网络的稳定状态。当输入 x^3 时，网络输出为

$$\operatorname{sgn}(Wx^3) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} = x^2 \neq x^3$$

如果，当输入一个观察向量 $x^* = (-1, 1, -1)$ 时，网络得输出为

$$\operatorname{sgn}(Wx^*) = \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix} = x^1$$

由于 $d_H(x^*, x^1) = \min d_H(x^*, x^i)$, $i \in \{1, 2, 3\}$, 可见, 未知的观察向量 x^* 是按 Hamming 距离最小的意义下, 收敛于网络的吸引子 x^1 , 也可认为未知向量 x^* 为记忆样本 x^1 受了干扰的不完全信息, 在 Hopfield 网络的联想作用下, 使信息得到了恢复。

【例 5-3】 考虑三阶离散 Hopfield 神经网络 $N(W, \theta)$, 其中

$$W = \begin{bmatrix} -0 & 1 & 8 \\ 1 & 1 & 1 \\ 8 & 1 & -1 \end{bmatrix}, \theta = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$$

判断系统属于哪一种工作方式是稳定的。

解: 由定理 1: 神经网络 $N(W, \theta)$ 中, 如果不等式

$$w_{ii} + c_i > 0 \quad i = 1, 2, \dots, n$$

成立, 则该网络是串行稳定的, 其中

$$c_i = \min \left\{ \left| \sum_{j=1}^n w_{ij} \alpha_j - \theta_i \right| \right\} \quad \alpha_j \in \{-1, 1\}$$

可得 $c_1 = 3$, $c_2 = 1$, $c_3 = 2$, 而 $w_{11} = -1$, $w_{22} = 1$, $w_{33} = -1$ 。从而有

$$w_{11} + c_1 = 2 > 0$$

$$w_{22} + c_2 = 2 > 0$$

$$w_{33} + c_3 = 1 > 0$$

根据定理 1, 该神经网络是串行稳定的。

又由定理 2: 如果矩阵

$$R \triangleq W + C$$

是正定的, 则神经网络 $N(W, \theta)$ 是并行稳定的。其中

$$C = \operatorname{diag} \{c_1, c_2, \dots, c_n\}$$

c_i 由定理 1 确定。

由定理 2 可得

$$R = W + C = \begin{bmatrix} 2 & 1 & 8 \\ 1 & 2 & 1 \\ 8 & 1 & 1 \end{bmatrix}$$

由于矩阵 R 不是正定的, 即该神经网络不满足定理 2 并行稳定的条件。若取 $x_1(0) = 1$, $x_3(0) = -1$, 验证该网络, 可以发现 $x_1(2k) = -x_3(2k) = 1$, $x_1(2k+1) = -x_3(2k+1) = -1$, $k = 0, 1, 2, \dots$ 。从而可知, 该神经网络是并行不稳定的。

【例 5-4】 考虑三阶离散 Hopfield 神经网络 $N(W, \theta)$, 其中

$$W = \begin{bmatrix} -1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \theta = \begin{bmatrix} 8 \\ 1 \\ 5 \end{bmatrix}$$

判断系统是否为并行稳定系统。

解：由定理 1 可得 $c_1 = 4$, $c_2 = 1$, $c_3 = 2$ 。再由定理 2 可得

$$R = W + C = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

矩阵 R 是正定的，根据定理 2 可知，该神经网络是并行稳定的。

【例 5-5】 离散型 Hopfield 网络用作联想存储器

图 5-3 给出了其用作联想存储时存储器的内容及联想情况，每一个存储项 [如图 5-3 (a)] 用 “+” 或 “-” 组成的向量来表示，其中 “+” 对应于状态为 “1”，“-” 对应于状态 “0”。即采用 $n = 24$ 个神经元节点构成的 Hopfield 网络，存储三对存储项，每对存储项都由两部分组成，一部分为名称，另一部分为颜色，为清晰起见，图中用 “|” 将两部分分开。每个向量（即每对输入模式）对应一个网络的稳定状态（即吸引子）。

在图 5-3 (b) 给出了网络输入某一存储项的一部分，而联想到另一部分的情况，即首先给网络一个测试模式，名称部分为树，而颜色部分未知（为空）。网络通过计算收敛后，其对应的状态就是要查找的存储项，即名称部分为 “树”，颜色部分为 “青色”，也就是说网络根据学习而存储的模式通过计算由 “树” 而联想到了 “青色”。

在图 5-3 (c) 给出了带有噪声的测试输入模式，此时测试模式的名称部分为带有噪声的 “天空”，颜色部分也是带有噪声的 “蓝色”，网络根据先前所记忆的模式，经过计算收敛后，其输出结果给出了正确的 “天空” 与 “蓝色” 的编码。这说明了 Hopfield 网络具有抗噪声干扰的能力。

输入样本模式		
树	- - + + + + - - - - + - - - + + - + + - - + - -	青色
西红柿	+ - - - + - - + + - + + - + - - - + + - + - + +	红色
天空	- + + - - + + + - - - + + - + + + + - - - - - +	蓝色
		(a)
输入测试模式		
树	- - + + + + - - - - + - - - - - - - - - - - - -	空
网络输出结果		
树	- - + + + + - - - - + - - - + + - + + - - + - -	青色
		(b)
输入带噪声的测试模式		
天空	- + + + - + + - - - - + + - - - - + - - - - - +	蓝色
网络输出结果		
天空	- + + - - + + + - - - + + - + + + + - - - - - +	蓝色
		(c)

图 5-3 Hopfield 网络用作联想存储器

(a) 存储输入模式；(b) 联想输出；(c) 抗噪声干扰输出

【例 5-6】 对于 $n = 4$ 的离散 Hopfield 网络，设训练样本向量为 $x^1 = [1 \ 1 \ 1 \ 1]^T$, $x^2 = [-1 \ -1 \ 1 \ 1]^T$, $x^3 = [1 \ 1 \ -1 \ -1]^T$ 。讨论离散网络的收敛性。

解：网络经训练后得权值矩阵为

$$W = \begin{bmatrix} 0 & 3 & -1 & -1 \\ 3 & 0 & -1 & -1 \\ -1 & -1 & 0 & 3 \\ -1 & -1 & 3 & 0 \end{bmatrix}$$

设测试样本 $x^* = [1 \ 1 \ \epsilon \ 1]^T = x^1 + [0 \ 0 \ \Delta \ 0]^T$, V 为网络输出。

(1) 当 $\epsilon > 2/3$, 或 $\Delta > -1/3$, 则网络状态收敛于 $x^1 = [1 \ 1 \ 1 \ 1]^T$ 。

(2) 当 $\Delta < -1/3$, 则演变过程为 $V = [1 \ 1 \ 1 \ -1]^T \rightarrow V = [1 \ 1 \ -1 \ 1]^T \rightarrow V = [1 \ 1 \ 1 \ -1]^T \cdots$ 无限振荡。

【例 5-7】 Hopfield 网络用于系统参数辨识。

设有一线性多变量离散系统的状态方程

$$\begin{cases} X(k+1) = GX(k) + HU(k) \\ X(0) = x_0 \end{cases} \quad (5-59)$$

其中 $X \in R^n$, $U \in R^m$, $G \in R^{n \times n}$, $H \in R^{n \times m}$ 。该系统是渐进稳定的, 完全能控和完全能观的。系统参数辨识就是已知输入输出数据 $\{U(k), X(k)\}$, 求系统 (5-59) 的系数矩阵 G, H 。

假设系统 (5-59) 的估计模型为

$$Y(k+1) = \hat{G}X(k) + \hat{H}U(k) \quad (5-60)$$

其中 $\hat{G} \in R^{n \times n}$, $\hat{H} \in R^{n \times m}$ 为待估计参数。

(1) 用 Hopfield 网络的神经元的状态输出值对应待识别参数, 并令

$$V = (\hat{G}_1, \hat{G}_2, \dots, \hat{G}_n, \hat{H}_1, \hat{H}_2, \dots, \hat{H}_n)^T \quad (5-61)$$

其中

$$\hat{G}_i = (\hat{g}_{i1}, \hat{g}_{i2}, \dots, \hat{g}_{in}), \hat{H}_i = (\hat{h}_{i1}, \hat{h}_{i2}, \dots, \hat{h}_{im}) (i = 1, 2, \dots, n)$$

$$V \in R^{q \times 1} (q = n^2 + mn)$$

(2) 构造能量函数。估计误差的动态方程为

$$e(k+1) = X(k+1) - Y(k+1) = (G - \hat{G})X(k) + (H - \hat{H})U(k) \quad (5-62)$$

当 $e(k+1)$ 趋于 0 时, \hat{G} 趋向于 G , \hat{H} 趋向于 H 。

因此, 相应的能量函数为

$$\begin{aligned} E &= \frac{1}{2} e^T(k+1) e(k+1) \\ &= \frac{1}{2} [x(k+1) - \hat{G}X(k) - \hat{H}U(k)]^T [X(k+1) - \hat{G}X(k) - \hat{H}U(k)] \\ &= \frac{1}{2} X^T(k+1) X(k+1) + f[\hat{G}, \hat{H}, X(k), U(k)] \\ &= f_0 + f[\hat{G}, \hat{H}, X(k), U(k)] \end{aligned} \quad (5-63)$$

由于 $\frac{\partial f_0}{\partial \hat{G}} = 0, \frac{\partial f_0}{\partial \hat{H}} = 0$, 因此能量函数可简化为

$$E = f[\hat{G}, \hat{H}, X(k), U(k)] \quad (5-64)$$

(3) 将 (5-64) 与标准能量函数式相比较, 可以确定 Hopfield 网络的权值矩阵 T 及偏

置输入 I 。

$$T(k) = -[P(k)Q(k)]^T[P(k)Q(k)] \quad (5-65)$$

其中

$$P(k) = \begin{bmatrix} X^T(k) & & \vdots \\ & \ddots & \\ \vdots & & X^T(k) \end{bmatrix}_{n \times n^2}$$

$$Q(k) = \begin{bmatrix} U^T(k) & & \vdots \\ & \ddots & \\ \vdots & & U^T(k) \end{bmatrix}_{n \times mn}$$

$$I(k) = I_1(k) I_2(k)$$

$$I_1(k) = X_i(k+1) X^T(k)$$

$$I_2(k) = X_i(k+1) U^T(k) \quad (i=1, 2, \dots, n)$$

(4) 启动网络运行, 当网络达到稳定状态时, 即

$$\epsilon = \sum_{i=1}^n [V_i(k) - V_i(k-1)]^2 \leq \epsilon_0 \quad (5-66)$$

则 Hopfield 网络的神经元输出即是需要辨识的参数。

第三节 连续型 Hopfield 网络

连续型 Hopfield 网络的基本结构仍如图 5-1 所示, 连续型 Hopfield 网络与离散型网络相比较, 主要的不同点有: ①网络中神经元的激活函数采用的不是阈值函数, 而是连续可微单调非减的 S 型函数; ②网络的输入状态和输出变量的取值不是离散的二值量, 而是在一定范围内变化的连续量。因此, 在连续网络的整个运行过程中, 所有神经元状态的更新不再是异步方式或同步方式, 而是连续方式, 即网络的所有神经元均连续地随时间 t 并行地更新。

由于连续型 Hopfield 网络的输入和输出都取连续值, 对于网络的单个神经元 (这里设为第 j 个神经元) 而言, 其输入与输出的关系为

$$S_j = \sum_{i=1}^n W_{ij} V_i + I_j \quad j = 1, 2, \dots, n \quad (5-67)$$

$$a_j u_j = -b_j \frac{du_j}{dt} + S_j \quad a_j, b_j > 0 \quad j = 1, 2, \dots, n \quad (5-68)$$

$$V_j = f(u_j) \quad j = 1, 2, \dots, n \quad (5-69)$$

在此, 神经元的输入加权和 S_j 和神经元的输入状态 u_j 用动态方程表示; 神经元的激活函数 $f(\cdot)$ 常用 $f(u) = 1/(1 + e^{-\lambda u})$ 或 $f(u) = \text{th}(\lambda u)$ 表示; 当 u_j 从 $-\infty$ 变化到 $+\infty$ 时, 神经元的输出状态 V_j 相应地为 $0 \sim +1$ 或 $-1 \sim +1$ 之间的连续值。其输入与输出的关系如图 5-4 所示。

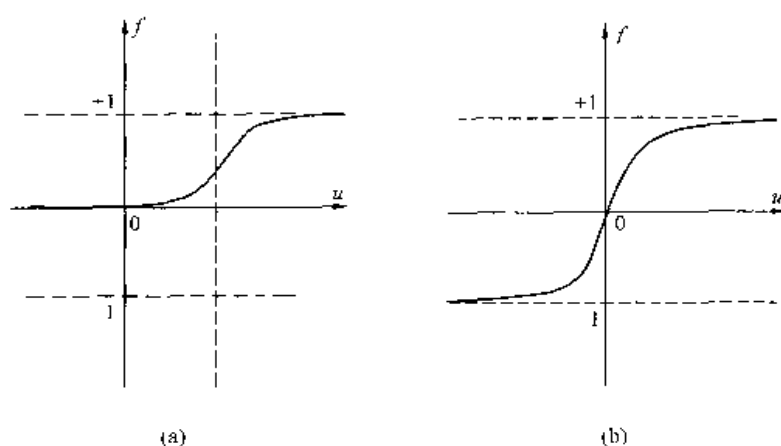


图 5-4 连续型 Hopfield 网络激活函数

一、连续型 Hopfield 网络的电路模型

Hopfield 网络是一种单层全反馈式动态神经网络，即它的每一个神经的输出都与其他神经元的输入相连接，根据网络的结构和式 (5-67) ~ 式 (5-68)，利用模拟电子线路与 Hopfield 网络之间存在的对应关系，建立连续型反馈人工神经网络的电路模型，如图 5-5 所示。

单个神经元的模拟电路，如图 5-6 所示，其输入状态 u_j 与输出 V_j 之间的关系是一个

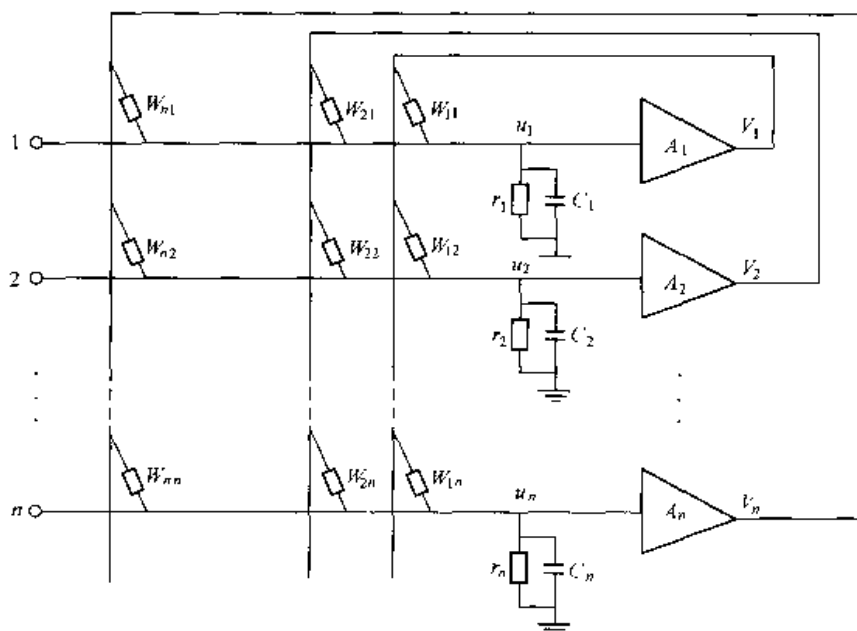


图 5-5 连续型 Hopfield 网络电路模型

A_i —是运算放大器，用来模拟网络神经元； r_j 、 C_j —是运算放大器的输入电阻与电容，其并联电路用来模拟神经元动态特性的时间常数； u_j —是第 j 个神经元的状态（即运算放大器 A_j 的总输入）； $u_{ij} = 1/R_{ij}$ —是第 i 个神经元输出 V_i 与第 j 个神经元输入之间的连接权值（导纳）； V_j —是第 j 个神经元 A_j 的输出； I_j —外加偏置电流，用来模拟神经元的阈值“ θ_j ”或外部的刺激

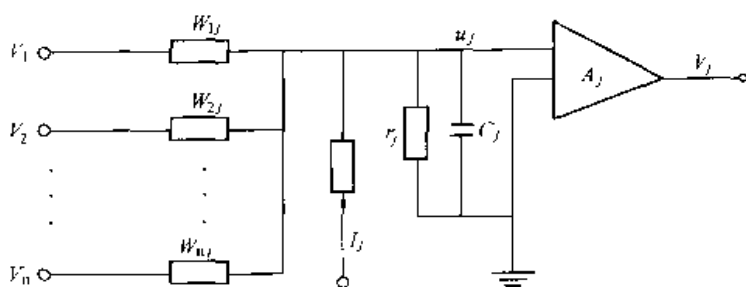


图 5-6 第 j 个神经元的模拟电路

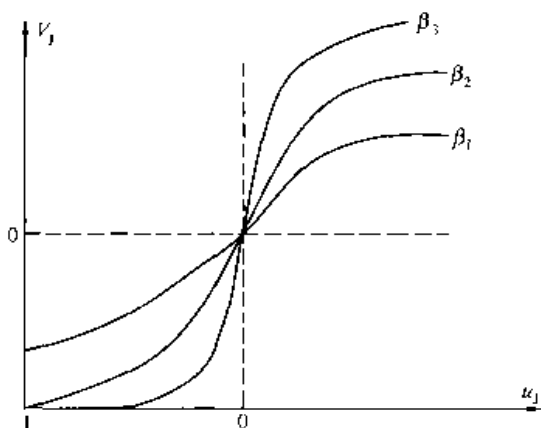


图 5-7 第 j 个神经元输入输出关系

单调上升的函数关系，记为 $V_j = f(u_j)$ ，如图 5-7 所示，其中 β 表示运算放大器的放大倍数。

在图 5-6 中，运算放大器模拟神经元的激活特性函数 f ；输入电阻决定各神经元之间的连接强度，即 $W_{ij} = 1/R_{ij}$ ；而电容 C_j 及电阻 r_j 决定神经元输入状态 u_j 与输出状态 V_j 之间的时间常数。

根据电路的克希荷夫定律可列写单个神经元的动态方程

$$C_j \frac{du_j}{dt} + \frac{u_j}{r_j} = \sum_{i=1}^n \frac{1}{R_{ij}} (V_i - u_j) + I_j$$

或

$$\begin{cases} C_j \frac{du_j}{dt} = -\frac{u_j}{r_j} + \sum_{i=1}^n W_{ij} (V_i - u_j) + I_j \\ V_j = f(u_j) \end{cases} \quad (5-70)$$

式 (5-70) 就是图 5-5 连续型 Hopfield 网络的模型方程。令 $\frac{1}{R_j} = \frac{1}{r_j} + \sum_{i=1}^n W_{ij}$ ，则式 (5-70) 可改写成下式并对 $j = 1, 2, \dots, n$ 都成立。即

$$\begin{cases} C_j \frac{du_j}{dt} = -\frac{u_j}{R_j} + \sum_{i=1}^n W_{ij} V_i + I_j \\ V_j = f(u_j) \end{cases} \quad (5-71)$$

若令 $a_j = \frac{1}{R_j}$, $b_j = C_j$ ，则式 (5-71) 可写成式 (5-67) ~ 式 (5-69) 形式。

二、连续型 Hopfield 网络能量函数和稳定性

由于式 (5-71) 是一个非线性动态系统，直接对网络进行稳定性分析往往比较困难，Hopfield 提出利用李雅普诺夫第二方法来研究网络的稳定性，为此引入了能量函数 E 。能量函数 E 是一个能够反映多维神经元状态的标量函数，可用于对单层反馈动态神经网络的稳定性进行判断。

1. Hopfield 能量函数的定义

对于式 (5-71) 描述的网络，Hopfield 定义了如下的能量函数

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} V_i(t) V_j(t) - \sum_{j=1}^n V_i(t) I_j + \sum_{j=1}^n \frac{1}{R_j} \int_0^{v_j(t)} f^{-1}(V) dV \quad (5-72)$$

式中 $f^{-1}(V)$ 表示 V 的逆函数, 即 $f^{-1}(V_j) = u_j$

写成向量矩阵的形式

$$E = -\frac{1}{2} V^T W V - I^T V + \rho^T S(V) \quad (5-73)$$

式中 $\rho = \left[\frac{1}{R_1}, \frac{1}{R_2}, \dots, \frac{1}{R_n} \right]^T$, $S(V) = \left[\int_0^{V_1} f^{-1}(V) dV, \int_0^{V_2} f^{-1}(V) dV, \dots, \int_0^{V_n} f^{-1}(V) dV \right]^T$

将式 (5-73) 与离散型 Hopfield 网络的能量函数表达式相比较可知, 连续型 Hopfield 网络的能量函数比离散型 Hopfield 网络的能量函数增加了一个积分项, 该积分项表示运算放大器输入 u_j 对输出 V_j 曲线包围的面积, 具有能量的量纲。由于放大器的放大系数较大, 积分项的值较小, 通常可忽略不计。于是式 (5-73) 可简化为

$$E = -\frac{1}{2} V^T W V - I^T V \quad (5-74)$$

式 (5-74) 称为具有能量量纲的“二次型”函数。

2. 网络的稳定性

稳定性定理: 对于连续 Hopfield 网络, 若网络对称 $W_{ij} = W_{ji}$, $W_{ii} = 0$, $i, j \in (1, 2, \dots, n)$ $C_j > 0$, 且神经元的激活函数 $f(\cdot)$ 或 $f^{-1}(\cdot)$ 为连续单调上升函数, 则有

$$\frac{dE(t)}{dt} \leq 0 \quad (5-75)$$

证明: 由式 (5-72) 可得

$$\frac{dE(t)}{dt} = \sum_{j=1}^n \frac{\partial E(t)}{\partial V_j(t)} \cdot \frac{dV_j(t)}{dt} \quad (5-76)$$

由于

$$\begin{aligned} \frac{\partial E(t)}{\partial V_j(t)} &= -\frac{1}{2} \sum_{i=1}^n W_{ij} V_i(t) - \frac{1}{2} \sum_{j=1}^n W_{ji} V_j - I_j + \frac{1}{R_j} u_j(t) \\ &= -\left[\sum_{i=1}^n W_{ij} V_i(t) - \frac{1}{R_j} u_j(t) + I_j \right] = -C_j \frac{du_j(t)}{dt} \end{aligned} \quad (5-77)$$

将式 (5-76) 代入式 (5-77), 得

$$\begin{aligned} \frac{dE(t)}{dt} &= -\sum_{j=1}^n C_j \frac{du_j(t)}{dt} \cdot \frac{dV_j(t)}{dt} = -\sum_{j=1}^n C_j \frac{du_j(t)}{dV_j(t)} \cdot \left(\frac{dV_j(t)}{dt} \right)^2 \\ &= -\sum_{j=1}^n C_j \left[\frac{dV_j(t)}{dt} \right]^2 \frac{d}{dV_j} f^{-1}(V_j) \end{aligned} \quad (5-78)$$

由于 $C_i > 0$, $f^{-1}(\cdot)$ 为连续单调上升函数, 故可得

$$\frac{dE}{dt} \leq 0$$

只有对于所有 j ($j = 1, 2, \dots, n$), $\frac{dV_j(t)}{dt} = 0$ 时, 才有 $\frac{dE(t)}{dt} = 0$, 所以, 随着网

络状态的变化,式(5-78)表达的能量函数总是朝着能量减小的方向运动,当所有神经元的状态不再变化时,能量达到极小点,该点就是网络的稳定平衡点。

Hopfield 能量函数 E 可正可负,但是有限,而且 $E(0)$ 不一定为零,所以, E 函数是一个广义的李雅普诺夫函数,只有当 E 函数满足 $E(V) > 0$, $E(0) = 0$, 及 $\frac{dE}{dt} \leq 0$ 条件时, Hopfield 能量函数才是李雅普诺夫函数,这时才可完全应用李雅普诺夫稳定性定理来分析网络的稳定性。

三、连续型 Hopfield 网络用于优化计算

对于渐进稳定的 Hopfield 网络,在其稳定点上的能量函数必然达到极小,将能量函数这一性质用于神经网络的优化计算。在优化计算过程中,首先应根据优化的目标函数和约束条件构造一个能量函数,通过对能量函数的分析去设计一个渐进稳定的 Hopfield 网络,由该网络来完成优化计算过程。由此可见,能量函数的设计是 Hopfield 神经网络优化计算的首要内容。

通常,求解最优化问题是在满足一定约束条件下求某个目标函数的极小(极大)值。即求 x 使

$$\left. \begin{array}{l} f(x) = f(x_1, x_2, \dots, x_n) = \min \\ \text{约束条件} \quad g(x) \geq 0 (\leq 0) \end{array} \right\} \quad (5-79)$$

上式中 x 为 n 维, g 为 m 维, f 为一维。

如果能把优化问题的目标函数和约束条件与 Hopfield 网络的能量函数 E 联系起来, E 的极小点也就是优化问题中满足约束条件下的目标函数的极小点,那么,就可以用该 Hopfield 网络来求解最优化问题了。

应用连续型 Hopfield 网络求解最优化问题的基本步骤可归纳如下:

(1) 根据要求的目标函数 $f(x)$ 写出能量函数的第 1 项。

(2) 根据约束条件 $g(x)$, 写出罚函数,作为能量函数的第 2 项,使在满足约束条件时,罚函数最小。

(3) 把 $\sum_{i=1}^n \frac{1}{R_j} \int_0^{V_j} f^{-1}(V) dV$ 作为能量函数的第 3 项,这一项是 Hopfield 网络电路实现上所需要的。因为在神经网络状态方程中存在一项 $\frac{u_j(t)}{R_j}$, 它是人工神经网络电路设计中产生出来的,是为了使神经网络优化计算的设计方案能在电路中得以实现。

(4) 对能量函数求导,可以得到一个关于状态变量 V_j 的微分方程。即神经网络的状态方程。

$$\frac{\partial E(t)}{\partial V_j(t)} = -C_j \frac{du_j(t)}{dt} \quad j = 1, 2, \dots, n \quad (5-80)$$

(5) 将状态方程式(5-80)与标准的 Hopfield 网络方程式(5-70)相比较,可求出网络的设计参数 W_{ij} 和 I_j , $i, j = 1, 2, \dots, n$ 。

(6) 根据已求得网络参数,参照图 5-5 的 Hopfield 网络电路,该网络电路的稳态值即

为优化计算目标函数的解。

第四节 连续型 Hopfield 网络的应用

【例 5-8】 基于连续型 Hopfield 网络的 A/D 转换器。

解：(1) 网络电路设计。

采用人工神经网络实现 A/D 转换，与传统的 A/D 转换电路相比较，具有结构简单，工作速度快等优点，下面以 4 位 A/D 转换器为例来说明网络电路的工作原理，Hopfield A/D 转换电路原理示意图如图 5-8 所示。

为了实现 4 位 A/D 转换功能，在图 5-8 中采用 4 个运算放大器模拟神经元构成 Hopfield 网络，将待转换的模拟量 V_x 以及参考电源 $-V_r$ 从图的左下端输入，各反馈线与运放输入线交叉处以小圆圈表示相应的电导值 W_{ij} ，设网络无自反馈，所以 $W_{ii} = 0$ ，各运放的输出即为 A/D 转换后的 4 位数字量。

(2) 建立网络的能量函数。

A/D 转换可作为优化问题来处理，要求变换的精度最高，或要求模拟量 V_x 转换后输出的数字量之差最小。现取最优性条件为方差最小，即取目标函数为

$$J(V) = \frac{1}{2} \left(V_x - \sum_{i=0}^3 V_i 2^i \right)^2 \quad (5-81)$$

式中， V_i ($i=0, 1, 2, 3$) 为各运放输出的数字量 $V_i \in \{0, 1\}$ ； 2^i 表示数字量 V_i 的二进制位数，显然 $J(V) \geq 0$ 。

通常，运算放大器的输出 V_i 为 0 至 1 之间变化的模拟量，为了确保电路输出的有效性，使 V_i 为最靠近 0 或 1 的数字量，于是约束条件为

$$g(V) = -\frac{1}{2} \sum_{i=0}^3 (2^i)^2 V_i (V_i - 1) \quad (5-82)$$

由上式可知：仅当输出为数字量 $V_i \in \{0, 1\}$ 时， $g(V) = 0$ ；否则当 V_i 为非数字量（即 V_i 为 0 与 1 之间的任意实数）时， $g(V) > 0$ 。式中系数的设置是为了保证网络无自反馈，即 $W_{ii} = 0$ 。

由于运放的放大系数 β 较大，式 (5-72) 的积分项数小，往往可以略去不计。于是将式 (5-81) 与式 (5-82) 相加，并略去与变量 V_i 无关的项（即常数项 $\frac{1}{2} V_x^2$ ）则可建立网络的能量函数为

$$\begin{aligned} E &= J(V) + g(V) \\ &= \frac{1}{2} \left(V_x - \sum_{i=0}^3 V_i 2^i \right)^2 - \frac{1}{2} \sum_{i=0}^3 (2^i)^2 V_i (V_i - 1) \\ &= -\frac{1}{2} \sum_{i=0}^3 \sum_{\substack{j=0 \\ i \neq j}}^3 (-2^{i+j}) V_i V_j - \sum_{i=0}^3 (-2^{2i-1} + 2^i V_x) V_i \end{aligned} \quad (5-83)$$

设： $n=3$ ， $i, j=0, 1, 2, 3$ 。将 Hopfield 能量函数的标准表达式写成如下形式

$$E = -\frac{1}{2} \sum_{i=0}^3 \sum_{\substack{j=0 \\ i \neq j}}^3 W_{ij} V_i V_j - \sum_{i=0}^3 V_i I_i \quad (5-84)$$

比较式 (5-83) 与 (5-84), 则可求得网络的结构参数为

$$W_{ij} = -2^{i+j} \quad (5-85)$$

$$I_i = -2^{2^{i-1}} + 2^i V_x \quad (5-86)$$

式 (5-85) 中的负号可利用运放的反相输出来实现。图 5-8 标注的其结构参数为

$$W_{01} = -2, W_{02} = -4, W_{03} = -8$$

$$W_{10} = -2, W_{12} = -8, W_{13} = -16$$

$$W_{20} = -4, W_{21} = -8, W_{23} = -32$$

$$W_{30} = -8, W_{31} = -16, W_{32} = -32$$

$$W_{00} = W_{11} = W_{22} = W_{33} = 0$$

式 (5-86) 中各项系数表示参考电源 V_r 和模拟量输入 V_x 输入解与各运放输入解之间的连接权值 (电导值), 图 5-8 中标注的参数为

$$I_0 = -0.5 + V_x$$

$$I_1 = -2 + 2V_x$$

$$I_2 = -8 + 4V_x$$

$$I_3 = -32 + 8V_x$$

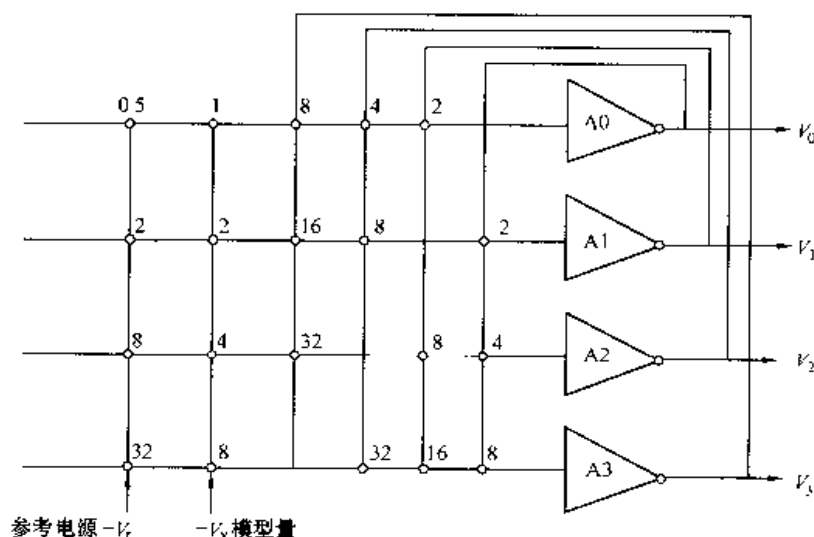


图 5-8 Hopfield A/D 转换电路原理示意图

【例 5-9】 旅行商问题 (Traveling Saleman Problem, 简称 TSP 问题)。

解: (1) 问题的提法。

设有 n 个城市 C_1, C_2, \dots, C_n ; C_i 到 C_j 的距离为 d_{ij} ($d_{ij} = d_{ji}$), 试求从某城市出发, 每一城市只经过一次并回到出发城市的一条最短路径。

(2) 用 Hopfield 网络的求解方式。

任何一个城市在一条路径上的位置 (次序) 可以用一个 n 维向量表示。以 5 个城市

为例, 如果 C1 是第 2 个访问, 则可表示成 01000, 即只有第 2 个神经元的输出为 1, 其余都是 0。为了表示所有的城市, 就需要一个 $n \times n$ 矩阵, 称为关联矩阵, 如果设 $n = 5$, 则 TSP 问题的一条有效路径的关联矩阵如表 5-1 所示。

表 5-1 关 联 矩 阵

城 市 x	次 序 i				
	1	2	3	4	5
C1	0	1	0	0	0
C2	0	0	0	1	0
C3	1	0	0	0	0
C4	0	0	0	0	1
C5	0	0	1	0	0

由表 5-1 可以看出, 关联矩阵表示了以 C3 为起点, 以 $C3 \rightarrow C1 \rightarrow C5 \rightarrow C2 \rightarrow C4$ 为顺序的一条路径, 路径的长度为

$$d = d_{C3C1} + d_{C1C5} + d_{C5C2} + d_{C2C4} + d_{C4C3}$$

为了描述一条有效的路径, 必须规定: ①关联矩阵的每一行只能有一个元素为 1, 其余均为 0, 这表示每个城市只访问一次; ②每一列也只有一个元素为 1, 其余均为 0, 这表示每一次只能访问一个城市; ③访问的城市的总数应为 n 个。

(3) 目标函数 $J(V)$ 。

若神经元的输出用 V_{xi} , V_{yi} ...表示, 下标 x 、 y 表示城市 C1、C2...; i 、 j 表示访问的次序。这样, 目标函数 $J(V)$ 可表示为

$$J(V) = \frac{A}{2} \sum_x \sum_{y \neq x} \sum_i d_{xy} V_{xi} (V_{y, i+1} + V_{y, i-1}) \quad (5-87)$$

式中 d_{xy} 表示两个不同城市 C_x 和 C_y 之间的距离;

如果 $V_{xi} = 0$, 那么这个输出对 $J(V)$ 没有贡献;

如果 $V_{xi} = 1$, 则寻找与 i 相邻次序的城市, 例如 $V_{C1, C2} = 1$, 那么在关联矩阵中可以找到与 $V_{C1, C2}$ 相邻列上的非零的 $V_{C3, C1}$ 和 $V_{C5, C3}$;

这时在 $J(V)$ 中得到 $d_{C1, C5}$ 和 $d_{C3, C1}$ 两个相加的量, 其余类推;

式 (5-87) 中 A 是大于零的常数, 由于每个距离计算两次, 故用 2 除。

(4) 约束条件 $g(x)$ (罚函数)。

总的罚函数有 3 项, 对应于有效路径的 3 个条件

$$g(V) = \frac{B}{2} \sum_x \sum_i \sum_{j \neq i} V_{xi} V_{xj} + \frac{C}{2} \sum_i \sum_x \sum_{y \neq x} V_{xi} V_{yi} + \frac{D}{2} \left(\sum_x \sum_i V_{xi} - n \right)^2 \quad B, C, D > 0 \quad (5-88)$$

式中, 等号右边的第一项为列约束条件, 保证每次只访问一个城市, 满足此条件时, 该项的值为零; 第二项为行约束条件, 保证每个城市只能访问一次; 第三项为全局约束条件, 确保 $V_{xi} = 1$ 的总数为 n , 因此, 当路径有效时, $g(V) = 0$ 。

(5) 构造总的能量函数。

将式 (5-87) 和式 (5-88) 相加, 即可得

$$E = J(V) + g(V) = \frac{A}{2} \sum_x \sum_{\substack{y \\ x \neq y}} \sum_i d_{xy} V_{xi} (V_{y,i+1} + V_{y,i-1}) + \frac{B}{2} \sum_i \sum_x \sum_{\substack{y \\ x \neq y}} V_{xi} V_{xj} \\ + \frac{C}{2} \sum_x \sum_i \sum_{\substack{j \\ i \neq j}} V_{xi} V_{xj} + \frac{D}{2} \left(\sum_x \sum_i V_{xi} - n \right)^2 \quad (5-89)$$

(6) 确定网络的结构参数。

将式 (5-89) 与 Hopfield 能量函数的规范表达式

$$E = -\frac{1}{2} \sum_i \sum_j W_{ij} V_i V_j - \sum_i V_i I_i \quad (5-90)$$

相比较, 并考虑到系数 B 只影响同一列各神经元之间的连接电导值, C 只影响同一行各神经元之间的连接电导值, D 与全部的连接电导值都有关, 而 A 只涉及路径顺序的相邻项, 于是可得网络的结构参数为

$$\begin{cases} W_{xy} = -A d_{xy} (\delta_{j,i+1} + \delta_{j,i-1}) - B \delta_{ij} (1 - \delta_{xy}) - C \delta_{xy} (1 - \delta_{ij}) - D \\ I_{xi} = D_n \end{cases} \quad (5-91)$$

$$\text{其中 } \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

根据参数值构造网络并将其运行, 则网络稳定时能量函数 E 达到极值, 这时网络的稳态输出值就对应 TSP 问题的最优解。

第五节 本章小结

(1) 反馈网络是人工神经网络中的一种, 它的特点是神经元的输出与输入相连, 形成一个时序性的动态系统。由于反馈网络具有丰富的动力学特性, 在一定的条件下, 系统在演变过程中能够逐渐收敛, 趋于稳定。因此, 反馈网络可用来模拟许多动态过程, 优化问题的求解等。

(2) 连续型 Hopfield 人工神经网络按照一定的状态轨迹, 沿着能量函数减小的方向运动, 最终能够达到网络的稳定状态 (或渐近稳定状态), 即能量 E 的极点。

(3) 能量函数的极小化过程表征了神经网络从初始状态到稳定状态的演化过程。如果建立目标函数和能量函数之间的关系, 并通过能量函数将约束化问题的解映射到神经网络的一个稳定状态上去, 那么就可以用神经网络的演化过程来实现优化计算, 这就是应用神经网络方法求约束优化或组合优化问题的基本思想。

(4) 联想记忆是离散型 Hopfield 人工神经网络的一个重要应用内容。要使反馈网络具有联想记忆的功能, 必须具备两个基本条件: ①网络必须能收敛到稳定状态, 并通过稳定状态来存储记忆信息; ②网络应具有回忆能力, 即能够从某一残缺的信息回忆起所属的完整的记忆信息。

第六章

自组织人工神经网络

神经网络系统中的一个主要特点是它具有学习功能。通过学习它可以提取一组数据中的重要特征或找出其内在规律和本质属性（如分布特征，或按某种特征聚类），不断调整本身的学习过程，通常把网络的这种功能称为自学习或自组织功能。这种无教师监督，能自动的向环境学习，并根据学习所得到的重要特征来记忆知识的网络统称为自组织神经网络。自组织神经网络有自组织特征映射网络、自适应共振理论网络、对传网络、神经认知机网络等。本章只介绍应用较多的自组织特征映射网络 and 自适应共振理论网络。

第一节 竞争学习的基本原理

本节讨论一种基本的竞争学习网络，如图 6-1 所示，该网络有 n 个输入节点，有时称 n 个输入节点为输入层，由 m 个神经元构成单个竞争层，由于竞争层中各神经元的输出即为网络的输出，因此，有时也称竞争层为输出层。在此设竞争层中的激活函数为二值型 $\{0, 1\}$ 函数。

由图 6-1 可以看出，基本的自组织竞争网络的权值有两类：一类是输入节点 i 至神经元 j 的权值 W_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$)，这些权值量通过训练可以被修正；另一类是竞争层中互相抑制的权值 Z_{kj} ($k = 1, 2, \dots, m$)，这类权值是固定不变的，且它满足一定的分布关系，如距离相近的抑制强，距离远的抑制弱。此外，它们是一种对称权值，即有 $Z_{kj} = Z_{jk}$ ，同时相同神经元之间的权值起加强的作用，即满足 $Z_{11} = Z_{22} = \dots = Z_{mm} > 0$ ，而不同神经元之间的权值相互抑制，对于 $k \neq j$ 时， $Z_{kj} < 0$ 。网络的输入向量为 $x = [x_1, x_2, \dots, x_n]^T$ ；对竞争网络的输出变量为 $y = [y_1, y_2, \dots, y_m]^T$ 。由于竞争网络中含有两种权值，因此，竞争层中各神经元激活函数的加权输入和也分为两部分：一部分来自输入节点的加权输入，另一部分来自竞争层内相互抑制的加权输入。对于第 j 个神经元有：

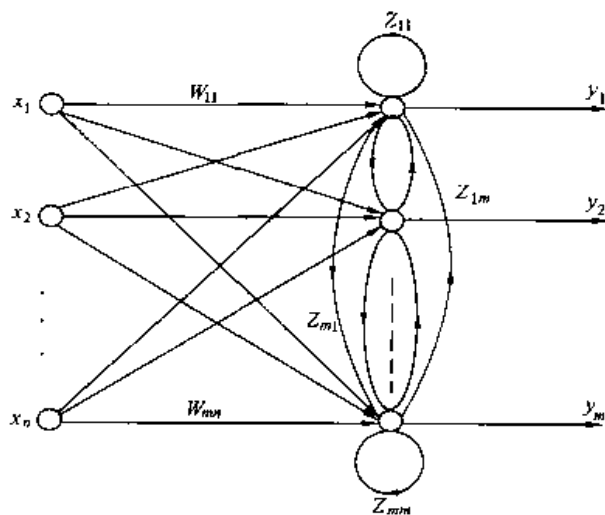


图 6-1 简单竞争网络结构图

$$(1) \text{ 来自输入节点的加权输入和为 } w_i = \sum_{j=1}^n w_{ij} x_i \quad (6-1)$$

$$(2) \text{ 来自竞争层内互相抑制的加权输入和为 } Z_j = \sum_{k \in D} z_{jk} f(y_k) \quad (6-2)$$

上式中的 D 表示竞争层中含有神经元节点的某个区域, 如果 D 表示的是整个竞争层, 则竞争后只能有一个神经元兴奋而获胜; 如果竞争层被分成若干个区域, 则竞争后每个区域可产生一个获胜者。

如果在竞争后, 第 j 个节点获胜, 则有 $k=j$, $y_k=1$, 而其他的节点输出均为零, 即 $y_k=0$, $k=1, 2, \dots, m$; $k \neq j$, 此时

$$Z_j = \sum_{k=1}^m z_{jk} f(y_k) = z_{jj} > 0 \quad (6-3)$$

如果在竞争后, 第 j 个节点失败, 而获胜的节点为 l , 则有 $k=l$, $y_l=1$, 而其余的节点输出

$$y_k=0, \quad k=1, 2, \dots, m, \quad k \neq l, \text{ 此时}$$

$$Z_j = \sum_{k=1}^m y_{jk} \cdot f(y_k) = w_{jl} < 0$$

所以, 对网络竞争层各节点的加权输入总和为

$$\begin{cases} s_l = W_l + W_{ll} & (\text{对于获胜节点 } l) \\ s_i = W_i - |W_{il}| & (\text{对于其他失败的节点 } i=1, 2, \dots, m, i \neq l) \end{cases} \quad (6-4)$$

由式 (6-4) 可知, 经过竞争后获胜节点的加权输入总和为最大, 即网络的输出为

$$y_k = \begin{cases} 1 & s_k = \max(s_i, i=1, 2, \dots, m) \\ 0 & \text{其他} \end{cases}$$

由于在网络学习的过程中, 只修正输入层中各输入节点与竞争中各神经元的连接权值 w_{ij} , 而竞争层内的权值 z_{jk} 是固定不变的, z_{jk} 对改善的结果只起到了加强或削弱作用, 即对获胜节点增加一个正值, 使其更能获胜, 若对失败节点增加一个负值, 使其更不能获胜。然而, 对改变节点竞争结果起决定性作用的还是输入层的加权和 W_j 。

$$\text{即} \quad W_{\text{胜}} = \max\left(\sum_{i=1}^n W_{ij} x_i\right) \quad (6-5)$$

第二节 几种基本学习算法

一、简单竞争学习 (Simple Competitive Learning)

简单竞争学习只允许单个胜者神经元出现, 学习只发生在该胜者一个神经元上, 其算法步骤如下:

(1) 对网络各权值赋以小的随机数作为学习初值。

(2) 输入学习样本向量 $x^k = [x_1^k, x_2^k, \dots, x_n^k]^T$, 计算它与竞争层中所有神经元的距离

$$d_j = \sum_{i=1}^n [x_i^k(t) - w_{ij}(t)]^2 \quad (6-6)$$

(3) 选取 d_j 最小的神经元作为胜者神经元, 其权向量系数 w_{ij} 按下式修正

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) \cdot [x_i(t) - w_{ij}(t)] \quad (6-7)$$

式中 $\alpha(t)$ ——学习参数, $0 < \alpha(t) < 1.0$;

t ——第 t 次学习。

(4) 输入一个新的学习样本, 转至步骤 (2), 直至网络收敛。

二、Kohonen 算法

Kohonen 算法中没有侧连接, 但却利用一种计算上的技巧达到墨西哥帽侧连接所实现的效果, 其权值的更新规则仍涉及输出阵列中的近邻关系。Kohonen 算法定义了胜者神经元的邻域, 将邻域中的神经元权向量向输入样本 X 靠拢, 其算法步骤如下:

(1) 对网络各权值赋予小的随机数作为初始值。

(2) 输入一个学习样本向量 $x = [x_1, x_2, \dots, x_n]^T$, 计算它与竞争层中所有神经元的距离

$$d_j = \sum_{i=1}^n [x_i^k(t) - w_{ij}(t)]^2 \quad (6-8)$$

(3) 选取 d_j 最小的神经元作为胜者神经元, 其权向量系数 w_{ij} 按下式修正

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) A(j, j^*) [x_i(t) - w_{ij}(t)]^2 \quad (6-9)$$

式中近邻函数 (neighborhood function) $A(j, j^*)$ 的定义是: 对于 $j = j^*$, j^* 为获胜单元, 函数 $A(j, j^*)$ 取值为 1, 随着输出阵列中单元 j 与 j^* 之间的距离 $|r_j - r_{j^*}|$ 的增加, 函数值减小。因此, 获胜单元以及其邻近的单元都明显权值, r_j, r_{j^*} 是相应获胜单元的位置。而那些 $A(j, j^*)$ 很小之处的较远的单元几乎不受什么影响。 $A(j, j^*)$ 的典型选择是

$$A(j, j^*) = \exp[-|r_j - r_{j^*}|^2 / 2\sigma(t)^2] \quad (6-10)$$

式中 σ 是逐渐减小的宽度参数。 $\alpha(t)$ 与 $\sigma(t)$ 之间的时间相关性可以取几种形式, 例如, $1/t$ 和 $a - bt$, 通常选择 $\alpha \propto t^{-a}$, $0 < a \leq 1$ 。

(4) 输入一新样本, 转至步骤 (2), 直至网络收敛。

三、频率修正竞争学习 (Frequency Sensitive Competitive Learning) 算法

频率修正竞争学习记住胜者神经元的激活次数, 这一胜者频率信息用来解决简单竞争学习中某些神经元始终处于休眠状态问题, 使所有神经元学习的次数接近相等, 其算法步骤如下:

(1) 对网络各权值赋以小的随机数作为初始值。

(2) 输入学习样本向量 $x = [x_1, x_2, \dots, x_n]^T$, 计算它与竞争层中所有神经元的距离。

$$d_j = \left| \sum_{i=1}^n [x_i(t) - w_{ij}(t)]^2 \right| x_{c_j}(t) \quad (6-11)$$

式中 $c_j(t)$ ——神经元 j 到 t 时的累计激活次数。

(3) 选取 d_j 最小的神经元作为胜者神经元, 其权向量系数 w_{ij} 按下式修正

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) [x_i(t) - w_{ij}(t)] \quad (6-12)$$

式中 $\alpha(t)$ ——学习参数, $0 < \alpha(t) < 1.0$;

t ——第 t 次学习。

(4) 输入一新的学习样本, 转至步骤 (2), 直至网络收敛。

四、Desieno 学习算法

该算法也属于频率修正竞争学习方法, 其算法步骤如下:

(1) 对网络各权值赋以小的随机数作为初始值。

(2) 输入一学习样本向量 $x = [x_1, x_2, \dots, x_n]^T$, 计算它与竞争层中所有神经元的距离

$$d_j = \left| \sum_{i=1}^n \|x_i(t) - w_{ij}(t)\| - b_j(t) \right| \quad (6-13)$$

式中 b_j ——与频率 f_j 有关的修正系数。

$$b_j(t) = c \left| \frac{1}{m} - f_j(t) \right| \quad (6-14)$$

式中 m ——竞争层的个数;

c ——一个衡量距离度和激活频率之间重要程度常数。

$$f_j(t+1) = f_j(t) + B[y_j - f_j(t)] \quad (6-15)$$

式中, $f_j(t)$ 为 t 的神经元 j 的频率, 胜者神经元 $y_j = 1$, 其他神经元 $y_j = 0$, B 为一常数, 即 $0 < B \leq 1$ 。

(3) 选取 d_j 最小的神经元作为胜者神经元, 其权向量系数 w_{ij} 按下式修正。

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)[x_i(t) - w_{ij}(t)] \quad (6-16)$$

式中 $\alpha(t)$ ——学习参数, $0 < \alpha(t) < 1.0$;

t ——第 t 次学习。

(4) 输入一个新的学习样本, 转达至步骤②, 直至网络收敛。

五、模糊竞争学习 (Fuzzy Competitive Learning) 算法

该算法首先定义一个模糊隶属度函数 U_j 来度量样本 x 对各神经元 j 的隶属程度, 根据隶属度函数 U_j 的大小进行竞争层神经元权向量 w_{ij} 的修正。其算法步骤如下:

(1) 对网络各权值赋予小的随机数作为初始值。

(2) 输入一个学习样本向量 $x = [x_1, x_2, \dots, x_n]^T$, 计算它与竞争层中所有神经元的距离

$$d_j = \sum_{i=1}^n [x_i(t) - w_{ij}(t)]^2 \quad (6-17)$$

(3) 选取 d_j 最小的神经元作为胜者神经元, 其权向量系数 w_{ij} 按下式修正

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) U_j^k [x_i(t) - w_{ij}(t)] \quad (6-18)$$

$$U_j^k(t) = \left| 1 - \frac{d_j(t)}{\max d_j(t)} \right| \quad (6-19)$$

式中 k ——模糊化指数。

上述的简单竞争学习、频率修正竞争学习及 Desieno 学习方法仅允许网络输出出现单个胜者, 其更一般的形式应为多个胜者出现, 学习发生在胜者集合域的各个神经元上,

Kohonen 学习和模糊竞争学习算法就属于这种协同式学习算法。

第三节 自组织特征映射网络

自组织特征映射 (Self-Organizing Feature Map, 简称 SOFM) 网络是由芬兰学者 T. Kohonen 在 1984 年提出的。它是一种无监督竞争学习, 它在模式识别、联想存储、样本分类、优化计算、机器人控制等领域中得到广泛应用。

一、SOFM 网络工作原理

SOFM 网络由输入和一个输出层组成, 如图 6-2 (a) 所示, 每个输入通过权值与输出

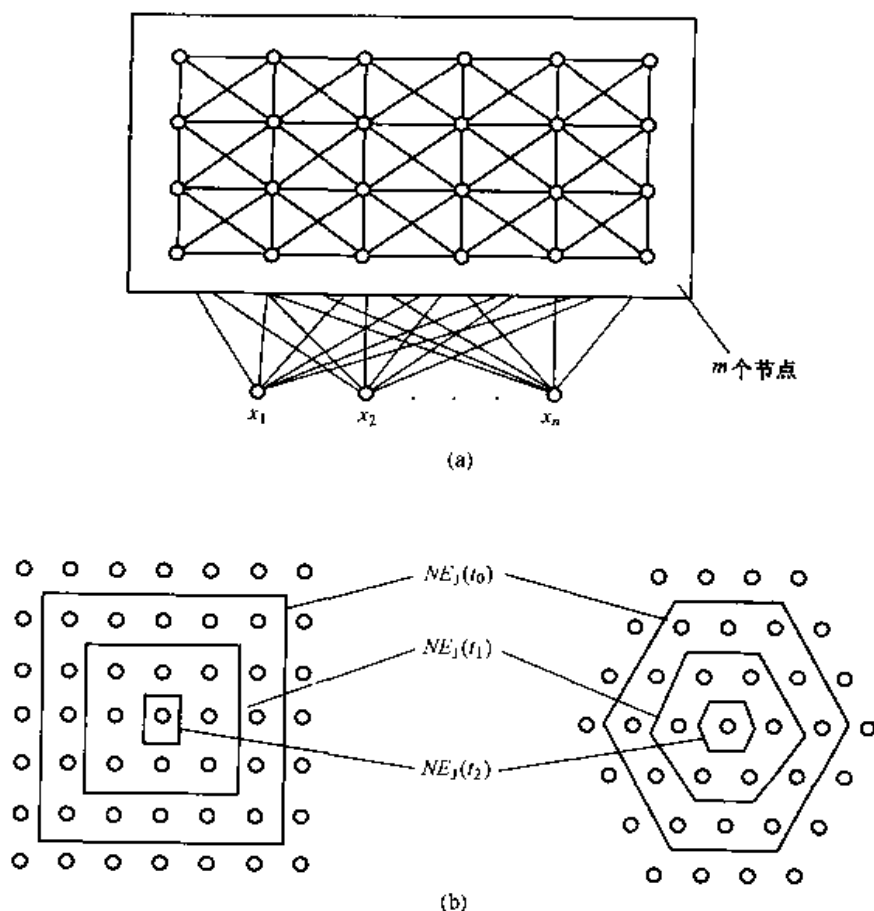


图 6-2 二维 SOFM 网络结构

(a) 二维阵列示意图; (b) 正方形或六角形邻域示意图

层的每一个神经元 (节点) 相联。与基本竞争网络不同之处是其竞争层可以一维或二维网络矩阵方式组成, 且权修正的策略也不同, 可实现对输入信号的非线性降维映射, 映射中保持拓扑不变性, 即把拓扑意义下相似的输入映射到相近的输出节点上。输出层中的每个神经元都是输入样本的代表。

在图 6-2 (a) 中, 其为二维网络结构, 网络上有输出节点 m 个, 按二维形式排成一个节点矩阵, 输入节点处于下方, 有 n 个节点, 所有节点到全部输出节点之间都有权值

连接, 而且在二维平面上的输出节点相互间也是局部连接的。

输出层是通过竞争进行学习的, 竞争过程如下: 设在 t 时刻输入一个样本, 经过竞争, 获胜的那个节点 j 以及其周围的邻域的节点 $NE_j(t)$ 在不同程度上得到兴奋, 而在 $NE_j(t)$ 以外的节点都被抑制, 这 $NE_j(t)$ 可以是正方形或六角形等形状, 如图 6-2 (b) 所示。 $NE_j(t)$ 包含以 j 为中心, 距离不超过某一半径的所有节点。 $NE_j(t)$ 是 t 的函数, 即随着训练过程的进行, $NE_j(t)$ 的半径逐渐减小, 最后剩下一个节点 (或一组节点), 它反映了一类样本的属性和特征。

设输入有 p 类模式 x^1, x^2, \dots, x^p , 而每一个样本 $x^k = [x_1^k, x_2^k, \dots, x_n^k]^T$ 为 n 维输入, x_i^k 与输出的第 j 个节点的连接权为 w_{ij} , 其输出为 y_j , 输入与输出满足如下方程

$$\frac{dy_j}{dt} = \sum_{i=1}^n w_{ij} x_i^k - r(y_j) \quad (6-20)$$

式中, $r(y_j)$ 为 y_j 有关的非线性函数, 其作用是使 y_j 变化速度变慢, 达到稳态时, $\frac{dy_j}{dt} = 0$, 这时

$$y_j = f\left(\sum_{i=1}^n w_{ij} x_i^k\right), f(\cdot) = r^{-1}(\cdot) \quad (6-21)$$

式中, $f(\cdot)$ 是一个单调上升的非线性函数, 竞争是在输出节点之间进行的, 当 y_j 成为最大的节点时, 它便是获胜的节点。

w_{ij} 的学习满足 Hebb 规则, 其变化率正比于输入与输出状态的乘积, 即

$$\frac{dw_{ij}}{dt} = \alpha y_j x_i^k - \beta(y_j) w_{ij} \quad (6-22)$$

式中, α 是常数, 上式右边第一项满足 Hebb 规则, 后一项中的 $\beta(y_j)$ 是一个遗忘因子, 在没有输入 x_i^k 时, w_{ij} 会随时间衰减。

考虑一种特殊情况: $y_j \in (0, 1)$, 且满足

$$\beta(y_j) = \begin{cases} \alpha & y_j = 1 \\ 0 & y_j = 0 \end{cases} \quad (6-23)$$

那么由式 (6-22) 可得

$$\frac{dw_{ij}}{dt} = \begin{cases} \alpha(x_i^k - w_{ij}) & y_j = 1 \\ 0 & y_j = 0 \end{cases} \quad (6-24)$$

由上式可知, 经过学习以后, 权值越来越接近输入的 x_i^k 。

在此, 网络权值的调整是在获胜节点的领域 $NE_j(t)$ 内进行的, 越靠近获胜节点的, 其权值调整系数 α 越大; 离获胜节点越远的, α 越小。

二、SOFM 网络的算法步骤

Kohonen 的自组织特征映射算法的基本步骤可简述如下:

(1) 权值初始化, 对所有 n 个输入到 m 个输出节点的连接权值赋予 $(0, 1)$ 区间上一个小随机值。

(2) 给网络输入样本模式 x^k 。

(3) 计算所有节点的距离, 即计算输入 x^k 至每个输出节点 j 之间的距离

$$d_j = \sum_{i=1}^n [x_i^k(t) - w_{ij}(t)]^2 \quad j = 1, 2, \dots, m \quad (6-25)$$

式中 $x_i^k(t)$ —— t 时节点 i 的输入。

(4) 选择具有最小距离的输出节点 j^* 为获胜节点

$$d_{j^*} = \min_{j=1}^m |d_j| \quad (6-26)$$

(5) 对节点 j^* 及其邻域内的节点的连接权值进行更新, 即

$$w_{ij}(t+1) = w_{ij}(t) + \alpha [x_i(t) - w_{ij}(t)] \quad (6-27)$$

$j \in NE_{j^*}(t)$; $i = 1, 2, \dots, n$; $0 < \alpha(t) < 1$, 并随时间减小, 达到稳态时, 则

$$w_{ij}(t+1) = w_{ij}(t) \quad (6-28)$$

$j \in NE_{j^*}(t)$, 设领域 $NE_{j^*}(t)$ 开始时为 $1/2$ 输出平面, 当中心点为 y_{j^*} 时, 可随时间对 y_{j^*} 的领域逐渐压缩。

(6) 返回步骤 (2), 重复进行, 直至网络收敛。

第四节 自适应共振理论网络

自适应共振理论 (Adaptive Resonance Theory, 简称为 ART) 网络是由美国 Boston 大学的 S. Grossberg 和 A. Carpenter 提出的。这种网络具有两种形式: ART1 用于处理二进制式双极型数据, ART2 用于处理连续值输入信号。ART 网络主要应用于模式识别、图像处理、语音处理、过程控制、故障诊断等领域。ART 的主要优点有:

(1) 可完成实时学习, 且可适应非平稳的环境。

(2) 对已学过的对象具有稳定的快速识别能力, 同时又能迅速适应未学习的新对象 (前部分通过注意子系统完成, 后部分通过取向子系统完成)。

(3) 具有自归一能力, 根据某些特征在全体中所占的比例, 有时作为关键特征, 有时又被当作噪声处理。

(4) 不需要事先已知结果, 可在无监督式下训练学习。

(5) 容量不受输入通道数的限制, 存储对象也不要求是正交的。

一、ART1 网络结构

ART1 网络一般由两层神经元和一些控制部分组成, 如图 6-3 所示。

1. 比较层 C (匹配子系统)

比较层 C 有 n 个节点, 每个节点接受输入信号 x_i , 由上一层返回的信号和增益控制器 G_1 的控制信号。节点的输出 s 根据 2/3 规则, 即“多数表决”产生, 就是说, 3 个信号中有 2 个为 1, 则输出为 1, 如图

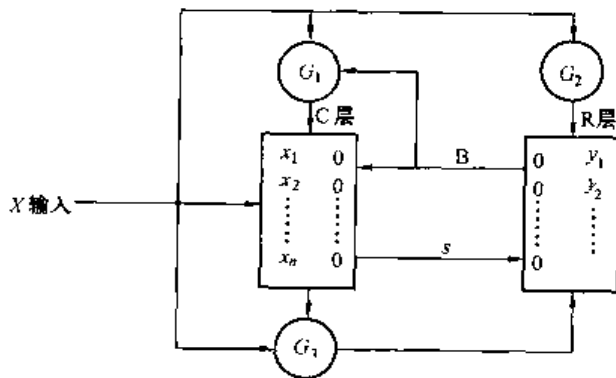


图 6-3 ART1 模型结构

6-4 所示。

2. 竞争层 R (竞争子网络)

网络的竞争层 (即输出层) 有 m 个节点, 代表 m 类输出模式。R 层节点接受来自 C 层的信号, 同时受控制器 G_3 的输出控制。C 层的信号送至 R 层后, 经过竞争, 在 R 层获胜的节点, 代表了输入向量的模式类别。

3. 控制部分

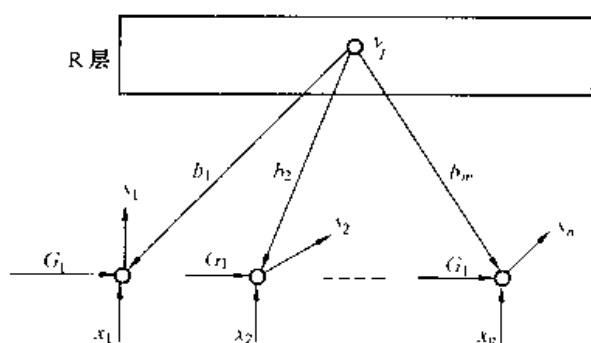


图 6-4 2/3 规则产生输出示意图

ART 网络的突出特点是在竞争学习模型中嵌入一个自动调节机构, 使其自治的学习和识别响应于任何输入模式序列而稳定地进行。

图 6-4 中的 G_1 为增益控制器, 它接受输入信号 x 和由 R 层返回的信号 b , 当输入信号 x_i 不全为零, R 层输出 b_j 全为零时, $G_1 = 1$, 给出加强信号; 在其他情况, $G_1 = 0$, 给出抑制信号。

G_2 为 R 层起动控制器。当 x 全为零时, $G_2 = 0$; x 不全为零时, $G_2 = 1$, 使 R 层工作。 G_3 为匹配控制器。在输入信号 x 和由 R 层返回的信号 b 匹配较好时, G_3 没有输出; 当不匹配时, G_3 输出一个复位信号, 抑制 R 层中获胜的那个节点, 这表示此次选择的模式不满足要求, 重新进行竞争。

二、ART1 网络运行过程

从网络输入模式数据开始, 到最后将模式存储在相应的模式类 (即相应的权值) 中, 网络的运行过程如下:

(1) 初始化阶段。设置由 C 层至 R 层的初始连接权值 w_{ij} 和由 R 层至 C 层的初始连接权值 \bar{w}_{ji} 。在输入样本之前, 设置 $x = 0$, $G_2 = 0$, R 层输出全为零, 使每个节点都有平等的竞争机会。

(2) 识别阶段。当输入样本数据 x , $G_2 = 1$, $G_1 = 1$, 由 2/3 规则, C 层第 i 个节点的输出为 $s_i(x_i) = x_i, i = 1, 2, \dots, n$ 。在 R 层第 j 个节点的输入为

$$T_j = \sum_{i=1}^n s_i(x_i) w_{ij} \quad j = 1, 2, \dots, m \quad (6-29)$$

这时由于没有 R 层的返回信号 b , x 与 b 不匹配, 则 G_3 输出一个复位信号。其网络识别过程如图 6-5 (a) 所示。

(3) 竞争阶段。在 R 层中的节点经过竞争算法, 得到一个输出最大的节点 j^* , 其输出 $y_{j^*} = 1$, 而所有其他的节点输出均为 $y_j = 0, j = 1, 2, \dots, m, j \neq j^*$ 。

(4) 比较阶段。R 层的输出信息返回 C 层, 这时 $G_1 = 0$, 输入 C 层的信号与由 x 和返回信号 b 组成, 根据 2/3 规则改变 C 层输出为 s_i , $i = 1, 2, \dots, n$ 。C 层第 i 个节点的返回信号为

$$b_i = \sum_{j=1}^m \bar{w}_{ji} y_j = \bar{w}_{ji^*} \quad (6-30)$$

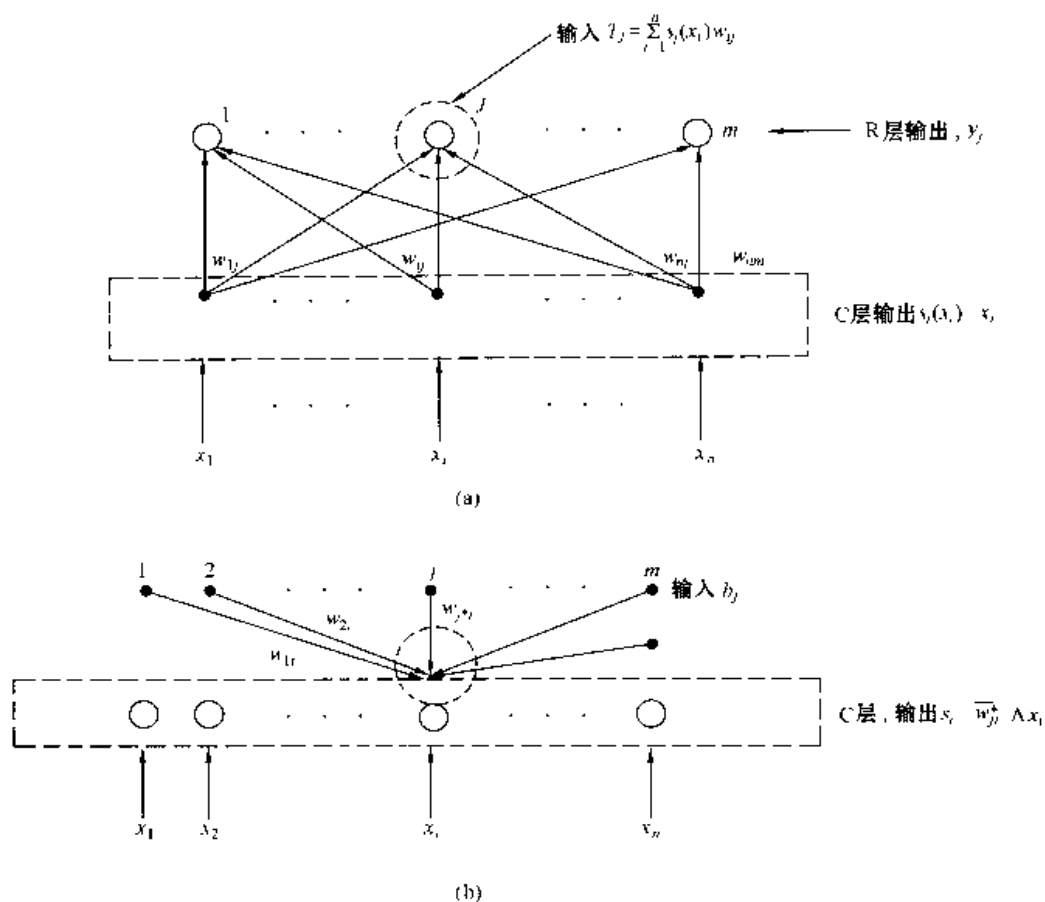


图 6-5 ART1 信号流向示意图
(a) 识别阶段; (b) 比较阶段

故 C 层的第 i 个节点的输出为

$$s'_i = b_i \wedge x_i = \bar{w}_{ji}^* \wedge x_i \quad (6-31)$$

式中 \wedge 表示逻辑与。C 层第 i 个节点新的状态 s' 反映了输入量 x 与获胜节点 j^* 匹配程度。如果 s' 给出了匹配较好的信息, 则表示竞争结果正确, 则 G_3 没有输出; 否则, G_3 输出一个复位信号, 使获胜的节点无效, 并使其在本次输入模式匹配中不能再获胜, 然后进行再竞争阶段。

(5) 再竞争阶段。

这时 R 层的输出全为零, 由于 G_3 输出了一个复位信号。此时 $G_1 = 1$, 在 C 层输出端又得到 $s_i = x_i, i = 1, 2, \dots, n$, 于是网络又进入竞争阶段, 竞争后, 又得到新的获胜节点, 再进入比较阶段。如此重复进行, 直到网络充分匹配为正。

三、ART1 网络学习算法

ART1 网络学习算法步骤归纳如下:

(1) 初始化。

设置 $w_{ij}(0) = \frac{1}{1+n}, \bar{w}_{ij}(0) = 1 \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m$

设置 ρ ($0 \leq \rho \leq 1$) 为警戒线, 为匹配程度的阈值。

(2) 输入样本模式数据。

$$x^k = (x_1, x_2, \dots, x_n)^T, x \in [0, 1]^n; k = 1, 2, \dots, P$$

(3) 计算 R 层节点输出。

$$y_j = \sum_{i=1}^n w_{ij} s_i(x_i) = \sum_{i=1}^n w_{ij} x_i \quad j = 1, 2, \dots, m \quad (6-32)$$

式中 y_j ——网络节点输出；

x_i ——输入节点 i 的输入，取值为 0 或 1。

(4) 选择获胜节点 j^* 。

$$y_{j^*} = \max_j \{y_j\} \quad (6-33)$$

这可通过输出节点的扩展抑制权达到。

(5) 比较与匹配检测。

设向量中不为零的个数用 $|x|$ 来表示，定义

$$|x| = \sum_{i=1}^n x_i \quad (6-34)$$

那么

$$|Tx| = \sum_{i=1}^n \overline{w_{j^*}}_i x_i \quad (6-35)$$

如果

$$\frac{|Tx|}{|x|} > \rho \quad (6-36)$$

则接受 j^* 为获胜节点，转步骤 (7)，否则转步骤 (6)。

(6) R 层节点的输出暂时设定为零，使之不再参加竞争，转至步骤 (3)。

(7) 调整网络权值

$$\begin{cases} \overline{w_{j^*}}_i(t+1) = \overline{w_{j^*}}_i(t) x_i \\ \overline{w_{ij}}(t+1) = \frac{\overline{w_{j^*}}_i(t) x_i}{\beta + \sum \overline{w_{j^*}}_i(t) x_i} \end{cases} \quad (6-37)$$

一般取 $\beta = 0.5$ 。

(8) 取消步骤 (6) 中对 R 层最佳节点的抑制，返至步骤 (2)。

上述的算法是一种快速学习算法，并且是边学习边运行，输出节点中每次最多只有一个为 1，每个获胜节点的输出可以看作是一类相近样本的代表模式。通过以上的运算过程，可以将输入样本 x^1, x^2, \dots, x^P 都记忆在权值 $\overline{w_{ij}}, \overline{w_{j^*}}_i$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$) 中。选择 ρ 的大小，可以调节输入样本分类能力。 ρ 越接近于 1，则输入一个新样本与已学习记忆过的样本稍有不同，就属于新的一类，即 ρ 小模式的类别少， ρ 大则模式的类别多。

第五节 自组织网络应用举例

【例 6-1】 SOFM 网络应用于炼钢炉炉温模式识别。

某钢铁厂用自组织映射网络识别高炉炉壁温度分布。网络用 65 个输入，来自炉壁沿上下方向和圆周方向设置的 65 个测温点的温度传感器；网络输出用 5×5 的节点阵列，代表 25 种炉壁温度分布模式，如图 6-6 所示。训练网络时，用 200 天的各测温点平均温度数据，用 SOFM 网络的算法来训练权值。经过学习后的网络的权值代表了 25 类炉壁温度的分布模式。使用训练完成以后的网络，输入一个月平均炉壁温度数据，由网络输出节点的输出可以判别是哪一种分布模式。

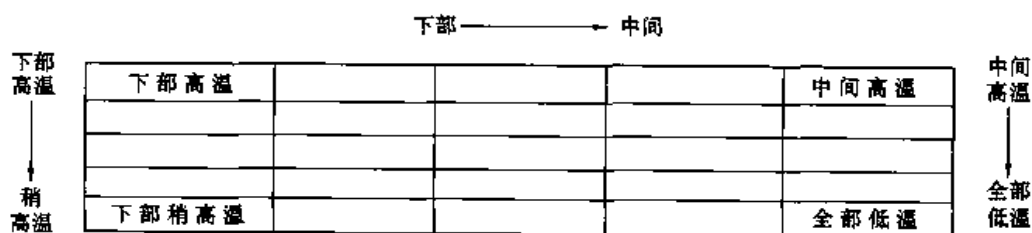


图 6-6 炉壁温度分布模式

Kohonen 自组织映射网络能将输入映射到低维输出空间，减少了维数，在对随机信号的分类和处理上有显著的优点。但网络要求的输出神经元数目多，连接权值也多，因此，要比 BP 网络规模大。

【例 6-2】 ART1 网络应用实例。

ART1 为无指导学习网络，可以实时进行学习，对于已学习过的模式，有快速的识别能力，目前主要用于图像处理，语音处理，模式识别和控制系统中。下面介绍 ART1 网络用于成组技术中的单元划分问题，单元划分是组织生产的重要方法。

设有 n 种产品和 m 台机器，需要将 m 台机器划分成 k 个单元， q_{ik} 为机器 i 上加工 k 产品的数量。在单元划分时，要找到产品与产品，机器与机器之间的相似程度，并使各单元之间负荷平衡。在此，定义 $A = [a_{ik}]_{m \times n}$ 为机器—产品连接矩阵，其中 $a_{ik} = 1$ 表示 i 需加工 k 产品，否则 $a_{ik} = 0$ 。这样， $A_i = (a_{i1}, a_{i2}, \dots, a_{in})^T$ 为机器 i 的加工模式，计算步骤如下：

- (1) 初始化。设置 $w_{kj}(0) = \frac{1}{n}$, $\bar{w}_{jk} = 1$ ，其中 n 为预定的单元数，并设置 ρ 值。
- (2) 输入一个加工模式 $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$ 。
- (3) 计算 R 层节点输出

$$y_j = \sum_{k=1}^n w_{kj} a_{ik} q_{ik} \quad j = 1, 2, \dots, m \quad (6-38)$$

- (4) 选择获胜节点 $y_j^* = \max\{y_j\}$ (6-39)

- (5) 验证 A_i 是否属于第 j 类，如

$$\sum_{k=1}^n \overline{w_{j^*k}} a_{jk} q_{ik} / \sum_{k=1}^n a_{ik} q_{ik} > \rho \quad (6-40)$$

则接受 j^* 为获胜节点, 转 (7); 否则, 转 (6)。

(6) 将 R 层节点 j^* 的输出暂时设定为零。

(7) 修正权值

$$\begin{cases} \overline{w_{j^*k}}(t+1) = \overline{w_{j^*k}}(t) a_{ik} \\ w_{kj^*}(t+1) = \frac{\overline{w_{j^*k}}(t) a_{ik}}{0.5 + \overline{w_{j^*k}}(t) a_{ik}} \end{cases} \quad (6-41)$$

(8) 恢复 (6) 中抑制的 R 层的节点, 返回 (2)。

分类计算过程中, 通过提高 ρ 值先将 m 台机器分成 k_1 ($k_1 > n$) 类, 从 k_1 类中选出负荷最大的 k_2 类作为基本类, 由获得的权值 w_{kj} , $\overline{w_{jk}}$ 进行新节点与基本类的匹配, 匹配中计算输入新节点在各类中的负荷, 使单元之间的负荷向减小方向聚类, 同时考虑各单元负荷是否达到或超过平均负荷 $\sum \sum q_{ij} / n$ 。若达到或超过, 则提高 ρ 。

【例 6-3】 自组织映射网络用于故障诊断。

如图 6-7 所示模拟电路, 其元件参数如下: $R_1 = 1\text{M}\Omega$, $R_2 = 10\text{M}\Omega$, $R_3 = 2\text{M}\Omega$, $R_4 = 1\text{M}\Omega$, $C_1 = 0.01\text{F}$, $C_2 = 1000\text{pF}$, $C_3 = 1000\text{pF}$, 利用 pspice 得到模拟电路的输出电压, 并采用蒙特卡罗分析法得到最坏情况下的容差。经过小波变换提取各种故障的特征量, 并进行归一化处理, 形成具有容差的故障集特征向量。

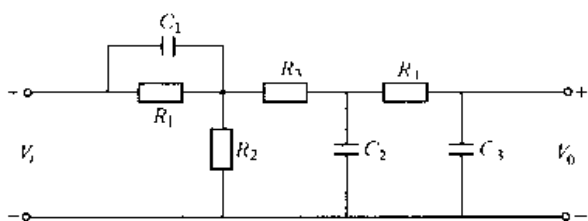


图 6-7 无源线性网络

为了解决故障定位中的容差问题, 把此故障集中各条记录作为一个输入模式, 分别映射到一维处理单元上, 经过网络训练后, 每一个故障特征量以一个神经元及其相邻的一个或两个神经元反映其结果, 使每个故障不仅仅依赖于故障集中的标称值, 而且具有上下容差, 形成故障字典, 如, 故障集中 (0.2712 0.3446

0.6588 0.5573 0.2449 0.0561) 代表故障 C_3 正偏 50% (C_3^+), 而经过 SOFM 网络后, 分别用 (0.2542 0.3398 0.6661 0.5560 0.2404 0.0546), (0.2622 0.3402 0.6602 0.5612 0.2399 0.0547), (0.2653 0.3343 0.6483 0.5816 0.2283 0.0466) 三条数据代表此故障。其中 (0.2622 0.3402 0.6602 0.5612 0.2399 0.0547) 为网络主神经元, 其余两项为其邻域神经元, 从网络的结构看与主神经元及其相邻的神经元相连的权向量代表此故障类, 主神经元代表最主要的特征。

【例 6-4】 SOFM 网络权值的实现。

根据 kohonen 学习算法, 当对某一输入模式 x_i , 若某输出节点 y_{j^*} 竞争获胜, 则 $d_{j^*} = \min(d_j)$, $j \in (1, 2, \dots, m)$ 。理想状况下, 当 $x_i = w_{ij^*}$, $i = 1, 2, \dots, n$, 则距离 $d_{j^*} = 0$ 为最小。图 6-8 即是对 x_i 与 w_{ij} 进行比较电路。当 $x_i = w_{ij}$ 时, 输出最大。输入特征 x_i 和内部权值特征 w_{ij} 分别用电压 V_1 , V_2 来表示。

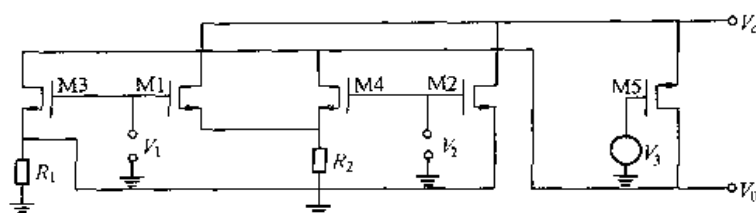


图 6-8 比较电路

图 6-8 中, M_3 和 R_1 , M_1 和 R_2 组成第一级输入, 为了减小其输出电阻, 把 M_1 和 M_3 的宽长比 (W/L) 取大一点。如选 $W/L = 180\mu/10\mu$, $R_1 = R_2 = 100k$, 可以获得很好的性能。 M_2 和 M_4 分别组成反相器结构, 其宽长比 $W/L = 10\mu/10\mu$, M_5 构成电流源, 实现一个有源电阻。由于电路对称, 对输入 V_1 , V_2 可分别表示权值和输入特征, 输出 V_0 的大小表示这两个特征的相似程度。当两个特征相同时, 即 $V_1 = V_2$, M_2 、 M_4 截止, 输出电压达到最大值, 接近电源电压。在电路工作过程中, M_2 、 M_4 始终只有一个导通, 从而能减少电路损耗。

图 6-9 是一个本质特征提取电路即学习电路，它是通过对一系列输入模式进行学习抽取其本质特征形成内部映射。为了简便描述，在此只对一个特征进行学习。设电压 V_1 代表输入特征 x_i ， V_2 表示内部映射特征 w_{ij} 。

图 6-9 中, M1、M2、M3 为 NMOS 开关, 当栅极电平为高电平时, 开关导通。M1 和 M2 分别由不相重叠的两相时钟控制, 与电容 C_1 组成学习采样电路。M3 为权值调整开关, 由相应的竞争输出节点控制。若某节点获胜, 则对应开关 M3 闭合, 以进行权值调整。

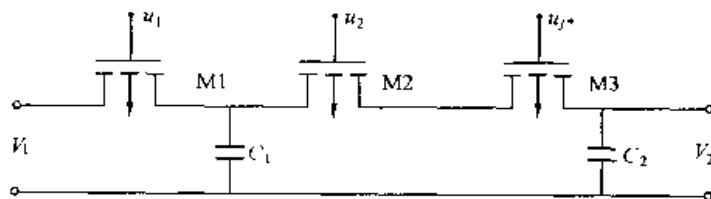


图 6-9 学习电路

若某一时刻 $t = t_0$, 电容 C_2 上存储电压 $V_{C2} = V_{w0}$, 输入特征 $V_1 = V_{x1}$, 当 u_1 为高电压时, M1 导通, 对电容 C_1 充电至 $V_{C1} = V_{x1}$, 当 u_2 为高电平以及对应输出节点获胜即 M3 导通, 则 C_1 、 C_2 进行电荷的重新分配即权值调整。经过一个学习周期, C_2 上电压即权值变为 V_{C2}^1 。

$$V_{G2}^I = k/(k+1) V_{x0} + V_{x1}/(k+1)$$

令 $t = t + 1$, 学习第二个模式中的同一特征 V_{x2} , 权值变为 V_{c2}^2

$$V_{C2}^2 = [k/(k+1)]^2 V_{w0} + V_{x2}/(k+1) + k/(k+1)^2 V_{x1}$$

若对同一模式学习 n 次之后, 权值变为 V_{ij}^n

$$V_{G2}^n = [k/(k+1)]^n V_{w0} + 1/(k+1) V_{x1} + k/(k+1)^2 V_{x1} + \dots + k^{n-1}/(k+1)^n V_{x1}$$

可得, 当 $n \rightarrow \infty$ 时有

$$V_{\Omega}^n = V_{\alpha 1}$$

由此可以看出,该网络通过学习具备了事物的某一特征或提取了事物的本质特征,其中 $k = C_2/C_1$ 控制学习速率。

【例 6-5】 三维 kohonen 网络用于黄土高原生态破坏程度预测。

三维 kohonen 网络如图 6-10 所示,该网络可将输出节点分布在空间,以提高网络的分类效果。

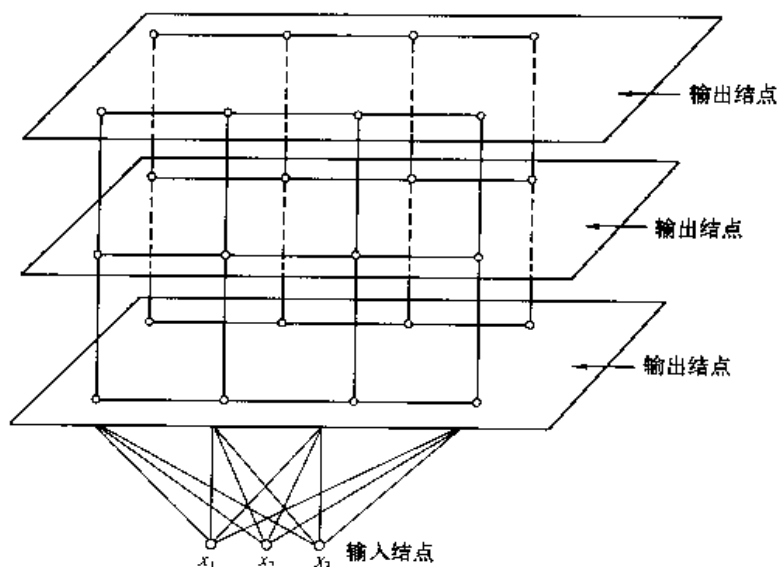


图 6-10 3 维 kohonen 网络示意图

以陕西省地处黄土高原的 33 个县(市)为统计样本,分别记为 S^p ($p = 1, 2, \dots, 33$),当 $p = 1, 2, \dots, 33$ 时,依次表示 33 个县(市)行政单位。对每个样本特征向量 S^p ,选取其相应的特征向量 X_i ($i = 1, 2, \dots, q$),这 q 个特征变量的含义如表 6-1 所示。

对每个 X_i ,取各县(市)生态破坏的经济总损失额为预测目标,记为 q_0 ,显然各县(市)生态破坏的经济损失是各种影响因素综合作用结果的总体效应。

根据 q_0 的大小进行分级。即

第一级 $q_0 < 1000$ (万元)

第二级 1000 (万元) $< q_0 < 3000$ (万元)

第三级 $q_0 > 3000$ (万元)

随机选取 30 个样本,如表 6-2 所示。

表 6-1 特征比较数列表

特征比较数列	含 义
X_1	滥伐森林、毁林开荒的生态经济损失(千元)
X_2	草场沙化、退化的生态经济损失(千元)
X_3	水土流失的生态经济损失(千元)
X_4	冲毁农田的生态经济损失(千元)
X_5	陡坡耕垦的生态经济损失(千元)
X_6	基本建设的生态经济损失(千元)
X_7	土壤盐渍化、沙化等的生态经济损失(千元)
X_8	农业总产值(千元)
X_9	人口密度(人/ km^2)

表 6-2

30 个学习样本

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	节点位置	类别
66	4039	774	14	6216	1112	2342	48780	73	4	2
0	1720	830	4	2477	323	4511	35000	67	5	1
0	1201	1174	5	5896	1769	9312	36980	74	19	2
260	2074	3935	9	4502	259	7590	32930	43	24	2
0	5012	129	0	5610	9	8034	29020	24	2	2
0	5692	4652	0	5529	780	16015	28310	26	29	3
2091	1908	1	13	1323	610	0	18950	26	15	1
0	2571	3	10	757	863	0	38200	29	0	1
0	2590	54	27	442	5819	753	64000	82	12	1
0	3704	21	2	1318	1972	45	29620	33	20	1
0	3120	2	17	395	9276	0	18680	19	7	2
0	1528	15	8	172	4577	162	29040	38	16	1
0	13096	591	50	9037	5507	32801	53600	35	14	3
0	8233	639	42	7297	2102	24401	27910	52	23	3
0	6169	1812	23	1445	8219	7930	51160	59	13	2
0	8661	13634	16	6144	5392	24250	67000	40	28	3
0	19097	1726	42	7724	2494	16315	47910	34	8	3
0	2939	310	11	7594	2424	6500	47870	148	18	2
0	1392	598	1	3738	1863	7010	32450	146	17	2
0	3157	274	1	4394	2169	12898	35280	97	9	2
0	680	50	27	784	927	2208	10930	156	1	1
0	2352	365	5	7629	2225	7792	40330	116	3	2
5691	0	23	6	455	7810	216	28040	498	11	2
0	529	21	1	1181	3420	228	51430	156	26	1
3484	1427	121	11	1374	1582	0	23700	57	21	1
910	0	66	29	2423	6159	740	37830	170	6	2
5362	0	0	44	2474	5908	969	44480	214	22	2
1235	0	6	11	760	1946	1002	32080	249	10	1
8705	0	18	43	1296	3703	2578	51680	126	27	2
875	0	148	11	725	3566	3284	35260	158	25	1

作为神经网络学习的特征向量, 即 $X_i (x_1, x_2, \dots, x_9)$ 作为输入。输出节点构成 $5 \times 3 \times 2$ 点阵。网络经过学习训练后, 能正确地划分这些样本, 建立了生态因素与其破坏程度之间的对应关系, 如图 6-11 所示。

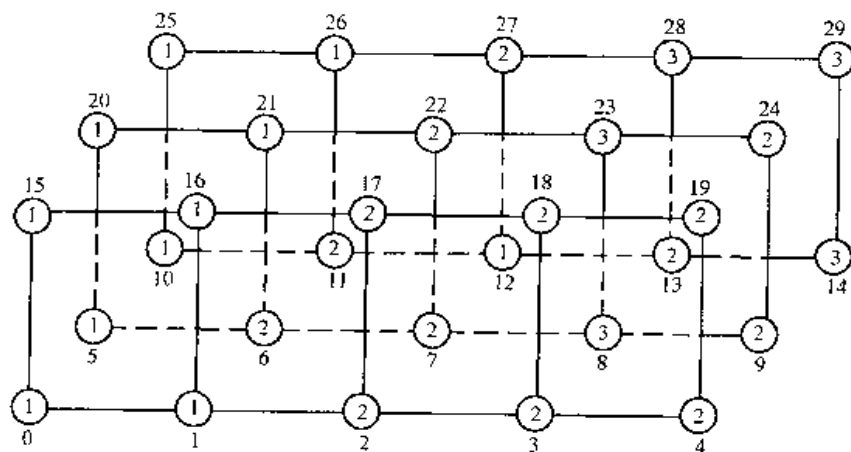


图 6-11 网络输出分类结果示意图

①, ②, ③…… 类别; 0~29—网络输出节点

第六节 本章小结

Kohonen 网络是一种自组织竞争人工神经网络, 该网络具有以下几个特点:

(1) 网络中权值是输入样本的记忆。如果输出神经元 j 与输入 n 个神经元之间的连接权用 w_j 表示, 对应某一类样本 x^p 输入, 使 y_j 达到匹配最大, 那 w_j 通过学习后十分靠近 x^p , 因此以后当 x^p 再次输入时, y_j 这神经元必定兴奋, y_j 是样本 x^p 的代表。

(2) 网络在学习过程中对权值的调整, 不只是对兴奋的那个神经元所对应的权值进行, 而同时对其周围 N_E 区域内的神经元进行调整。因此对于在 N_E 内神经元可以代表不只是一个样本 x^p , 而是与 x^p 比较相近的样本都可以在 N_E 内得到反映, 因此, 这种网络对于样本的畸变和噪声的容声量比较大。

(3) 网络学习的结果使比较相近的输入样本在输出二维平面的位置上也比较接近。

ART 网络也是一种自组织竞争人工神经网络, 该网络和算法具有较大的灵活性, 可适应新输入模式, 同时能避免对网络先前所学习过的模式的修改, 其记忆容量可以随样本的增加而自动增大, 且可在不破坏原记忆样本的情况下学习新的样本。其主要特点如下:

(1) 对已学过的模式具有稳定的快速识别能力, 并能迅速适应对新模式的学习。

(2) 不需要事先已知样本结果, 可进行无监督式的学习。

(3) 可根据某些特征量在全体样本中所占的比例, 实现自归一功能。

参考文献

1. Andreas Z. Simulation neuronaler netze. Boon: Addison-Wesley (Deutschland) GmbH, 1994
2. Venkatasubramanian V, Chan K. A Neural Network Methodology for Process Fault Diagnosis. AIChE J, 1989, 35: 1993 ~ 2002
3. Hoskins J C, Himmelblau D M. Fault Detection and Diagnosis Using Artificial Neural Networks. San Diego: Academic Press. CA, 1990, 123 ~ 130
4. Hoskins J C, J. C. Hoskins, K. M. Kaliyur, and David M. Himmelblau. Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks. AIChE J, 1990, 37: 137 ~ 141
5. V. Venkatasubramanian, R. V. aidyanathan, and Y. yamamoto. Process Fault Detection and Diagnosis Using Neural Networks. Computer & Chemical Engineering, 1990, 14: 699 ~ 712
6. Rumelhart D, Hinton G, Williams R. Learning Representations by Back-propagating Errors. Nature, 1986, 323: 533 ~ 538
7. 朱群雄, 麻德贤. 神经网络过程模型辨识. 化工学报, 1997, 48 (5): 547 ~ 552
8. 张勤鑫, 江涓, 吴肖群, 戴擎鏊. 人工神经网络辅助丙烷氨氧化工艺条件优化设计. 化学反应工程与工艺, 1999, 65 (3): 282 ~ 287, 294
9. 杨启平, 薛五德, 蓝之达. 基于人工神经网络的变压器故障诊断. 变压器, 2000, 37 (3): 33 ~ 36
10. 何雨侯, 闫桂荣等. 基于神经网络的电机状态监测与早期故障诊断. 电工技术学报, 1997, 12 (5): 41 ~ 44
11. 张津春, 杨伟东, 蔡耀淳. 采用人工神经网络技术对高压线缆绝缘破坏预测的研究. 河北工业大学学报, 1998, 27 (1): 20 ~ 24
12. 何悦盛, 叶春, 杨波, 忻建华. BP 网络在电站故障诊断系统中的应用研究. 电力自动化设备, 2002, 22 (5): 7 ~ 9
13. 范澍, 毛承雄, 陆继明. 基于 BP 神经网络的最优励磁控制器. 电力系统自动化, 2000, 24 (2): 29 ~ 32
14. 王建辉, 顾树生. 一种基于神经网络的直流传动控制系统. 东北大学学报 (自然科学版), 1998, 19 (1): 36 ~ 39
15. 姜齐荣, 闵勇, 韩英铎. 基于人工神经网络的励磁调节、快门控制和电阻掣动的协调控制. 清华大学学报, 1997, 37 (7): 67 ~ 70
16. 袁宇春, 张保会等. 用 ANN 决定电力系统的切负荷控制. 中国电力, 1997, 30 (7): 3 ~ 5, 20
17. 王钢, 李遥等. 基于 ANN 的超高压输电线路方向高频保护. 电力系统自动化, 1998, 22 (5): 18 ~ 21
18. 段玉倩, 贺家李. 基于人工神经网络的距离保护. 中国电机工程学报, 1999, 19 (5): 67 ~ 70
19. 徐斌, 张国基. 连续型 Hopfield 神经网络求解设备布置问题. 华南理工大学学报 (自然科学版), 2001, 29 (9): 9 ~ 12
20. 彭天波. 基于 Hopfield 网络的滞水梯级水电站实时优化调度. 华中电力, 1997, 10 (3): 14 ~ 18
21. 孙春鹏, 魏文秋, 王伶俐. 基于 Hopfield 网络的排放口最优规划算法研究. 水文, 2001, 21 (6): 18 ~ 19, 17

22. 张杰, 李俊民, 王珏. 基于 Hopfield 神经网络的状态观测器设计. 西安电子科技大学学报 (自然科学版), 2002, 29 (2): 271 ~ 274
23. 宣士斌. 动态 Hopfield 神经网络模型及其学习算法. 广西民族学院学报 (自然科学版), 1999, 5 (3): 5 ~ 6, 14
24. Narendra K S, Parthasarathy K. Identification and control of dynamical system using neural networks. IEEE Trans on Neural Networks, 1990, 1 (1): 4 ~ 27
25. 张国忠, 叶霞, 胡益民, 陈宇峰. 基于 BP 神经网络的湖北省水电发电量预测模型. 中国农村水利水电, 2001, (9): 52 ~ 53
26. 胥悦红, 顾培亮. 基于 BP 神经网络的产品成本预测. 管理工程学报, 2000, 14 (4): 61 ~ 64
27. 邬红娟, 林子扬, 郭生练. 人工神经网络方法在资源与环境预测方面的应用. 长江流域资源与环境, 2000, 9 (2): 237 ~ 241
28. 赵瑞, 申金山, 郭院成, 关柯. 神经网络在城市用水量预测中的应用研究. 郑州工业大学学报, 2000, 21 (1): 75 ~ 77
29. 吴义虎, 陈荫山, 侯志祥等. 基于神经网络的中国汽车保有量建模与预测. 西安公路交通大学学报, 2001, 21 (2): 91 ~ 93
30. 史雪岩, 陈祥光, 傅若农等. 人工神经网络在气相色谱保留值预测上的应用. 分析科学学报, 2000, 16 (3): 196 ~ 200
31. A. Berthod, E. Y. Zhou, D. W. Armstrong. Determination and use of Rohrschneider McReynolds Constants for Chiral Stationary Phase Used in Capillary Gas Chromatography. Anal. Chem., 1995, 67, 849-857
32. M. H. Abraham. Characterization of Some GLC Chiral Stationary Phase LFER Analysis Anal. Chem., 1997, 69, 613-617
33. 陈祥光, 薛锦诚, 傅若农, 顾峻岭. 引入模拟辅助样本提高 BP 网络的泛化能力. 分析科学学报, 2002, 18 (2): 137 ~ 141
34. 蔡煜东, 姚林声. 人工神经网络方法预测气相色谱保留值. 分析仪器, 1994, (4): 12 ~ 16
35. 陈祥光, 黄聪明. 改进 BP 反传算法在智能控制系统中的应用. 北京理工大学学报, 1997, 17 (4): 481 ~ 487
36. 陈祥光, 黄聪明. 神经网络智能控制系统辨识模型结构的研究. 北京理工大学学报, 1999, 19 (1): 53 ~ 57
37. 皮陵霞, 李家启. RBF 神经网络在油田累计产量系统的建模和预测中的应用. 交通与计算机, 2001, 19 (101): 92 ~ 94
38. 刘芳, 李海涛等. 基于径向基函数网络的微合金钢流变应力的预测. 塑性工程学报, 2001, 8 (4): 9 ~ 12
39. 王旭东, 邵惠鹤. 神经元网络建模与软测量技术. 化工自动化及仪表, 1996, 23 (2): 28 ~ 31
40. 马野, 李楠. 应用自组织映射神经网络进行模拟电路故障诊断的仿真研究. 系统仿真学报, 2001, 13 (5): 582 ~ 584
41. 肖军, 吴杰, 尹新. 自组织特征映射神经网络权值的实现. 系统工程与技术, 1997, (12): 1 ~ 3, 11
42. 蔡煜东, 许伟杰. 应用自组织人工神经网络预报黄土高原生态破坏程度的初探. 农村生态环境学报, 1995, 11 (2): 28 ~ 31