

# 人工神经网络原理及应用

朱大奇 史 慧 编著

科学出版社

## 内 容 简 介

本书介绍了人工神经网络的基本原理及其应用。重点阐述了 9 种常见神经网络的结构组成、工作原理、设计方法及应用实例。其中心内容包括前馈型 BP 神经网络；反馈型 Hopfield 神经网络和双向联想记忆 BAM 神经网络；局部逼近的 CMAC 小脑神经网络和径向基函数 RBF 神经网络；竞争学习的自组织 SOM 神经网络、对偶传播 CPN 神经网络、ART 自适应谐振理论及量子神经网络。

本书可作为电子、自动化、仪器仪表、计算机及相关专业研究生教材，书中介绍的相关算法及应用实践，对相关理论研究者和工程技术人员也具有一定的指导意义。

### 图书在版编目 (CIP) 数据

人工神经网络原理及应用/朱大奇，史慧编著.

—北京：科学出版社，2006

ISBN 7-03-016570-5

. 人... . 朱... 史... . 人工神经网络 .TP.183

中国版本图书馆 CIP 数据核字 (2005) 第 144552 号

责任编辑：洪 英 / 责任校对：科 海

责任印刷：科 海 / 封面设计：林 陶

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码：100717

<http://www.sciencep.com>

北京科普瑞印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

\*

2006 年 3 月第一版

开本：16 开

2006 年 3 月第一次印刷

印张：14.25

印数：1—3000

字数：347 千字

定价：24.00 元

(如有印装质量问题，我社负责调换)

# 前 言

“人脑是如何工作的？”、“人类如何从现实世界获取知识和运用知识？”、“我们能否制作模仿人脑的人工神经系统？”。多少年来，人们从医学、生物学、生理学、信息学、认知学等各个角度试图认识并解答这些问题。

心理学家和认知科学家从探索人脑加工、存储和搜索信息的机制出发，试图弄清人脑功能机理，建立人类认识过程的微结构理论；生物学、医学专家试图通过人脑的生理学研究推动脑科学向定量、精确和系统化方向发展，寄希望于临床医学的新突破；信息处理与计算机科学家则试图寻求新的途径以解决目前计算机不能解决或不善解决的大量问题，构造更加逼近人脑功能的新一代计算机模型。各学科的发展和相互渗透，逐渐形成了一个新兴的多学科交叉的技术领域——人工神经网络。

作为一门新兴的信息处理科学，人工神经网络以人的大脑工作模式为基础，研究自适应、非程序的信息处理方法。这种工作机制的特点表现在通过网络中大量神经元的相互作用来体现它自身的处理功能，从模拟人脑的结构和单个神经元功能出发，达到模拟人脑处理信息的目的。

近十几年来，人工神经网络一直是科学界的研究热点，无论是理论研究还是应用实践都有大量的成果报道。国内陆续出版了一些有关人工神经网络的书籍，这对人工神经网络在我国的发展起到了很好的推动作用，它们大多是以专著或译著的形式出版，随着人工神经网络学科的发展与普及，人工神经网络已成为高等学校电类及相关专业研究生的必修课程，但目前出版的人工神经网络书籍适合作研究生教材的还较少。本书从研究生教学的实际出发，结合作者多年来在高校从事神经网络研究和教学工作的经验，重点阐述人工神经网络的基本原理和实际应用，同时兼顾近年来神经网络的新进展。在内容阐述上有以下特点：

- 通俗易懂，条理清晰，便于自学。注重物理概念内涵的阐述，尽量避免繁琐的数学推导。
- 注重理论联系实际，加强神经网络应用实例的介绍。以期读者在学习过程中得到启发，进而将神经网络模型应用到自己的研究课题之中。对选自于科技论文中的应用实例，都进行了改编、分析与说明。
- 按神经网络类型，逐个阐述其工作原理、网络特点及局限性，并提供应用实例，每章自成体系，便于取舍和教学。

全书内容共分 10 章。其中：

第 1 章为人工神经网络基础，阐述了人工神经网络的历史、研究概况及发展趋势，介绍了人工神经网络的基本模型、基本要素及其功能，包括转移（激励）函数、学习算法等。

为后续章节的学习奠定基础。

第2章介绍前馈型误差反传BP神经网络,重点阐述了BP神经网络的基本结构、标准学习算法及其局限性、改进的BP学习算法及BP神经网络的一般性设计原则。同时介绍了BP网络在各方面的应用。

第3章分别对离散型和连续型Hopfield反馈神经网络的结构、工作原理及稳定性进行了分析;最后从优化计算及联想记忆两方面介绍了Hopfield神经网络的应用。

第4章针对Hopfield神经网络自联想记忆,讨论了BAM异联想记忆神经网络,重点介绍了离散型双向联想记忆神经网络的结构、工作原理、权矩阵设计方法及稳定性。

第5章在讨论CMAC小脑神经网络基本结构及原理的基础上,从CMAC网络输出计算阶段和网络权值调整阶段两个方面,重点介绍了模糊CMAC神经网络,基于信度分配的CMAC神经网络,基于平衡学习的CMAC神经网络及基于信度分配的模糊CMAC神经网络的结构、工作原理及权值调整学习算法,并给出了具体的仿真算例。本章内容得到了江苏省自然科学基金“动态非线性系统在线故障诊断的神经网络学习算法研究”课题的资助(NO.BK2004021)。

第6章讨论径向基函数神经网络RBF的结构、工作原理,在此基础上,介绍了一种基于免疫算法的改进RBF神经网络隐层数据中心个数的确定、隐层数据中心位置的确定方法和输出权值的学习调整算法。最后讨论了RBF神经网络在热工过程辨识方面的具体应用。

第7章讨论竞争算法的学习过程,在此基础上进一步介绍SOM自组织特征映射神经网络的结构、工作原理。最后讨论了SOM神经网络在聚类分析、复合材料的损伤检测及模拟电路故障识别中的具体应用。

第8章在介绍简单CPN神经网络拓扑结构及训练算法的基础上,进一步讨论了双向CPN神经网络的结构和工作原理。最后分析了CPN神经网络在模拟电路故障识别中的具体应用。本章内容得到了教育部科学技术研究重点项目“鲁棒控制与故障检测及其集成应用”的资助(NO.105088)。

第9章分别讨论了简单的ART1型神经网络拓扑结构及训练算法、用于处理连续型模拟信号的ART2型神经网络的结构与工作原理,以及具有3R功能的综合ART系统。最后介绍了ART神经网络在图像处理及故障诊断中的具体应用。本章内容得到了总装备部国防科技预研基金“神经网络智能故障诊断专家系统”课题的资助(NO.413170203)。

第10章介绍了近些年受到广泛关注的量子神经网络的一些基础知识及研究现状。并介绍了相关的一些应用。

本书由江南大学智能控制研究所朱大奇教授和中国航天科工集团史慧研究员共同撰写。其中第1章、第2章、第3章、第4章、第5章、第7章、第8章、第10章由朱大奇教授执笔,第6章、第9章由史慧研究员执笔。全书由朱大奇教授统稿。

本书可作为电子、自动化、仪器仪表、计算机及相关专业研究生教材,书中介绍的相关算法及应用实践,对相关理论研究者 and 工程技术人员也具有一定的指导意义。

由于作者水平所限,时间仓促,书中难免存在不足和错误之处,敬请广大读者和专家批评指正。

编 者

# 致 谢

本书由东南大学博士生导师曹进德教授和江南大学博士生导师、“太湖学者”王士同教授主审，两位专家在审阅过程中对本书提出的宝贵意见与建议使作者受益匪浅，在此深表感谢！

书中引用了一些学者的论著和研究成果，为此向他们表示深深的谢意！没有这些研究成果的加入，本书难成体系。

在本书的撰写过程中，作者的学生杨兴、易健雄、苏圣超、陈昕静、刘永安等人帮助进行了文字录入与绘图。

作者同时要感谢出版社的邹华跃老师、陈跃琴老师，没有他们的大力支持和辛勤工作，就没有本书的出版。

书中相关章节内容得到了江苏省自然科学基金 (NO.BK2004021)、教育部科学技术研究重点项目 (NO.105088)、总装备部国防科技预研基金 (NO.413170203) 及江南大学研究生精品课程项目资助。

P137

表 6-1 为 IRBF 与 CMAC 学习结果的误差分析，其误差计算按式 (6-16) 进行，从中可见 IRBF 在学习 5 次、10 次、20 次及 30 次时，其逼近误差均小于 CMAC，特别是在初始训练阶段尤其明显。也就是说，基于免疫算法的 IRBF 神经网络在非线性函数逼近中，不仅计算量小、速度快、实时性好，而且精度高、误差小，对动态非线性系统的在线故障辨识来说，是一种实用而有效的辨识算法。

## 第1章 人工神经网络的基础知识

人类关于认知的探索由来已久。早在公元前 400 年左右,希腊哲学家柏拉图(Plato)和亚里士多德(Asidtole)等,就曾对人类认知的性质和起源进行过思考,并发表了有关记忆和思维的论述。在此及以后很长的一段时间内,由于科学技术发展水平所限,人们对人脑的认识主要停留在观察和猜测的基础之上,缺乏有关人脑内部及其工作原理的科学依据,因而进展缓慢。直到 20 世纪 40 年代,随着神经解剖学、神经生理学以及神经元的电生理过程等的研究取得突破性进展,人们对人脑的结构、组成及最基本工作单元有了越来越充分的认识,在此基本认识的基础上,以数学和物理方法以及信息处理的角度对人脑神经网络进行抽象,并建立简化的模型,称为人工神经网络 ANN(Artificial Neural Network),为叙述方便将人工神经网络直接称之为神经网络(以下同)。

目前,关于神经网络的定义尚不统一,按美国神经网络学家 Hecht Nielsen 的观点,神经网络的定义是:“神经网络是由多个非常简单的处理单元彼此按某种方式相互连接而形成的计算机系统,该系统靠其状态对外部输入信息的动态响应来处理信息”。综合神经网络的来源、特点和各种解释,它可简单地表述为:人工神经网络是一种旨在模仿人脑结构及其功能的信息处理系统<sup>[1]</sup>。

作为一门活跃的边缘性交叉学科,神经网络的研究与应用正成为人工智能、认识科学、神经生理学、非线性动力学等相关专业的热点。近十几年来,针对神经网络的学术研究大量涌现,它们当中提出上百种的神经网络模型,其应用涉及模式识别、联想记忆、信号处理、自动控制、组合优化、故障诊断及计算机视觉等众多方面,取得了引人注目的进展。

### 1.1 人工神经网络发展的历史及现状

#### 1.1.1 人工神经网络研究阶段的划分

纵观神经网络的发展历史,其发展过程大致可以概括为如下 3 个阶段。

##### 1. 第一阶段——启蒙时期

这是神经网络理论研究的奠基阶段,1943 年,神经生物学家 McCulloch W.S.和青年数学家 Pitts W.A.合作,提出了第一个人工神经元模型,并在此基础上抽象出神经元的数理模型<sup>[2]</sup>,开创了人工神经网络的研究,以 McCulloch W.S.和 Pitts W.A.提出人工神经元的数理模型(即神经元的阈值模型,简称 MP 模型)为标志,神经网络拉开了研究的序幕。为了模拟起连接作用的突触的可塑性,神经生物学家 Hebb 于 1949 年提出了连接权值强化的 Hebb 法则<sup>[3]</sup>。这一法则告诉人们,神经元之间突触的联系强度是可变的,这种可变性是学习和记忆的基础。Hebb 法则为构造有学习功能的神经网络模型奠定了基础。1952 年英国生物学家 Hodgkin 和 Huxley 建立了著名的长枪乌贼巨大轴索非线性动力学微分方程,即 H-H 方程。这一方程可用来描述神经膜中所发生的非线性现象如自激震荡、混沌及多重稳定性等问题,所以有重大的理论与应用价值。1954 年,生物学家 Eccles 提出了真实突触的分流模型<sup>[4]</sup>,这一模型由于通过突触的电生理实验得到证实,因而为神经网络模拟突触的功能提供了原型和生理学的证据。1956 年,Uttley 发明了一种有处理单元组成的推理机,用以模拟行为及条件反射。70 年代中期他把该推理机用于自适应模式识别,并认为该模型能反映实际神经系统工作原理。

1958 年 Rosenblatt 在原有 MP 模型的基础上增加了学习机制<sup>[5]</sup>。他提出的感知器模型,首次把神经网络理论付诸工程实现,他的成功之举大大激发了众多学者对神经网络的兴趣。

Rosenblatt 证明了两层感知器能够对输入进行分类,他还提出了带隐层处理元件的三层感知器这一重要的研究方向。Rosenblatt 的神经网络模型包含了一些现代神经计算机的基本原理,从而形成神经网络方法和技术的重大突破。神经网络的研究迎来了第一次高潮期。

1960 年 Widrow 和 Hoff 提出了 ADALINE 网络模型<sup>[6]</sup>,这是一种连续取值的自适应线性神经网络模型,可以用于自适应系统。他们针对输入为线性可分的问题进行了研究,得出期望响应与计算响应的误差可能搜索到全局最小值;在研究神经网络中会出现的回响现象时,意大利科学家 Caianiello 在神经元模型中引入了不应期特性;为了研究思维和大脑结合的理论问题, Grossberg 从信息处理的角度,研究了自组织性、自稳定性和自调节规律<sup>[7-8]</sup>;日本科学家 Amari 注重把生物神经网络的行为与严格的数学描述相结合,在数学求解研究上得到一定成果;Willshaw 等人提出了一种称为全息音的模型<sup>[9]</sup>,为利用光学原理实现神经网络奠定了理论基础;Nilsson 对多层机即有隐层的广义认知机作了精辟的论述等<sup>[10]</sup>。上述成果足以表明神经网络研究已获得了广泛的成功。

## 2. 第二阶段——低潮时期

正当一些科学家怀着极大的热情追求神经网络那遥远但并非不可及的目标时,人工智能的创始人之一 Minsky 和 Papert 对以感知器为代表的网络系统的功能及局限性从数学上做了深入研究,于 1969 年发表了轰动一时《Perceptrons》一书,指出简单的线性感知器的功能是有限的,它无法解决线性不可分的两类样本的分类问题,如简单的线性感知器不可能实现“异或”的逻辑关系等。这一论断给当时人工神经网络的研究带来沉重的打击,以致美国及前苏联的一些科研机构纷纷停止对此类项目提供资助,而使得这个领域的许多学者不得不转向其他课题的研究。开始了神经网络发展史上长达 10 年的低潮期。

使神经网络研究处于低潮的另外一个原因是,20 世纪 70 年代以来,集成电路和微电子技术的迅猛发展,使传统的 Von Neumann 计算机进入全盛时期,基于逻辑符号处理方法的人工智能得到了迅速发展并取得显著成绩,它们的问题和局限性尚未暴露,因此暂时掩盖了发展新型计算机和寻求新的神经网络的必要性和迫切性。

也许 Minsky 的评论是过于苛刻了,不过这一评论一定程度上暴露出当时神经网络研究的局限性,因而有一定的启发性。可喜的是,仍有少数具有远见卓识的科学家持之以恒的继续这一领域的研究,另有一些科学家在此期间新投入到这一领域中,成为低潮期神经网络研究的亮点。

1976 年,美国 Grossberg 教授提出了著名的自适应共振理论 ART(Adaptive Resonance Theory),其学习过程具有自组织和自稳定的特征。其后的若干年中,他与 Carpenter 一起研究了 ART 网络<sup>[11]</sup>,并有 ART1、ART2 和 ART3 三个 ART 系统的版本,ART1 网络只能处理二值的输入,ART2 比 ART1 复杂并且能处理连续型输入。ART3 网络纳入了生物神经元的生物电-化学反应机制,其结构更接近人脑的工作过程。

1972 年,有两位学者分别在欧洲和美洲异地发表了类似的神经网络开发结果。一位是芬兰的 Kohonen T.教授,提出了自组织神经网络 SOM(Self-Organizing feature map);另一位是美国的神经生理学家和心理学家 Anderson J.,提出了一个类似的神经网络,称为“交互存储器”。后来的神经网络主要是根据 Kohonen T.的工作来实现的<sup>[12]</sup>。SOM 网络是一类无导师学习网络,主要用于模式识别、语音识别及分类问题。它采用一种“胜者为王”的竞争学习算法,与先前提出的感知器有很大的不同,同时它的学习训练方式是无指导训练,是一种自组织网络。这种学习训练方式往往是在不知道有那些分类类型存在时,用作提取分类信息的一种训练。

低潮期另一位重要人物是日本的福岛邦彦(Kunihiko Fukushima)。他开发了一些神经网络结构和训练算法,其中最有名的是 1980 年发表的“新认知机”(Neocognitron)。“新认知机”是视觉模式识别机制模型,它与生物视觉理论相结合,其目的在于综合出一种神经网络

模型，使它像人类一样具有进行模式识别能力。

在整个低潮时期，上述许多重要研究成果虽然未能得到应有的重视，但其科学价值不可磨灭，他们的工作为日后神经网络理论研究的又一次高涨打下了坚实的基础。

### 3. 第三阶段——复兴时期

这是神经网络理论研究的主要发展时期。1982 年，美国国家科学院的刊物上发表了著名的 Hopfield 模型的理论<sup>[13]</sup>。Hopfield 神经网络是如下的一组非线性微分方程

$$C_i dU_i / dt = \sum_{j=1}^N T_{ij} f_j[u_j] - u_i / R_i + I_i, (i = 1, 2, \dots, N)$$

其中  $U_i$  是第  $i$  个神经元的膜电位； $C_i$ ， $R_i$  分别是输入电容和电阻； $I_i$  是电路外的输入电流；

$T_{ij}$  是第  $j$  个神经元对第  $i$  个神经元的联系强度； $f(u)$  是  $u$  的非线性函数。Hopfield 构造出

Laypunov 函数，并证明了在  $T_{ij} = T_{ji}$  情况下，网络在平衡点附近的稳定性，还将这种模型

用电子电路来实现。Hopfield 的模型不仅对人工神经网络信息存储和提取功能进行了非线性数学概括，提出了动力方程和学习方程，还对网络算法提供了重要公式和参数，使人工神经网络的构造和学习有了理论指导，在 Hopfield 模型的影响下<sup>[14]</sup>，大量学者又激发起研究神经网络的热情，积极投身于这一学术领域中，神经网络理论研究很快便迎来了第二次高潮。

同年，Marr 开辟了视觉和神经科学研究的新篇章<sup>[15]</sup>，他对视觉信息加工和过程进行了全面、系统和深刻的描述，并与神经实现机制联系起来。1983 年，Kirkpatrick 等人认识到模拟退火算法可用于 NP 完全组合优化问题的求解<sup>[16]</sup>，这种模拟高温物体退火过程来寻找全局最优解的方法最早由 Metropolis 等人 1953 年提出的。1984 年，Hinton 与年轻学者 Sejnowski 等合作提出了大规模并行网络学习机，并明确提出隐单元的概念，这种学习机后来被称为 Boltzmann 机<sup>[17]</sup>。Poggio 等人以 Marr 视觉理论为基础，提出了初级视觉的正则化方法<sup>[18-19]</sup>。我国生物物理学家汪云九提出了视觉神经元的广义 Gabor 函数模型以及有关立体视觉、纹理检测、运动方向检测、超视觉度现象等的计算模型。

1986 年，由 Rumelhart 和 McClelland 主编的有 16 位作者参加撰写的《Parallel Distributed Processing: Exploration in the Microstructures of Cognition》出版<sup>[20]</sup>，该书中，他们建立了并行分布处理理论，主要致力于认知的微观研究，同时对具有非线性连续转移函数的多层前馈网络的误差反向传播算法即 BP 算法进行了详尽的分析，解决了长期以来没有权值调整有效算法的难题。可以求解感知机所不能解决的问题，回答了《Perceptrons》一书中关于神经网络局限性的问题，从实践上证实了人工神经网络有很强的运算能力，BP 算法是目前最引人注目、应用最广泛的神经网络算法之一。该书的出版表明 PDP 理论已达到一个新水平，在世界各地产生了广泛的影响。

在神经网络的物理实现方面，Mead 和 Conway、Mahowald 等人合作，研制一种动物神经系统的电子电路模拟，即硅神经系统，他们以人的视网膜中锥体细胞的方式来连接一块 VLSI 芯片。1988 年，Chua 和 Yang 提出了细胞神经网络 (CNN) 模型<sup>[21-22]</sup>，它是一个细胞自动机特性的大规模非线性计算机仿真系统；Kosko 建立了双向联想存储模型 (BAM)<sup>[23-24]</sup>，它具有非监督学习能力。

90 年代中后期，神经网络研究步入了一个新的发展时期，一方面已有理论在不断深化和得到进一步推广，另一方面，新的理论和方法也从未停止过其不断开拓的步伐。Edelman 提出的 Darwinism 模型在 90 年代初产生了很大的影响，他建立了一种神经网络系统理论，以 Darwinism 为例，其组成包括输入阵列、Darwin 网络和 Nallance 网络，而这两个网络



又包含一些功能不同的子网络等；1991 年，Haken 把协同学引入神经网络，在他的理论框架中，他认为，认知过程是自发的，并断言模式识别过程即是模式形成过程；我国学者吴佑寿、廖晓昕、曹进德、黄德双、戴先中等在这一时期对神经网络进行了大量的研究，1991 年吴佑寿等人提出了一种激励函数可调的神经网络模型，试图对先验知识加以利用。1994 年，廖晓昕关于细胞神经网络的数学理论与基础的提出，带来了这个领域新的进展<sup>[25]</sup>。通过拓广神经网络的激活函数类，给出了更一般的时滞细胞神经网络(DCNN)<sup>[61]</sup>、Hopfield 神经网络(HNN)<sup>[62]</sup>、双向联想记忆网络(BAM)模型，并借助杨格(Young)不等式、琴生(Jensen)不等式和 Lyapunov 泛函方法，通过巧妙引入参数，分别就神经网络系统含有常滞量、变滞量的二种情况，证明了神经网络模型的全局渐近稳定性，得到了一系列的充分判据，对全局稳定的神经网络的设计和应用有重要指导意义。这一时期，另一些新的发展方向也非常的引人关注。

以光学方法来实现神经网络，即光学神经网络，由于能充分发挥光学强大的互连能力和并行处理能力，因而受到重视。Wunsch 在 90 OSA 年会上提出一种光电 ART，它的主要计算强度由光学硬件完成。1995 年，Jenkins 等人研究了光学神经网络(PNN)，建立了光学二维并行互连与电子学混合的光学神经网络系统<sup>[26]</sup>。其他还有 McAulay 等学者致力于将电子俘获材料用于光学神经网络等等<sup>[27-28]</sup>。

鉴于非线性系统控制问题的复杂性，一些学者试图通过神经网络方法来解决此类问题。1990 年 Narendra 和 Parthasarathy 提出了一种推广的动态神经网络系统即其连接权的学习算法<sup>[29]</sup>，增强了非线性系统控制的鲁棒性。戴先中等人提出了连续非线性系统的神经网络( $\alpha$ 阶逆系统控制方法)<sup>[30]</sup>。Miller 等人在小脑模型关节控制器(CAMC)基础上，提出的非线性系统控制方法<sup>[31]</sup>，具有局部逼近和修改权极小的特点，但存在由于采用间断超平面对非线性曲面进行逼近而有时会出现精度不够的问题。1993 年 Bulsari 提出以乘积 Sigmoidal 函数作为激活函数<sup>[32]</sup>，给出了非线性系统用神经网络逼近的构造性描述等等。

随着人们发现人脑中存在着混沌现象，一些科学家进而又提出了混沌神经网络理论。1991 年 Aibara 等在前人推导和实验的基础上，给出一个混沌神经元模型<sup>[33]</sup>。1991 年 Inoue 等提出用耦合混沌振荡子作为单个神经元构造混沌神经网络模型的方法<sup>[34-35]</sup>。耦合混沌振荡子的同步和异步分别对应神经元激活和抑制两个状态。虽然混沌是由简单的确定性规则产生的，但它包含规则性和不规则性两个方面，耦合的混沌振荡子的同步来自规则性，而不规则性可产生随机搜索能力。

考虑到人的思维及表达上常具有模糊性，一些学者把神经网络的研究与模糊系统联系起来<sup>[36-37]</sup>，从而导致了模糊神经网络的产生。Hiroyuki Qkada 等人把模糊神经网络模型用于金融风险评估，他们提出了由 Sigmoid 型节点和线性节点构成的模糊神经网络模型<sup>[38]</sup>，其中的模糊规则由领域专家给出。该模型具有网络结构简单、模糊规则易于理解、有学习能力及能充分利用专家知识等特点，不足在于该网络连接结构及其权值的确定过分依赖领域专家的知识，而专家知识的获取、有时会比较困难的。Cai Yaling 等人提出了由三种不同类型节点构成的模糊神经网络模型<sup>[39]</sup>，该网络具有能够对学习样本快速记忆，不需要领域专家知识指导等优点，但在学习样本较多时，规则数量较大时应用不理想。模糊神经网络在工业控制、金融时间序列分析、风险评估等方面预计有较大的应用潜力。

经过多年的发展，目前已有上百种的神经网络模型被提出，表 1-1 列出了神经网络发展过程中起过重要作用的十几种著名神经网络，它也是神经网络发展史的一个缩影。

表 1-1 重要影响神经网络

网络名称	发明者	时间	特点	局限性	应用领域
感知器 (Perceptron)	Frank Rosenblatt	1958	最早的神经网络， 有学习能力，只能	不能识别复杂 字符，与输入模	文字识别 、声音识

	(康奈尔大学)		进行线性分类	式的大小、平移和旋转敏感	别和学习记忆等
自适应线性单元 (Adaline)	Bernard Widrow (斯坦福大学)	1960-1962	学习能力较强,较早开始商业应用	要求输入-输出之间是线性关系	雷达天线控制、自适应回波抵消等
小脑自动机 (Cerebellatron)	Marr D. (麻省理工学院)	1969-1982	能调和各种指令系列,按需要缓慢地插入动作	需要复杂的控制输入	控制机器人的手臂运动
误差反传网络 BP(Back Propagation)	Werbs P. (哈佛大学) Rumelhart D. (斯坦福大学) McClelland (斯坦福大学)	1974-1985	多层前馈网络,采用最小均方差学习方式,是目前应用最广泛的网络	需要大量输入-输出数据,训练时间长,易陷入局部极小	语音识别、过程控制、模式识别等
自适应共振理论 ART (Adaptive Resonance Theory)	Carpenter G. Grossberg S. (波士顿大学)	1976-1990	可以对任意多个和任意复杂的二维模式进行自组织学习	受平移、旋转和尺度的影响;系统较复杂。	模式识别,长于识别复杂、未知模式
盒中脑 BSB 网络 (Brain State in a Box)	James Anderson (布朗大学)	1977	具有最小均方差的单层自联想网络,类似于双向联想记忆,可对片断输入补全	只能作一次性决策,无重复性共振	解释概念形成,分类和知识处理
新认知机 (Neocognition)	Fukushima K. (日本广播协会)	1978-1984	多层结构化字符识别网络,与输入模式的大小、平移和旋转无关,能识别复杂字型	需要大量加工单元和联系	手写字母识别
自组织特征映射网络 SOM (Self-Organizing feature map)	Tuevo Kohonen (芬兰赫尔辛基技术大学)	1980	对输入样本自组织聚类,可映射样本空间的分布	模式类型数需要事先知道	语音识别、机器人控制,图像处理等
Hopfield 网络	John Hopfield (加州理工学院)	1982	单层自联想网络,可从缺损或有噪声输入中恢复完整信息	无学习能力,权值要预先设定	求解 TSP 问题,优化计算及联想记忆等
玻尔兹曼机 (Boltzman machine);	Hinton J. (多伦多大学) Sejnow-	1985-1986	采用随机学习算法的网络,可实现全局最优	玻尔兹曼机训练时间长;柯西机在某些统计	图像、声纳和雷达等的模式

柯西机(Cauchy machine)	ski T. (霍布金斯大学)			分布下产生噪声	识别
双向联想记忆网 BAM(Bi-directional Associative Memory)	Bart Kosko (南加州大学)	1985-1988	双向联想式单层网络, 有学习功能, 简单易学	存储的密度低, 数据必须能编码	内容寻址的联想记忆
双向传播网 CPN(Counter Propagation)	Robert Hecht-nielsen(美国)	1986	一种在功能上作为统计最优化和概率密度函数分析的网络	需要大量处理单元和连接, 要高度准确	神经网络计算机, 图像处理和统计分析

### 1.1.2 人工神经网络国内外研究状况

随着人工神经网络 20 世纪 80 年代在世界范围内的复苏, 国内也逐步掀起了研究热潮。1989 年 10 月和 11 月分别在北京和广州召开了神经网络及其应用讨论会和第一届全国信号处理-神经网络学术会议; 1990 年 2 月由国内八个学会, 即中国电子学会、人工智能学会、自动化学会、通信学会、物理学会、生物物理学会和心理学会联合在北京召开“中国神经网络首届学术会议”。这次大会以“八学会联盟, 探智能奥秘”为主题, 收到了 300 多篇学术论文, 开创了中国人工神经网络及神经计算机方面科学研究的新纪元。2004 年 10 月在合肥召开的“人工神经网络学术会议”已是第 14 届学术年会了; 2004 年 8 月在中国大连召开的 ISNN2004(International Symposium on Neural Networks)国际会议, 引起了国内外神经网络研究者的广泛关注, 产生了较大的影响; 另外, 国内外许多相关的学术会议都设有神经网络专题, 如国内的 WCICA、CIAC、CDC、CCC、CAA 及国外的 ACC、CCA、IDEAC 等。经过十几年的发展, 中国学术界和工程界在人工神经网络的理论研究和应用方面取得了丰硕成果, 学术论文、应用成果和研究人员逐年增加。

在国际上, 1987 年, 在美国加州召开了第一届国际神经网络学术会议。此后每年召开的国际联合神经网络大会 IJCNN (International Joint Conference on Neural Networks), 成为神经网络研究者的重要学术交流平台。另外, 十几种国际著名的神经网络学术刊物相继问世, 如:《IEEE Transactions on Neural Networks》、《IEEE Transactions on Circuit and Systems》、《IEEE Transactions on Systems, Man, and Cybernetics》、《Journal of Artificial Neural Networks》、《Journal of Neural Systems》、《Neural Networks》、《Neural Computation》、《Networks: Computation in Neural Systems》、《Machine Learning》等, 至此神经网络理论研究在国际学术领域获得了其应有的地位。

### 1.1.3 神经网络的发展展望

经过近半个世纪的发展, 神经网络理论在模式识别、自动控制、信号处理、辅助决策、人工智能等众多研究领域取得了广泛的成功。关于学习、联想和记忆等具有智能特点过程的机理及其模拟方面的研究正受到越来越多的重视。目前神经网络研究与发展主要集中在以下几个方面<sup>[48]</sup>:

#### 1. 神经生理学、神经解剖学研究的发展

通过神经网络研究的发展, 我们对人脑一些局部功能的认识已经有所提高, 如对感知器的研究, 对视觉处理网络的研究, 对存储与记忆问题的研究等都取得一定的成功, 但遗憾的是, 这些成功一方面还远不够完善, 另一方面, 在对人脑作为一个整体的功能的解释上还几乎起不上任何帮助。科学家已经积累了大量关于大脑组成、大脑外形、大脑运转基本要素等

知识,但仍无法解答有关大脑信息处理的一些实质问题<sup>[40]</sup>。整体功能决不是局部功能的简单组合而是一个质的飞跃,人脑的知觉和认知等过程是包含着一个复杂的动态系统中对大量神经元活动进行整合的统一性行动。由于我们对人脑完整工作过程几乎没有什么认识,连一个稍微完善的令人可以接受的假设也没有,这造成神经网络研究始终缺乏一个明确的大方向。这方面如果不能有所突破,神经网络研究将始终限于模仿人脑局部功能的缓慢的摸索过程当中,而难以达到研究水平的质的飞跃。

## 2. 与之相关的数学领域的研究与发展

神经元的以电为主的生物过程在认识上一般采用非线性动力学模型,其动力学演变过程往往是非常复杂的,神经网络这种强的生物学特征和数学性质,要求有更好的数学手段<sup>[42-43]</sup>。而我们知道,对解决非线性微分方程这样的问题,稍微复杂一些的便没有办法利用数学方法求得完整的解<sup>[41]</sup>。这使得在分析诸如一般神经网络的自激振荡、稳定性、混沌等问题时常常显得力不从心,更不用说,当我们面对人脑这样的由成千上万个神经网络子系统组成的巨系统,而每个子系统(具有某种特定功能)又可能由成千上万个神经元组成,每个神经元本身是一个基本的非线性环节。因此可以认为,当今神经网络理论的发展,已经客观要求有关数学领域必须有所发展,并大胆预期一种更简洁、更完善和更有效的非线性系统表达与分析的数学方法是这一领域数学发展的主要目标之一。

## 3. 神经网络应用的研究与发展

从神经网络发展过程来看,理论研究经常走在前列,有时会超出实际使用阶段。虽然说理论研究和实际应用可以相辅相成,但实际需求总是科技发展的主要推动力。目前,在神经网络实用上,虽然有不少实际应用成果报道,如智能控制、模式识别、机器人控制及故障诊断等<sup>[50-52]</sup>。但真正成熟的应用还比较少见。

## 4. 神经网络硬件的研究与发展

要真正实现神经网络计算机,则神经网络芯片设计与生产技术必须有实质性的进展。目前,在单片上集成数百个神经元的制作技术已经没有困难,但这种水平与神经网络实际应用的要求尚有较大距离。神经网络硬件设计和理论研究相比,要落后很多。因此,这也是神经网络研究发展的重要方向之一。在这方面,光学技术是实现神经网络及神经计算机的一个比较理想的选择。因为光学技术具有非常好的固有特性,主要体现在:高驱动性、较高的通信带宽、以光速并行传递信息等。虽然光学神经计算机实现技术目前还不成熟,其商品化大规模实现还有待时日,但一些光学神经元器件、光电神经计算机研究已表现出广阔的发展和应用潜力,并引起相应领域的充分关注。

## 5. 新型神经网络模型的研究

为了推动神经网络理论的发展,除了期待神经生理学等研究突破外,将神经网络与其他理论交叉结合,研究新型神经网络模型,也是神经网络研究发展方向之一。如将之与混沌理论相结合产生的混沌神经网络理论<sup>[34-35]</sup>;再如将量子力学与神经网络的结合,研究量子神经网络<sup>[44-47]</sup>,实现功能强大的量子神经计算就是目前神经网络研究的热点之一。

人工神经网络的研究涉及相当广泛的应用数学工具,除线性代数、集合论、微分与差分方程、状态空间及数值分析等基本方法外,往往需要运用非线性动态系统稳定性理论、概率统计方法和随机过程、优化理论、非线性规划、自适应控制及信息论的初步概念。从研究方法上看,目前尚未形成统一、完整的理论体系。各种网络模型及算法的形成、构造、设计及性能评估都是只就具体问题进行分析,主要依赖计算机模拟的实验结果,大多数还不能给出严密、科学的一般规律和方法。为推动人工神经网络研究的进展,迫切需要宏观的理论指导。另一方面,为开展应用系统的研究,迫切需要硬件制作技术的新突破,目前,在这方面的进展还遇到不少困难。

虽然神经网络理论研究有着非常广阔的发展前景,但历来这个领域的研究就是既充满诱

惑又不乏挑战的。没有人能肯定它的发展不会再经受挫折,也没有人知道一旦成功实现其最终目标会给科技界带来多大的辉煌和巨变。有理由相信,只要坚持不懈地努力,来自神经网络理论研究的一些新理论和新方法必定会给 21 世纪科学研究带来源源不断的动力。

## 1.2 人工神经网络的基本模型及其功能

### 1.2.1 人工神经元的模型<sup>[49]</sup>

人工神经元的研究源于脑神经元学说,19 世纪末,在生物、生理学领域,Waldegger 等人创建了神经元学说。人们开始认识到,复杂的神经系统是由数目繁多的神经元组合而成。大脑皮层神经元的类型有多种,它的基本结构如图 1-1 所示。

神经元由细胞及其发出的许多突起构成。细胞体内有细胞核,突起的作用是传递信息。作为引入输入信号的若干个突起称为“树突”或“晶枝”(dendrite),而作为输出端的突起只有一个称为“轴突”(axon)。

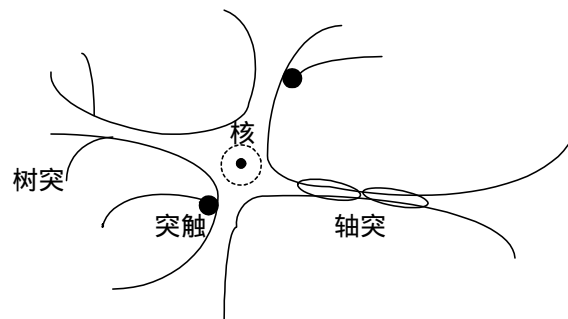


图 1-1 神经元结构示意图

树突是细胞体的延伸部,它由细胞体发出后逐渐变细,全长各部位都可与其它神经元的轴突末梢相互联系,形成所谓“突触”(synapse)。在突触处两神经元并未连通,它只是发生信息传递功能的结合部。联系界面之间隙约为 $(15 \sim 50) \times 10^{-9} \text{m}$ 。突触可分为兴奋性与抑制性两种类型,它相应于神经元之间耦合的极性。每个神经元的突触数目有所不同,最高可达  $10^5$  个,各神经元之间的连接强度和极性有所不同,并且都可调整,基于这一特性,人脑具有存储信息的功能。

对于这样一种多输入、单输出的基本单元可以进一步从生物化学、电生物学、数学等方面给出描述其功能的模型。从信息处理观点考察,为神经元构造了各种形式的数学模型。下面初步介绍经典的 McCulloch-Pitts 模型,图 1-2 给出这种模型的示意结构。对于第  $j$  个神经元,接受多个其它神经元的输入信号  $x_i$ 。各突触强度以实系数  $w_{ij}$  表示,这是第  $i$  个神经元对第  $j$  个神经元作用的加权值。利用某种运算把输入信号的作用结合起来,给出它们的总效果,称为“净输入”,以  $N_{etj}$  或  $I_j$  表示。净输入表达式有多种类型,其中,最简单的一种形式是线性加权求和,即:

$$N_{etj} = \sum w_{ij} x_i \quad (1-1)$$

此作用引起神经元  $j$  的状态变化,而神经元  $j$  的输出  $y_j$  是其当前状态的函数。

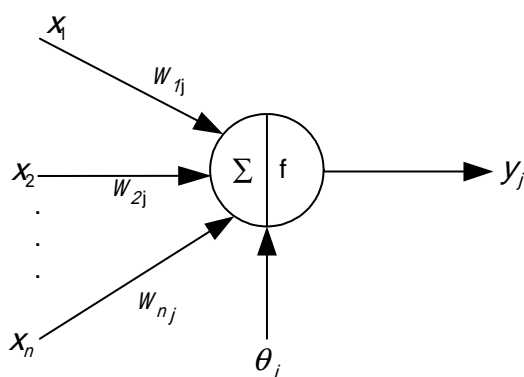


图 1-2 M-P 模型示意结构

M-P 模型的数学表达式为：

$$y_j = \text{sgn} \left( \sum_i w_{ij} x_i - \theta_j \right) \quad (1-2)$$

式中， $\theta_j$  为阈值，sgn 是符号函数，当净输入超过阈值时， $y_j$  取+1 输出，反之为-1 输出。如果考虑输出与输入的延时作用，表达式可修正为

$$y_j(t+1) = \text{sgn} \left( \sum_i w_{ij} x_i(t) - \theta_j \right) \quad (1-3)$$

利用大量神经元相互连接组成的人工神经网络，将显示出人脑的若干特征，人工神经网络也具有初步的自适应与自组织能力。在学习或训练过程中改变突触权重  $w_{ij}$  值，以适应周围环境的要求。同一网络因学习方式及内容不同可具有不同的功能。人工神经网络是一个具有学习能力的系统，可以发展知识，以至超过设计者原有的知识水平。通常，它的学习(或训练)方式可分为两种，一种是有监督(supervised)或称有导师的学习，这时利用给定的样本标准进行分类或模仿；另一种是无监督(unsupervised)学习或称无导师学习，这时，只规定学习方式或某些规则，而具体的学习内容随系统所处环境(即输入信号情况)而异，系统可以自动发现环境特征和规律性，具有更近似于人脑的功能。

### 1.2.2 人工神经网络的特性

人工神经网络与人脑以及冯·诺依曼计算机相比有如下特点<sup>[50]</sup>：

#### 1. 大规模并行处理

人脑神经元之间传递脉冲信号的速度远低于冯·诺依曼计算机的工作速度，前者为毫秒量级，后者的时钟频率通常可达  $10^8\text{Hz}$  或更高的速率。但是，由于人脑是一个大规模并行与串行组合处理系统，因而在许多问题上可以做出快速判断、决策和处理，其速度可以远高于串行结构的冯·诺依曼计算机。人工神经网络的基本结构模仿人脑，具有并行处理的特征，可以大大提高工作速度。

#### 2. 分布式存储

人脑存储信息的特点是利用突触效能的变化来调整存储内容，也即信息储存在神经元之间连接强度的分布上，存储区与运算区合为一体。虽然人脑每日有大量神经细胞死亡，但不影响大脑的功能，局部损伤可能引起功能衰退，但不会突然丧失功能。

冯·诺依曼计算机具有相互独立的存储器和运算器，知识存储与数据运算互不相关，只有通过人的编程给出指令使之沟通，这种沟通不能超越程序编写者的预想。元件的局部损伤或程序中的微小错误都可能引起严重的失常。

#### 3. 自适应(学习)过程

人类大脑有很强的自适应与自组织特性。后天的学习与训练可以开发许多各具特色的活动功能。如盲人的听觉和触觉非常灵敏，聋哑人善于运用手势，训练有素的运动员可以表现出非凡的运动技巧等等。

冯·诺依曼计算机强调程序编写，系统的功能取决于程序给出的知识和能力。显然，对于上述智能活动要加以总结并编制程序将十分困难。

人工神经网络也具有初步的自适应与自组织能力。在学习和训练过程中改变突触权重值，以适应周围环境的要求。同一网络因学习方式及内容不同可具有不同功能。人工神经网络是一个有学习能力的系统，可以发展知识，以致超过设计者原有的知识水平。

### 1.2.3 人工神经网络的基本功能<sup>[53]</sup>

人工神经网络是一种旨在模仿人脑结构及其功能的信息处理系统。因此，它在功能上具有某些智能特点。

#### 1. 联想记忆功能

由于神经网络具有分布存储信息和并行计算的性能，因此它具有对外界刺激和输入信息进行联想记忆的能力。这种能力是通过神经元之间的协同结构及信息处理的集体行为而实现的。神经网络通过预先存储信息和学习机制进行自适应训练，可以从不完整的信息和噪声干扰中恢复原始的完整的信息。这一功能使神经网络在图像复原、语音处理、模式识别与分类方面具有重要的应用前景。

联想记忆又分自联想记忆和异联想记忆两种：

自联想记忆 (Associative memory)：假定有  $m$  个样本矢量  $X^{(i)}$ ，其中  $i = 0, 1, 2, \dots, m-1$ ，

若网络输入  $X^j = X^k + \Delta$ ， $X^k$  表示第  $k$  个样本， $\Delta$  是由于噪声、干扰或图形缺损等

因素引起的偏差。要求输出  $Y = X^k$ ，也即去掉偏差使信号按样本复原。神经网络所具有的此种功能称为自联想记忆功能。

异联想记忆 (Hetero-associative memory)：与自联想记忆不同，它涉及两组样本，若样本  $X^k$  与样本  $Z^k$  一一对应，当具有偏差的输入信号为  $X^j = X^k + \Delta$  时，输出为

$Y = Z^k$ ，此联想为异联想功能。

#### 2. 分类与识别功能

神经网络对外界输入样本有很强的识别与分类能力。对输入样本的分类实际上是在样本空间找出符合分类要求的分割区域，每个区域内的样本属于一类。假定神经网络输入  $X$  有  $m$  类样本，样本元素为  $n$ ，输出  $y_k$  相应与  $m$  类样本之一， $k = 0, 1, 2, \dots, m-1$ ，输入  $X$  与输出  $Y$  之间关系如图 1-3 所示。

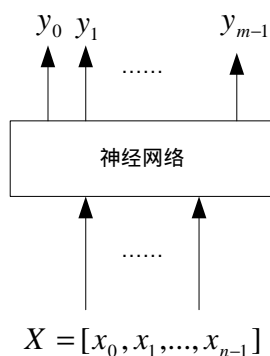


图 1-3 神经网络分类器

对于  $X \in R_j$  则：

$$y_k = \begin{cases} 1 & \text{当 } k = j \\ 0 & k \neq j \end{cases}$$

此关系表明，当输入样本与标准样本匹配时即可归类，神经网络完成分类功能。

### 3. 优化计算功能

优化计算是指在已知的约束条件下，寻找一组参数组合，使该组合确定的目标函数达到最小。将优化约束信息（与目标函数有关）存储于神经网络的连接权矩阵之中，神经网络的工作状态以动态系统方程式描述。设置一组随机数据作为起始条件，当系统的状态趋于稳定时，神经网络方程的解作为输出优化结果。优化计算在 TSP 及生产调度问题上有重要应用。

### 4. 非线性映射功能

在许多实际问题中，如过程控制、系统辨识、故障诊断、机器人控制等诸多领域，系统的输入与输出之间存在复杂的非线性关系，对于这类系统，往往难以用传统的数理方程建立其数学模型。神经网络在这方面有独到的优势，设计合理的神经网络通过对系统输入输出样本进行训练学习，从理论上讲，能够以任意精度逼近任意复杂的非线性函数。神经网络的这一优良性能使其可以作为多维非线性函数的通用数学模型。如文献[54]的小脑模型神经网络在线辨识算法，就是一典型的非线性映射。

#### 1.2.4 人工神经网络的应用领域<sup>[53]</sup>

1988 年，在 Darpa 的“神经网络研究报告”<sup>[55]</sup>中列举了各种神经网络的应用。其中第一个应用就是大约在 1984 年的自适应频道均衡器。这个设备在商业上取得了极大的成功。它用一个单神经元网络来稳定电话系统中长距离传输的声音信号。Darpa 报告还列出了其它一些神经网络在商业领域的应用，如单词识别器、过程监控器、声纳分类器、风险分析系统等。目前神经网络的应用领域正在不断扩大，它不仅广泛应用于工程、科学和数学领域，也可广泛应用于医学、商业、金融甚至于文学领域。文献[53]在这方面做了比较好的总结。

#### 1. 信息领域

神经网络作为一种新型智能信息处理系统，其应用贯穿信息的获取、传输、接收与加工利用等各个环节。

##### 信号处理

神经网络被广泛应用于自适应信号处理和非线性信号处理。前者如信号的自适应滤波、时间序列预测、谱估计、噪声消除等；后者如非线性滤波、非线性预测、非线性编码、调制/解调等。在信号处理方面神经网络有过许多成功应用的实例。第一个成功应用的实例就是电话线中回声的消除，其他还有雷达回波的多目标分类、运动目标的速度估计、多探测器信



息的融合等等。

#### 模式识别

模式识别涉及模式的预处理变换和将一种模式映射为其他类型的操作,神经网络在这两个方面都有成功的应用。神经网络不仅可以处理静态模式如固定图像、固定图谱等,还可以处理动态模式如视频图像、连续语音等。静态模式识别成功的例子有手写汉字的识别,人体的生物学识别,如指纹识别系统、脸庞识别系统、虹膜识别系统及故障模式识别<sup>[56]</sup>等;动态模式识别的成功实例有语音信号识别。目前电脑市场上随处可见的手写输入和语音输入系统进一步表明神经网络在模式识别方面的应用已经商品化。

#### 数据压缩

在数据传送与存储时,数据压缩至关重要。神经网络可对待传送(或存储)的数据提取模式特征,只将该特征传送(或存储),接收后(或使用)再将其恢复原始模式。

### 2. 自动化领域

20 世纪 80 年代以来,神经网络和控制理论相结合,发展为自动控制领域的一个前沿学科——神经网络控制。它是智能控制的一个重要分支,为解决复杂的非线性、不确定、不确定系统的控制问题开辟了一条新的途径。神经网络用于控制领域,已取得了以下主要进展。

#### 系统辨识

在自动控制问题中,系统辨识的目的是为了建立被控对象的数学模型。多年来控制领域对于复杂的非线性对象的辨识,一直未能很好的解决。神经网络所具有的非线性特性和学习能力,使其在系统辨识方面有很大的潜力,为解决具有复杂的非线性、不确定性和不确定对象的辨识问题开辟了一条有效途径。基于神经网络的系统辨识是以神经网络作为工具辨识对象的模型,利用其非线性特性,可建立非线性的静态或动态模型。如在动态非线性系统容错控制中,一般来说要进行非线性系统在线故障辨识,这时可以应用局部逼近神经网络 CMAC 或 RBF 进行在线快速故障学习<sup>[57]</sup>,并在此基础上进行系统控制规律的重构,实现容错控制。

#### 神经控制器

控制器在实时控制系统中起着“大脑”的作用,神经网络具有自学习自适应等智能特点,因而非常适用于控制器设计。对于复杂非线性系统,神经网络控制器所达到的控制效果明显好于常规控制器。近年来,神经网络控制器在工业自动化,航空航天以及智能机器人等领域有较为广泛的应用。

#### 智能监测

所谓智能监测一般包括干扰量的处理、传感器输入输出特性的非线性补偿、零点和量程的自动校正以及故障自动诊断等。这些智能检测功能可以通过传感元件和信号处理元件的功能集成来实现。如复合检测、特征提取及识别等,而这类信息处理问题正是神经网络的强项。在对综合指标的检测(例如对环境舒适度这类综合指标的检测)中,以神经网络作为智能监测中的信息处理元件便于对多个传感器的相关信息(如温度、湿度、风向和风速等)进行复合、集成、融合、联想等数据融合处理,从而实现单一传感器所不具备的功能。如文献[58]即是利用 BP 网络对多传感器数据进行融合,来提高电子设备故障诊断的准确率。

### 3. 工程领域

20 世纪 80 年代以来神经网络的理论研究已在众多的工程领域取得了丰硕的应用成果,汽车工程

汽车在不同状态参数下运行时,能获得最佳动力性与经济性的档位称为最佳档位。利用神经网络的非线性映射能力,通过学习优秀驾驶员的换挡经验数据,可自动提取蕴含在其中的最佳换挡规律。神经网络在汽车刹车自动控制系统中也有成功的应用,该系统能在给定刹车距离、车速和最大减速情况下,以人体感受到最小冲击实现平稳刹车,而不受路面坡度和车重的影响。随着国内外对能源短缺和环境污染问题日趋关切,燃油消耗率和排烟度愈来愈

受到人们的关注。神经网络在载重车柴油机燃烧系统方案优化中的应用,有效地降低了油耗和排烟度,获得了良好的社会效益。

#### 军事工程

神经网络同红外搜索与综合系统配合后可发现和跟踪飞行器。一个成功的例子是,利用神经网络检测空间卫星的动作状态是稳定、倾斜、旋转还是摇摆,正确率可达 95%。利用声纳信号判断水下目标是潜艇还是礁石是军事上常用的办法。借助神经网络的语音分类与信号处理上的经验对声纳信号进行分析研究,对水下目标的分辨率可达 90%。密码学研究一直是军事领域中的重要研究课题,利用神经网络的联想记忆特点可设计出密钥分散保管方案;利用神经网络的分类能力可提高密钥的破解难度;利用神经网络还可以设计出安全的保密开关,如语音开关、指纹开关等等。

#### 化学工程

20 世纪 80 年代中期以来,神经网络在制药、生物化学、化学工程等领域的应用与研究蓬勃发展,取得了不少成果。例如,在谱分析方面,应用神经网络在红外谱、紫外谱、折射光谱和质谱与化合物的化学结构间建立某种确定的对应关系方面的成功应用比比皆是。此外,还有将神经网络用于判定化学反应的生成物;用于判定钾、钙、硝酸、氯等离子浓度;用于研究生命体中的某些化合物的含量与其生物活性的对应关系等大量应用实例。

#### 水利工程

近年来,我国水利工程领域的科技人员已成功地将神经网络的方法用于水力发电过程辨识和控制、河川径流预测、河流水质分类、水资源规划、混凝土性能预估、拱坝优化设计、预应力混凝土桩基等结构损伤诊断、砂土液化预测、岩体可爆破分级及爆破效应预测、地下工程围岩分类、大坝等工程结构安全监测、工程造价分析、等许多实际问题中。

#### 制造工程

生产流程控制,产品设计与分析,过程和机器故障的诊断,可视质量监督,焊接质量分析,计算机芯片质量分析,机器性能分析,项目投标、计划与管理等,均是神经网络的应用领域。

#### 航空航天工程

在现代的交通调度管理系统中,神经网络有相当规模的应用。典型的如航空优化调度系统,就是利用神经网络的优化计算功能,实现航空运输资源的最佳调配,以提高航空运输能力,提高生产效益。另一方面,神经网络在航空航天设备上也有广泛应用。如高性能飞行器驾驶仪,飞行路径模拟,飞行控制系统,飞行器部件故障检测器等。

#### 气象工程

现代的天气预报,地球环境的气候变化与预测,都是一个非常复杂的综合的非线性系统,神经网络在气象模式识别,环境气候数据分析方面都有很好的应用。

### 4. 医学领域

#### 检测数据分析

许多医学检测设备的数据都是连续波形的形式,这些波的极性和幅值常常能够提供有意义的诊断结果。神经网络在这方面的应用非常普遍,一个成功的应用实例是用神经网络进行多通道脑电棘波的检测。很多癫痫病人常规治疗往往无效,但其早期发现可得益于脑电棘波检测系统。脑电棘波的出现通常意味着脑功能的某些异常,棘波的极性和幅值提供了异常的部位和程度信息,因而神经网络脑电棘波检测系统可用来提供脑电棘波的实时检测和癫痫的预报;另外,在智能人工呼吸系统中,也可以利用神经网络对呼吸机的参数进行设置<sup>[59]</sup>。

#### 生物活性研究

用神经网络对生物学检测数据进行分析,可提取致癌物的分子结构特征,建立分子结构和致癌活性之间的定量关系,并对分子致癌活性进行预测。分子致癌性的神经网络预测具有

生物学检测不具备的优点,它不仅可对新化合物的致癌性和致突变性预先作出评价,从而避免盲目投入造成浪费,而且检测费用低,可作为致癌物的大面积预筛的工具。

#### 医学专家系统

专家系统在医疗诊断方面有许多应用。虽然专家系统的研究与应用取得了重大进展,但由于知识“爆炸”和冯·诺依曼计算机的“瓶颈”问题使其应用受到严重挑战。以非线性并行分布式处理为基础的神经网络为专家系统的研究开辟了新的途径,利用其学习功能、联想记忆功能和分布式并行信息处理功能,来解决专家系统中的知识表示、获取和推理等问题取得了良好的效果。

#### 生物信息学

生物信息学是 20 世纪末发展起来的一极具发展潜力的新型学科。人类的基因中蕴涵着大量有用信息,利用神经网络可以对其蕴涵的信息进行识别与分类。从而得到许多有用的信息。如利用神经网络分析疾病与基因序列的关系等。

另外,神经网络在乳房癌细胞分析,EEG 和 ECG 分析,修复设计等方面均有应用。

### 5. 经济领域

#### 信贷分析

这类问题中,信用评估机构要针对不同申请公司的各自特点提出信用评价,判断失误的例子经常发生,给信贷机构带来巨大的损失。采用神经网络评价系统不仅评价结果具有较高的可信度,而且可以避免信贷分析人员的主观好恶和人情关系造成的错误。神经网络评价系统将公司贷款申请表中的关键数据编码为输入向量,将实际的信用情况作为输出评价,用数以千计的历史数据对网络进行训练后,可给出准确客观的评价结果。因此基于神经网络的评价系统在金融风险分析领域应用十分广泛。

#### 市场预测

市场预测问题可归结为对影响市场供求关系的诸多因素的综合分析,以及对价格变化规律的掌握。应用神经网络进行市场预测的一个实例是期货市场的神经网络预测。其做法是根据某期货市场每月平均期货价格、价格不定性和市场心理指标量等因素,建立较为准确可靠的市场模型。模型的训练数据覆盖了多年时间,它不仅能判断价格的未来走势,而且能在走势持续一段时间后预测到价格的反转。神经网络市场预测在股票走势预测中也有广泛的应用。

## 1.3 人工神经网络的基本要素

在人工神经网络设计及应用研究中,通常需要考虑三个方面的内容,即神经元功能函数、神经元之间的连接形式和网络的学习(训练)。

### 1.3.1 神经元功能函数

神经元在输入信号作用下产生输出信号的规律由神经元功能函数  $f$  (Activation Function) 给出,也称激活函数,或称转移函数,这是神经元模型的外特性。它包含了从输入信号到净输入、再到激活值、最终产生输出信号的过程。综合了净输入、 $f$  函数的作用。 $f$  函数形式多样,利用它们的不同特性可以构成功能各异的神经网络。

常见的神经元功能函数有:

#### 1. 简单线性函数

神经元功能函数  $f$  连续取值,输入  $x$  由连接矩阵  $W$  加权产生输出。

$$f(x) = x \quad (1-4)$$

#### 2. 对称硬限幅函数

这是一种非线性模型，输出只取二值，如+1 或-1(或 1 与 0 为硬限幅函数)，当净输入大于某一阈值 时，输出取+1，反之，输出取-1，这一作用可借助符号函数表示，如图 1-4(a)，此时有：

$$f(x) = \operatorname{sgn}(x - \theta) \quad (1-5)$$

### 3. sigmoid 函数 (s 形函数)

神经元输出是限制在两个有限值之间的连续非减函数，表达式可写为：

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (1-6)$$

式中 sigmoid 曲线由双曲正切函数构成，最大值与最小值分别取+1 与-1。如图 1-4(a)所示。

如果将双曲函数向上平移，即换用以下函数即单极性 S 型函数。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1-7)$$

则曲线之最大与最小值分别取 1 和 0，如图 1-4(b)所示。

常用的神经网络功能函数如表 1-2 所示。

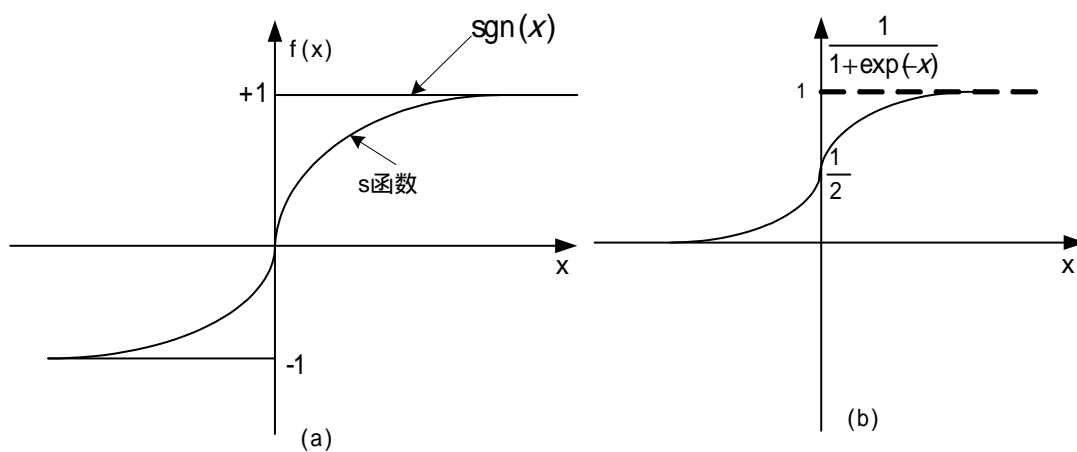


图 1-4 sigmoid 函数曲线

表 1-2 常用的神经网络功能函数

名称	输入输出函数关系	MATLAB 函数
硬限幅函数	$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$	hardlim
对称硬限幅函数	$f(x) = \operatorname{sgn}(x - \theta) = \begin{cases} -1 & x < \theta \\ +1 & x \geq \theta \end{cases}$	hardlims
线性函数	$f(x) = x$	purelin

饱和线性函数	$f(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$	satlin
对称饱和线性函数	$f(x) = \begin{cases} -1 & x < -1 \\ x & -1 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$	satlins
单极性 S 型函数	$f(x) = \frac{1}{1 + e^{-x}}$	logsig
双曲正切 S 型函数	$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$	tansig
正线性函数	$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$	poslin
竞争函数	$f(x) = \begin{cases} 1 & \text{点积最大神经元} \\ 0 & \text{所有其它神经元} \end{cases}$	compet

### 1.3.2 神经元之间的连接形式

前已述及，神经网络是一个复杂的互连系统，单元之间的互连模式将对网络的性质和功能产生重要影响。互连模式种类繁多，这里介绍一些典型的网络结构。

#### 1. 前向网络(前馈网络)

网络可以分为若干“层”，各层按信号传输先后顺序依次排列，第  $i$  层的神经元只接受第  $(i-1)$  层神经元给出的信号，各神经元之间没有反馈。前馈型网络可用一有向无环路图表示，如图 1-5。可以看出，输入节点并无计算功能，只是为了表征输入矢量各元素值。各层节点表示具有计算功能的神经元，称为计算单元。每个计算单元可以有任意个输入，但只有一个输出，它可送到多个节点作输入。称输入节点层为第零层。计算单元的各节点层从下至上依次称为第 1 至第  $N$  层，由此构成  $N$  层前向网络。(也有把输入节点层称为第 1 层，于是对  $N$  层网络将变为  $N+1$  个节点层序号。)

第一节点层与输出节点统称为“可见层”，而其他中间层则称为隐含层(hidden layer)，这些神经元称为隐节点。BP 网络就是典型的前向网络。

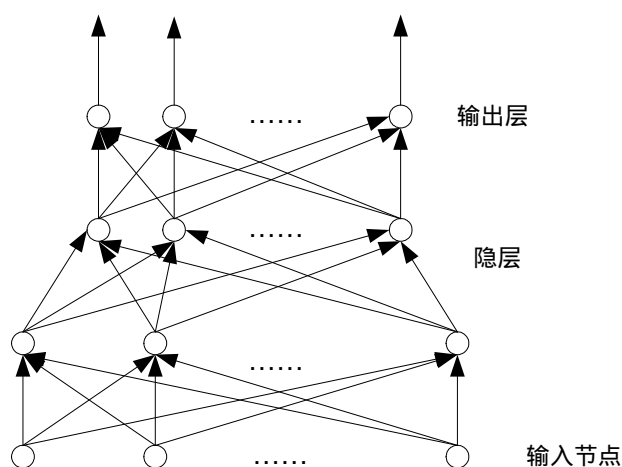


图 1-5 前向网络结构

## 2. 反馈网络

典型的反馈型神经网络如图 1-6(a)，每个节点都表示一个计算单元，同时接受外加输入和其它各节点的反馈输入，每个节点也都直接向外部输出。Hopfield 网络即属此种类型。在某些反馈网络中，各神经元除接受外加输入与其它各节点反馈输入之外，还包括自身反馈。有时，反馈型神经网络也可表示为一张完全的无向图，如图 1-6(b)。图中，每一个连接都是双向的。这里，第  $i$  个神经元对于第  $j$  个神经元的反馈与第  $j$  至  $i$  神经元反馈之突触权重相等，也即  $w_{ij} = w_{ji}$ 。

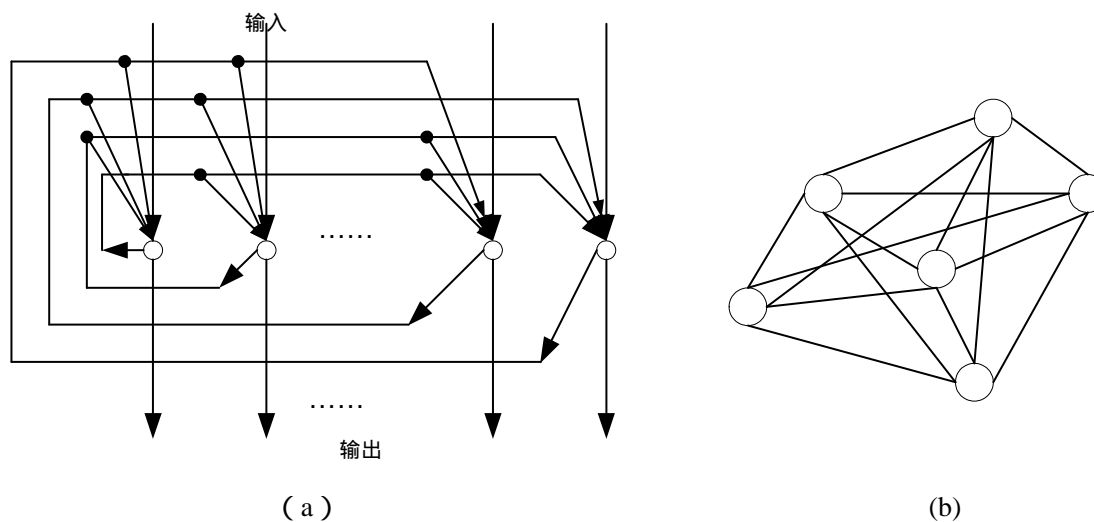


图 1-6 反馈网络

以上介绍了两种最基本的人工神经网络结构，实际上，人工神经网络还有许多种连接形式，例如，从输出层到输入层有反馈的前向网络，同层内或异层间有相互反馈的多层网络……等等。

### 1.3.3 人工神经网络的学习(训练)

学习功能是神经网络最主要的特征之一。各种学习算法的研究，在人工神经网络理论与实践发展过程中起着重要作用。当前，人工神经网络研究的许多课题都致力于学习算法的改进、更新和应用。

四十年代末，D. O. Hebb 首先提出了一种神经网络学习算法，称为 Hebb 规则。以此规则为基础，出现了多种形式的学习算法。下面初步介绍这一规则。

#### 1. Hebb 规则

在 Hebb 学习规则中，学习信号简单地等于神经元的输出：

$$y_j = f\left(\sum_{i=1}^n w_{ij} x_i\right) \quad (1-8)$$

权向量的调整公式为：

$$w_{ij}(t+1) = w_{ij}(t) + \eta f x_i = w_{ij}(t) + \eta y_j x_i \quad (1-9)$$

上式表明，权值调整量与输入输出的乘积成正比。显然，经常出现的输入模式将对权向量有较大的影响。在这种情况下，Hebb 学习规则需预先设置权饱和值，以防止输入和输出正负始终一致时出现权值无约束增长。

Hebb 学习规则代表一种纯前馈、无导师学习。该学习规则至今在各种神经网络模型中起作重要作用。典型的应用如利用 Hebb 规则训练线性联想器的权矩阵。如果假定网络的权矩阵初始化为 0，然后将  $m$  个输入输出对依此应用式 (1-9)，取学习率  $\eta=1$ ，得权矩阵  $W$ ：

$$W = \sum_{k=1}^m Y_k * X_k^T \quad (1-10)$$

假设输入向量  $X_k$  为标准正交向量（向量之间正交，每个向量的长度为单位长）。如果将  $X_j$  输入到网络，则网络产生的输出为：

$$O_k = W * X_j = \left[ \sum_{k=1}^m Y_k * X_k^T \right] * X_j = \sum_{k=1}^m Y_k (X_k^T X_j) \quad (1-11)$$

由于  $X_k$  为标准正交向量， $X_k^T * X_k = 1$ ，则  $O_k = Y_k$ （ $j = k$ ）； $X_k^T * X_j = 0$ （ $j \neq k$ ）。此时，网络的输出等于其相应的目标输出。也就是说，如果输入原型向量是标准正交向量，Hebb 学习规则就能为每个输入生成正确的输出结果。

但是当输入原型向量不是正交向量时，又将如何？假设输入向量  $X_k$  为单位向量但不是正交向量，则 (1-11) 式变为：

$$O_k = W * X_j = \left[ \sum_{k=1}^m Y_k * X_k^T \right] * X_j = \sum_{k=1}^m Y_k (X_k^T X_j) = Y_k + \sum_{j \neq k} Y_k (X_k^T X_j) \quad (1-12)$$

由于这些向量不是正交的，所以网络的输出有误差。误差的大小取决于原型输入模式之间的相关总和。

例 1-1 假设神经网络的原型输入输出向量为： $x_1 = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$ ， $y_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ ；

$x_2 = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$ ， $y_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 。请利用 Hebb 规则设计线性联想器。

解： 网络的权矩阵设计

可以验证这两个输入向量为标准正交向量。则网络的权矩阵为：

$$W = \sum_{k=1}^2 Y_k * X^T = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \end{bmatrix}$$

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

线性联想器验证

用上述两个原型输入验证该权矩阵有：

$$O_1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} = y_1 ;$$

$$O_2 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = y_2 \circ$$

网络的线性联想获得成功，实际输出与目标输出完全相同。对非标准正交的情况。可对输入向量进行规格化处理后，设计权矩阵，当采用原型模式进行输入联想时，一般说来其输出会接近目标输出，但与目标输出不能完全匹配。

## 2. 误差修正法学习算法

下面介绍另一种学习算法，称为误差修正法，权值的调整与网络的输出误差有关，它包括 学习规则、Widrow-Hoff 学习规则、感知器学习规则和误差反向传播的 BP(Back Propagation)学习规则等。此处仅介绍前面几种算法，BP 算法在本书第 2 章有详细分析。

感知器学习规则

1958 年 美国学者 Frank Rosenblatt 首次定义了一个具有单层计算单元的神经网络结构，取名为感知器(Perceptron)。如果包括输入层在内，应为两层。单计算节点感知器结构如图 1-7 所示。

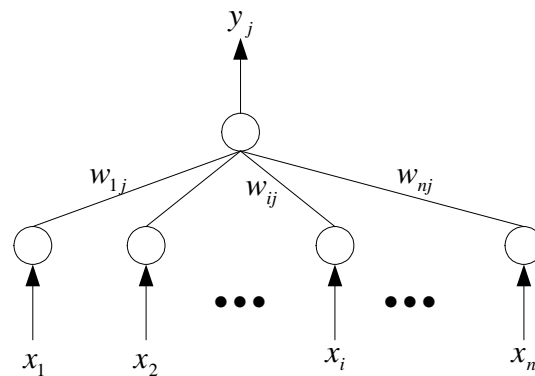


图 1-7 单计算节点感知器

在给定样本的条件下，首先随机设置初始权重值(包括  $w$  和  $b$  值)。然后，加入样本矢量。对于第  $j$  神经元，假设  $x_i$  为其输入矢量的第  $i$  元素，而  $w_{ij}$  是相应的权重值。如果期望输出为



$d_j$ ，而实际输出是  $y_j$ ，那么，在训练过程中， $w_{ij}$  的调整规则由下式给出：

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - y_j)x_i \quad (1-13)$$

或

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - y_j)x_i \quad (1-14)$$

对离散感知器，其中

$$y_j = f(W_j^T X - \theta) = \text{sgn}(W_j^T X - \theta) = \begin{cases} 1, & W_j^T X - \theta \geq 0 \\ -1 & W_j^T X - \theta < 0 \end{cases} \quad (1-15)$$

式中， $\eta$  为调整步幅系数， $\eta > 0$ ，离散感知器学习规则只适用二进制神经元，初始权值可取任意值。单计算节点感知器仅对线性可分问题具有分类能力，而无法解决线性不可分问题。如简单的线性感知器不可能实现“异或”的逻辑关系等。这也就是人工智能的创始人之一 Minsky 和 Papert 在《Perceptrons》一书中指出的简单的线性感知器的局限性。下面对这一问题作一简单分析。

(1) 设输入向量  $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ，对逻辑运算节点  $j$  的输出 (1-15) 式可写为：

$$y_j = \begin{cases} 1 & w_{1j}x_1 + w_{2j}x_2 - \theta \geq 0 \\ 0 & w_{1j}x_1 + w_{2j}x_2 - \theta < 0 \end{cases} \quad (1-16)$$

(2) 逻辑“与”和逻辑“异或”关系，其真值表如下：

表 1-3 真值表

逻辑“与”			逻辑“异或”		
$x_1$	$x_2$	$y$	$x_1$	$x_2$	$y$
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	0

对逻辑“与”，由式 (1-16) 得：

$$\begin{cases} w_{1j} * 0 + w_{2j} * 0 - \theta < 0 \\ w_{1j} * 0 + w_{2j} * 1 - \theta < 0 \\ w_{1j} * 1 + w_{2j} * 0 - \theta < 0 \\ w_{1j} * 1 + w_{2j} * 1 - \theta \geq 0 \end{cases} \Rightarrow \begin{cases} \theta > 0 \\ w_{2j} < \theta \\ w_{1j} < \theta \\ w_{1j} + w_{2j} \geq \theta \end{cases} \Rightarrow 0 < \theta \leq w_{1j} + w_{2j} < 2\theta \quad (1-17)$$

可见总存在一组  $w_{1j}, w_{2j}, \theta$  满足式 (1-17)，也就是说，单计算节点感知器对逻辑“与”问题具有分类能力。

对逻辑“异或”，由式 (1-16) 得：

$$\begin{cases} w_{1j} * 0 + w_{2j} * 0 - \theta < 0 \\ w_{1j} * 0 + w_{2j} * 1 - \theta \geq 0 \\ w_{1j} * 1 + w_{2j} * 0 - \theta \geq 0 \\ w_{1j} * 1 + w_{2j} * 1 - \theta < 0 \end{cases} \Rightarrow \begin{cases} \theta > 0 \\ w_{2j} \geq \theta \\ w_{1j} \geq \theta \\ w_{1j} + w_{2j} < \theta \end{cases} \Rightarrow \begin{cases} w_{1j} + w_{2j} \geq 2\theta \\ w_{1j} + w_{2j} < \theta \\ \theta > 0 \end{cases} \quad (1-18)$$

很明显，式 (1-18) 是无解的，不存在任何一组  $w_{1j}, w_{2j}, \theta$  满足式 (1-18)，也就是说，简单的线性感知器不可能实现“异或”的逻辑关系。

单层感知器只能求解线性问题，对求解非线性问题时，需要用到多层感知器，即网络应具有隐层，但对隐层神经元的学习规则尚无所知。就感知器学习规则来说，其权值的调整取决于网络期望输出与实际输出之差，而对各隐层节点来说，不存在期望输出，因而该学习规则对隐层权值调整不适用。此时需要用到误差反向传播的 BP 学习规则。有关 BP 学习规则将在第 2 章作详细分析。

学习规则

与离散感知器相对应的是连续感知器学习规则，也称为 学习规则。其权值调整为：

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - y_j)f'(W_j^T X)x_i \quad (1-19)$$

或

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - y_j)f'(W_j^T X)x_i \quad (1-20)$$

对 学习规则，其中  $y_j = f(W_j^T X)$ ， $f'(W_j^T X)$  是功能函数  $f(W_j^T X)$  的导数，要求功能函数可导，因此它只适用于有导师学习中，定义功能函数为连续函数的情况。

Widrow-Hoff 学习规则

1962 年，Bernard Widrow 和他的研究生 Marcian Hoff 引入了 ADLINE（自适应线性神经元）网络和 LMS（Least Mean Square，最小二乘法）算法的学习规则。

Widrow-Hoff 学习规则的权值修正关系为：

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - W_j^T X)x_i \quad (1-21)$$

或

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - W_j^T X)x_i \quad (1-22)$$

$\eta$  为学习速度，与算法的稳定性有关，文献[60]已从数学上证明二次函数的最大稳定学习速度为： $\eta < \frac{1}{\lambda_{\max}}$ ，其中  $\lambda_{\max}$  是有关输入向量的矩阵特征值。

例 1-2 假设神经网络的原型输入输出向量为： $x_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$ ， $y_1 = [-1]$ ；

$x_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$ ， $y_2 = [1]$ 。请利用 Widrow-Hoff 学习规则设计线性模式分类器。

解：(1) 最大稳态学习速度  $\eta$

通过求解输入相关矩阵的特征值可以求得  $\eta$ 。假定输入向量是以相等概率随机产生，则可以如下计算输入相关矩阵：

$$R = \frac{1}{2} \sum_{i=1}^2 x_i x_i^T = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}。R 的特征值为  $\lambda_1 = 1.0, \lambda_2 = 0.0, \lambda_3 = 2.0$ 。故最大稳态学习速度$$

为： $\eta < \frac{1}{\lambda_{\max}} = \frac{1}{2} = 0.5$ ，若保守些，可以取  $\eta = 0.4$ 。

(2) 权值学习训练

开始时可将所有权值设为 0，然后输入样本进行训练。直到网络权值最后稳定下来。

输入  $x_1, y_1$ ： $y(0) = W^T X = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = 0$ ， $e(0) = d(0) - y(0) = -1$ ；则新的

权矩阵为：

$$W(1) = W(0) + \eta e(0) x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 0.4(-1) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.4 \\ 0.4 \\ 0.4 \end{bmatrix}；$$

下一次输入  $x_2, y_2$ ： $y(1) = W^T X = \begin{bmatrix} -0.4 & 0.4 & 0.4 \end{bmatrix} \bullet \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = -0.4$ ，

$e(1) = d(1) - y(1) = 1 - (-0.4) = 1.4$ ；则新的权矩阵为：

$$W(2) = W(1) + \eta e(1) x_2 = \begin{bmatrix} -0.4 \\ 0.4 \\ 0.4 \end{bmatrix} + 0.4(1.4) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.16 \\ 0.96 \\ -0.16 \end{bmatrix}；$$

下一次输入  $x_1, y_1$ ： $y(2) = W^T X = \begin{bmatrix} 0.16 & 0.96 & -0.16 \end{bmatrix} \bullet \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = -0.64$ ，

$e(2) = d(2) - y(2) = -1 - (-0.64) = -0.36$ ；则新的权矩阵为：

$$W(3) = W(2) + \eta e(2) x_2 = \begin{bmatrix} 0.16 \\ 0.96 \\ -0.16 \end{bmatrix} + 0.4(-0.36) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.016 \\ 1.104 \\ -0.016 \end{bmatrix}$$

按此过程经多次训练

后，算法将收敛于： $W(\infty) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ 。

### (3) 线性分类器验证

用上述两个原型输入验证该权矩阵有：

$$O_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = [-1] = y_1; \quad O_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = [1] = y_2。$$

以上几种学习规则都属于监督类型的学习。可以将这类学习过程用图 1-8 表示。图中，样本训练数据加到网络输入端，同时将相应的期望输出与网络输出相比较得到误差信号，以此控制权重连接强度的调整，经计算至收敛后给出确定的  $w$  值。当样本情况发生变化时，经学习可修正  $w$  值以适应新的环境。

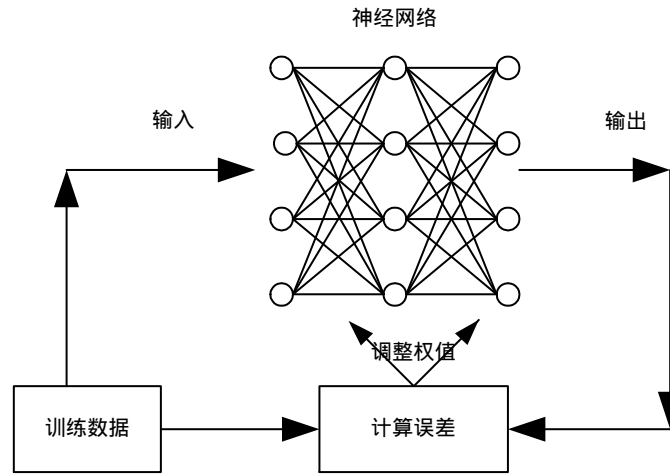


图 1-8 神经网络学习过程示意

### 3. 胜者为王(Winner-Take-All)学习规则

对于无监督学习的情况，事先不给定标准样本，直接将网络置于“环境”之中，学习(训练)阶段与应用(工作)阶段成为一体。此时，学习规律服从连接权重  $w$  的演变方程

$$\frac{dw}{dt} = f[w, x] \quad (1-23)$$

选定初始值  $w_0$  之后，由于环境不断改变， $x, w$  随之逐渐改变，对于平稳环境  $w$  可达稳定状态。如果环境发生变化， $w$  也随之改变。这种边学习边工作的特征与人脑学习过程更相似。此处介绍的胜者为王学习规则，一般将网络的某一层定为竞争层，对于一个特定的输入  $X$ ，竞争层的所有  $p$  个神经元均有输出响应，其中响应值最大的神经元为竞争中获胜神经元，即：

$$W_m^T X = \max_{i=1,2,\dots,p} (W_i^T X) \quad (1-24)$$

只有获胜神经元才有权调整其权向量  $W_m$ ，调整量为：

$$\Delta W_m = \alpha(X - W_m) \quad (1-25)$$

式中  $\alpha \in (0,1]$  是学习常数，一般其值随着学习的进展而减小。由于两个向量的点积越大，

表明两者越相似，所以调整获胜神经元权值的结果是使  $w_m$  进一步接近当前输入  $X$ 。显然，当下次出现与  $x$  相象的输入模式时，上次获胜的神经元更易获胜。在反复的竞争学习过程中，竞争层的各神经元所对应的权向量被逐渐调整为输入样本空间的聚类中心。

在有些应用中，如自组织神经网络，以获胜神经元为中心定义一个获胜邻域，除获胜神经元要调整权值外，邻域内的其他神经元也不同程度地调整各自的权值。有关胜者为王的竞争学习规则，在本书的第 7 章有更详细分析。

对以上介绍的神经网络常规学习算法的总结，如表 1-4 所示。

表 1-4 神经网络常规学习算法

学习规则	权值调整	权值初始 化	学习方 式	功能函 数
Hebb 规则	$\Delta w_{ij} = \eta f(W_j^T X) x_i$	0	无导师	任意
离散感知器	$\Delta w_{ij} = \eta (d_j - \text{sgn}(W_j^T X - \theta)) x_i$	任意	有导师	二进制
	$\Delta w_{ij} = \eta (d_j - f(W_j^T X)) f'(W_j^T X) x_i$	任意	有导师	连续
Widrow-Hoff	$\Delta w_{ij} = \eta (d_j - W_j^T X) x_i$	任意	有导师	任意
Winner-Take-All	$\Delta W_m = \alpha (X - W_m)$	随机， 归一化	无导师	连续

## 本章小结

本章介绍人工神经网络的基础性知识。主要包括三个方面内容：一是人工神经网络发展的历史及现状，介绍了神经网络从启蒙到低潮，再到重新复兴的曲折发展历程，并对神经网络的未来发展作出展望；二是人工神经网络基本概念、特性及应用的介绍；三是重点介绍了人工神经网络的三大基本要素，神经网络的结构形式、功能函数及学习规则。通过对本章的学习，使读者对神经网络这门学科有一个整体的认识，为以下各种神经网络的学习打下一个基础。

## 参考文献：

- [1] 董军, 胡上序. 混沌神经网络研究进展和展望. 信息与控制, 1997, 26(5): 360-368.
- [2] McCulloch W S, Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 1943, 10(5): 115-133.
- [3] Hebb D O. The Organization of Behavior. New York: Wiley, 1949.
- [4] Eccles J C. Cholinergic and Inhibitory Synapses in A Pathway from Motor-axon Collaterals to Motorneurons. Journal Physiology, 1954, 126: 524.
- [5] Rosenblatt F. The Perception: A Probabilistic Model for Information storage and Organization in the Brain. Psychological Review, 1958, 65: 386-408.
- [6] Widrow B, Hoff M E. Adaptive Switching Circuits. IRE WESCON convention record: part 4. Computers: Machine Systems, Los Angeles: 96-104. 1960
- [7] Grossberg S. On the Serial Learning of Lists. Bio-science, 1969, 10(4): 201-253.
- [8] Grossberg S. Some Networks that Can Learn, Remember and Reproduce any Number of Compialted Space-time Patterns. , study Applied Mathematics

- 1970,11(49):135-166.
- [9]Willshaw D J, Buneman O P. A holographic Neural networks, Nature,1969, 222:960.
- [10]Nilsson N J. Learning Machines: Foundations of Trainable Pattern Classifying Systems. New York: McGraw-hill, 1965.
- [11]Carpenter G.A, Grossberg S. The ART of adaptive pattern recognition by self-organizing neural network, Trans. IEEE on Computer,1988,15(1):77-88.
- [12]Kohonen T. Self-organization and associative memory, New York: third Edition Spring-Verlag,1989.
- [13]Hopfield J. Neural networks and physical systems with emergent collective computer abilities, Proc. Natl. Acad. Sci., USA, 1982, 79(6):2554-2558.
- [14]Hopfield J. Computing with neural circuits: A Model, Science, 1986,233:625-633.
- [15]Marr D. Vision and neural networks. San Francisco: W H Freeman,1982.
- [16]Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by Simulated Annealing.Science,1983,220(4598):671-681.
- [17]Hinton G E, Sejnowski T J, Ackley D H. Boltzmann Machines: Constraint Satisfaction Networks that Learn. Carnegie Mellon University, Tech, Report CMU-CS-84-119,1984.
- [18]Poggio T. An analog model of Computation for ill-posed Problems of Early Vision. Artificial Intelligence Lab Memo,783,MIT,1984.
- [19]Poggio T. Computational Vision and Regularization Theory. Nature, 1985,21(3):314-319.
- [20]McClelland J.L., Rumelhart D.E. Exploration in parallel distributed processing, A Handbook of Models, Programs, and Exercises, Cambridge: MIT Press, 1986.
- [21]Chua L O,Yang L. Cellular Neural Networks: Application, IEEE Transactions on Circuits and Systems,1988,28(35):1257-1272.
- [22]Chua L O,Yang L. Cellular Neural Networks: Theory. IEEE Transactions on Circuits and Systems, 1988, 28(35):1237-1290.
- [23]Kosko B. Adaptive Bidirectional Associative Memories. Applied Optical, 1987, 26(23):4667-4680.
- [24]Kosko B. Bidirectional Associative Memories. IEEE Transactions On Man, System and Cybernetics. 1988, 26(18):49-59.
- [25]廖晓昕。 细胞神经网络的数学理论 ( I ) ( II ) 中国科学 ( A 辑 ), 1994 , 24 ( 9 ): 902-910 ; 24 ( 10 ): 1037-1046。
- [26]Jenkins B K, A R Tanguay. Jr Optical Architectures for Neural Network Implementation, Handbook of Neural Computing and Neural Networks. Boston: MIT Press, 1995:673-677.
- [27]McAulay A D, Wang, Ma C. Optical Hetero-associative Memory Using Spatial Light Re-broadcasters. Applied Optical, 1990, 29(14):2067-2073.
- [28]阮昊, 陈述春, 戴凤妹, 千福熹。 利用电子俘获材料实现 IPA 神经网络。 光学学报, 1997, 17 ( 6 ): 766-771。
- [29]Narendra K, Parthasarathy K. Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks, March 1990, 1(1): 4-27.

- [30]戴先中, 刘军, 冯纯伯. 连续非线性系统的神经网络 阶逆系统控制方法. 自动化学报, 1998, 24(4): 463-468.
- [31]Miller W T, Real-time Application of Neural Networks for Sensor- based Control of Robbts, With Vision. IEEE, Transactions on Man, System and Cybernitics, 1989, 27(19):825-831.
- [32]Bulsari A. Some Analytical Solutions to the General Approximation Problem for Feedback Neural Networks. Neural networks, 1993, 28(6):991-996
- [33]Aibara K. Chaotic Neural Networks. Physical letter A, 1991, 158(8):373-376.
- [34]Inoue M, Nagayoshi A. A Chaos Neural-computer. Physical Letter A, 1991, 158(8):373-376.
- [35]Inoue M, Nakamoto K. Epilepsy in a Chaos Neural -computer Model, Chaos in Biology and Medicine, SPIE 1993, 236,:77-84.
- [36]Satoru Isaka. On neural approximation of fuzzy system. In: Proceedings of INCNN ' 92, New York: IEEE, 1992:1263-1268.
- [37]Jokinen Petri A. On the relations between radial basis function networks and fuzzy system[C]. In: Proceedings of INCNN ' 92, New York: IEEE, 1992, 1220-1225.
- [38]Okada Hiroyuki. Initializing multilayer neural network with fuzzy logic. In: Proceedings of INCNN ' 92, New York: IEEE, 1992, 1239-1244.
- [39]Cai yaling, Hon Keung Kwan. A fuzzy neural network with fuzzy classification. In : proceedings of ASME SCI ' 94, Wuhan: Press of Huazhong University of Science and Technology, 1994:894-899.
- [40]胡守仁, 余少波, 戴葵, 神经网络导论. 长沙: 国防科技大学出版社, 1992.
- [41]冯纯伯, 费树岷. 非线性控制系统分析与设计. 北京: 电子工业出版社, 1998.
- [42]Simon Haykin 著, 叶世伟, 史忠植译, 神经网络原理, 机械工业出版社, 2004 年 5 月, 北京.
- [43]廖晓昕. 论 Hopfield 神经网络中物理参数的数学内蕴, 中国科学(A 辑), 2003, 33(2): 127-136.
- [44]Perus M. Ecimovic P. Memory and pattern recognition in associative neural networks, International Journal of Applied Science and Computation, 1998(4):283-310.
- [45]Ventura D. and Martinez R. Quantum associative Memory, Information Sciences, 2000,124:147-148.
- [46]Ventura D. Quantum computing and neural information processing, Information Sciences, 2000,128:273-296.
- [47]解光军, 杨俊安, 庄镇泉, 基于量子双缝干涉实验的神经网络模型, 模式识别与人工智能, 2003, 16(1): 28-33.
- [48]朱大奇, 人工神经网络研究现状及其展望, 江南大学学报, 2004, 3(1): 103-110.
- [49]朱大奇, 陈小平, 计算机过程控制技术, 南京, 南京大学出版社, 2001.12.
- [50]Rovithakis G.A., Gaganis V.I., Perrakis S.E., and Christodoulou M.A., Real-time control of manufacturing cells using dynamic neural networks, Automatica, 1999,35(1):139-149.
- [51]Tao T. and Su S.F., CMAC-based online learning for robust controllers, in Proc. Automatic Control Conference, Apr. 1998, pp.447-452.
- [52]朱大奇, 航空电子设备故障诊断新技术研究, 博士论文, 南京航空航天大学, 2002.7.

- [53] 韩力群, 人工神经网络理论、设计及应用, 化学工业出版社, 2002 年 1 月, 北京。
- [54] Shun-Feng S., Ted T., and Hung T.H., Credit assigned CMAC and its application to online learning robust controllers, IEEE Trans. On Systems, Man, and Cybernetics----Part B: Cybernetics, vol. 33, no.2, pp. 202-213, 2003.
- [55] Darpa, Neural network study, Lexington, MA:MIT Lincoln Laboratory, 1988.
- [56] Daqi zhu, Yangqing and Wuzhao li, Blind fault diagnosis algorithm for integrated circuit based on the CPN neural networks, Lecture Notes in Computer Science 3174, pp.619-626, Springer-Verlag Berlin Heidelberg 2004.
- [57] 朱大奇, 张伟, 基于平衡学习的 CMAC 神经网络非线性辨识算法, 控制与决策, 2004, 19 ( 12 ) :1425-1428。
- [58] Zhu Daqi, Yu Shenglin, Neural Networks Data Fusion Algorithm of Electronic Equipment Fault Diagnosis, The 5<sup>th</sup> World Congress on Intelligent Control and Automation(WCICA2004), june, 14-18, 2004, HangZhou, China. PP.1815-1818.
- [59] Shunshoku K., Zi-Jiang Y. and Kiyoshi W., Estimation of Pulmonary Elastance Based on RBF Expression, Lecture Notes in Computer Science 3174, pp.507-512, Springer-Verlag Berlin Heidelberg 2004.
- [60] Martin T.H., Howard B.D., Mark H.B.著, 戴葵等译, 神经网络设计, 机械工业出版社, 2002 年 9 月, 北京。
- [61] Jinde Cao, Global exponential stability and periodic solutions of delayed cellular neural network, Journal of Computer and System Science, 2000, 60:38-46.
- [62] Jinde Cao, Global exponential stability of Hopfield neural network, International Journal of System Science, 2001, 32(2):233-236.

### 思考题

1. 请介绍人工神经网络的特征与功能。
2. 简要介绍离散感知器学习规则和  $\delta$  学习规则。
3. 简要介绍 Widrow-Hoff 学习规则。
4. 请简要介绍常见的四种神经网络功能函数。
5. 为什么单计算节点感知器不能实现“异或”的逻辑关系？
6. 设有某神经网络的功能函数为  $f(net) = \text{sgn}(net)$ , 学习率为  $\eta = 1$ , 初始权向量为

$$W(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{ 两对输入样本为: } X_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, d_1 = -1; X_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, d_2 = 1。 \text{ 试}$$

用感知器学习规则对以上样本进行 4 次训练, 并写出每步的输出结果和权向量。

7. 设有某神经网络的功能函数为双极性  $s$  型函数, 学习率为  $\eta = 0.2$ , 初始权向量为

$$W(0) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \text{ 两对输入样本为: } X_1 = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}, d_1 = -1; X_2 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}, d_2 = 1。$$

试用  $\delta$  学习规则和 Widrow-Hoff 学习规则对以上样本进行训练, 并写出前 3 步的输出结果和权向量。



## 第2章 BP 误差反传神经网络

BP(Error Back Propagation Network)神经网络是目前应用最为广泛和成功的神经网络之一。它是在 1986 年,由 Rumelhart 和 Mclelland 提出的。是一种多层网络的“逆推”学习算法。其基本思想是,学习过程由信号的正向传播与误差的反向传播两个过程组成。正向传播时,输入样本从输入层传入,经隐层逐层处理后,传向输出层。若输出层的实际输出与期望输出(教师信号)不符,则转向误差的反向传播阶段。误差的反向传播是将输出误差以某种形式通过隐层向输入层逐层反传,并将误差分摊给各层的所有单元,从而获得各层单元的误差信号,此误差信号即作为修正各单元权值的依据。这种信号正向传播与误差反向传播的各层权值调整过程,是周而复始地进行。权值不断调整过程,也就是网络的学习训练过程。此过程一直进行到网络输出的误差减少到可以接受的程度,或进行到预先设定的学习次数为止。

### 2.1 BP 神经网络模型及其学习算法<sup>[1-4]</sup>

#### 2.1.1 BP 网络结构

图 2-1 是 BP 网络的结构图。它有输入层、输出层和中间层(或称隐层)组成。

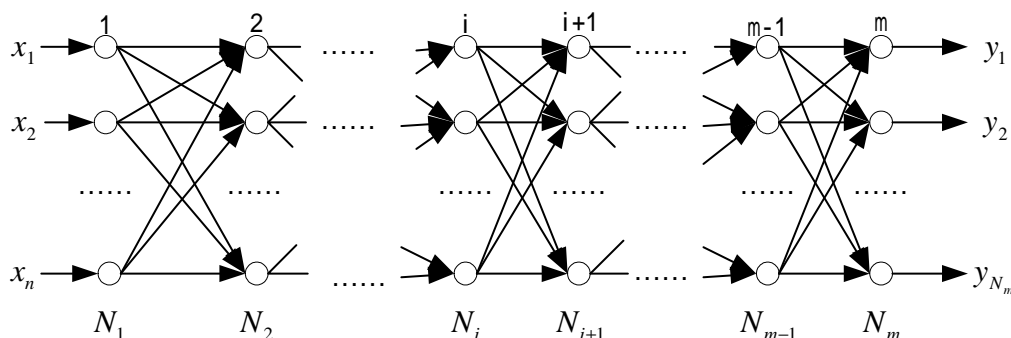


图 2-1 BP 网络的结构图

其中： $x_i$ ：神经网络输入； $y_i$ ：神经网络实际输出； $d_i$ ：神经网络期望输出； $w_{ijk}$ ：第  $i$

层第  $j$  个神经元到第  $i+1$  层第  $k$  个神经元连接权值； $o_{ij}$ ：第  $i$  层第  $j$  个神经元输出； $\theta_{ij}$ ：

第  $i$  层第  $j$  个神经元阈值； $net_{ij}$ ：第  $i$  层第  $j$  个神经元总输入； $N_i$ ：第  $i$  层神经元节点数。

#### 2.1.2 BP 网络学习算法

##### 1. 标准 BP 算法

以图 2-1 的 BP 网络的结构图为例,来推导标准 BP 算法。

**BP 网络前向传播计算：**

$$net_{ij} = \sum_{k=1}^{N_{i-1}} O_{(i-1)k} * W_{(i-1)kj} \quad (2-1)$$

$$O_{ij} = f_s(net_{ij}) = \frac{1}{1 + \exp[-(net_{ij} - \theta_{ij})]} \quad (2-2)$$

**BP 网络后退算法 (BP 算法)**

基本思想：如果神经元  $j$  在输出层，则  $O_{ij}$  就是网络的实际计算输出，记为  $y_j$ ，通过  $y_j$  与所期望的输出  $d_j$  之间的误差反向传播来修改各权值。

$$\text{误差定义：} e_j = d_j - y_j \quad (2-3)$$

$$\text{网络目标函数为：} E = \frac{1}{2} \sum_j (d_j - y_j)^2 \quad (2-4)$$

网络的权值沿  $E$  函数梯度下降的方向修正：

$$\Delta W_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}} \quad \text{其中 } 0 < \eta < 1 \text{ 为学习效率} \quad (2-5)$$

对此计算要使其能直接编制程序进行计算，必须求出  $\Delta W_{ijk}$  与神经元输出之间的递推关系：

$$\Delta W_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}} = -\eta \frac{\partial E}{\partial net_{(i+1)k}} * \frac{\partial net_{(i+1)k}}{\partial w_{ijk}} = \eta \delta_{ik} \frac{\partial net_{(i+1)k}}{\partial w_{ijk}} \quad (2-6)$$

$$\text{其中：} \delta_{ik} = -\frac{\partial E}{\partial net_{(i+1)k}}$$

$$1) \frac{\partial net_{(i+1)k}}{\partial w_{ijk}} \text{ 求解：}$$

$$\frac{\partial net_{(i+1)k}}{\partial w_{ijk}} = \frac{\partial}{\partial w_{ijk}} \left( \sum_{h=1}^{N_i} O_{ih} * W_{ihk} \right) = O_{ij} \quad (2-7)$$

$$\Delta W_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}} = -\eta \delta_{ik} O_{ij} \quad (2-8)$$

2) 计算  $\delta_{ik}$ ：

$$\delta_{ik} = -\frac{\partial E}{\partial net_{(i+1)k}} = -\frac{\partial E}{\partial O_{(i+1)k}} * \frac{\partial O_{(i+1)k}}{\partial net_{(i+1)k}} \quad (2-9)$$

$$\text{A. } \frac{\partial O_{(i+1)k}}{\partial net_{(i+1)k}} = f'(net_{(i+1)k}) \text{ 由 } f'(x) = \frac{e^{-(x-f)}}{[1 + e^{-(x-f)}]^2} = f(x)[1 - f(x)] \text{ 得：}$$

$$\begin{aligned} \frac{\partial O_{(i+1)k}}{\partial net_{(i+1)k}} &= f'(net_{(i+1)k}) = f(net_{(i+1)k})(1 - f) \\ &= O_{(i+1)k}(1 - O_{(i+1)k}) \end{aligned} \quad (2-10)$$

$$B. \frac{\partial E}{\partial O_{(i+1)k}} :$$

$$i) \text{ 当 } O_{(i+1)k} \text{ 为输出节点时: } \frac{\partial E}{\partial O_{(i+1)k}} = y_k - d_k ; [E = \frac{1}{2} \sum_j (d_j - y_j)^2] \quad (2-11)$$

$$\begin{aligned} \delta_{ik} &= -\frac{\partial E}{\partial net_{(i+1)k}} = -\frac{\partial E}{\partial O_{(i+1)k}} * \frac{\partial O_{(i+1)k}}{\partial net_{(i+1)k}} = (d_k - y_k) O_{(i+1)k} (1 - O_{(i+1)k}) \\ &= (d_k - y_k) y_k (1 - y_k) = (d_k - O_{mk}) O_{mk} (1 - O_{mk}) \end{aligned} \quad (2-12)$$

ii) 当  $O_{(i+1)k}$  为隐层节点时 :

$$\frac{\partial E}{\partial O_{(i+1)k}} = \sum_{h=1}^{N_{i+2}} \frac{\partial E}{\partial net_{(i+2)h}} * \frac{\partial net_{(i+2)h}}{\partial O_{(i+1)k}} = -\sum_{h=1}^{N_{i+2}} \delta_{(i+1)h} w_{(i+1)kh} \quad (2-13)$$

$$net_{ij} = \sum_{k=1}^{N_{i-1}} O_{(i-1)k} * w_{(i-1)kj} \quad (2-14)$$

$O_{(i+1)k}$  为隐节点时, 知道节点的实际输出, 但无法提前知道其正确输出, 只是总的误差与隐层输出相关, 同时隐层的输出, 必影响下一隐层的各节点的输入。

$$\delta_{ik} = O_{(i+1)k} (1 - O_{(i+1)k}) \sum_{h=1}^{N_{i+2}} \delta_{(i+1)h} w_{(i+1)kh} \quad (2-15)$$

BP 算法的权值调整公式为 :

$$\Delta W_{ijk} = \begin{cases} \eta (d_k - y_k) y_k (1 - y_k) O_{ij} & i+1 \text{ 层为输出层} \\ \eta O_{(i+1)k} (1 - O_{(i+1)k}) * (\sum_{h=1}^{N_{i+2}} \delta_{(i+1)h} w_{(i+1)kh}) O_{ij} & i+1 \text{ 层为隐层} \end{cases} \quad (2-16)$$

第  $i$  层神经元

$$\delta_{ik} = \begin{cases} (d_k - y_k) y_k (1 - y_k) & i+1 \text{ 层为输出层} \\ O_{(i+1)k} (1 - O_{(i+1)k}) (\sum_{h=1}^{N_{i+2}} \delta_{(i+1)h} * w_{(i+1)kh}) & i+1 \text{ 层为隐层} \end{cases} \quad (2-17)$$

$$W_{ijk}(t+1) = W_{ijk}(t) + \Delta W_{ijk} = W_{ijk} + \eta \delta_{ik} O_{ij} \quad (2-18)$$

同样证明 :

$$\Delta \theta_{ik} = -\eta \frac{\partial E}{\partial \theta_{ik}} = \begin{cases} \eta (d_k - y_k) y_k (1 - y_k) & i+1 \text{ 层为输出层} \\ \eta O_{(i+1)k} (1 - O_{(i+1)k}) (\sum_{h=1}^{N_{i+2}} \delta_{(i+1)h} * w_{(i+1)kh}) & i+1 \text{ 层为隐层} \end{cases} \quad (2-19)$$

## 2. 改进的 BP 算法

### 误差曲面与 BP 算法的局限性

多层前向 BP 网络的误差是各层权值和输入样本对的函数，因此可以表达为：

$$E = F(X^p, W, \theta, d^p) \quad (2-20)$$

从式 (2-20) 可以看出，误差函数  $E$  中可调整（改变）参数个数  $N_w$  等于各层权值数加上阈值数。是多维空间中一个形状极为复杂的曲面，该曲面的每一个点的“高度”对应一个误差值，每个点的坐标向量对应着  $N_w$  个权值，称这样的空间为误差权空间。为直观描述误差曲面在权空间的起伏状况，图 2-2 给出了二维权空间的误差曲面分布情况。通过这样一个简单的情况可以看出，误差曲面的分布有以下两个特点，这也是标准 BP 算法局限性之所在。

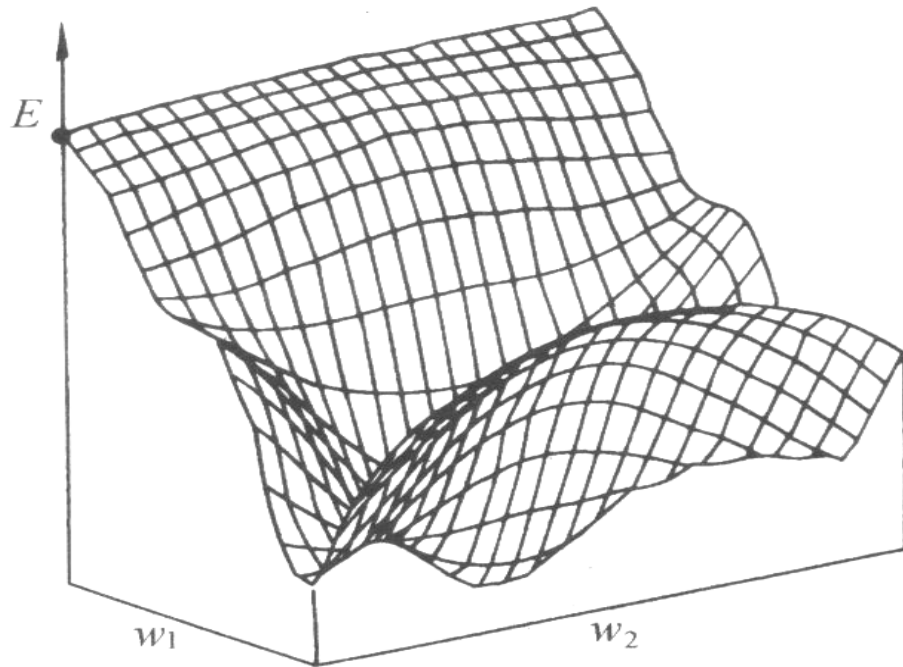


图 2-2 二维权空间的误差曲面分布情况

#### (1) 局限 1

在误差曲面上有些区域平坦，此时误差对权值的变化不敏感，误差下降缓慢，调整时间长，影响收敛速度。这时误差的梯度变化很小，即使权值的调整量很大，误差仍然下降很慢。造成这种情况的原因与各节点的净输入过大有关，以输出层为例分析，由误差梯度表达式

$$\Delta W_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}} = -\eta \delta_{ik} O_{ij} \text{ 可知，误差梯度小意味着 } \delta_{ik} \text{ 接近零。而 } \delta_{ik} \text{ 的表达式为：}$$

$\delta_{ik} = (d_k - y_k) y_k (1 - y_k)$ ，因此， $\delta_{ik}$  接近零有三种可能：一是  $y_k$  充分接近样本  $d_k$ ，此时对应于误差的某个谷点（极小点）；二是  $y_k$  充分接近于零；三是  $y_k$  充分接近于 1。对后两种情况， $E$  可以是任意值，但梯度很小，误差曲面出现平坦区。

$y_k$  0 或 1 的原因在于 S 转移函数具有饱和特性造成的，从 S 转移函数的表达式可知，

当净输入的绝对值  $\left| \sum_{j=0}^m W_{jk} * y_j \right| > 3$  时,  $y_k$  将处于接近“1”或“0”的饱和区。此时对权值的变化不敏感, BP 算法是严格按照误差梯度下降的原则调整权值的, 训练进入平坦区后, 尽管  $y_k - d_k$  仍然很大, 但误差下降缓慢, 调整时间长, 迭代次数多, 影响收敛速度。

## (2) 局限 2

存在多个极小点。从两维权空间的误差曲面可以看出, 其上存在许多凸凹不平, 其低凹部分就是误差函数的极小点。可以想象多维权空间的误差曲面, 会更加复杂, 存在更多个局部极小点, 它们的特点都是误差梯度为 0。BP 算法权值调整依据是误差梯度下降, 当梯度为 0 时, BP 算法无法辨别极小点性质, 因此训练常陷入某个局部极小点而不能自拔, 使训练难以收敛于给定误差。图 2-3 是某一单权值调整为例的局部极小点。

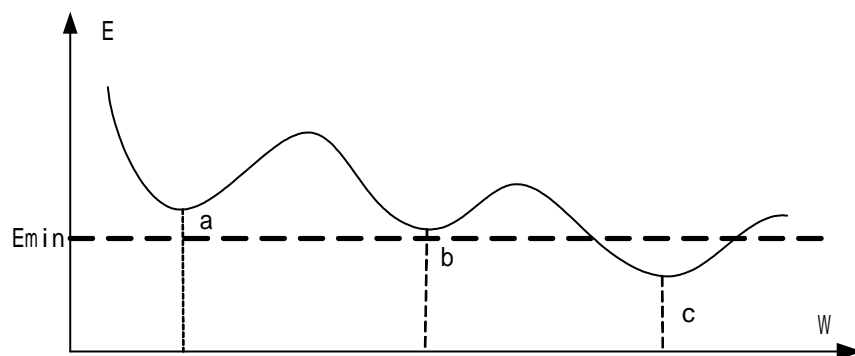


图 2-3 单权值调整为例的局部极小点

### 标准 BP 算法改进

从上面的分析可知, 误差曲面的平坦区将使误差下降缓慢, 调整时间加长, 迭代次数增多, 影响收敛速度; 而误差曲面存在的多个极小点会使网络训练陷入局部极小, 从而使网络训练无法收敛于给定误差。以上两个问题是 BP 网络标准算法的固有缺陷。

针对这一问题, 目前国内外不少学者提出了许多改进算法, 下面就介绍几种典型的改进算法。

#### (1) 增加动量项

标准 BP 算法在调整权值时, 只按  $t$  时刻误差的梯度下降方向调整, 而没有考虑  $t$  时刻以前的梯度方向, 从而常使训练过程发生振荡, 收敛缓慢。为了提高训练速度, 可以在权值调整公式中加一動量項:

$$\Delta W(t) = \eta \delta O + \alpha \Delta W(t-1) \quad (2-21)$$

式中  $W$  为某层权矩阵,  $O$  为某层输出向量,  $f$  称为动量系数,  $f \in (0, 1)$  定义动量项  $\alpha \Delta W(t-1)$  反映了以前积累的调整经验。当误差梯度出现局部极小时, 虽然

$$\Delta W_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}} = -\eta \delta_{ik} O_{ij} \rightarrow 0, \text{ 但 } \Delta W(t-1) \neq 0, \text{ 使其跳出局部极小区域, 加快迭代收敛速度。}$$

目前, 大多数 BP 算法中都增加了动量项, 以至于有动量项的 BP 算法成为一种新的标准算法。

#### (2) 可变学习速度的反向传播算法 (variable learning rate back propagation, VLBP)

正如我们所见的, 多层网络的误差曲面不是二次函数。曲面的形状随参数空间区域的不同而不同。可以在学习过程中通过调整学习速度来提高收敛速度。技巧是决定何时改变学习速度和怎样改变学习速度。

可变学习速度的 VLBP 算法有许多不同的方法来改变学习速度。这里介绍一种非常直

观的批处理过程<sup>[11-12]</sup>，它的学习速度是根据算法的性能改变的。可变学习速度反向传播算法 VLBP 的规则如下：

1) 如果平方误差（在整个训练集上）在权值更新后增加了，且超过了某个设置的百分数  $\xi$ （典型值为1%至5%），则权值更新被取消，学习速度被乘以一个因子  $\rho$  ( $0 < \rho < 1$ )，并且动量系数  $\alpha$ （如果有的话）被设置为0。

2) 如果平方误差在权值更新后减少了，则权值更新被接受，而且学习速度将被乘以一个因子  $\eta > 1$ 。如果  $\alpha$  被设置为0，则恢复到以前的值。

3) 如果平方误差的增长小于  $\xi$ ，则权值更新被接受，但学习速度保持不变。如果  $\alpha$  过去被设置为0，则恢复到以前的值。

### (3) 学习速率的自适应调节

上面给出的可变学习速度 VLBP 算法，需要设置多个参数，算法的性能对这些参数的改变往往十分敏感，另外，处理起来也较麻烦。此处给出一简洁的学习速率的自适应调节算法。学习率的调整只与网络总误差有关。

学习速率  $\eta$  也称步长，在标准 BP 中是一常数，但在实际计算中，很难给定出一个从始至终都很合适的最佳学习速率。从误差曲面可以看出，在平坦区内  $\eta$  太小会使训练次数增加，这时候希望  $\eta$  值大一些；而在误差变化剧烈的区域， $\eta$  太大会因调整过量而跨过较窄的“凹坑”处，使训练出现振荡，反而使迭代次数增加。为了加速收敛过程，最好是能自适应调整学习率  $\eta$ ，使其该大则大，该小则小。比如可以根据网络总误差来调整：在网络经过一批次权值调整后，若  $E_{\text{总}}$ ，则本次调整无效。且：

$$\eta = \beta * \eta \quad 0 < \beta < 1 \quad (2-22)$$

若  $E_{\text{总}}$ ，则有效，且：

$$\eta = \alpha \eta \quad \alpha > 1 \quad (2-23)$$

### (4) 引入陡度因子——防止饱和

从前面的分析可知，误差曲面上存在着平坦区。其权值调整缓慢的原因在于 S 转移函数具有饱和特性造成的。如果在调整进入平坦区后，设法压缩神经元的净输入，使其输出退出转移函数的饱和区，就可改变误差函数的形状，从而使调整脱离平坦区。实现这一思路的具体作法是在转移函数中引进一个陡度因子  $\lambda$

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (2-24)$$

当  $E \rightarrow 0$  而  $d_k - y_k$  仍较大时，进入平坦区，此时令  $\lambda > 1$ ；当退出平均区后，再令  $\lambda = 1$ ，使转移函数恢复原状。

## 2.2 BP 神经网络设计的一般原则

虽然 BP 神经网络是目前应用最广泛，研究较多的一种网络。但关于它的开发设计到目前为止还没有一套完整的理论。大多数是根据使用者的经验来设计网络结构、功能函数、学

习算法、样本等。本节在参考诸多文献<sup>[4,13]</sup>的基础上，结合作者的研究成果<sup>[2,10]</sup>，给出BP神经网络设计的一些共性的原则，供读者参考。

### 2.2.1 BP网络参数设计

#### 1. BP网络输入与输出参数的确定

输入量的选择

(1) 输入量必须选择那些对输出影响大且能够检测或提取的变量。

(2) 各输入量之间互不相关或相关性很小。

从输入、输出量性质分类来看，可以分为两类：数值变量和语言变量。数值变量又分为连续变量或离散变量。如常见的温度，压力，电压，电流等就是连续变量；语言变量是用自然语言表示的概念。如红，绿，蓝；男，女；大，中，小，开，关，亮，暗等。一般来说，语言变量在网络处理时，需要转化为离散变量。

(3) 输入量的表示与提取

大多数情况下，直接送给神经网络的输入量无法直接得到，常常需要用信号处理与特征提取技术从原始数据中提取能反映其特征的若干参数作为网络输入。以下是几个典型特征处理实例。

例2-1 文字符号输入——英文字母识别 (C I T)。

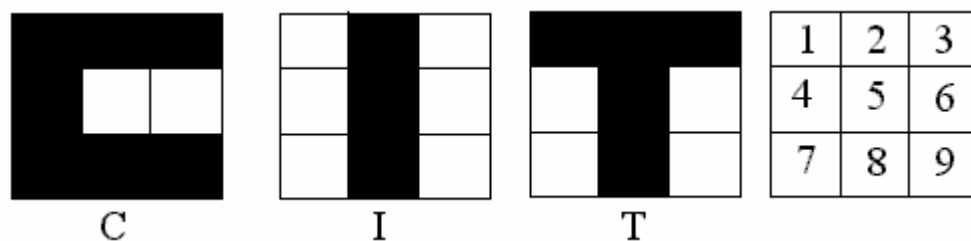


图 2-4 字符的神经网络表示

如图 2-4，将每个字符纳入 3\*3 网络，用数字 1-9 表示网络序号。设计一个具有 9 个分量的输入向量  $X$ ，其中每个分量的下标与网络的序号相对应，取值“1”或“0”表示网络内字符笔记有无。则有：

$$X_c = (111100111)^T, \quad X_I = (010010010)^T, \quad X_T = (111010010)^T,$$

对应的期望值输出应为：C 类，I 类，T 类。其中输出也可用编码方式实现，如字母 A，B，C，...，Z 共有 26 个，可用 5 位二进制表示（或用 26 位二进制数表示）。这样对应的输出向量  $Y$  为：

$$Y_c = (00010)^T, \quad Y_I = (00100)^T, \quad Y_T = (10011)^T$$

例 2. 模糊数学的隶属度函数表示。

对许多模拟信号，一般可以利用模糊数学的隶属度函数来表示。在模式识别和故障诊断中常用到这种处理方法。常见的隶属度函数有正态分布、三角函数分布和梯形分布等。

1) 正态分布：用  $f(x) = e^{-k(x-a)^2}$  ( $k > 0$ ) 来表示“大约为  $a$ ”的模糊概念，如图 2-5。

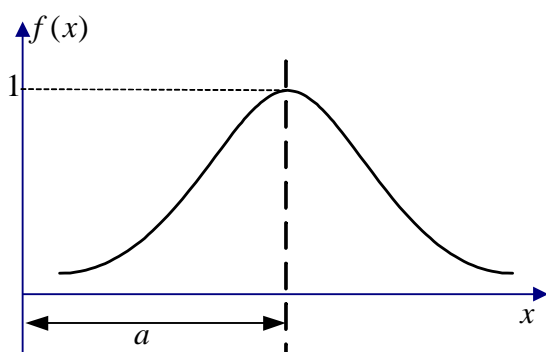


图 2-5 正态分布函数

$a$  为某模式的特征值， $k$  为调节参数。

## 2) 三角函数与梯形分布

三角函数分布表达式与梯形分布表达式分别为：

$$f_1(x) = \begin{cases} \frac{x-b}{a-b} & b \leq x \leq a \\ 0 & b \geq x, \text{ or } x \geq c \\ \frac{c-x}{c-a} & a \leq x \leq c \end{cases}; f_2(x) = \begin{cases} 0 & x \leq b, \text{ or } x \geq c \\ \frac{x-b}{d-b} & b \leq x \leq d \\ \frac{c-x}{c-e} & e \leq x \leq c \\ 1 & d \leq x \leq e \end{cases} \quad (2-25)$$

其相应的曲线图如图 2-6 和 2-7 所示。表示“大约为  $a$ ”的模糊概念，对图 2-6， $a$  为某模式的特征值， $b$  和  $c$  为参数的极限偏差；对图 2-7， $a$  为某模式的特征值， $b$  和  $c$  为参数的极限偏差， $d$  和  $e$  为待诊断设备参数的正常变化范围。

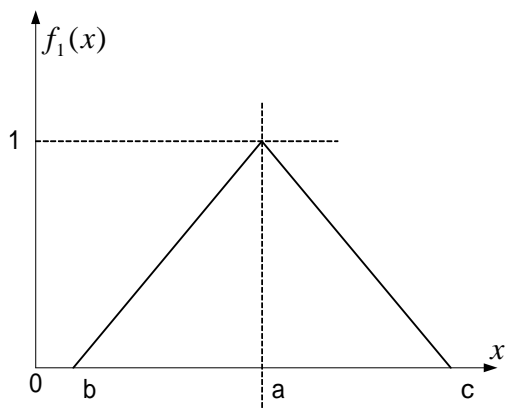


图 2-6 三角函数分布

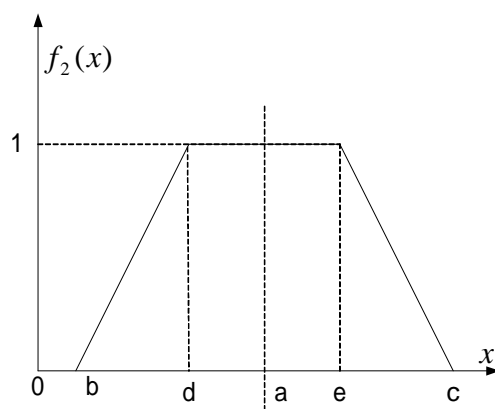


图 2-7 梯形分布

### 输出量选择与表示

(1) 输出量一般代表系统要实现的功能目标，如分类问题的类别归属等。

(2) 输出量表示可以是数值也可能是语言变量：

1) “ $n$  中取 1”表示法。(如分类问题)

例 优，良，中，差 ---- 0001, 0010, 0100, 1000

直观，简便，当分类的类别不是太多时经常采用。

2) “ $n-1$ ”表示法。考虑全 0 的情况

3) 数值表示法：用 0~1 或 -1~1 之间的连续的数据表示。即模糊隶属函数表示方法。如：

好 1, 较好 0.75, 中等 0.5, 较差 0.25, 差 0。



## 2. 训练样本集的设计

网络的性能与训练用的样本密切相关,设计一个好的训练样本集既要注意样本规模,有要注意样本质量。

### 样本数目的确定

一般来说样本数  $n$  越多,训练结果越能正确反映其内在规律,但样本的获取往往有一定困难,另一方面,当样本数  $n$  达到一定数量后,网络的精度也很难提高。

选择原则:网络规模越大,网络映射关系越复杂,样本数越多。一般说来,训练样本数是网络连接权总数的 5~10 倍,但许多情况难以达到这样的要求。

### 样本的选择和组织

(1) 样本要有代表性,注意样本类别的均衡。

(2) 样本的组织要注意将不同类别的样本交叉输入。

(3) 网络的训练测试,测试标准是看网络是否有好的泛化能力。测试做法:不用样本训练集中数据测试。一般是将收集到的可用样本随机地分成两部分,一部分为训练集,另一部分为测试集。若训练样本误差很小,而对测试集的样本误差很大,泛化能力差。

## 3. 初始权值的设计

网络权值的初始化决定了网络的训练从误差曲面的哪一点开始,因此初始化方法对缩短网络的训练时间至关重要。

神经元的作用函数是关于坐标点对称的,若每个节点的净输入均在零点附近,则输出均出在作用函数的中点,这个位置不仅远离作用函数的饱和区,而且是其变化最灵敏的区域,必使网络学习加快。从神经网络净输入表达式来看,为了使各节点的初始净输入在零点附近,如下两种方法被常常使用。

方法一:取足够小的初始权值;

方法二:使初始值为+1 和-1 的权值数相等。

### 2.2.2 BP 网络结构参数设计

#### 1. 隐层结构设计

##### 隐层数设计

理论证明:具有单隐层的前馈网络可以映射所有连续函数,只有当学习不连续函数时才需要两个隐层,故一般情况隐层最多需要两层。一般方法是先设一个隐层,当一个隐层的节点数很多,仍不能改善网络性能时,再增加一个隐层。最常用的 BP 神经网络结构是 3 层结构,即输入层、输出层和 1 个隐层。

##### 隐层节点数设计

隐层节点数目对神经网络的性能有一定的影响。隐层节点数过少时,学习的容量有限,不足以存储训练样本中蕴涵的所有规律;隐层节点过多不仅会增加网络训练时间,而且会将样本中非规律性的内容如干扰和噪声存储进去。反而降低泛化能力。一般方法是凑试法:

(1) 先由经验公式确定

$$m = \sqrt{n+l} + \alpha \quad \text{or} \quad m = \sqrt{nl} \quad (2-26)$$

$m$  为隐层节点数,  $n$  为输入节点  $l$  输出节点  $\alpha$  为调节常数在 1~10 之间。

(2) 改变  $m$ , 用同一样本集训练,从中确定网络误差最小时对应的隐层节点数。

## 2.3 BP 神经网络的应用

### 2.3.1 BP 神经网络在模拟电路故障诊断中应用<sup>[5, 14]</sup>

电子设备故障检测与诊断是一十分复杂的模式识别问题,特别是模拟电路的故障诊断存在广泛的非线性问题,另外模拟电路还存在大量的反馈回路和容差,这都增加了故障模式识

别的复杂性。人工神经网络由于其本身信息处理特点，如并行性、自学习、自组织性、联想记忆及分类功能等，使其能够出色解决那些传统模式识别方法难以圆满解决的问题，近年来电子设备故障诊断成为人工神经网络的重要应用领域之一。

对于一般规模电路的故障诊断，诊断工作可以分为测前工作与测后工作两部分进行。测前工作，根据经验，在一定的激励条件下，将常见的各种故障状态及正常状态所对应的理论值用实验或理论计算求出。并以此作为神经网络的样本，输入给神经网络，将神经网络训练好，实际诊断时，在同样的激励条件下，测出实际的输出，经处理后输入给神经网络。其输出即为所对应的故障。由于神经网络具有联想推理的能力，具有很强的泛化能力，它不仅能识别已经训练过的样本，而且能通过联想推理，识别未出现过的样本。综上所述，用 BP 网络进行电路级故障诊断时，其步骤可以总结为：

- (1)测前，将电路的正常状态及各种故障状态所对应的理论值求出或通过实验测定好，并进行数据预处理作为样本输入给 BP 网络，训练好 BP 网络。
- (2)加测试激励，检测实际状态输出，并提供给 BP 网络，BP 网络经过学习后，它不仅能识别已经训练过的样本，而且能通过联想推理，识别未出现过的样本，这正是神经网络的泛化能力的体现。

### 1. 问题提出：

某装备雷达喇叭控制继电器板为一由三极管组成的电路。其中有 5 个关键电压测试点，9 种元件故障。要求用一训练的 BP 网识别任何一组测试信号，指出故障元器件（模式）。电路原理图略。

### 2. BP 网设计

**输入、输出层和隐层节点数确定。**

利用三层 BP 网络来识别电路故障。取输入节点  $N_1=5$ ，代表 5 个关键测试点电压信号；取输出节点  $N_3=10$ ，分别代表 9 种故障元件和正常情况。其 BP 网络结构如图 2-8 所示。根据 BP 网络隐层节点设计的经验公式，隐层节点数  $N_2$  可取为：

$$N_2 = \sqrt{N_1 * N_3} + \alpha = \sqrt{50} + \alpha = 12 \quad (\alpha = 1 \sim 10, N_1, N_2 \text{ 分别为输入和输出层节点数}) \quad (2-27)$$

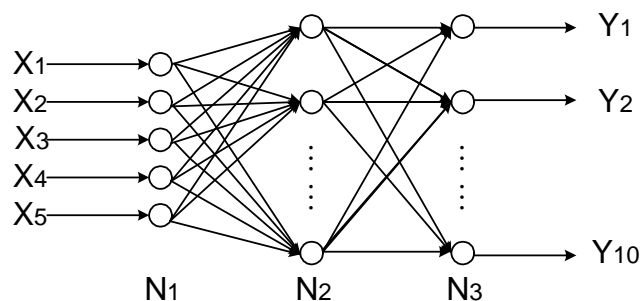


图 2-8 三层 BP 网络结构

### 故障特征提取、数据变换

测试点待测的是电压信号，一般要进行特征提取——模糊化处理。在模拟电路测试中，同一参数对于不同故障有不同的反映，同一参数在故障的不同阶段也有不同反映。因此对电路的某一特征参数来说，其模糊子集的故障隶属度函数一般是不同的。常用方法是使用精确

事实规则。例如对于“大约为  $a$ ”的模糊概念，可用一正态分布函数  $f(x) = e^{-k(x-a)^2} (k > 0)$

表示，如图 2-9 (a)。简化形式也可用三角函数如图 2-9(b)或梯形分布如图 2-9(c)。

$f(x) \in (0,1)$ ，表示“ $x$  大约为  $a$ ”的程度。 $k$  为调节参数。

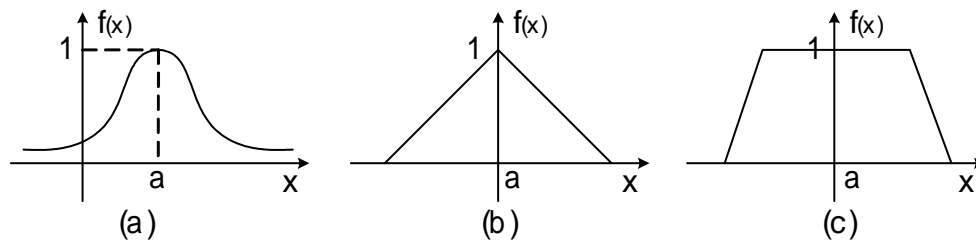


图 2-9 分布函数

### 3. 输入输出样本及其模糊处理。

#### 原始数据

电路原始测试数据如表 2-1 所示。

表 2-1 电路原始测试数据

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	故障形式
11.264	14.134	10.631	14.779	11.331	$Y_1$ (无故障)
0.171	14.78	0	14.78	0.248	$Y_2$ ( $R_1$ 开路)
11.234	10.465	10.452	10.881	10.882	$Y_3$ ( $R_2$ 开路)
11.264	14.134	10.833	14.779	11.536	$Y_4$ ( $R_3$ 开路)
10.264	14.18	14.757	14.78	13.355	$Y_5$ ( $R_4$ 开路)
0.741	14.653	0.153	14.78	0.002	$Y_6$ (短路)
10.341	14.78	9.512	14.78	9.973	$Y_7$ ( $BG_1$ 集电极开路)
11.264	14.78	9.989	14.78	10.689	$Y_8$ ( $BG_1$ 射集开路)
11.333	11.341	12.305	13.020	13.012	$Y_9$ ( $BG_1$ 基集与集电极短路)
11.264	14.058	14.058	14.776	14.767	$Y_{10}$ ( $BG_1$ 发射极与集电极短路)

#### 模糊处理

采用  $f(x) = e^{-k(x-a)^2}$  ( $k > 0$ ) 其中  $k$  对不同测试点选不同数值。此处： $k_1=10$   $k_2=0.3$

$k_3=2.5$   $k_4=0.2$   $k_5=1$ ；得到网络的训练样本如表 2-2 所示。

表 2-2 BP 网络输入、输出训练样本

序号		1	2	3	4	5	6	7	8	9	10
故障征兆	X <sub>1</sub>	1.0000	0.0000	0.9910	1.0000	1.0000	0.0000	0.0020	1.0000	0.9535	1.0000
	X <sub>2</sub>	1.0000	0.8823	0.0176	1.0000	0.8823	0.9223	0.8823	0.8823	0.0963	0.9983
	X <sub>3</sub>	1.0000	0.0000	0.9230	0.9030	0.0000	0.0000	0.0437	0.3569	0.0009	0.0000
	X <sub>4</sub>	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5386	1.0000
	X <sub>5</sub>	1.0000	0.0000	0.8174	0.9588	0.0071	0.0000	0.1582	0.6622	0.0593	0.0000
故障模式	Y <sub>1</sub>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>2</sub>	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>3</sub>	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>4</sub>	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>5</sub>	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>6</sub>	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
	Y <sub>7</sub>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
	Y <sub>8</sub>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Y <sub>9</sub>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000

	$Y_{10}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
--	----------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

取学习率  $\eta=0.5$  ; 动量因子  $\alpha=0.5$  ;  $E<0.02$  起用增加动量项的 BP 算法, 利用表 2-3 给出的样本训练 BP 网。

#### 4. BP 网识别结果分析

先用样本自身数据送入训练完成的样本中, 其识别结果如表 2-3 所示, 从中可见自身数据检测正确率为 100%。

表 2-3 自身数据识别结果

$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$	$Y_6$	$Y_7$	$Y_8$	$Y_9$	$Y_{10}$
0.0001	0.0000	0.0039	0.0338	0.0000	0.0028	0.0049	0.0044	0.0000	0.0001
0.0024	0.9285	0.0001	0.0019	0.0064	0.5333	0.0144	0.0004	0.0004	0.0009
0.0012	0.0004	0.9930	0.0000	0.0063	0.0000	0.0027	0.0000	0.0069	0.0000
0.0000	0.0000	0.0021	0.9637	0.0000	0.0076	0.0039	0.0086	0.0001	0.0003
0.0001	0.0039	0.0004	0.0001	0.9820	0.0000	0.0000	0.0009	0.0083	0.0106
0.0021	0.0598	0.0001	0.0076	0.0003	0.9442	0.0160	0.0006	0.0003	0.0130
0.0022	0.0019	0.0001	0.0052	0.0005	0.0080	0.9730	0.0093	0.0001	0.0007
0.0002	0.0000	0.0013	0.0112	0.0001	0.0017	0.0148	0.9847	0.0000	0.0018
0.0004	0.0141	0.0055	0.0003	0.0065	0.0000	0.0000	0.0006	0.9898	0.0000
0.0001	0.0000	0.0010	0.0005	0.0156	0.0120	0.0000	0.0087	0.0019	0.9843

任取 2 组实验数据如表 2-4 所示, 看其故障识别结果。

表 2-4 任意两组实验数据

	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$
(1)	11.33	11.331	12.305	13.002	13.012
(2)	11.260	14.778	14.747	14.78	13.315

表 2-5 模糊化数据

	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$
(1)	0.9574	0.0947	0.0009	0.5386	0.0593
(2)	0.9998	0.8830	0.0000	1.0000	0.0166

表 2-6 输出诊断结果

$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$	$Y_6$	$Y_7$	$Y_8$	$Y_9$	$Y_{10}$
0.0004	0.0137	0.0055	0.0003	0.0065	0.0000	0.0000	0.0006	0.9899	0.0000
0.0001	0.0032	0.0004	0.0002	0.9792	0.0000	0.0000	0.0010	0.0081	0.0120

输入模糊化后得数据如表 2-5 所示。再将其送入训练完成的样本中, 得输出结果如表 2-6 所示。可见: 第 1 组为  $Y_9$  故障模式——BG1 基数与集电极短路。第 2 组为  $Y_5$  故障模式——R4 开路。由此可以看出, 本网络不仅能识别出样本自身故障, 而且能准确诊断样本以外数据故障, 也就是说, BP 神经网络在进行模拟电路故障诊断时, 不仅有很强的学习能力, 而且有一定的泛化能力。

### 2.3.2 BP 神经网络在多传感器信息融合故障诊断中应用<sup>[6-10]</sup>

#### 1. 多传感器神经网络信息融合故障诊断原理

神经网络信息融合, 是将神经网络(如 BP 网)引入信息融合之中, 同时结合模糊集合论进行故障判断。利用多传感器信息, 来提高故障诊断的准确性。其具体过程为: 通过多传感器测试被诊断对象, 求出每一传感器对故障集中各类故障的隶属度值, 将所有传感器的故障隶属度值矢量作为神经网络的输入, 网络输出即为融合后该症状属于各类故障的隶属度值矢

量,最后利用基于规则的判定原则进行故障决策,其融合诊断过程如图 2-10 所示。其中  $A_1, A_2, \dots, A_n$  为待诊断的故障模式。 $\mu_j(A_1) \dots \mu_j(A_n)$  为传感器  $j$  测得的各故障模式  $A_1, A_2, \dots, A_n$  所得的隶属度值,  $\mu(A_1) \dots \mu(A_n)$  为融合的隶属度值。

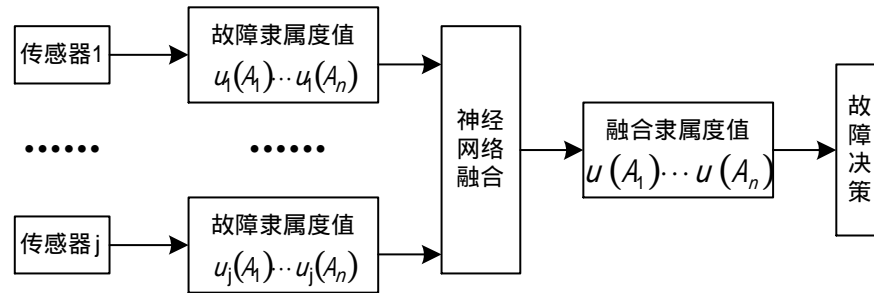


图 2-10 神经网络数据融合故障诊断

### 各传感器故障隶属度值确立

通过传感器测试待诊断对象的症状参数,经过一定的变换处理,得到各传感器所测症状属于各类故障的隶属度值。 $\mu_j(A_1) \dots \mu_j(A_1) \dots \mu_j(A_n)$ ; 图 2-10 中的  $\mu_j(A_i)$  表示传感器  $j$  测得症状属于故障  $A_i$  的隶属度值,  $n$  为故障的类型总数。

### 神经网络的训练

先用标准样本对神经网络进行训练,其样本由实验测定,训练时网络的输入端为各传感器测试出的各待诊断对象属于故障的隶属度,网络输出端为各待诊断对象已知的故障隶属度值。网络的实际结构形式视实际问题而定。

### 神经网络信息融合

对某一待诊断对象,将各传感器测试的故障隶属度矢量作为训练后网络的输入,输出即为融合后的故障隶属度矢量。由此输出可以判断实际故障模式。

### 故障决策:

对网络融合后输出的各待诊断模式属于故障的隶属度值,在故障决策时,采用基于规则的方法,其基本原则有:

- (1) 判定的故障模式应具有最大的隶属度值。
- (2) 判定的故障模式的隶属度值要大于某一阈值。此阈值过小,易出现故障误判现象,而此阈值过大时又易出现故障漏判。一般来说此值至少要大于  $1/n$  ( $n$  为待诊断故障模式个数),阈值越大故障判断的准确性越高,只是取得过大时测试得到的故障隶属度值无法满足要求,因此必须针对实际被诊断设备,根据实际诊断结果准确率的统计数据来选择一大于  $1/n$  的适当数值。
- (3) 判定的目标模式和其它模式的隶属度值之差要大于某个门限。此门限的取值类似于 (2)。

## 2. 简单模拟电路信息融合故障诊断实例

### 问题的提出

图 2-11 为实验用待诊断电路原理图,此电路为一弱信号放大电路,其中的三个运放芯片 OP07 为待诊断元件,称为  $A_1$ :元件 1,  $A_2$ :元件 2,  $A_3$ :元件 3。取  $U_1, U_2, U_3$  分别为元件 1、2、3 的电压测试点,以各元件断路(不供电)为故障形式。用运多传感器模糊信息融合方法进行故障元件搜寻,并与单传感器诊断进行比较,从而通过这一简单电路的故障诊断说明多传感器信息融合在电路故障诊断中的有效性和优越性。

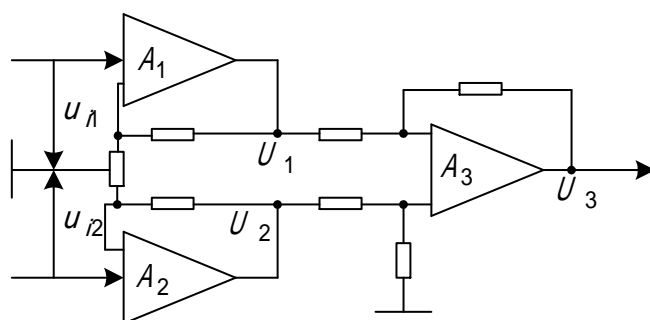


图 2-11 实验电路原理图

### 融合诊断原理

实验中所采用的多传感器信息融合结构框图如图 2-12 所示。首先利用探针测出各待诊断元件关键点的电压信号，利用热像仪（Inframetrics 600）测试出电路板待诊断元件的工作温度信号，对每一传感器来讲，被测元件属于故障的可能性可分别用一组隶属度值来表示，这样会得到两组共六个隶属度值。由于电路中前后元件的相互影响，同一传感器测得的不同元件的隶属度有的相互接近，如用一种传感器判别故障元件，往往会出现误判，解决的办法是：应用模糊集合论对两组隶属度值进行融合处理，得到两传感器融合后各待诊断元件属于故障的隶属度值，再根据一定的判定准则进行故障元件的判定。

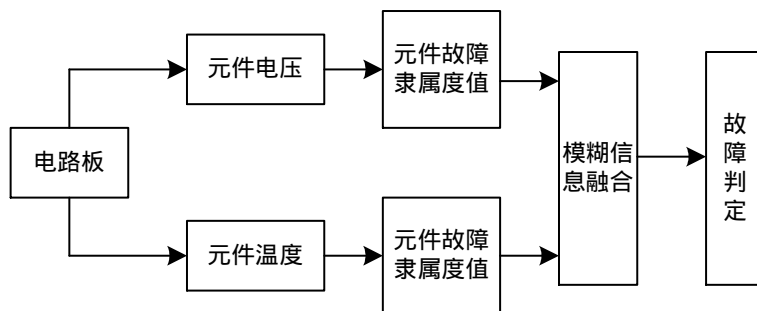


图 2-12 基于多传感器模糊信息融合的故障诊断框图

### 隶属度函数的形式

隶属度函数主要由传感器本身的工作特性及被测参数的特性而定，对电子电路系统某一特定元器件，当系统正常工作时，其关键点电压值应是稳定的，在环境温度一定时，其芯片温度值也是一稳定数值，当元器件出现故障时，一般地说其电压值会偏离正常范围，温度信号也会发生变化（无论是升高还是降低），偏差越大相对来说元器件出现故障的可能性会变大，为讨论方便，同时结合文献[14]的隶属度函数待定系数法及具体的实验结果，

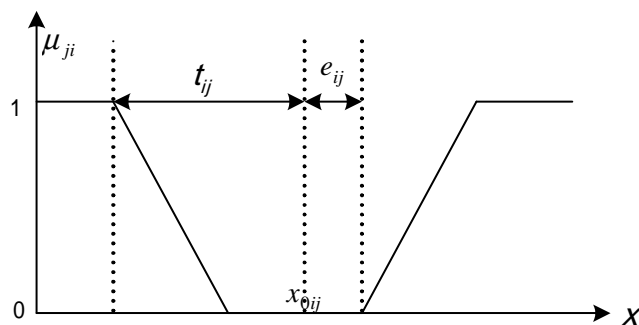


图 2-13 隶属函数分布

定义隶属度函数  $\mu_{ji}(x)$  的分布如图 2-13 所示。其表达式如下：

$$\mu_{ji} = \begin{cases} 1 & x_j \leq x_{0ij} - t_{ij} \\ -\alpha(x_j - x_{0ij} + e_{ij}) / (t_{ij} - e_{ij}) & x_{0ij} - t_{ij} < x_j \leq x_{0ij} - e_{ij} \\ 0 & x_{0ij} - e_{ij} < x_j \leq x_{0ij} + e_{ij} \\ \alpha(x_j - x_{0ij} - e_{ij}) / (t_{ij} - e_{ij}) & x_{0ij} + e_{ij} < x_j \leq x_{0ij} + t_{ij} \\ 1 & x_j > x_{0ij} + t_{ij} \end{cases} \quad (2-28)$$

其中  $x_{0ij}$  为电路工作正常时被测元件的标准参数值，

$e_{ij}$  为待诊断元件参数的正常变化范围，即容差。

$t_{ij}$  为待诊断元件参数的极限偏差， $\alpha$  为修正系数。

$\mu_{ji}$  为传感器  $j$  测定被诊断元件  $i$  属于故障的隶属度， $x_j$  为传感器  $j$  测定的实际数值。

在此，为处理问题的方便结合实际测试数据取电压传感器  $\alpha=1$ ，温度传感器  $\alpha=10$ ；

$e_{ij} = 0$ ， $t_{ij} = x_{0ij}$ ，其中表 2-7 为原始的温度和电压测试数据，其中  $T_1$ 、 $T_2$ 、 $T_3$  分别为元

件 1、2、3 的工作温度， $U_1$ 、 $U_2$ 、 $U_3$  分别为元件 1、2、3 的工作电压。它是在人为设置  $A_1$ 、 $A_2$  和  $A_3$  故障（即分别使其不供电）的情况下，通过改变输入信号  $U_{i1}$ 、 $U_{i2}$  得到的多组实验数据。而表 2-8 为按(2-28)式计算后的故障隶属度表格。表 2-9 为归一化的故障隶属度表格

表 2-7 温度和电压测试数据

序号	状态	电压数据(V)			温度数据( )		
		$U_1$	$U_2$	$U_3$	$T_1$	$T_2$	$T_3$
1	$A_1$ 故障	9.00	4.01	7.81	10.41	11.33	11.25
	正常状态	10.33	4.07	9.35	11.01	11.10	11.30
2	$A_2$ 故障	10.51	6.50	6.35	11.01	10.30	11.21
	正常状态	10.33	4.07	9.35	11.01	11.10	11.30
3	$A_3$ 故障	10.40	3.99	4.32	11.11	11.01	10.54
	正常状态	10.33	4.07	9.35	11.01	11.10	11.30
4	$A_1$ 故障	6.65	2.05	7.52	10.70	11.61	11.45
	正常状态	7.48	2.00	8.32	11.40	11.71	11.55
5	$A_2$ 故障	7.40	3.25	6.72	11.40	10.81	11.45
	正常状态	7.48	2.00	8.32	11.40	11.71	11.55
6	$A_3$ 故障	7.41	2.06	4.03	11.60	11.61	10.70
	正常状态	7.48	2.00	8.32	11.40	11.71	11.55
7	$A_1$ 故障	5.05	3.45	2.75	11.81	12.60	12.55
	正常状态	5.50	3.50	3.21	12.61	12.50	12.55
8	$A_2$ 故障	5.45	4.40	1.90	12.32	11.60	12.45
	正常状态	5.50	3.50	3.21	12.61	12.50	12.55
9	$A_3$ 故障	5.44	3.55	1.80	12.50	12.41	11.80
	正常状态	5.50	3.50	3.21	12.61	12.50	12.55
10	$A_1$ 故障	8.50	3.65	8.50	11.80	12.66	12.55
	正常状态	9.49	3.60	9.56	12.55	12.76	12.55
11	$A_2$ 故障	9.55	5.20	7.40	12.40	11.85	12.55
	正常状态	9.49	3.60	9.56	12.55	12.76	12.55

12	A <sub>3</sub> 故障	9.56	3.65	5.50	12.60	12.46	11.74
	正常状态	9.49	3.60	9.56	12.55	12.76	12.55
13	A <sub>1</sub> 故障	7.70	2.45	9.05	12.30	13.12	13.12
	正常状态	8.62	2.50	9.95	13.00	13.02	13.12
14	A <sub>2</sub> 故障	8.65	4.07	7.75	12.90	12.21	13.12
	正常状态	8.62	2.50	9.95	13.00	13.02	13.12
15	A <sub>3</sub> 故障	8.70	2.60	6.54	13.00	12.91	12.32
	正常状态	8.62	2.50	9.95	12.91	13.02	13.12
16	A <sub>1</sub> 故障	5.32	1.45	6.91	11.83	12.74	12.65
	正常状态	6.25	1.52	7.82	12.43	12.44	12.35
17	A <sub>2</sub> 故障	6.30	2.35	6.52	12.63	11.93	12.66
	正常状态	6.25	1.52	7.82	12.43	12.44	12.35
18	A <sub>3</sub> 故障	6.32	1.60	5.22	12.82	12.64	11.94
	正常状态	6.25	1.52	7.82	12.43	12.44	12.35

表 2-8 故障隶属度

序号	故障元件	电压故障隶属度			温度故障隶属度		
		$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$	$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$
1	A <sub>1</sub>	0.1288	0.0147	0.1647	0.5450	0.2072	0.0443
2	A <sub>2</sub>	0.0174	0.6225	0.3209	0.0000	0.7207	0.0797
3	A <sub>3</sub>	0.0068	0.0197	0.5380	0.0908	0.0810	0.6726
4	A <sub>1</sub>	0.1110	0.0250	0.0962	0.6140	0.0854	0.0866
5	A <sub>2</sub>	0.0107	0.5854	0.1923	0.0000	0.7686	0.0866
6	A <sub>3</sub>	0.0094	0.0300	0.5156	0.1754	0.0854	0.7359
7	A <sub>1</sub>	0.0818	0.0143	0.1433	0.6344	0.0800	0.0000
8	A <sub>2</sub>	0.0091	0.2571	0.4081	0.2300	0.7200	0.0797
9	A <sub>3</sub>	0.0110	0.0143	0.4393	0.0872	0.0720	0.5976
10	A <sub>1</sub>	0.1043	0.0139	0.1108	0.5976	0.0784	0.0000
11	A <sub>2</sub>	0.0063	0.4444	0.2260	0.1195	0.7132	0.0000
12	A <sub>3</sub>	0.0074	0.0139	0.4247	0.0398	0.2351	0.6454
13	A <sub>1</sub>	0.1067	0.0200	0.0905	0.5385	0.0614	0.0000
14	A <sub>2</sub>	0.0035	0.6820	0.2211	0.0769	0.6221	0.0000
15	A <sub>3</sub>	0.0093	0.0400	0.3427	0.0000	0.0845	0.6098
16	A <sub>1</sub>	0.1488	0.0461	0.1164	0.4827	0.2412	0.2429
17	A <sub>2</sub>	0.0080	0.5460	0.1662	0.1609	0.4100	0.2510
18	A <sub>3</sub>	0.0112	0.0526	0.3325	0.3138	0.1608	0.3320

表 2-9 归一化故障隶属度

序号	故障元件	电压故障隶属度			温度故障隶属度		
		$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$	$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$
1	A <sub>1</sub>	0.4179	0.0477	0.5344	0.6842	0.2614	0.0554
2	A <sub>2</sub>	0.0181	0.6479	0.3340	0.0000	0.9004	0.0996
3	A <sub>3</sub>	0.0121	0.0349	0.9530	0.1075	0.0959	0.7966
4	A <sub>1</sub>	0.4780	0.1077	0.4143	0.7812	0.1087	0.1101



5	A <sub>2</sub>	0.0136	0.7425	0.2439	0.0000	0.8987	0.1013
6	A <sub>3</sub>	0.0169	0.0541	0.9290	0.1760	0.0857	0.7383
7	A <sub>1</sub>	0.3417	0.0597	0.5986	0.8880	0.1120	0.0000
8	A <sub>2</sub>	0.0135	0.3813	0.6052	0.2234	0.6992	0.0774
9	A <sub>3</sub>	0.0237	0.0308	0.9455	0.1152	0.0951	0.7897
10	A <sub>1</sub>	0.4555	0.0607	0.4838	0.8840	0.1160	0.0000
11	A <sub>2</sub>	0.0093	0.6567	0.3340	0.1435	0.8565	0.0000
12	A <sub>3</sub>	0.0166	0.0312	0.9522	0.0433	0.2555	0.7012
13	A <sub>1</sub>	0.4913	0.0921	0.4166	0.8976	0.1024	0.0000
14	A <sub>2</sub>	0.0041	0.7366	0.2593	0.1100	0.8900	0.0000
15	A <sub>3</sub>	0.0237	0.1020	0.8743	0.0000	0.1217	0.8783
16	A <sub>1</sub>	0.4780	0.1481	0.3739	0.4993	0.2495	0.2512
17	A <sub>2</sub>	0.0111	0.7581	0.2308	0.1958	0.4988	0.3054
18	A <sub>3</sub>	0.0283	0.1327	0.8390	0.3890	0.1994	0.4116

### 神经网络训练样本

训练样本的具体数据如表 2-10 所示。表中输入部分左边三行为电压传感器的归一化隶属度值，右边三行为温度传感器的归一化隶属度值。其输出部分中，0 代表元件正常，1 代表元件故障。

表 2-10 BP 神经网络训练样本

序号	输入						输出
	电压			温度			A <sub>1</sub> A <sub>2</sub> A <sub>3</sub>
	$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$	$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$	
1	0.4179	0.0477	0.5344	0.6842	0.2614	0.0554	1,0,0
2	0.0181	0.6479	0.3340	0.0000	0.9004	0.0996	0,1,0
3	0.0121	0.0349	0.9530	0.1075	0.0959	0.7966	0,0,1
4	0.4780	0.1077	0.4143	0.7812	0.1087	0.1101	1,0,0
5	0.0136	0.7425	0.2439	0.0000	0.8987	0.1013	0,1,0
6	0.0169	0.0541	0.9290	0.1760	0.0857	0.7383	0,0,1
7	0.3417	0.0597	0.5986	0.8880	0.1120	0.0000	1,0,0
8	0.0135	0.3813	0.6052	0.2234	0.6992	0.0774	0,1,0
9	0.0237	0.0308	0.9455	0.1152	0.0951	0.7897	0,0,1
10	0.4555	0.0607	0.4838	0.8840	0.1160	0.0000	1,0,0
11	0.0093	0.6567	0.3340	0.1435	0.8565	0.0000	0,1,0
12	0.0166	0.0312	0.9522	0.0433	0.2555	0.7012	0,0,1
13	0.4913	0.0921	0.4166	0.8976	0.1024	0.0000	1,0,0
14	0.0041	0.7366	0.2593	0.1100	0.8900	0.0000	0,1,0
15	0.0237	0.1020	0.8743	0.0000	0.1217	0.8783	0,0,1

### 神经网络信息融合故障判定规则

对神经网络融合后输出的各元件属于故障的隶属度值，在故障元件决策时，其基本原则与模糊融合判定规则基本相同，只是阈值可以取得更大一些：

判定的故障元件应具有最大的隶属度值。

判定的故障元件的隶属度值要大于某一阈值。对此处给出的被诊断电路，从实验中发现当此值取 0.75 时，其融合结果能准确判断故障元件。

判定的故障元件和其它元件的隶属度值之差要大于某个门限。对此诊断实例取其值为 0.6。

### 神经网络信息融合诊断结果讨论

表 2-11 为利用热像仪和探针测试出的的隶属度值分配表及单传感器和双传感器神经网络融合故障识别结果。对表中每一种故障元件来说,其中第一行,第二行为从电压和温度所测值得的各元件属于故障的隶属度值,第三行为融合后的隶属度值。从表中可以明显看出,在本实验电路被诊断的三种故障元件中,两种传感器所提取的故障隶属度值有的很相近,如果只用一种传感器的隶属度值来识别故障元件,会出现无法确定哪个元件故障的状况。如元件 1 故障时,两种传感器所提取的故障隶属度值十分接近,单独识别时无法判定故障元件,但融合后元件 1 的隶属度值大为提高,且与另两种元件的隶属度值相差很大,此时就能很准确地识别出故障元件。也就是说,融合后的隶属度值和单传感器的隶属度值相比,增加了实际故障元件的隶属度

表 2-11 各传感器单独故障识别和神经网络融合故障识别结果比较

故障元件	传感器及融合	故障隶属度			诊断结果
		$\mu(A_1)$	$\mu(A_2)$	$\mu(A_3)$	
元件 1 $A_1$	电压	0.4780	0.1481	0.3739	不定
	温度	0.4993	0.2495	0.2512	不定
	融合	0.8892	0.0045	0.1065	$A_1$ 故障
元件 2 $A_2$	电压	0.0111	0.7581	0.2308	不定
	温度	0.1958	0.4988	0.3054	不定
	融合	0.0518	0.9418	0.0078	$A_2$ 故障
元件 3 $A_3$	电压	0.0283	0.1327	0.8390	$A_3$ 故障
	温度	0.3890	0.1994	0.4116	不定
	融合	0.0097	0.0439	0.9836	$A_3$ 故障

分配值,相对减少了其它元件的隶属度分配值,这必使待诊断对象的不确定性大幅度降低,消除了由于单传感器提供信息量少而产生的误诊断现象,在本例这种人为设置的简单电路故障诊断中,故障识别的准确率达到了 100%。由此可见,基于 BP 网和模糊数学的多传感器信息融合故障诊断方法,使实际故障元件的隶属度值大为增加,待诊断对象的可分析性增强,故障元件定位的准确率大为提高。

### 2.3.3 BP 神经网络在工业生产中的应用<sup>[15]</sup>

#### ——基于人工神经网络的热轧带钢热流密度预测

在热轧带钢生产中,终轧温度是一个重要的技术指标,其控制精度直接影响整卷带钢力学性能和微观组织的均匀性。而决定终轧温度控制精度的重要参数是机架间水冷区的带钢热流密度。轧件在精轧区的温度变化是一个复杂的过程。实际冷却过程中释放的带钢相变潜热难以用数学模型精确表达,只能依赖经验模型和模型的自适应功能进行修正,因此,生产中终轧温度控制超差的现象普遍存在。人工神经网络的出现给提高终轧温度的控制精度提供了可能性。神经网络是以模拟脑神经传递信息的方法建立起来的信息智能化处理系统,具有自学习、自组织、自适应和非线性动态处理等特性,特别适合处理复杂的非线性过程。国内在上世纪开始有人工智能在轧制领域应用研究的报导,如冷连轧机组压下规程设定、多辊轧机板形控制、利用 BP 网络进行板形识别、综合利用神经网络和模糊逻辑进行板形控制、工字钢孔型设计专家系统<sup>[16-18]</sup>等。此处针对国内某钢铁厂热连轧机架间冷却系统,采用神经网络与数学模型相结合的方法,通过分析处理实测数据给出优化的带钢热流密度,并据此预测带钢的终轧温度。

### 1. 机架间喷水冷却的数学模型

机架间冷却属于低压喷水冷却,是强迫对流的一种形式。此处采用数学模型描述了建立在无限大平板热传导方程基础上的冷却过程。在冷却区中,带钢上某一点的表面温度可以由下面的时间函数近似描述:

$$T(t) = T_a + (T_f - T_a)e^{p \cdot t} \quad (2-29)$$

其中,  $T(t)$  为时刻  $t$  带钢某点的表面温度;  $t$  为带钢进入冷却区的时间;  $T_a$  为环境温度;  $T_f$  为进入冷却区时带钢的温度;  $p$  为模型系数,由式(2-30)确定:

$$p = \frac{2\alpha}{\rho h C} \quad (2-30)$$

式中,  $\alpha$  为对流散热系数;  $\rho$  为带钢的密度;  $h$  为带钢的厚度;  $C$  为带钢的热容。为了简化计算,将式(2-29)线性化:

$$\Delta T = \frac{1000Qt}{3600Cph} \quad (2-31)$$

式中,  $Q$  为带钢的热流密度。可以看出,带钢热流密度对终轧温度的计算精度影响很大。

同时,根据现场实际生产经验可知,带钢的热流密度又受带钢的导热系数、对流散热系数、带钢厚度、水压和带钢速度的影响。可见,只有准确计算出带钢的热流密度,才能提高终轧温度的预测精度。

### 2. 神经网络模型

问题的描述

BP神经网络可以任意精度逼近连续函数。对于精轧模型参数,如果能设计出适当的网络结构和算法,就可得到预测模型的优化结果<sup>[17]</sup>。但神经网络在处理问题过程中类似黑箱,无法表达过程的物理意义,若能将其与能表达某些物理意义的数学模型相结合,对于描述工艺过程更为有利。带钢热流密度是机架间冷却温降模型的重要参数,可利用神经网络对其进行预测,并将结果代入数学模型用于计算带钢的终轧温度,其处理过程见图2-14。这将有利于提高终轧温度的预报和控制精度,从而达到优化终轧温度数学模型的目的。

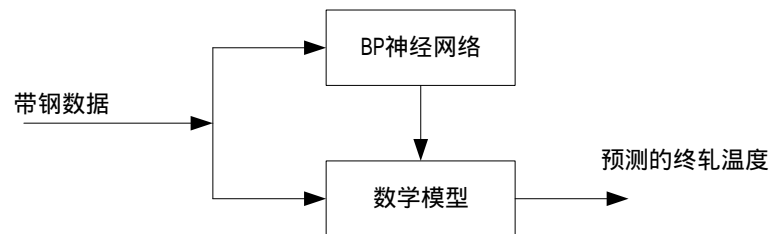


图 2-14 神经网络与数学模型的结合

影响带钢热流密度的物理参数很多,如对流散热系数、带钢的厚度、带钢的导热系数、冷却区的冷却能力、冷却水流量、水压和带钢速度。带钢热流密度与这些影响因素之间存在着复杂的非线性映射关系,各影响因素对控制目标作用的权重不同。而多层前馈神经网络(BP神经网络)具有的非线性逼近能力以及用不同的权值表述各权重的大小的特性可解决上述问题。另一方面,BP网络的自学习和调整功能,使得实际输出不断逼近期望输出,且性能稳定可靠,显然BP网络是最佳选择之一。

## 模型结构的确定

### (1) 输入层和输出层的设计

BP 网络输入节点的数目要适量。过多会导致 BP 网络结构过于庞大，不可避免引入更多的噪声信息；过少则不能保证网络所必须的信息量。带钢热流密度与许多物理量有关，但有些物理量的实测值很难得到，因此这些物理量不宜作为输入量。此处将带钢厚度、带钢宽度、轧制速度、入精轧机时带钢的温度、目标终轧温度、水温共 6 个物理参数作为神经网络的输入量，带钢热流密度作为神经网络的输出量。因此，设定输入层节点数为 6，输出层节点数为 1。

### (2) 隐层数和节点数的选择

1989 年，Robert Hecht-Nielson 证明了对于任何闭区间内的一个连续函数都可以用一个隐层的 BP 网络来逼近。因为一个 3 层的 BP 网络可以完成任意的  $n$  维到  $m$  维的连续映射，故本模型采用单隐层。隐层单元数的选择往往靠经验确定。此处应用前述公式 (2-26)

$m = \sqrt{n + l} + \alpha$  式中， $m$  为隐层单元数， $n$  为输入层单元数， $l$  为输出层单元数， $\alpha$  为 1 ~ 10 之间的常数。据此可取  $m = 7$ 。在实际运行网络时未发现某一节点的权值总无变化，而且神经网络能很好地满足要求，故最终确定隐层单元数为 7。中间隐层神经元的激发函数取为 Sigmoid 函数。

### (3) 其他参数的确定

样本数：一般来说，样本数据越多，学习和训练的结果越能正确反映输入值与输出值之间的关系，但是这样会加大收集、分析数据的难度，同时使网络训练的误差加大。这里选择样本数为 600，其中 325 个训练样本，225 个测试样本，训练 6000 次。

学习步长：学习步长选得足够小可使网络的总误差函数达到最小值。但是太小的学习步长会使得网络的学习速度非常慢；若学习步长较大，则权值更新幅度较大，有可能加快收敛速度。但学习步长过大有可能造成算法的不稳定。这里初始步长选为 0.1，在实际运行时加以调整：若误差函数  $E < 0$ ，步长乘 2；若误差函数  $E > 0$ ，步长乘 0.5。

综上所述，所用神经网络为  $6 \times 7 \times 1$  的 3 层网络结构，见图 2-15。图中  $w_{ij}^1$  和  $w_{ij}^2$  分别表示输入层与隐层以及隐层与输出层之间的权值。

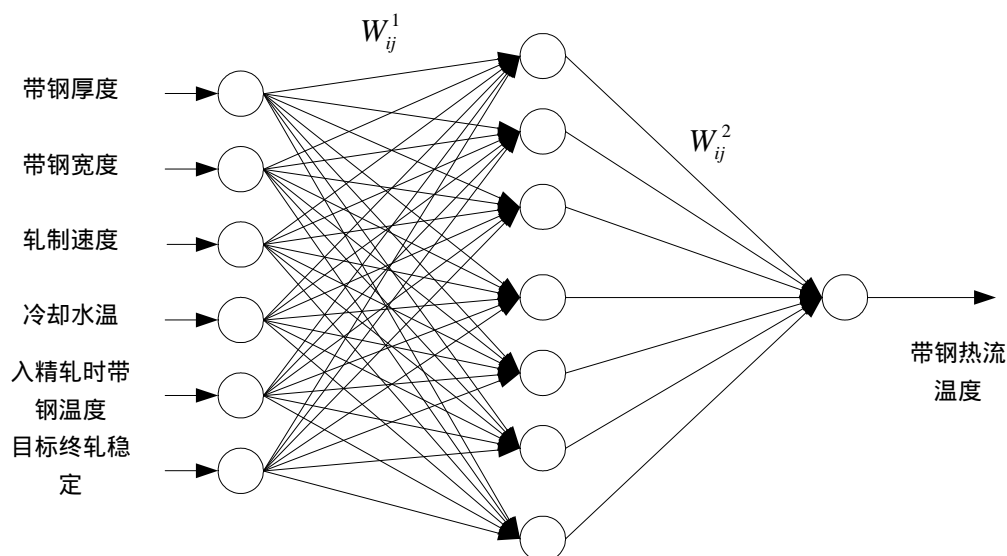


图 2-15 BP 神经网络的结构图

### 3. 应用结果及分析

采用前面所选定的神经网络模型、确定的参数和采自国内某热轧厂 50 卷带钢的生产数据的样本数据,用式(2-31)可以得到训练和学习样本中带钢热流密度的期望值。先用神经网络软件得到初步预测结果,再利用带钢热流密度的预测值计算出的终轧温度与实测值进行对比。可以看出:神经网络预测终轧温度计算值与实测值符合得很好。为了能够定量地评价预报精度,把预测值与实测值的标准差作为评价指标。通过对比计算可以得到统计结果:采用 BP 神经网络预报带钢热流密度后再计算的终轧温度预测值与实测值的标准差比只用数学模型的方法降低了 14.08%。由此可见:

将 BP 神经网络与数学模型相结合用来预测带钢终轧温度的方法效果明显,预测值与实测值的标准差比仅用数学模型的方法降低了 14.08%。

另外,此方法具有良好的在线应用前景。先通过建立实测数据库获取现场生产数据;再借助离线训练获得的不断更新的网络权值矩阵来计算机架间水冷区带钢的热流密度,就可在在线预测带钢终轧温度,达到提高终轧温度控制精度和产品质量的目的。

#### 2.3.4 BP 神经网络在工程建设安全管理中的应用<sup>[19]</sup>

##### ——基于人工神经网络的工程建设安全管理效果评价

安全因素被列为建设工程的三大主题(质量、安全、环境)之一,一直是困扰建设项目顺利实施的重要因素。安全管理作为建设项目管理的组成部分,越来越受到行业主管部门和施工企业的重视。工程建设安全管理关系到项目本身、施工企业、社会等方方面面的问题,是人民生命与健康和国家财产的基本保障。特别是在 1999 年 4 月第 15 届世界职业安全卫生大会上,国际劳工组织(ILO)负责人指出,IL0 将像贯彻 ISO9000 和 ISO14000 进行认证那样,研究企业职业安全卫生管理的评价。由此工程建设安全管理效果的评价成为亟待解决的问题。此处对反映工程建设安全管理效果的状态因素进行分析与处理,应用 BP 神经网络评价具体工程的安全管理效果,以期对工程建设安全管理实践有所帮助。进而提高建筑行业主管部门和施工企业的安全管理水平,促进我国企业进入国际市场。

##### 1. 评价指标的确定

评价指标的初选

反映工程建设安全管理效果的指标有很多,各个指标之间又相互关联、相互影响。本例指标的设置以施工现场安全状态检查表 JGJ59—99《建筑施工安全检查标准》和专家意见为基础进行修正、补充和综合得出<sup>[20~21]</sup>。具体选择的指标如下:

$t_1$ : 公司管理机构对施工安全的重视;  $t_2$ : 施工复杂性;  $t_3$ : 安全计划;  $t_4$ : 工程现场环境条件;  $t_5$ : 安全检查,包括检查的次数和质量;  $t_6$ : 特殊自然环境施工条件下的施工安排与措施;  $t_7$ : 安全预算;  $t_8$ : 专职安全管理人员,包括数量和素质,职责划分和授权等;  $t_9$ : 操作人员的素质;  $t_{10}$ : 分包公司的管理;  $t_{11}$ : 施工管理人员的经验;  $t_{12}$ : 安全教育;  $t_{13}$ : 机械设备的状态与检查维修计划;  $t_{14}$ : 安全会议,包括和业主、监理、分包公司之间的正式安全会议。

对于不同的工程项目和具体情况(如特殊环境、特殊工程)相应的指标可以有所增减。

评价指标的分析与处理

反映工程建设安全管理水平的指标由于具体工程的不同会有所不同,同时他们所起的作用也各不相同,且各指标之间有着不同程度的关联,因此需要进行一定的处理。

首先对各项指标进行赋值,然后应用主成份分析方法选出那些与安全管理水平关联最密切的指标。即对赋值的指标进行主成份分析,将主成份对总体贡献较小的指标先删除掉,直至最小指标的特征值大于事先约定的特征值为止。这时保留下来的指标就是要入选的“最佳指标”集合,这些指标基本保留了反映安全管理水平的大部分信息。这种指标处理的方法可以使应用的数据更客观准确。

(1) 对指标赋值应该遵循以下原则:

指标的赋值应尽可能具有客观性。即赋值的原始信息应当直接来自客观环境,赋值大小应能体现指标量大小和它们之间的关系。

(2) 用主成份分析法确定最佳指标集合

设安全管理水平用  $m$  个状态参量  $x_1, x_2, \dots, x_m$  来表征。每一个指标有  $n$  个典型工程的测值,这  $m$  个指标的  $n$  个测值构成一个表征安全管理水平的状态矩阵

$$C = \{c_{ij}\}_{n \times m} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{bmatrix} \quad (2-32)$$

式中  $C_{ij}$ ——代表第  $j$  个指标在第  $i$  个工程的规范值。

其中  $C_{ij}$  的值是由专家打分后经过一定处理得出的。因为当指标由专家评判给出时,在很多情况下,专家很难给出一个确定的评价值,尤其是在指标的含义具有较大的模糊性时,专家更容易给出一个评价区间。为了使  $C_{ij}$  的确定更具客观性,对专家给出的区间值做如下处理:

设有  $k$  个专家,第  $h$  个专家的评价区间为  $[u_1^{(h)}, u_2^{(h)}]$ , 其中  $u_1^{(h)} \neq u_2^{(h)}$ , 若  $u_1^{(h)} = u_2^{(h)} = u^*$ , 则将  $u^*$  按公式(2-33), (2-34)区间化处理成  $[u_1^*, u_2^*]$  的形式

$$u_1^* = u^* - \frac{1}{2k} \sum_{h=1}^{k'} [u_2^{(h)} - u_1^{(h)}] \quad (2-33)$$

$$u_2^* = u^* + \frac{1}{2k} \sum_{h=1}^{k'} [u_2^{(h)} - u_1^{(h)}] \quad (2-34)$$

式中  $u_1^*, u_2^*$ ——分别表示当专家打分为单一值,即  $u_1^{(h)} = u_2^{(h)} = u^*$  时,对  $u^*$  区间化处理后的两个端点值;  $k'$  - 专家打分为区间值时(即  $u_1^{(h)} \neq u_2^{(h)}$ )的区间个数。

此时,对于  $u_1^{(h)} = u_2^{(h)} = u^*$  来说,区间化后所确定的区间为  $[u_1^*, u_2^*]$ 。根据集值统计方法,专家对某个指标的群体评价值取为:

$$\bar{u} = \frac{1}{2} \frac{\sum_{h=1}^k [(u_2^{(h)})^2 - (u_1^{(h)})^2]}{\sum_{h=1}^k [u_2^{(h)} - u_1^{(h)}]} \quad (2-35)$$

为了考虑指标的客观性和可靠性,  $\bar{u}$  乘上一个表示指标可靠程度的权值  $b_i$  作为  $x_{ij}$  的值, 即

$$x_{ij} = \bar{u} * b_i \quad (2-36)$$

$b_i$  为指标  $x_i$  的总置信度 ( $i = 1, 2, \dots, m$ ), 计算公式:

$$b_i = \sum_{j=1}^n b_i^j / n \quad (2-37)$$

其中,  $n$  为典型工程的个数;  $b_i^j$  为评价工程  $j$  时得到的指标  $x_{ij}$  的置信度

$$b_{ji} = \frac{1}{1 + g_i} \quad (2-38)$$

$g_i$  为识别指标  $x_i$  时得到的  $g$ :

$$g = \frac{1}{3} \frac{\sum_{h=1}^k [(u_2^{(h)} - \bar{u})^3 - (u_1^{(h)} - \bar{u})^2]}{\sum_{h=1}^k [u_2^{(h)} - u_1^{(h)}]} \quad (2-39)$$

对处理完的  $x_{ij}$  按对数级差标准化公式规范化

$$C = \frac{\ln x_{ij} - \ln x_{j \min}}{\ln x_{j \max} - \ln x_{j \min}} \quad (i=1, 2, \dots, n; j=1, 2, \dots, m) \quad (2-40)$$

其中:  $x_{ij}$  代表第  $j$  个指标在第  $i$  个工程上处理后的指标值;  $x_{j \min}$  和  $x_{j \max}$  分别代表第  $j$  个指标相应于工程评价指数为 0 和 100 时的标准值。

根据相关关系公式

$$r_{ij} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{[\sum (x - \bar{x})^2][\sum (y - \bar{y})^2]}} \quad (2-41)$$

计算指标之间的相关系数, 由  $m$  个指标的两两相关系数可构成一个相关系数矩阵

$$R = \{r_{ij}\}_{m \times n} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{bmatrix} \quad (2-42)$$

求解相关系数矩阵尺的特征方程

$$R\gamma = \lambda\gamma \quad (2-43)$$

计算出  $R$  的特征值  $\lambda_j$  和相应的特征向量  $\gamma(j)$  ( $j = 1, 2, \dots, m$ )。则  $\gamma(j)$  是属于特征值  $\lambda_j$

的特征向量。将求出来的  $m$  个特征值从大到小顺序排列  $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ ，相应的特征值

$\lambda_j$  在  $m$  个特征总量  $\sum_{i=1}^m \lambda_i$  中占有的百分比就代表该主成分占有原始变量的总的信息量。一

般说来，前面少数几个主成份对于保留原始变量的信息程度最大。因此，可由前  $k$  个特征值的累积量占特征值的百分比

$$\rho = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^m \lambda_j} \quad (2-44)$$

来确定应选的主成份数， $\rho$  值可根据实际情况确定。

## 2. 评语集确定

设  $S = \{S_1, S_2, S_3, S_4\}$  为工程建设安全管理效果的状态集合， $S_i$  表示的状态分别为很好、好、一般、差。这些状态的取值由专家根据自己对工程的总体评价和经验通过评分确定。

设专家  $h$  对安全管理效果的评价区间为  $[v_1^{(h)}, v_2^{(k)}]$ ，则  $m$  个专家的群体评价值取为  $\bar{v}$ ， $\bar{v}$  的

具体数值根据上文的公式(2-33) ~ (2-39)计算确定。由  $\bar{v}$  的取值可以判定该工程安全管理效果的状况。但安全管理效果是一个模糊的概念，为了使  $\bar{v}$  能够更好地反映安全管理效果，用一个确切的数字区分两个相邻级别的状态标准是不适当的，两个相邻状态之间应具有模糊的

分界。所以，按上述安全管理效果状态集合  $S = \{S_1, S_2, S_3, S_4\}$  给出的四种状态实际上是四

个模糊子集，它们可分别用下列四个隶属函数及图 2-16 表示：

$$\mu_{S_1} = \begin{cases} 1 & 0 \leq \bar{v} < 50 \\ -\frac{\bar{v} - 60}{10} & 50 \leq \bar{v} < 60 \\ 0 & 60 \leq \bar{v} \end{cases} \quad (2-45)$$



$$\mu_{s_2} = \begin{cases} 0 & 50 > \bar{v} \\ -\frac{\bar{v}-50}{10} & 50 \leq \bar{v} < 60 \\ -\frac{\bar{v}-80}{20} & 60 \leq \bar{v} < 80 \\ 0 & 80 \leq \bar{v} \end{cases} \quad (2-46)$$

$$\mu_{s_3} = \begin{cases} 0 & 60 > \bar{v} \\ \frac{\bar{v}-60}{20} & 60 \leq \bar{v} < 80 \\ -\frac{\bar{v}-90}{10} & 80 \leq \bar{v} < 90 \\ 0 & 90 \leq \bar{v} \end{cases} \quad (2-47)$$

$$\mu_{s_4} = \begin{cases} 0 & 80 > \bar{v} \\ \frac{\bar{v}-80}{10} & 80 \leq \bar{v} < 90 \\ 1 & 90 \leq \bar{v} \end{cases} \quad (2-48)$$

以  $\bar{v}$  的值计算所得的  $\{\mu_{s_1}, \mu_{s_2}, \mu_{s_3}, \mu_{s_4}\}$  作为输出  $\{S_1, S_2, S_3, S_4\}$  的取值。

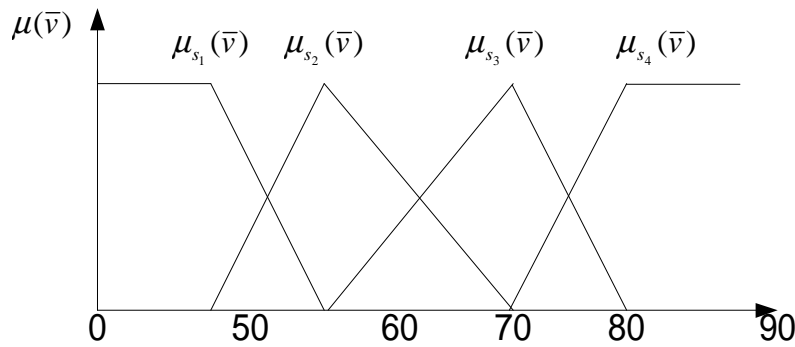


图 2-16 很好、好、一般、差四个状态的隶属函数

### 3. 评价方法

选用  $F$  个工程作为典型工程，应用 BP 神经网络进行训练。其输入根据上文“评价指标的确定”中得到的最佳指标集  $T = \{T_1, T_2, \dots, T_n\}$  确定；输出根据上文“评语集的确定”中得到的  $\{\mu_{s_1}, \mu_{s_2}, \mu_{s_3}, \mu_{s_4}\}$  确定。BP 神经网络训练的过程如下：

对于  $F$  个典型工程，得到  $F$  个样本。把  $T_i = \{t_{ij}\}$  作为网络的样本输入向量， $S_i$  作为样本的输出向量，对网络进行训练。选择图 2-17 所示的网络结构。

训练结束后将稳定的训练结果作为知识库，用以评价待识别工程的安全管理效果。即将待识别工程的最佳指标集  $T_{F+1}$  的数值输入稳定的 BP 神经网络，得到识别结果，对该工程的安全管理效果做出评价。

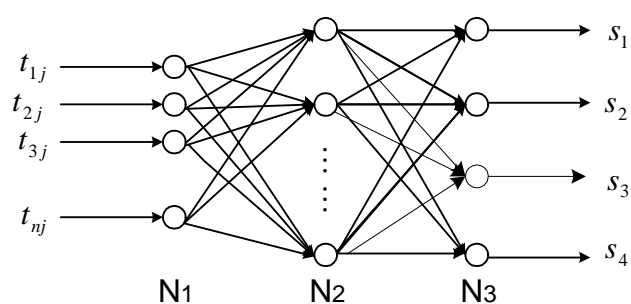


图 2-17 网络结构

#### 4. 工程实例

本例选取 8 个典型类似工程作为数据收集点，数据来源为协作单位协助提供。根据数据进行处理后采用的评价指标体系为  $\{t_1, t_3, t_5, t_7, t_{11}, t_{12}, t_{13}\}$ 。8 个典型工程的观测值处理后作为 8 个样本，对应的每个样本的相应的输入值  $t_{ij}$  为：

$$T_1 = [0.95 \ 0.88 \ 1.00 \ 0.85 \ 0.98 \ 0.99 \ 0.89]$$

$$T_2 = [0.96 \ 0.85 \ 0.96 \ 0.85 \ 0.87 \ 0.65 \ 0.72]$$

$$T_3 = [0.71 \ 0.86 \ 0.94 \ 0.92 \ 0.94 \ 0.96 \ 0.85]$$

$$T_4 = [0.88 \ 0.80 \ 0.63 \ 0.77 \ 0.79 \ 0.55 \ 0.70]$$

$$T_5 = [0.84 \ 0.96 \ 0.95 \ 0.84 \ 0.83 \ 0.91 \ 0.95]$$

$$T_6 = [0.85 \ 0.93 \ 0.46 \ 0.35 \ 0.85 \ 0.70 \ 0.74]$$

$$T_7 = [0.77 \ 0.78 \ 0.81 \ 0.80 \ 0.84 \ 0.86 \ 0.95]$$

$$T_8 = [0.81 \ 0.77 \ 0.98 \ 0.87 \ 0.84 \ 0.96 \ 0.86]$$

式中  $T_j$ ——为第  $j$  个工程的样本输入值 ( $j = 1, 2, \dots, 8$ )。

以专家对每个工程的评语集打分结果作为样本输出值，具体结果见表 2-12

表 2-12 输出值及输出状态计算结果

	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$
$\bar{v}$	0.95	0.89	0.83	0.77	0.86	0.68	0.80	0.85
$\mu_{s_1}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\mu_{s_2}$	0.00	0.00	0.00	0.15	0.00	0.60	0.00	0.00
$\mu_{s_3}$	0.00	0.10	0.70	0.85	0.40	0.40	1.00	0.50

$\mu_{s_4}$	1.00	0.90	0.30	0.00	0.60	0.00	0.00	0.50
评价	很好	很好	好	好	很好	中	好	(很)好

对网络进行训练，训练过程如下：

(1) 初始化。此处选用具有 7 个输入节点，5 个中间节点和 4 个输出节点的三层 BP 神经网络。

取动量项校正系数  $\alpha = 0.5$ ，初始权值取随机数，误差  $E < 0.02$ 。对网络进行训练。网络稳定后的权值作为知识库。

(2) 取  $T_1, T_2, T_3, T_4, T_5, T_6$  六组数值作为样本输入， $F_1 \sim F_6$  的相应数值  $\mu_{s_1} \sim \mu_{s_4}$  作为输出

值，对网络进行训练。用  $T_7 - T_8$  两个样本作为校验样本，在 BP 神经网络达到稳定后用

$T_7 - T_8$  作为待识别的工程进行检验，得到输出分别为 (0.0000, 0.0037, 0.9925, 0.0038)，

(0.0000, 0.0002, 0.5001, 0.4997)。输出结果与相应的由专家打分结果得到的输出

$\{\mu_{s_1}, \mu_{s_2}, \mu_{s_3}, \mu_{s_4}\}$  状态对比可知，所输出的数值误差满足精度要求且能够正确反映工程的实际状态。

(3) 取  $T_3, T_4, T_5, T_6, T_7, T_8$  六组数值作为样本输入， $F_3 \sim F_8$  的相应数值  $\mu_{s_1} \sim \mu_{s_4}$  作为网络

的输出值，对网络重新进行训练。在网络达到稳定时，用  $T_1 - T_2$  两个样本作为校验样本

得到输出分别为 (0.0000, 0.0000, 0.0000, 1.0000) 和 (0.0000, 0.0000, 0.0985, 0.9015)。

把输出结果和  $T_1, T_2$  两个样本的原来专家评价结果对比可知，误差同样满足精度要求且能够正确反映工程的实际状态。

(4) 取 8 个样本对该 BP 神经网络进行训练，待 BP 神经网络稳定后作为知识库。将待识别工

程的指标集  $T_9 = [0.88 \ 0.85 \ 0.91 \ 0.87 \ 0.82 \ 0.81 \ 0.85]$  输入，得到的输出结果为

[0.0000, 0.0000, 0.0810, 0.9190]。由此可以判定该工程的安全管理效果对很好的隶属度为 0.919，对好的隶属度为 0.081，总体评价为很好。

## 本章小结

本章介绍误差的反向传播 BP 神经网络的基本结构、标准学习算法及其局限性；改进的 BP 学习算法及 BP 神经网络的一般设计原则。最后通过四个实例展示了 BP 神经网络在故障模式识别、多传感器数据融合、工业过程自动化及管理工程方面的应用。通过本章的学习使读者不仅对 BP 算法有一较全面的了解，而且掌握如何利用神经网络来解决研究课题中遇到的实际问题。

## 参考文献

- [1] 朱大奇，计算机过程控制技术，南京，南京大学出版社，2001.12
- [2] 朱大奇，航空电子设备故障诊断新技术研究，博士论文，南京航空航天大学，2002.6.
- [3] 徐丽娜，神经网络控制，电子工业出版社，2003 年 2 月，北京。
- [4] 韩力群，人工神经网络的理论、设计及应用，北京，化学工业出版社，2002.1。
- [5] 朱大奇，电子设备故障诊断的原理与实践，电子工业出版社，2004 年 1 月，北京。

- [6] Zhu Daqi, Yu Shenglin, Neural Networks Data Fusion Algorithm of Electronic Equipment Fault Diagnosis, The 5<sup>th</sup> World Congress on Intelligent Control and Automation(WCICA2004), june, 14-18, 2004, HangZhou, China.
- [7] Zhu Daqi, Yu Shenglin, Shi Yu, The Studies of Analog Circuit Fault Diagnosis Based Multi-Sensors Neural Network Data Fusion Technology, DCDIS Proceedings 1(2003)73-77, Intelligent and Complex Systems, Watam Press.
- [8] 朱大奇, 纪志成, 基于多智能体的复杂工程系统故障诊断研究, 计算机集成制造系统—CIMS, 2004, 10(6): 699-704.
- [9] Martin T.H., Howard B.D., Mark H.B.著, 戴葵等译, 神经网络设计, 机械工业出版社, 2002年9月, 北京。
- [10] 朱大奇, 于盛林, 电子电路故障诊断的神经网络数据融合算法, 东南大学学报, 2001, 31(6): 87-91.
- [11] Vogl T.P., Mangis J.K., Zigler A.K., Accelerating the convergence of the backpropagation method, Biological Cybernetics., 1988, 59(9):256-264.
- [12] Tollenaere T., SuperSAB: Fast adaptive back propagation with good scaling properties, Neural Networks, 1990, 3(5):561-573.
- [13] Simon Haykin 著, 叶世伟, 史忠植译, 神经网络原理, 机械工业出版社, 2004年5月, 北京。
- [14] 王浩, 庄钊文, 模糊可靠性分析中的隶属函数确定, 电子产品可靠性与环境试验, 2000, 10(4): 2-7.
- [15] 张敏, 周旭东, 刘相华, 王国栋, 基于人工神经网络的热轧带钢热密度预测, 钢铁研究学报, 2004, 16(3): 75-78.
- [16] 艾立群, 人工神经网络在钢铁工业中的应用, 钢铁研究学报, 1997, 9(4): 60-63.
- [17] Liu Z.Y., Wang W.D., Gao W., Prediction of mechanical properties of hot-rolled C-Mn steels using artificial neural networks, Journal of materials processing technology, 1996, 57(2):332-336.
- [18] Sun X.G., Application of synergetic artificial intelligence to scheduling in finishing train of strip mills, Journal of materials processing technology, 1996, 60(5):405-408.
- [19] 苏义坤, 张守健, 孙三龙, 工程建设安全管理效果评价, 佳木斯大学学报, 2003, 21(4): 383-389.
- [20] 苏振民, 建筑施工安全状态的识别, 南京建筑工程学院学报, 2000, 10(3): 15-18.
- [21] Edward J., Jaselsks, Stuard D., Strategies for achieving excellence in safety performance, J. Constr. Engrg. And Magmt., 1996(1):61-70.

### 思考题

1. BP 网有哪些优点和不足? 试各列出三条。
2. 什么是 BP 网的泛化能力? 用自己理解的语言阐述。
3. 在标准 BP 算法中加入动量项, 其目的是什么?
3. 设计一个神经网络字符分类器对下图中的英文字母进行分类。输入向量含 16 个分量, 输出向量分别用  $(1, -1, -1)^T$ 、 $(-1, 1, -1)^T$  和  $(-1, -1, 1)^T$  代表字符 A、I、O。试用标准 BP 学习算法训练网络, 训练时可选择不同隐节点数及不同的学习率, 对达到同一训练误差的训练次数进行对比。

4. 以本章的实例 1 ( BP 神经网络在模拟电路故障诊断中应用 ) 的数据 , 在 MATLAB 中编写程序 , 对其结果进行验证。
5. 请给出一 BP 网络的应用实例。

## 第 3 章 Hopfield 反馈神经网络

Hopfield 递归网络是美国加州理工学院物理学家 J.J.Hopfield 教授于 1983 年提出的,他对神经网络学科的发展史颇具影响。作为一个著名的物理学家, J.J.Hopfield 在早期学术活动中,他曾研究光和固体之间的相互作用。后来,他集中精力研究生物分子间的电子转移机制。他在数学和物理学上的学术研究和他在生物学上经验的结合,为他在神经网络上提出的概念和所作出的贡献奠定了基础。

J.J.Hopfield 分别在 1982 年和 1984 年发表了两篇非常有影响的研究论文<sup>[1-2]</sup>, 在这两篇文章中集中了许多前人的观点,如 MP 神经网络模型<sup>[3]</sup>, Grossberg 的改进模型<sup>[4]</sup>, Anderson 和 Kohonen 的线性联想器模型等<sup>[5-6]</sup>, 并把一些重要思想结合起来进行简明的数学分析,其重要内容之一就是**在反馈神经网络中引入了“能量函数”的概念,这一概念的提出对神经网络的研究有重大意义,它使神经网络运行稳定性的判定有了可靠的依据。并应用 Hopfield 网络成功地求解了优化组合问题中最有代表性的 TSP 问题<sup>[7]</sup>**,从而开创了神经网络用于智能信息处理的新途径。本章讨论 Hopfield 网络的基本原理及其应用。

Hopfield 网络按网络输入和输出的数字形式不同可分为离散型和连续型两种网络,即:

离散型 Hopfield 神经网络——DHNN(Discrete Hopfield Neural Network);

连续型 Hopfield 神经网络——CHNN(Continues Hopfield Neural Network)。

下面介绍其原理与结构。

### 3.1 离散 Hopfield 神经网络模型

#### 3.1.1 DHNN 结构与工作方式

##### 1. DHNN 结构

它是一种单层全反馈网络,共有  $n$  个神经元。每个神经元都通过连接权接收所有其它神经元输出反馈来的信息,其目的是为了任一神经元的输出能接受所有神经元输出的控制,从而使各神经元能相互制约。其结构如图 3-1 所示。

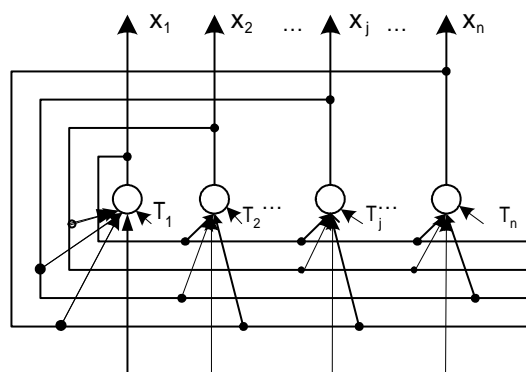


图 3-1 DHNN 结构

$T_j$ ——神经元  $j$  的阈值;  $net_j$ ——神经元  $j$  的净输入;  $w_{ij}$ ——神经元  $i$  与  $j$  的连接权值。

##### 2. DHNN 网络状态及工作方式

###### 网络状态

DHNN 中每个神经元都有相同的功能,其输出称为状态,状态用  $x_j$  表示 ( $x_j$ ——神经

元  $j$  状态输出), 所有神经元状态的集合就构成反馈网络的状态  $X = [x_1, x_2, \dots, x_n]$ , 反馈网络的输入就是网络的初始状态值, 可以表示为:

$X(0) = [x_1(0), x_2(0), \dots, x_n(0)]$ 。网络在外界激发作用下, 从初始状态进入动态演变过程,

$x_j(t+1) = f(net_j)$ ,  $f(\cdot)$ ——转移函数, DHNN 的转移函数常用符号函数表达, 也有用对称饱和线性函数来表达, 符号函数:

$$x_j(t+1) = \text{sgn}(net_j) = \begin{cases} 1 & net_j \geq 0 \\ -1 & net_j < 0 \end{cases} \quad (3-1)$$

对称饱和线性函数:

$$x_j(t+1) = \text{satlins}(net_j) = \begin{cases} -1 & net_j < -1 \\ net_j & -1 \leq net_j \leq 1 \\ 1 & net_j > 1 \end{cases} \quad (3-2)$$

净输入  $net_j = \sum_{i=1}^n (w_{ij}x_i - T_j)$ ; 对于 DHNN 来说, 一般有  $w_{ii} = 0, w_{ij} = w_{ji}$ ——对称矩阵

网络稳定时, 每个神经元的状态都不再改变, 此时的稳定状态就是网络的输出。

DHNN 异步方式 ——串行工作方式

每次只有一个神经元按式 (3-1) 或 (3-2) 进行状态的调整计算。其他保持不变。

$$x_j(t+1) = \begin{cases} \text{sgn}[net_j(t)] & j = i \\ x_j(t) & j \neq i \end{cases} \quad (3-3)$$

$$x_j(t+1) = \begin{cases} \text{satlins}[net_j(t)] & j = i \\ x_j(t) & j \neq i \end{cases} \quad (3-4)$$

神经元状态的调整次序可以按某种规定的次序进行, 也可随机选定。每次神经元在调整状态时, 根据其当前的净输入值的大小决定下一时刻的状态, 因此其状态可能发生变化, 也可能保持原状。下次调整其他神经元状态时, 本次的调整结果即在下一个神经元的净输入中发挥作用。

DHNN 同步方式——并行工作方式

所有神经元同时按式 (3-1) 或 (3-2) 进行状态的调整计算。即:

$$x_j(t+1) = \text{sgn}[net_j(t)] \quad j = 1, 2, \dots, n \quad (3-5)$$

$$x_j(t+1) = \text{satlins}[net_j(t)] \quad j = 1, 2, \dots, n \quad (3-6)$$

和异步工作方式相同, 每次神经元在调整状态时, 根据其当前的净输入值的大小决定下一时刻的状态, 下次调整其他神经元状态时, 本次的调整结果即在下一个神经元的净输入中发挥作用。网络稳定时, 每个神经元的状态都不再改变, 此时的稳定状态就是网络的输出。

为直观地说明离散型 Hopfield 神经网络的运行过程, 此处给出一离散型 Hopfield 神经网络进行简单模式分类的实例<sup>[8]</sup>。

例 3-1 基于离散型 Hopfield 神经网络的简单模式分类器

### (1) 问题提出

某水果货仓存储着各种水果，不同类型的水果可能会混淆在一起，所以希望设计一台能够将水果自动分类的仪器。分类时可以从水果三个方面进行识别：外形、质地和重量。如果水果基本上是圆形的，其外形测试传感器输出为“1”；如果水果更接近于椭圆形的，其外形测试传感器输出为“-1”。如果水果表面光滑，其质地测试传感器输出为“1”；如果水果表面比较粗糙，其质地测试传感器输出为“-1”。当水果的重量超过1磅时，重量测试传感器的输出为“1”；当水果的重量轻于1磅时，重量测试传感器的输出为“-1”。

将三个测试传感器的输出输入到设计好的 Hopfield 神经网络分类器中，待网络稳定后，其输出则对应水果的类型。为简单起见，现假设只有两类水果：苹果和橘子。每个水果用一三维向量表示：

$$X = \begin{bmatrix} \text{外形} \\ \text{质地} \\ \text{重量} \end{bmatrix} \quad (3-7)$$

一个标准的橘子可以表示为：

$$X_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \quad (3-8)$$

一个标准的苹果可以表示为：

$$X_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \quad (3-9)$$

### (2) Hopfield 神经网络分类器设计

Hopfield 神经网络的设计在本章第3.3节中将有较仔细的讨论。此处为了说明网络的作用和工作过程，直接给出网络的权矩阵和阈值矩阵：

$$W = \begin{bmatrix} 0.2 & 0.0 & 0.0 \\ 0.0 & 1.2 & 0.0 \\ 0.0 & 0.0 & 0.2 \end{bmatrix}, \quad T = \begin{bmatrix} -0.9 \\ 0.0 \\ 0.9 \end{bmatrix} \quad (3-10)$$

对式(3-10)给出的参数，利用对称饱和和线性函数可以将 Hopfield 神经网络的运算关系写为：

$$\begin{cases} x_1(t+1) = \text{satlins}(0.2x_1(t) + 0.9) \\ x_2(t+1) = \text{satlins}(1.2x_2(t)) \\ x_3(t+1) = \text{satlins}(0.2x_3(t) - 0.9) \end{cases} \quad (3-11)$$

这里希望网络输出要么收敛于橘子的标准模式  $X_1$ ，要么收敛于苹果的标准模式  $X_2$ 。两个模式的标准向量的第一个元素均为1，第三个元素均为-1，两者不同之处在于第二个元素。所以无论给网络输入什么模式，均希望输出模式的第一个元素均为1，第三个元素均为-1，而第二个元素要么收敛于-1，要么收敛于1，使之最接近输入向量的第二个元素。

从式(3-11)看，无论网络初始值是多少，第一个元素的值将不断增加直到最后的值为1；第三个元素的值将不断减少直到最后的值为-1；第二个元素乘上一个大于1的数，当第二个元素的初始值为负数时，它将收敛于-1，当第二个元素的初始值为正数时，它将收敛



于 1。必须指出这里给出的 (3-10) 式并非唯一。

### (3) 网络分类功能检测

此处用椭圆形橘子的实例对 Hopfield 神经网络分类器进行检测。此时待识别水果的特征向量可以表示为：

$$x(0) = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \text{ 也即 Hopfield 神经网络初始输入。输入设计好的网络, 前三个迭代过程}$$

结束时, Hopfield 神经网络分类器的输出分别为：

$$x(1) = \begin{bmatrix} 0.7 \\ -1 \\ -1 \end{bmatrix}; \quad x(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}; \quad x(3) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \quad (3-12)$$

从式 (3-12) 可见, Hopfield 神经网络最终收敛于橘子模式。也是网络的稳定状态。完成了水果的分类功能。

### 3.1.2 网络的稳定性与吸引子<sup>[15-16]</sup>

作为一个非线性动力学系统, Hopfield 神经网络在任一初态下最终可能收敛于一个稳定状态, 也可能产生振荡构成极限环。研究神经网络的演变过程, 了解稳定状态存在的规律有助于进一步认识 Hopfield 神经网络用于优化计算及联想记忆的许多性能。

对联想记忆来说, 如果把稳定状态视为一个记忆样本, 那么从初态朝稳态收敛的过程就是寻找记忆样本的过程。初态可以认为是给定样本的部分信息, 神经网络的演变过程可从初态部分信息找到全部信息, 以实现联想记忆; 对优化计算来说, 如果将稳定状态与某种优化计算的目标函数相对应, 作为神经网络能量函数的极小点, 那么优化计算过程就是从初态寻找能量函数极小点的过程, 优化计算的目的就在系统的演变过程中自动完成。

无论是优化计算还是联想记忆, 只有神经网络是稳定的时候, 才具有实际的意义。反馈网络作为非线性动力学系统, 具有丰富的动态特性, 如稳定性、有限环状态和混沌 (chaos) 状态等。下面着重讨论异步工作方式的稳定性。

#### 1. 吸引子的基本概念

由网络的工作状态的分析可知, DHNN 网实质上是一个离散的非线性动力学系统, 网络从初态  $X(0)$  开始, 若能经有限次递归后, 其状态不再发生变化, 即  $X(t+1) = X(t)$ , 则称该网络是稳定的。网络的稳定状态也称为“吸引子” (attractor or fixed-point)。如果网络是稳定的, 它可以从任一初态收敛到一个稳态, 如图 3-2 (a) 所示; 若网络是不稳定的, 由于 DHNN 网每个节点的状态只有 1 和 -1 两种情况, 网络不可能出现无限发散的情况, 而只能出现限幅的自持振荡, 这种网络称为有限环网络, 图 3-2 (b) 给出了它的相图。如果网络状态的轨迹在某个确定的范围内变迁, 但既不重复也不停止, 状态变化为无穷多个, 轨迹也不发散到无穷远, 这种现象称为混沌, 其相图如图 3-2 (c) 所示。对于 DHNN 网, 由于网络的状态是有限的, 因此不可能出现混沌现象。

网络达到稳定时的状态  $X$ , 称为网络的吸引子。一个动力学系统的最终行为是由它的吸引子决定的, 吸引子的存在为信息的分布存储记忆和神经优化计算提供了基础。如果把吸引子视为问题的解, 那么从初态朝吸引子演变的过程便是求解计算的过程, 若把需记忆的样本信息存储于网络不同的吸引子, 当输入含有部分记忆信息的样本时, 网络演变过程便是从部分信息寻找全部信息, 即联想回忆的过程。

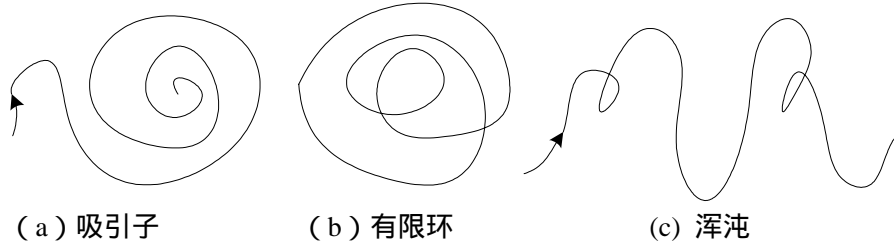


图 3-2 反馈网络的 3 种象图

下面给出 DHNN 网吸引子的定义和定理。

**定义 3-1** 若网络的状态  $X$  满足  $X = f(WX - T)$ ，则称  $X$  为网络的吸引子。

**定理 3-1** 按异步方式调整网络状态，对于任意初态离散 Hopfield 神经网络，网络都最终收敛到一个吸引子。

证明：为了证明定理 3-1 首先需要证明能量函数  $E(t)$  经有限次异步演变会收敛到一个常数。

能量函数  $E(t)$  为：

$$E(t) = -\frac{1}{2} X^T(t)WX(t) + X^T(t)T \quad (3-13)$$

令网络能量函数的改变量为  $\Delta E$ ，网络状态的改变量为  $\Delta X$ ，有

$$\Delta E(t) = E(t+1) - E(t) \quad (3-14)$$

$$\Delta X(t) = X(t+1) - X(t) \quad (3-15)$$

将式 (3-13) 和式 (3-15) 代入 (3-14)，并展开，则有：

$$\begin{aligned} \Delta E(t) &= E(t+1) - E(t) \\ &= -\frac{1}{2} [X(t) + \Delta X(t)]^T W [X(t) + \Delta X(t)] + [X(t) + \Delta X(t)]^T T - \left[ -\frac{1}{2} X^T(t)WX(t) + X^T(t)T \right] \\ &= -\Delta X^T(t)WX(t) - \frac{1}{2} \Delta X^T(t)W\Delta X(t) + \Delta X^T(t)T \\ &= -\Delta X^T(t)[WX(t) - T] - \frac{1}{2} \Delta X^T(t)W\Delta X(t) \end{aligned} \quad (3-16)$$

对异步工作方式来说，第  $t$  个时刻只有一个神经元调整状态，设该神经元为  $j$ ，将  $\Delta X(t) = [0, 0, \dots, 0, \Delta x_j(t), 0, \dots, 0]^T$  代入上式，并考虑到  $W$  为对称矩阵，有：

$$\Delta E(t) = -\Delta x_j(t) \left[ \sum_{i=1}^n (w_{ij} - T_j) \right] - \frac{1}{2} \Delta x_j^2(t) w_{jj} \quad (3-17)$$

对 DHNN 来说，各种神经元不存在自反馈，有  $w_{jj} = 0$ ，利用式 (3-3)，上式可简化为

$$\Delta E(t) = -\Delta x_j(t) net_j(t) \quad (3-18)$$

下面分析上式中可能出现的所有情况。

情况 1 :  $x_j(t) = -1, x_j(t+1) = 1$  , 由式 (3-15) 得  $\Delta x_j(t) = 2$  , 由式 (3-1) 知 ,  $net_j(t) \geq 0$  , 代入式 (3-18) , 得  $\Delta E(t) \leq 0$ 。

情况 2 :  $x_j(t) = 1, x_j(t+1) = -1$  , 由式 (3-15) 得  $\Delta x_j(t) = -2$  , 由式 (3-1) 知 ,  $net_j(t) < 0$  , 代入式 (3-18) , 得  $\Delta E(t) < 0$ 。

情况 3 :  $x_j(t) = x_j(t+1)$  , 所以  $\Delta x_j(t) = 0$  , 代入式 (3-17) , 则有  $\Delta E = 0$ 。

以上三种情况包括了式 (3-18) 可能出现的所有情况 , 由此可知在任何情况下均有  $\Delta E(t) \leq 0$ 。也就是说 , 在网络动态演变过程中。能量总是在不断下降或保持不变。由于网络中各节点的状态只能取 1 或 -1 , 能量函数  $E(t)$  作为网络状态的函数是有下界的 , 因次网络能量函数最终将收敛于一个常数 , 此时  $\Delta E(t) = 0$ 。

下面分析当  $E(t)$  收敛于常数时 , 是否对应于网络的稳态。当  $E(t)$  收敛于常数时 , 有  $\Delta E(t) = 0$ 。此时对应于以下两种情况。

情况 1 :  $x_j(t) = x_j(t+1) = 1$  , 或  $x_j(t) = x_j(t+1) = -1$  , 这种情况下神经元  $j$  的状态不再改变 , 表明网络已进入稳态 , 对应的网络状态就是网络的吸引子。

情况 2 :  $x_j(t) = -1, x_j(t+1) = 1, net_j = 0$  , 这种情况下网络继续演变时 ,  $x_j = 1$  将不会变化。因为如果  $x_j$  由 1 变到 -1 , 则有  $\Delta E(t) < 0$  , 与  $E(t)$  收敛于常数的情况 (  $\Delta E(t) = 0$  ) 相矛盾。

综上所述 , 当网络工作方式和权矩阵均满足定理 3-1 条件时 , 网络最终将收敛到一个吸引子。

事实上 , 对  $w_{jj}$  的规定是为了数学推导的简便 , 如不做此规定 , 上述结论仍然成立。此外当神经元状态取 1 和 0 时 , 上述结论也将成立。

**定理 3-2** 对于 DHNN 网 , 若按同步方式调整网络状态 , 且连接权矩阵  $W$  为非负定对称阵 , 则对于任意初态 , 网络都最终收敛到一个吸引子。

**证明 :** 由式 (3-16) 得 :

$$\begin{aligned}\Delta E(t) &= E(t+1) - E(t) = \Delta X^T(t) [WX(t) - T] - \frac{1}{2} \Delta X^T(t) W \Delta X(t) \\ &= -\Delta X^T(t) net(t) - \frac{1}{2} \Delta X^T(t) W \Delta X(t) \\ &= -\sum_{j=1}^n \Delta x_j(t) net_j(t) - \frac{1}{2} \Delta X^T(t) W \Delta X(t)\end{aligned}$$

前已证明，对于任何神经元  $j$ ，有  $-\Delta x_j(t)net_j(t) \leq 0$ ，因此上式第一项不大于 0，只要  $W$  为非负定对称阵，第二项也不大于 0，于是有  $\Delta E(t) \leq 0$ 。也就是说  $E(t)$  最终将收敛到一个常数值，对应的稳定状态是网络的一个吸引子。

比较定理 3-1 和 3-2 可以看出，网络采用同步方式工作时，对权值矩阵  $W$  的要求更高，如果  $W$  不能满足非负定对称阵的要求，网络会出现振荡，异步方式比同步方式有更好的稳定性，应用中较多采用，但其缺点是失去了神经网络并行处理的优势。

## 2. 吸引子的性质

下面介绍吸引子的几个性质，主要介绍吸引子可能存在或不可能存在的条件。

**性质 3-1** 若  $X$  是网络的一个吸引子，且阈值  $T = 0$ ，在  $\text{sgn}(0)$  处， $x_j(t+1) = x_j(t)$ ，则  $-X$  也一定是该网络的一个吸引子。

**证明：** 因为  $X$  是吸引子，即  $X = f(WX)$ ，从而有

$$f[W(-X)] = f[-WX] = -f[WX] = -X$$

所以  $-X$  也是该网络的吸引子。

**性质 3-2** 若  $X^a$  是网络的一个吸引子， $w_{ii} = 0$ ，且  $\text{sgn}(0) = 1$  的条件下，则与  $X^a$  的海明距离  $d_H(X^a, X^b) = 1$  的  $X^b$  一定不是吸引子。

**证明：** 两个向量的海明距离  $d_H(X^a, X^b)$  是指两个向量中不相同元素的个数。不妨设

$x_1^a \neq x_1^b, x_j^a = x_j^b, j = 2, 3, \dots, n$ 。因为  $w_{11} = 0$ ，由吸引子定义，有：

$$x_1^a = f\left(\sum_{i=2}^n w_{1i}x_i^a - T_1\right) = f\left(\sum_{i=2}^n w_{1i}x_i^b - T_1\right)$$

$$x_1^b \neq f\left(\sum_{i=2}^n w_{1i}x_i^b - T_1\right)$$

## 3. 吸引域的基本概念

为正确实现联想记忆，当样本状态在一定范围内逃离吸引子之后，应有能力自动调整返回原状态。或者说，对于引入噪声、产生缺损之样本状态，网络应有能力自动联想而恢复到吸引子状态，以实现正确的识别。概括来讲，吸引子应具有一定的吸引范围或称吸引域 (domain of attraction)，否则，对于吸引子的研究将失去意义。下面给出吸引域的基本定义。

**定义 3-2** 设  $X^a$  为吸引子，若存在一条路径可以从  $X$  演变到  $X^a$ ，则称  $X$  弱吸引到  $X^a$ ；

若从  $X$  开始每条演变路径都可以到达  $X^a$ ，则称这些  $X$  强吸引到  $X^a$ ；

**定义 3-3** 若对某些  $X$ ，有  $X$  弱吸引到  $X^a$ ，则称这些  $X$  的集合为  $X^a$  的弱吸引域；若对某些  $X$ ，有  $X$  强吸引到  $X^a$ ，则称这些  $X$  的集合为  $X^a$  的强吸引域。

**例 3-2** 有一 DHNN 网络,  $n=4$ ,  $T_j=0$ ,  $j=1, 2, 3, 4$ , 向量  $X^a$ 、 $X^b$  和权值矩阵  $W$  分别为

$$X^a = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad X^b = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 2 & 2 \\ 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{bmatrix}$$

检验  $X^a$  和  $X^b$  是否为网络的吸引子, 并考察其是否具有联想记忆能力。

**解:** 检验吸引子

由吸引子定义

$$f(WX^a) = f \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} \text{sgn}(6) \\ \text{sgn}(6) \\ \text{sgn}(6) \\ \text{sgn}(6) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = X^a$$

所以  $X^a$  是网络的吸引子, 因为  $X^b = -X^a$ , 由吸引子的性质 3-1 知,  $X^b$  也是网络的吸引子。

考察联想记忆能力

设有样本  $X^1 = (-1, 1, 1, 1)^T$ ,  $X^2 = (1, -1, -1, -1)^T$ ,

$X^3 = (1, 1, -1, -1)^T$  试考察网络以异步方式工作时两个吸引子对三个样本的吸引能力。

令网络初态  $X(0) = X^1 = (-1, 1, 1, 1)^T$ 。

设神经元状态调整次序  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , 则有:

$X(0) = X^1 = (-1, 1, 1, 1)^T \rightarrow X(1) = (1, 1, 1, 1)^T = X^a$ 。可以看出该样本

比较接近吸引子  $X^a$ , 事实上只按异步方式调整了一步, 样本  $X^1$  即收敛于  $X^a$ 。

令网络初态  $X(0) = X^2 = (1, -1, -1, -1)^T$ 。

设神经元状态调整次序  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , 则有:

$X(0) = X^2 = (1, -1, -1, -1)^T \rightarrow X(1) = (-1, -1, -1, -1)^T = X^b$ 。可以

看出该样本比较接近吸引子  $X^b$ , 事实上只按异步方式调整了一步, 样本  $X^2$  即收敛于  $X^b$ 。

令网络初态  $X(0) = X^3 = (1, 1, -1, -1)^T$ 。它与两个吸引子的海明矩阵相等。若设神经

元状态调整次序为  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , 则有:

$X(0) = X^3 = (1, 1, -1, -1)^T \rightarrow X(1) = (-1, 1, -1, -1)^T \rightarrow X(2) =$

$(-1, -1, -1, -1)^T = X^b$ 。若将神经元状态调整次序改为  $3 \rightarrow 4 \rightarrow 1 \rightarrow 2$ ，则有：

$$X(0) = X^3 = (1, 1, -1, -1)^T \rightarrow X(1) = (1, 1, 1, -1)^T \rightarrow X(2) = (1, 1, 1, 1)^T = X^a。$$

从本例可以看出，当网络的异步调整次序一定时，最终稳定于哪个吸引子与其初态有关；而对于确定的初态，网络最终稳定于哪个吸引子与其异步调整次序有关。

### 3.1.3 DHNN 的设计原则

吸引子的分布是由网络的权值（包括阈值）决定的，设计吸引子的核心就是如何设计一组合适的权值。为了使所设计的权值满足要求，权值矩阵应符合以下要求：

- （1） 为保证异步方式工作时网络收敛， $W$  应为对称阵；
- （2） 为保证同步方式工作时网络收敛， $W$  应为非负定对称阵；
- （3） 保证给定的样本是网络的吸引子，并且要有一定的吸引域。

具体设计时，可以采用以下不同的方法。

#### 1. 联立方程法

以 3 节点 DHNN 网为例，说明权值设计的联立方程法。要求设计的吸引子为  $X^a = (010)^T$  和  $X^b = (111)^T$ ，权值和阈值在  $[-1, 1]$  区间取值，试求权值和阈值。

考虑到  $w_{ij} = w_{ji}$ ， $w_{ii} = 0$ ，对稳态  $x(t+1) = x(t)$ 。对于状态  $X^a = (010)^T$ ，各节点净输入应满足：

$$net_1 = w_{12} \times 1 + w_{13} \times 0 - T_1 = w_{12} - T_1 < 0 \quad (3-19)$$

$$net_2 = w_{12} \times 0 + w_{23} \times 0 - T_2 = -T_2 > 0 \quad (3-20)$$

$$net_3 = w_{13} \times 0 + w_{23} \times 1 - T_3 = w_{23} - T_3 < 0 \quad (3-21)$$

对于  $X^b = (111)^T$  状态，各节点净输入应满足：

$$net_1 = w_{12} \times 1 + w_{13} \times 1 - T_1 > 0 \quad (3-22)$$

$$net_2 = w_{12} \times 1 + w_{23} \times 1 - T_2 > 0 \quad (3-23)$$

$$net_3 = w_{13} \times 1 + w_{23} \times 1 - T_3 > 0 \quad (3-24)$$

联立以上 6 项不等式，可求出 6 个未知量的允许取值范围。如取  $w_{12} = 0.5$ ，则由式 (3-19)，有  $0.5 < T_1 \leq 1$ ，取  $T_1 = 0.7$ ；由式 (3-22)，有  $0.2 < w_{13} \leq 1$ ，取  $w_{13} = 0.4$ ；由式 (3-20)，有  $-1 < T_2 < 0$ ，取  $T_2 = -0.2$ ；由式 (3-23)，有  $-0.7 < w_{23} \leq 1$ ，取  $w_{23} = 0.1$ ；由式 (3-24)，

有  $-1 < T_3 < 0.5$ ，取  $T_3 = 0.4$ ；可以验证，利用这组参数构成的 DHNN 网对于任何初态最终

都将演变到两吸引子  $X^a = (010)^T$  和  $X^b = (111)^T$ 。

## 2. 外积和法

当所需要的吸引子较多时，可采用下面的 Hebb 规则的外积和法。设给定  $m$  个模式样本  $X^p$ ， $p = 1, 2, \dots, m$ ， $x \in \{-1, 1\}^n$ ，并设样本两两正交，且  $n > m$ ，则权值矩阵为记忆样本的外积和

$$W = \sum_{p=1}^m X^p (X^p)^T \quad (3-25)$$

若取  $w_{jj} = 0$ ，上式应写为

$$W = \sum_{p=1}^m [X^p (X^p)^T - I] \quad (3-26)$$

式中  $I$  为单位矩阵。上式写成分量元素形式，有

$$W_{ij} = \begin{cases} \sum_{p=1}^m x_i^p x_j^p, & i \neq j \\ 0, & i = j \end{cases} \quad (3-27)$$

按以上外积和规则设计的  $W$  阵必然满足对称性要求。下面检验所给样本能否称为吸引子。

因为  $m$  个样本  $X^p$  ( $p = 1, 2, \dots, m$ )， $x \in \{-1, 1\}^n$  是两两正交的，有：

$$\sum_p (X^p)^T X^k = \begin{cases} 0, & p \neq k \\ n, & p = k \end{cases}, \text{ 所以}$$

$$WX^k = \sum_{p=1}^m [X^p (X^p)^T - I] X^k = \sum_{p=1}^m [X^p (X^p)^T X^k - X^k]$$

$$= X^k (X^k)^T X^k - mX^k = nX^k - mX^k = (n-m)X^k。 \text{ 因为 } n > m, \text{ 所以有：}$$

$$f(WX^p) = f[(n-m)X^p] = \text{sgn}[(n-m)X^p] = X^p。 \text{ 可见给定样本 } X^p (p = 1, 2, \dots, m)$$

是吸引子。需要指出的是，有些非给定样本也是网络的吸引子，它们并不是网络设计所要求的解，这种吸引子称为伪吸引子。

## 3.2 连续 Hopfield 神经网络

在连续型 Hopfield 神经网络中,所有神经元都随时间  $t$  并行更新,网络状态随时间连续改变。

### 1. 连续型 Hopfield 神经网络的结构

基于模拟电子线路的连续型 CHNN 拓扑结构如图 3-3 所示。

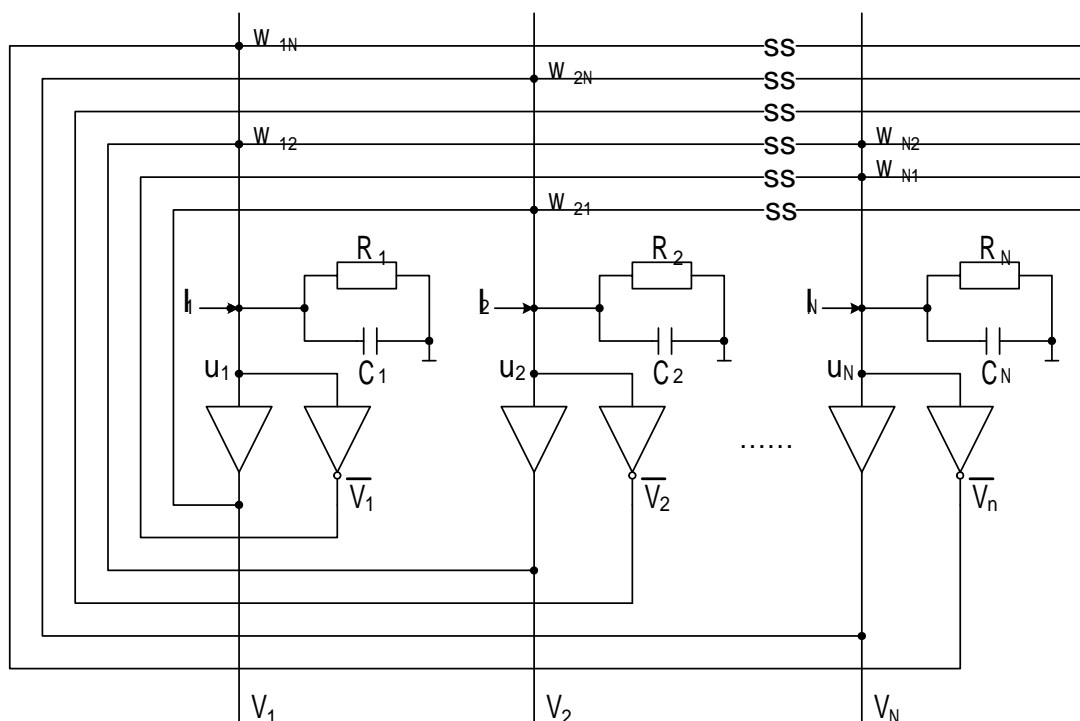


图 3-3 连续型 Hopfield 神经网络结构

图中电子线路与神经网络之间关系为：运算放大器——神经元；神经元输入——运放输入电压  $u_i$ ；神经元输出——运放输出电压  $V_i$ （输出有正向输出  $V_i$  和反向输出  $\bar{V}_i$ ）；

连接权  $W_{ij}$ ——输入端电导；阈值——输入偏置电流  $I_i$ 。

## 2. 连续型 Hopfield 神经网络的结构模型

图 3-4 为某种神经元  $j$  的结构图。由基尔霍夫定理可得：

$$C_j \frac{du_j}{dt} + g_j u_j = \sum_{i=1}^n W_{ij} (V_i - u_j) + I_j \quad (3-28)$$

$$\text{令 } \frac{1}{R_i} = \sum_{j=1}^n W_{ij} + g_j \text{ 得到:}$$

$$C_j \frac{du_j}{dt} = \sum_{i=1}^n W_{ij} V_i - \frac{u_j}{R_j} + I_j \quad (3-29)$$



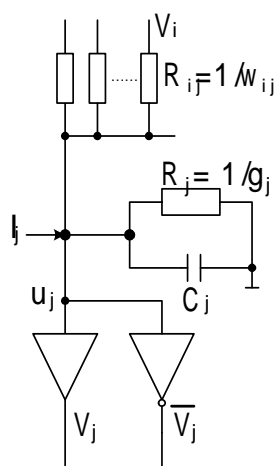


图 3-4 CHNN 神经元结构

### 3. 连续型 Hopfield 神经网络的能量函数与稳定性分析

CHNN 能量函数

$$\text{定义: } E = \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n W_{ij} V_i V_j - \sum_{j=1}^n V_j I_j + \sum_{i=1}^n \frac{1}{R_j} \int_0^{V_j} f^{-1}(V) dV \quad (3-30)$$

$f^{-1}$  为神经元转移函数的反函数。

CHNN 稳定性分析

**定理 3-3** 若神经元的转移函数  $f$  存在反函数  $f^{-1}$ ,  $f^{-1}$  是单调连续递增且有  $w_{ij} = w_{ji}$ , 则由

任意初态开始, CHNN 网络的能量函数总是单调递减的, 即  $\frac{dE}{dt} \leq 0$ , 当且仅当

$\frac{dV_j}{dt} = 0$  时, 有  $\frac{dE}{dt} = 0$  因而网络最终能达到稳态。

$$\text{证明: } \frac{dE}{dt} = \sum_{j=1}^n \frac{\partial E}{\partial V_j} \cdot \frac{dV_j}{dt} \quad \text{由能量函数的定义可得: } \frac{\partial E}{\partial V_j} = -\sum_{i=1}^n W_{ij} - I + \frac{u_j}{R_j} \Big|_{\substack{u_j = f^{-1} \\ W_{ij} = W_{ji}}}$$

$$\text{由(3-29)式可得: } \frac{\partial E}{\partial V_j} = -C_j \frac{du_j}{dt} \Rightarrow$$

$$\begin{aligned} \frac{dE}{dt} &= -\sum_{j=1}^n \frac{dV_j}{dt} C_j \frac{du_j}{dt} = -\sum_{j=1}^n C_j \cdot \frac{du_j}{dt} \cdot \frac{dV_j}{dt} \\ &= -\sum_{j=1}^n C_j \frac{du_j}{dV_j} \cdot \left( \frac{dV_j}{dt} \right)^2 = -\sum_{j=1}^n C_j f^{-1}(V_j) \left( \frac{dV_j}{dt} \right)^2 \end{aligned} \quad (3-31)$$

由于  $C_j > 0$ , 单调递增函数  $f^{-1}(V_j) > 0 \Rightarrow \frac{dE}{dt} \leq 0$

只有对于所有  $j$  均满足  $\frac{dV_j}{dt} = 0$  时有  $\frac{dE}{dt} = 0$

故网络稳定。即随着状态的演变，网络的能量总是降低的。当网络中所有节点的状态不再改变时，能量才不再改变，此时到达能量的某一局部极小点或局部最小，该能量对应着网络的某一稳定状态。

#### 4. Hopfield 网络的主要功能

Hopfield 神经网络的提出就是与其实用应用密切相关。其主要功能在以下两个方面。

##### 联想记忆

输入—输出模式的各元素之间，并不存在一对一的映射关系，输入—输出模式的维数也不要求相同；联想记忆时，只给出输入模式部分信息，就能联想出完整的输出模式。即具有容错性。如前面介绍的应用实例——离散 Hopfield 神经网络的模式分类器。

##### CHNN 的优化计算功能

优化计算在实际问题中有广泛的应用。如我们前面提到的 TSP 问题，工业生产和交通运输中的调度问题等。应用 Hopfield 神经网络来解决优化计算问题的一般步骤为：

- (1) 分析问题：网络输出与问题的解相对应。
- (2) 构造网络能量函数：构造合适的网络能量函数，使其最小值对应问题最佳解。
- (3) 设计网络结构：将能量函数与标准式相比较，定出权矩阵与偏置电流。
- (4) 由网络结构建立网络的电子线路并运行，稳态——优化解或计算机模拟运行。

为说明 Hopfield 神经网络的优化计算功能，此处给出一 10 城市的 TSP 问题的实例。

#### 例 3-3 Hopfield 神经网络用于求解 TSP 问题

##### (1) 问题描述

旅行商最优路径问题是一个经典的人工智能难题。常称为 TSP(Travelling Salesman Problem)。它假定有  $n$  个城市  $A, B, C, \dots$ ，它们之间的相互距离分别为  $d_{AB}, d_{AC}, \dots, d_{BC} \dots$ 。

要求寻找一条闭合路径，此路径历经每个城市且仅经过一次，返回起始城市，要求此路径最短。图 3-5 为  $n = 10$  的一种可能路径，此时，路径的总长度为：

$$d = d_{AJ} + d_{JC} + d_{CB} + \dots + d_{EA} \quad (3-32)$$

不难求得在给定  $n$  的条件下，可能存在的闭合路径数目为  $\frac{1}{2}(n-1)!$ 。为了寻找最短路径，按照传统的求解方法利用搜索法需要找出全部路径的组合，随着  $n$  的增大，计算量会急剧增大，成为一个利用计算机也难以圆满解决的问题。

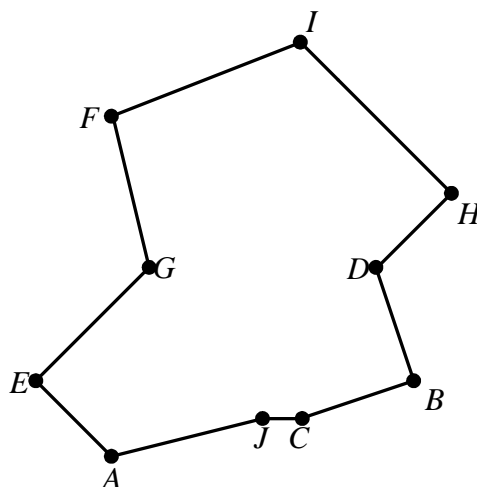


图 3-5  $n = 10$  的 TSP 路径示例

##### (2) TSP 问题的 Hopfield 神经网络的优化计算

## 1) 置换矩阵

为了建立用于表明 TSP 状态与结果的人工神经网络模型，首先需要构造  $n \times n$  矩阵，以  $n = 5$  为例，此时矩阵如表 3-1 所示。

表 3-1 置换矩阵

	1	2	3	4	5
A	0	1	0	0	0
B	0	0	0	1	0
C	1	0	0	0	0
D	0	0	0	0	1
E	0	0	1	0	0

表中 A、B、C、D、E (对应各行) 表示城市名称，1、2、3、4、5 (对应各列) 表示路径顺序。矩阵中元素“1”者表明该城市在路径全程中所居顺序，其余元素均为“0”。此处路径顺序为  $C \rightarrow A \rightarrow E \rightarrow B \rightarrow D \rightarrow C$ ，路径总长度为：

$$d = d_{CA} + d_{AE} + d_{EB} + d_{BD} + d_{DC} \quad (3-33)$$

表 3-1 给出的矩阵称为置换矩阵，它具有如下特点：

- \*每行只有一个“1”，其余元素均为“0”；
- \*每列只有一个“1”，其余元素均为“0”；
- \*全部元素中“1”的总和为  $n$ 。

利用  $n \times n$  个神经元组成 Hopfield 神经网络，网络达到稳定状态时各个神经元之状态对应置换矩阵的各个元素值。各城市间的距离  $d_{AB}, d_{AC}, \dots, d_{BC} \dots$  作为一组约束信息决定神经元之间的突触连接强度  $w_{ij}$ 。期望网络演变最终结果给出最优解，也即以置换矩阵表明最短距离  $d_{\min}$  条件下路径之顺序。

## 2) 能量函数

为便于和置换矩阵对应，各神经元输出电压  $V$  以双下标表示，写为  $V_{Xi}$ ，其中下标第一位  $X$ 、 $Y$  等表示城市名称，下标第二位  $i, j, \dots$  表示路径顺序。相应的放大器输入端电压用双下标表示，写作  $u_{Xi}$ 。

(a) 各城市距离信息将通过建立能量函数表达式产生对突触权重之约束。能量函数的建立原则上应满足以下两方面的要求：

- \*在最低能量状态神经元输出符合置换矩阵的形式；
- \*在最低能量状态置换矩阵对应最短路径。为满足第一项要求，能量函数表达式中应包括以下部分：

$$\frac{A}{2} \sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} + \frac{B}{2} \sum_i \sum_X \sum_{Y \neq X} V_{Xi} V_{Yi} + \frac{C}{2} (\sum_X \sum_i V_{Xi} - n)^2 \quad (3-34)$$

上式中 A、B、C 为实系数。A 系数项表示置换矩阵中所有各行每行内各元素两两交叉相乘，不包含自乘的各种组合。如果同一行中出现两个或两个以上的元素为“1”，则此项不能为零；当同一行内只有一个元素为“1”时，此项为零，这时在能量函数中 A 系数项达最低值。

B 系数项表示置换矩阵中所有各列每列内各元素两两交叉相乘，不包含自乘的各种组

合。如果同一列中出现两个或两个以上的元素为“1”，则此项不能为零；当同一列内只有一个元素为“1”时，此项为零，这时在能量函数中 B 系数项达最低值。

C 系数项表示置换矩阵中全部元素取和的情况，当全部元素取和等于  $n$  时，也即有  $n$  个元素为“1”时，此项为零，这时在能量函数中 C 系数项达最低值。

(b) 为满足距离信息之约束，能量函数中应包括以下部分：

$$\frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{Xi} (V_{Y,i+1} + V_{Y,i-1}) \quad (3-35)$$

为了符合置换矩阵形式，式 (3-35) 中  $V_{Xi} = 1$  时， $V_{Y,i+1}$  与  $V_{Y,i-1}$  只能有一个为“1”，另一个为“0”，这表明 Y 城市与 X 城市的路径顺序相邻。 $d_{XY}$  表示 X 城市与 Y 城市距离，如果 X 城市与 Y 城市不相邻，当  $V_{Xi} = 1$  时， $V_{Y,i+1}$  与  $V_{Y,i-1}$  都为“0”，此项消失。全部相邻的  $d_{XY}$  取和之后得到某种路径组合。对应最小距离  $d_{\min}$  在能量函数中 D 系数项取得最小值。

(c) 综上所述，得 TSP 建立的能量函数表达式为：

$$E = \frac{A}{2} \sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} + \frac{B}{2} \sum_i \sum_X \sum_{Y \neq X} V_{Xi} V_{Yi} + \frac{C}{2} (\sum_X \sum_i V_{Xi} - n)^2 + \frac{D}{2} \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{Xi} (V_{Y,i+1} + V_{Y,i-1}) \quad (3-36)$$

为了便于对照比较，将 (3-30) 式在积分项为零时的能量函数表达式重写如下：

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} V_i V_j - \sum_i I_i V_i \quad (3-37)$$

由以上两式不难求得利用 A、B、C、D 和  $d$  描述各突触权重  $w$  与偏置电流  $I$  的表达式，以约束 A、B、C、D 对各  $w$  值的不同贡献，系数 A 只影响同一行各元素之间突触权重  $w$  值，系数 B 只影响同一列各元素之间突触权重  $w$  值，C 与全部的突触权重  $w$  值都有关，而 D 只涉及路径顺序相邻项。则有：

$$w_{Xi,Yj} = -A \delta_{XY} (1 - \delta_{ij}) - B \delta_{ij} (1 - \delta_{XY}) - C - D d_{XY} (\delta_{j,i+1} + \delta_{j,i-1}) \quad (3-38)$$

$$I_{Xi} = Cn \quad (3-39)$$

$$\text{其中 } \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (3-40)$$

(3) TSP 神经网络的状态方程

借助 (3-31) 且  $C = C_j$  最后求得：

$$\frac{du_{Xi}}{dt} = -\frac{u_{Xi}}{t} - A \sum_{j \neq i} V_{Xj} - B \sum_{Y \neq X} V_{Yi} - C (\sum_X \sum_i V_{Xi} - n)$$

$$-D \sum_Y d_{XY} (V_{Y,i+1} + V_{Y,i-1}) \quad (3-41)$$

$$V_{Xi} = f(u_{Xi}) = \frac{1}{2} [1 + \tanh(\frac{u_{Xi}}{u_0})] \quad (3-42)$$

这是  $n \times n$  个神经元状态方程的通用表达式, 为求得 TSP 的优化结果, 需要求解  $n \times n$  个非线性一阶联立微分方程式, 以得到置换矩阵中  $n \times n$  个元素的全部状态。

#### (4) 实例分析

Hopfield 给出了利用数字计算机求解方程组的实例, 其参数选择如下:

$\tau = 1, A = B = 500, C = 200, D = 500, u_0 = 0.02, n = 10$ 。给出各城市位置和相互距离如图

3-5, 起始条件为随机噪声, 令起始  $u_{Xi}$  如下式:

$$\begin{cases} u_{Xi} = u_{00} + \delta u_{Xi} \\ -0.1u_0 \leq \delta u_{Xi} \leq 0.1u_0 \end{cases} \quad (3-43)$$

其中  $u_{00}$  满足在  $t = 0$  时,  $\sum_X \sum_i V_{Xi} = 10$  以利于收敛。利用数值计算方法对此微分方程组求

解, 经若干次迭代可求得网络各神经元的最终状态。图 3-6 为计算稳定后的置换矩阵示意图。其中每行或每列都是只有一个“1”(用一圆圈表示), 其它元素均为“0”(用一点表示)。本例给出的最优(或接近最优)的路径为  $D \rightarrow H \rightarrow I \rightarrow F \rightarrow G \rightarrow E \rightarrow A \rightarrow J \rightarrow C \rightarrow B \rightarrow D$ 。

Hopfield 对实例计算的统计结果表明, 当选择 20 组不同数据进行计算时, 其中大部分结果收敛于接近最优的路径, 包括最优路径。对于  $n = 10$ , 路径的构成共有:

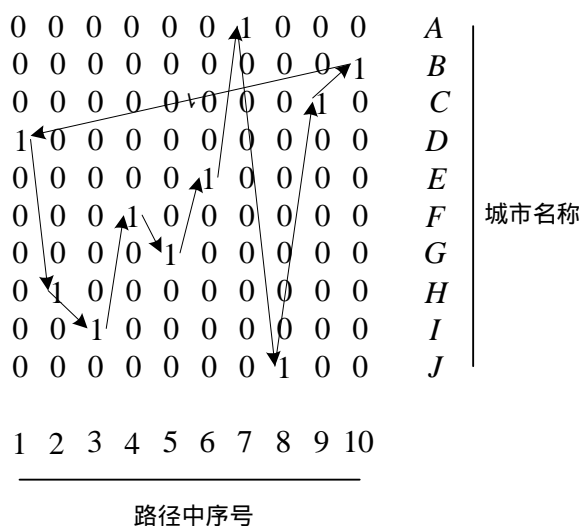


图 3-6 路径顺序 DHIFGEAJCBD

$\frac{1}{2}(n-1)! = \frac{1}{2}(10-1)! = 181440$  种可能方案, 因而, 这种方法改善的计算效率约为:

$\frac{20}{181440} \approx \frac{1}{10^5}$ , 当  $n$  增大时, 从理论上讲计算效率会更好。但网络结构将更加复杂。

### 3.3 Hopfield 神经网络的应用

Hopfield 神经网络在诸多问题上已有应用报道，此处给出两个代表性应用实例：基于 Hopfield 神经网络的 A/D 变换器；基于 Hopfield 神经网络的模式识别问题。

### 3.3.1 Hopfield 神经网络在 A/D 转换器上的应用<sup>[9-11]</sup>

作为 Hopfield 神经网络 A/D 转换器是 Hopfield 神经网络在优化计算中应用的又一实例。和传统的 A/D 转换电路相比较，它具有结构简单，转换速度快的优点。

#### 1. 问题描述

以 4 位 A/D 转换器为例，图 3-7 为 Hopfield 神经网络 A/D 转换器原理电路图，

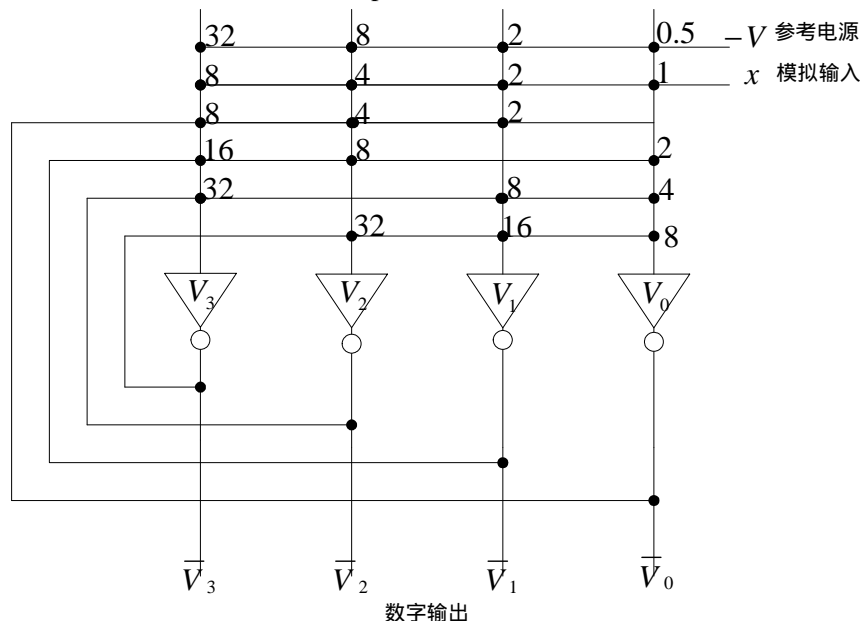


图 3-7 Hopfield 神经网络 A/D 转换器

此处为实现 4 位 A/D 转换之功能以 4 个运放作为神经元构成 Hopfield 神经网络。待转换的模拟量  $x$  以及参考电压源  $-V$  自右上端输入。经 A/D 转换作用由 4 个运放给出 4 位数字量以  $\bar{V}_3, \bar{V}_2, \bar{V}_1, \bar{V}_0$  表示。其运放特性仍为 sigmoid 曲线。这里，取反向输出，其变化范围从 0 到 -1。此处取反向输出有利于反馈网络之简化。在反馈线与各运放输入线之交叉点以一实心圆点表示由运放  $j$  到运放  $i$  的突触电导  $w_{ij}$ ，并注明了  $w_{ij}$  的数值。

#### 2. Hopfield 神经网络电路参数

为设计 A/D 变换电路，需要建立 Hopfield 神经网络的能量函数方程式，由此计算各突触电导值  $w_{ij}$ 。当 A/D 转换功能结束时，模拟量  $x$  与数字量  $V_i$  之间应满足如下关系：

$$\sum_{i=0}^3 V_i 2^i \approx x \quad (3-44)$$

能量函数之中包含下式：

$$\frac{1}{2} (x - \sum_{i=0}^3 V_i 2^i)^2 \quad (3-45)$$

当近似关系满足最小方差意义下最优条件时，上式取最小值。此外，由于运放输出  $V_i$  取 0

至 1 之间的连续值，为保证输出数字量，使  $V_i$  最靠近 0 或 1，在能量函数中需附加下式：

$$-\frac{1}{2} \sum_{i=0}^3 V_i (V_i - 1) (2^i)^2 \quad (3-46)$$

当  $V_i=1$  或  $V_i=0$  时, 此式取得极小值。其中系数的选择是为了保证突触电导矩阵  $W$  对角元素为 0, 也即  $w_{ij} = 0 (i = j)$ 。将上面两式合并, 整理并除去与变量  $V_i$  无关的项, 最后建立能量函数方程为:

$$E = -\frac{1}{2} \sum_{j=0}^3 \sum_{\substack{i=0 \\ i \neq j}}^3 (-2^{i+j}) V_i V_j - \sum_{i=0}^3 (-2^{(2i-1)} + 2^i x) V_i \quad (3-47)$$

当  $V_i$  作为  $x$  的最优数字描述时, 能量函数  $E$  取得极小值。将此处能量函数表达式与 Hopfield

神经网络的能量函数表达式  $E = -\frac{1}{2} \sum_i \sum_j w_{ij} V_i V_j - \sum_i I_i V_i$  相比较即可求得图 3-7 电路中各电导值。

$$\begin{cases} w_{ij} = -2^{(i+j)} \\ I_i = -2^{(2i-1)} + 2^i x \end{cases} \quad (3-48)$$

上式中,  $w_{ij}$  表示反馈网络中各突触之电导值, 利用运放的反相输出  $\bar{V}_i$  刚好实现式中的负号;

$I_i$  中的各项系数表示参考电压源和  $x$  引入线需要连接的电导。若参考电压源为 -1V, 则由此

引入的电导值为  $2^{(2i-1)}$ ; 而  $x$  输入至各运放经过的电流值为  $2^i$ 。按以上规律求得各电导值如图 3-7 所注。很明显, 这是一组归一化电导值, 如果考虑电路的实际参数还要作如下的修正:

(1) 当运放输出电压为 0 到  $V_{BB}$  时, 则:  $w_{ij} = -2^{(i+j)} / V_{BB}$ ;

(2) 当参考电压源为  $-V_R$  时, 则相应的一组电导值为:  $2^{(2i-1)} / V_R$ ;

(3) 当输入信号  $x$  的最大值为  $V_{BB}$ , 则相应的一组电导值为:  $\frac{2^i}{V_H} = \frac{2^{4+i}}{V_H}$ 。

16

### 3. Hopfield 神经网络 A/D 转换器性能分析

与一般的串行 A/D 变换电路相比, 由于 Hopfield 神经网络 A/D 转换器并行工作特性, 因而它可以达到更高的工作速度; 与一般的并行 A/D 变换电路相比, 它的优点是需要的运放数目较少, 电路结构简单。图 3-8 给出了 4 位 A/D 转换器正常工作时模拟量与数字量之间的对应关系。

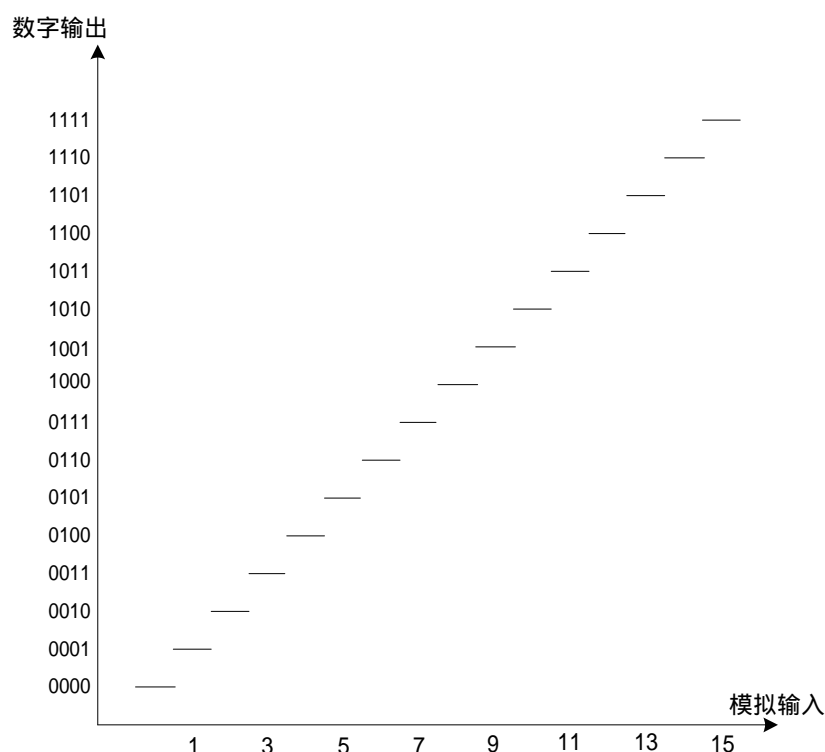


图 3-8 4 位 A/D 转换器模拟量与数字量对应关系

但这种电路的严重缺陷是存在迟滞(hysteresis)现象。所谓迟滞是指当输入模拟量  $x$  连续增长后再连续减小时,对应的数字量是不唯一的,以至 A/D 转换功能出现错误。产生这种现象的原因是由于在选定  $x$  之条件下,此电路的能量函数存在多个极小值,其中只有一个是全局最小解,其余为局部最小解,当系统收敛到局部最小解时,可能出现模拟量与数字量的错误映射。有关如何避免迟滞现象的方法可参考文献[10-11]。此处不再阐述。

### 3.3.2 Hopfield 神经网络在字符识别上的应用<sup>[12]</sup>

无论是在日常生活,还是在生产实践中,人们经常会遇到被噪声污染的字符。如:印刷文字的笔划断开、粘连、油墨深浅不均等;纸张洁白度、光洁度和版面上的污点、行高低不齐、文字歪曲变形等都会影响对文字的辨识。又如,机器的铭牌或车辆牌照由于长期使用而造成的磨损和被油污或泥水笼盖,都会使一些字符模糊难辨。可见,找到一种行之有效的方法实现对噪声字符的识别是有其现实意义的。

应用神经网络进行噪声字符识别已有报道<sup>[13-14]</sup>。此处采用连续 Hopfield 神经网络方法进行噪声字符识别,仿真结果表明,该方法识别率高,鲁棒性好。

#### 1 布尔神经网络(BNN)

文献[14]中提出的基于布尔神经网络的监督分类器(NTE),是一个以 $\{1\}$ -近邻 $\{1\}$ 准则为依据、以汉明距离作为度量的监督分类器。考虑二值特征值的输入模式,结构图如图 3-9 所示。

它是一个三层神经网络。第一层是输入层,共有  $n$  个神经元,对应着输入模式的  $n$  维分量;第二层是类别层,共有  $c$  个神经元,对应着输入模式的类别数;第三层是输出层,有 2 个神经元,一个是模糊神经元  $CA$ ,另一个是拒识神经元  $CR$ 。布尔神经网络的监督分类器 NTE 学习算法包括以下步骤:



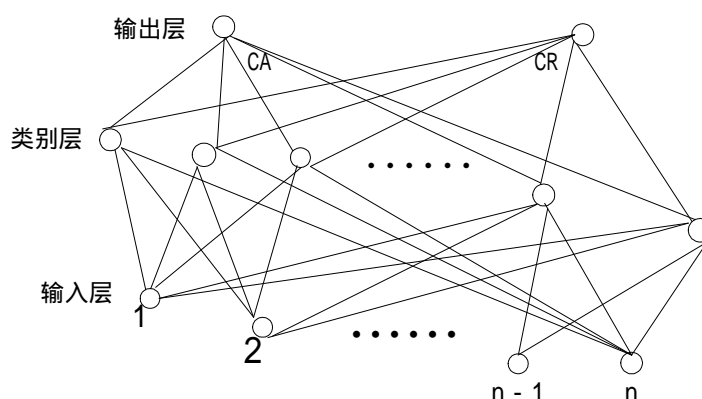


图 3-9 布尔网络结构图

给出训练输入模式  $e_k$  ( $k = 1, 2, \dots, c$ )，其中  $e_k = (e_{k1}, e_{k2}, \dots, e_{kn})^T$ ；

计算类别层神经元与输入层神经元之间的权， $w_k = 2e^k - 1 = (w_{k1}, w_{k2}, \dots, w_{kn})^T$ ，其中

$w_{ki} \in (-1, 1)$ ；

设置类别层第  $k$  个神经元的门限  $\theta_k = \sum_{i=1}^n e_{ki} w_{ki} - \eta_k$  ( $k = 1, 2, \dots, c$ )，式中  $\eta_k$  是类别层第

$k$  个神经元的吸引半径。

当对所有训练的输入模式完成以上学习过程后，就可以对待识别模式进行测试。

## 2. 连续型 Hopfield 神经网络 (CHNN)

图 3-10 是 Hopfield 动态神经元模型的电路图，图中电阻  $R_{i0}$  和电容  $C_i$  并联，模拟生物神经元的时延特性；电阻  $R_{ij}$  ( $j = 1, 2, \dots, n$ ) 模拟生物神经元之间的突触特性；利用运算放大器的非线性特性来模拟生物神经元的非线性特性，其输入、输出关系常用如下的两种单调递增非线性函数表示，对应于  $(0, 1)$  和  $(-1, 1)$  观点下的输出分别为：

$$V_i = f(U_i) = \frac{1}{1 + e^{-U_i}} \quad (3-49)$$

$$V_i = \varphi(U_i) = \tan(U_i) \quad (3-50)$$

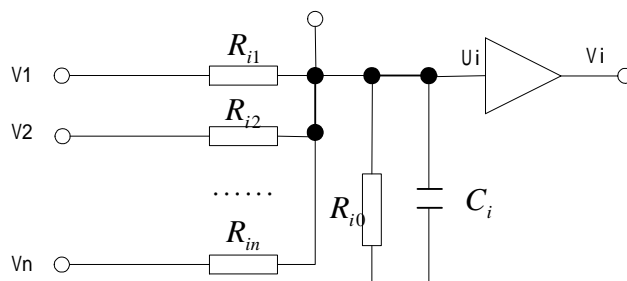


图 3-10 Hopfield 动态神经元模型图

CHNN 系统的结构如图 3-11 所示，由与待识别字符的像素相同数量的 Hopfield 动态神经元并联构成，每个神经元可由同相端或反相端输出。当由反向端输出时，它对其他神经元将起抑制作用。对于每一个神经元而言，自己的输出信号经过其他神经元又反馈到自己，所

以 CHNN 是一个连续的非线性动力学系统。

CHNN 的基本结构与 DHNN 相似，但 CHNN 中所有神经元都同步工作，各输入输出量均是随时间连续变化的模拟量。这就使得 CHNN 比 DHNN 在信息处理的并行性和实时性等方面更接近于实际生物神经网络的工作机理。同时 CHNN 模型可与电子线路直接对应起来，每一个神经元可以用一个运算放大器的输入电压  $U_i$  和输出电压  $V_i$  ( $i = 1, 2, 3, \dots, n$ ) 表示。而连接权  $W_{ij}$  用输入端的电导表示，其作用是把第  $i$  个神经元的输出反馈到第  $j$  个神经元，作为输入之一。每一个运放均有一个正相输出端和一个反相输出端。与正相输出端相连的电导表示

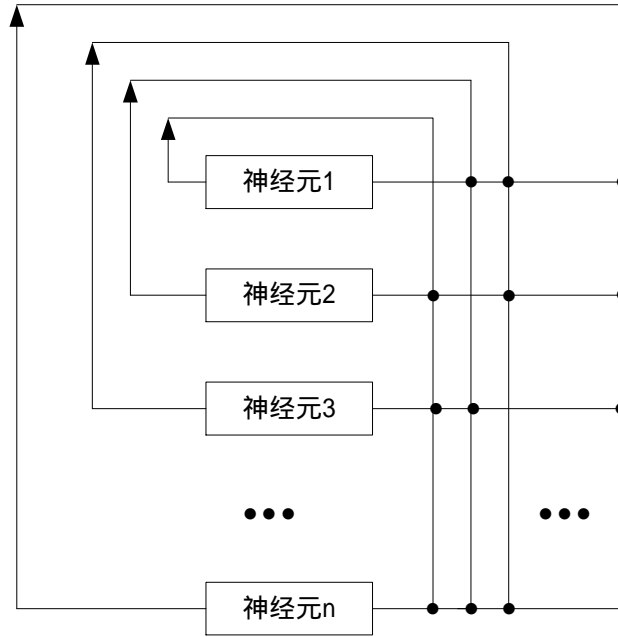


图 3-11 CHNN 网络结构图

兴奋性突触，而与反相输出端相连的电导表示抑制性突触。另外，每个神经元还有一个用于设置激活电平的外界输入偏置电流  $I_f$ ，其效果相当于阈值作用。由图 3-5，对第  $i$  个神经元运用基尔霍夫电流定律可写出以下方程：

$$C_i \frac{dU_i}{dt} + \frac{U_i}{R_{i0}} = \sum_{j=1}^n (W_{ij} V_j - U_i) + I_i \quad (3-51)$$

上式整理，此处令  $\frac{1}{\tau_i} = \frac{1}{R_{i0} C_i} + \sum_{j=1}^n \frac{1}{R_{ij} C_i}$ ，得：

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau_i} + \sum_{j=1}^n W_{ij} V_j + \theta_i \quad (3-52)$$

其中： $W_{ij} = \frac{1}{R_{ij} C_i}$  且  $\theta_i = \frac{I_i}{C_i}$  对于由  $n$  个神经元构成的 CHNN，各放大器输入输出关系

可以用矩阵方程来描述：

$$\dot{U} = -\tau^{-1}U + WV + \theta \quad (3-53)$$

$$\text{如果令上式中 } \dot{U} = 0, \text{ 则有: } \dot{U} = \tau WV + \tau \theta \quad (3-54)$$

### 3. 分析与仿真

Hopfield 神经网络状态的演变过程是一个非线性动力学过程, 在确定连接权之后, 若输入某个矢量, 系统将趋向某一定态, 即状态空间中的定点吸引子, 这个定态便是网络输出矢量。系统的稳定性可用能量函数, 即李雅普诺夫或哈密顿函数来进行分析。在满足一定条件下, 能量函数在网络运行过程中不断地减少, 最后趋于稳定的平衡状态。如果将系统的稳定点视为一个记忆的话, 那么从初态朝这个稳定点的演变过程就是寻找该记忆的过程, 从而实现从片段中对数据的完全恢复。

网络的稳定性分析是基于网络的能量函数。如本章前面的分析, 定义 CHNN 能量函数为:

$$E = -\frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n w_{ij} V_i V_j - \sum_{j=1}^n V_j I_j + \sum_{j=1}^n \frac{1}{R_j} \int_0^{V_j} \varphi^{-1}(V) dV \quad (3-55)$$

式(3-55)中  $\varphi^{-1}(\bullet)$  是神经元转移函数  $V = \varphi(U)$  的反函数。

对于上式中定义的能量函数, 存在以下定理: 若神经元转移函数  $V = \varphi(U)$  存在反函数  $\varphi^{-1}(\bullet)$ , 且  $\varphi^{-1}(\bullet)$  是单调连续递增的, 同时网络权值对称, 则由任意初态开始, CHNN 网络的能量函数总是单调递减的, 即  $\frac{dE}{dt} \leq 0$  (当且仅当  $\frac{dE}{dt} = 0$  时, 取等号)。因而网络最终能达到稳态。

如前面所介绍, 网络从某一初始状态按一定的运行规则最终可能都稳定在同一个吸引子上。那么, 称这种凡是经过一定时间能够稳定在同一吸引子的所有初始状态集合为该吸引子的吸引域。吸引半径是指包含在吸引域的最大 Hamming 球半径。

在那些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, 由于能量函数的单调下降特性, 保证状态的运动方向能从远离吸引点处, 不断的趋于吸引点, 直到达到稳定点。如果把欲恢复的模式设置为网络的一个稳定平衡点, 那么当网络从距该平衡点较近的某个初始状态(相当于被噪声污染的待识别状态)出发, 便能联想出该模式, 从而达到模式识别的目的。

取文献[14]中数据的  $i_0 i_1 \sim i_9 i_{10}$  十个字符, 则  $c = 10$ ,  $n = 63$ 。对不同失真率的数据在两种情况下用 MATLAB 编程进行仿真, 并与 NTE 方法结果进行了比较。若把一个字符设定为  $7 \times 9$  大小的黑白图象, 每幅 63 个像素, 用 1 表示黑, 0 表示白, 设给定吸引半径为  $R = 12$ , 选择适当的电路参数。其原理框图如图 3-12, 仿真结果如表 3-2 表 3-3。

表 3-2 两种失真率下给定吸引半径的识别结果

失真率	方法	识别结果			
		$R$	$R+2$	$R+4$	$R+10$
10%	NTE	全部正确	9 个正确 1 个模糊	9 个正确 1 个模糊	8 个正确 2 个模糊
	CHNN	全部正确	全部正确	全部正确	全部正确
20%	NET	全部正确	9 个正确 1 个模糊	3 个正确 7 个模糊	2 个正确 8 个模糊
	CHNN	全部正确	全部正确	全部正确	全部正确

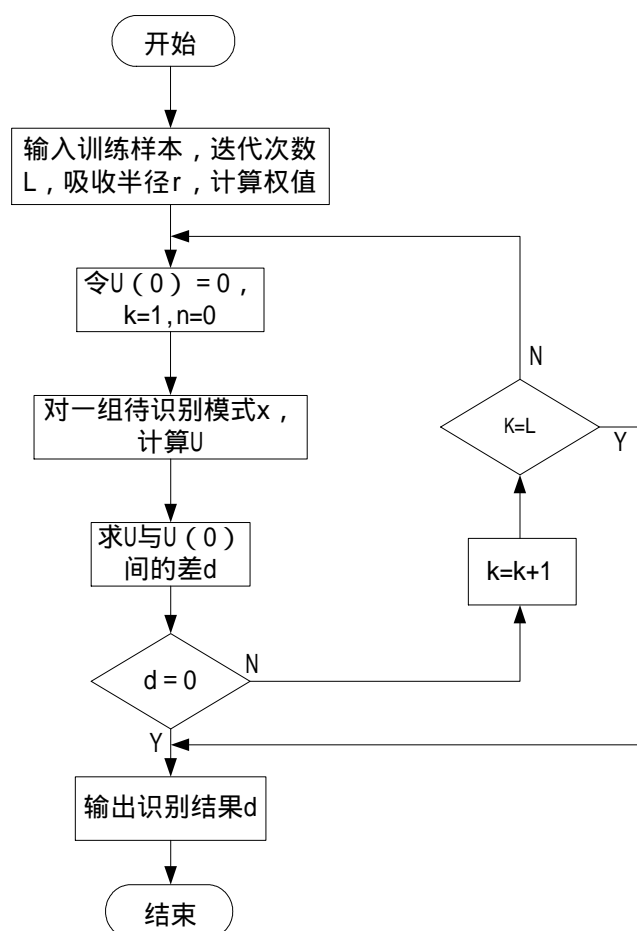


图 3-12 仿真计算原理框图

由仿真结果可得出 CHNN 方法比 NTE 方法有更高的识别率。由于字符识别的特征向量提取方法的不同，CHNN 和 NTE 具有不同的识别机理。NTE 监督分类器以  $i_1$  - 近邻  $i_1$  准则为依据，在模式识别理论中， $i_1$  - 近邻  $i_1$  准则是准最优的。NTE 算法对字符的识别有很高的识别率，但较高的识别率是建立在已知待识别模式的失真率及其相应的吸引半径的基础上的。利用网络稳定的平衡点来存储记忆样本，按照反馈动力学运动规律唤起记忆，来实现对缺损目标的恢复，使得 CHNN 联想记忆实现方法有更高的识别率。

表 3-3 两种失真率下自定义给定吸引半径的识别结果比较

失真率	方法	识别结果			
		$R = 30$	$R = 60$	$R = 10$	$R = 18$
10%	NTE	全部模糊	全部模糊	全部正确	6 个正确 4 个模糊
	CHNN	全部正确	全部正确	全部正确	全部正确
20%	NET	全部模糊	1 个正确 9 个模糊	1 个正确 9 个拒识	6 个正确 3 个模糊 1 个拒识
	CHNN	全部正确	全部正确	全部正确	全部正确

同时，由仿真结果也可看出，NTE 方法对吸引半径敏感，随着吸引半径的增大，误识率明显增加。而 CHNN 方法在失真率较大和吸引半径变化较大的情况下具有较强的鲁棒性。这是因为反馈网络要具有联想能力，每个吸引子都应该具有一定的吸引域。在较大的吸引半径情况下，对于带有一定噪声或缺损的初始样本，CHNN 经过动态演变而稳定到某一吸引子状

态，从而对吸引半径不敏感。

由此可见，用连续型 Hopfield 神经网络对噪声字符进行识别，不仅在失真率较大和吸引半径变化较大的情况下，都能正确识别，而且与 NTE 方法比较，求解质量稳定，速度快，且对吸引半径不敏感，有较强的鲁棒性。该方法能更有效地对噪声字符进行识别，有一定的实用价值。

#### 参考文献：

- [1]Hopfield J.J., Neural networks and physical systems with emergent collective computational properties, Proceedings of the National Academy of Sciences, 1982, Vol.79, pp.2554-2558.
- [2]Hopfield J.J., Neurons with graded response have collective computational properties like those of two-state neurons, Proceedings of the National Academy of Sciences, 1984, Vol.81, pp.3088-3092.
- [3]McCulloch W., Pitts W., A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics., 1943, Vol.5, pp.115-133.
- [4]Grossberg S., Nonlinear difference-differential equations in prediction and learning theory, Proceedings of the National Academy of Sciences, 1967, Vol.58, pp.1329-1334.
- [5]Anderson J., A simple neural network generating an interactive memory, Mathematical Biosciences, 1972, Vol.14, pp.197-220.
- [6]Kohonen T., Correlation matrix memories, IEEE Transactions on Computers, 1972, Vol.21, pp.353-359.
- [7]Hopfield J.J., Tank D.W., Neural computation of decisions in optimization problems, Biological Cybernetics, 1985, Vol.52, pp.141-154.
- [8]Martin T.H., Howard B.D., Mark H.B.著,戴葵等译,神经网络设计,机械工业出版社,2002年9月,北京。
- [9]Tank D.W., Hopfield J.J., Simple neural optimization networks: An A/D Converter, Signal Decision Circuit, and a linear programming circuit, IEEE Trans. CAS, 1986, 33(5):533-541.
- [10]Lee B.W., Shen B.J., Design of a neural-based A/D converter using modified Hopfield network, IEEE J. of Solid-State circuits, 1989, 24(4):1129-1135.
- [11]张宁,郑君里,非对称人工神经网络 A/D 变换器,电路与系统学报,1991,10(1):45-49.
- [12]王滔,基于连续 Hopfield 神经网络的噪声字符识别,系统仿真学报,2003,15(9):1288-1290.
- [13]王启瑞,朱婉玲,一种基于 MATLAB 的带噪声字符识别算法实现,计算机仿真,2002,19(4):91-94.
- [14]Gazula S., Kabuka M.R., Design of supervised Classifiers Using Boolean neural networks, IEEE Transaction on PAMI., 1995, 17(12):1239-1246.
- [15]韩力群,人工神经网络的理论、设计及应用,北京,化学工业出版社,2002.1。
- [16]Simon Haykin 著,叶世伟,史忠植译,神经网络原理,机械工业出版社,2004年5月,北京。

#### 本章小结

Hopfield 递归网络对人工神经网络的发展有重要而特殊的影响。本章分别对离散型和连续型 Hopfield 神经网络的结构,工作原理及稳定性进行了分析;Hopfield 神经网络与前章介绍的 BP 神经网络相比,在学习方式上,网络权值不是经过反复学习而获取,而是按一定规则进行设计,其权值一经设定就不再改变;运行过程中网络各神经元的状态不断更新演变,

网络运行达到稳定时各神经元的状态便是问题的解。

最后从优化计算及联想记忆两方面介绍了 Hopfield 神经网络的应用，重点阐述了 Hopfield 递归网络在经典的旅行商最优路径问题和 Hopfield 神经网络 A/D 转换器上的应用。

### 思考题

1. Hopfield 神经网络与 BP 神经网络相比，在学习方式和工作原理上有何不同？
2. 何为吸引子？何为吸引域？
3. Hopfield 神经网络主要应用领域有哪些？请简要介绍。
4. 何为网络能量函数？能量函数的设计原则是什么？
5. DHNN 网络权值如下：

$$W = \begin{bmatrix} 0 & 1 & -1 & -1 & -3 \\ 1 & 0 & 1 & 1 & -1 \\ -1 & 1 & 0 & 3 & 1 \\ -1 & 0 & 3 & 0 & 1 \\ -3 & -1 & 1 & 1 & 0 \end{bmatrix}, \text{ 已知各神经元阈值为 } 0. \text{ 试计算网络状态为:}$$

$$X^1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \text{ 和 } X^2 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \text{ 时的能量值。}$$

6. 有一 DHNN 网络， $n=4$ ， $T_j=0$ ， $j=1, 2, 3, 4$ ，向量  $X^a$ 、 $X^b$  和权值矩阵  $W$  分别为

$$X^a = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}, \quad X^b = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 2 & 2 \\ 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{bmatrix}$$

检验  $X^a$  和  $X^b$  是否为网络的吸引子，并考察其是否具有联想记忆能力。

7. 以 3 节点 DHNN 网为例，要求设计的吸引子为  $X^a = (011)^T$  和  $X^b = (101)^T$ ，权值和阈值在  $[-1, 1]$  区间取值，试求权值和阈值。

## 第 4 章 BAM 双向联想记忆神经网络

神经网络的联想记忆功能可以分为两种，一是自联想记忆，另一种是异联想记忆。第 3 章讨论的 Hopfield 神经网络就属于自联想记忆。本章讨论异联想记忆神经网络，重点介绍由 Kosko B.1988 年提出的双向联想记忆神经网络 BAM(Bidirectional Associative Memory)。BAM 有离散型、连续型和自适应型等多种形式。此处主要介绍离散型 BAM 神经网络。

### 4.1 BAM 结构、算法及稳定性

#### 4.1.1 BAM 结构及工作原理

若  $N$  维矢量  $A$  与  $M$  维矢量  $B$  的表达式分别如下：

$$A = [a_0, a_1, \dots, a_{N-1}]^T, \quad A \in \{-1, 1\}^N \quad (4-1)$$

$$B = [b_0, b_1, \dots, b_{M-1}]^T, \quad B \in \{-1, 1\}^M \quad (4-2)$$

构成一组矢量对  $(A_k, B_k)$ ， $k = 0, 1, \dots, P-1$ ，共有  $P$  对样本矢量。将它们存入双向联想存储器即可进行由  $A$  到  $B$  或由  $B$  到  $A$  的双向联想。即给定  $A$ （或  $B$ ）可以经联想作用得到对应的标准样本  $B$ （或  $A$ ）。当有噪声或缺损时，联想功能可以使样本对复原。

#### 1. BAM 神经网络结构

利用人工神经网络实现的 BAM 有多种形式，此处介绍一种最基本的形式，也称 Kosko 型 BAM<sup>[1-2]</sup>。图 4-1 给出了由双层双向网络构成的 BAM 结构，其中，与矢量  $A$  相应的一层有  $N$  个节点，另一层对应矢量  $B$ ，由  $M$  个节点构成。两层之间双向连接。假定由  $B$  向  $A$  的传输为正向，正向的突触权重矩阵为  $W$ ；反之，由  $A$  向  $B$  的传输为反向，反向的突触权重矩阵为  $W^T$ 。

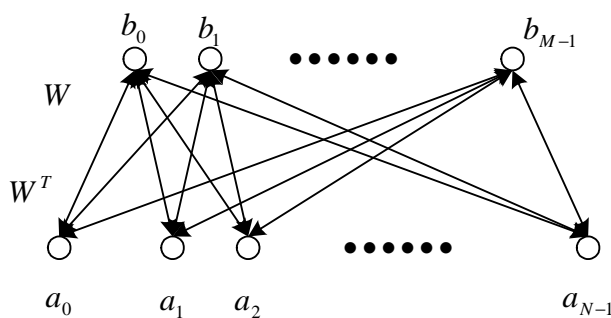


图 4-1 BAM 双向联想记忆网络模型

#### 2. BAM 神经网络工作原理<sup>[8]</sup>

如果输入矢量由上层加入，且相应于网络中  $B$  的状态，则经  $W$  的作用产生  $A$  的稳定状态。同样，如果输入矢量由下层加入，且相应于网络中  $A$  的状态，则经  $W^T$  的作用产生  $B$  的稳定状态。当任意矢量输入时，网络要经过若干次迭代计算演变到稳定状态。为了说明此过程，可以将 BAM 与自联想记忆原理相对比，对自联想来说，其演变过程为：

$$WX(t) \rightarrow X(t+1) \quad (4-3)$$

$$WX(t+1) \rightarrow X(t+2) \quad (4-4)$$

•••••

$\rightarrow X(t+k)$ 。经  $k$  次迭代后收敛，对应网络的稳定状态。

对 BAM 双向联想神经网络，当任意矢量输入时，网络迭代演变过程为：

$$WB(t) \rightarrow A(t+1) \quad (4-5)$$

$$W^T A(t+1) \rightarrow B(t+2) \quad (4-6)$$

$$WB(t+2) \rightarrow A(t+3) \quad (4-7)$$

•••••

直到  $A$ 、 $B$  均为稳定状态，演变过程结束。

#### 4.1.2 BAM 权矩阵设计及稳定性分析

##### 1. BAM 权矩阵设计

BAM 网络的学习仍按 Hebb 规则进行。若给定  $P$  个双极性矢量对： $(A_0, B_0), (A_1, B_1) \dots (A_{P-1}, B_{P-1})$ 。由这些样本计算权重矩阵表达式为：

$$W = \sum_{k=0}^{P-1} A_k B_k^T \quad (4-8)$$

$$W^T = \sum_{k=0}^{P-1} B_k A_k^T \quad (4-9)$$

##### 2. BAM 稳定性分析

如果 BAM 网络神经元函数的阈值等于零，则称为齐次 BAM 网络。定义齐次 BAM 网络的能量函数为：

$$E(A, B) = -\frac{1}{2} A^T W B - \frac{1}{2} B^T W^T A \quad (4-10)$$

$$\text{因为 } B^T W^T A = B^T (A^T W)^T = (A^T W B)^T = A^T W B \quad (4-11)$$

$$\text{所以有： } E(A, B) = -A^T W B \quad (4-12)$$

一般情况下，神经元非线性函数阈值非零，此时构成非齐次 BAM 网络。它的能量函数表达式为：

$$E(A, B) = -A^T W B + A^T \theta + B^T \mu \quad (4-13)$$

式 (4-13) 中：

$$\theta = (\theta_0, \theta_1, \dots, \theta_{N-1}) \quad (4-14)$$

$$\mu = (\mu_1, \mu_2, \dots, \mu_{M-1}) \quad (4-15)$$

$\theta_i$  为  $A$  的第  $i$  元素  $a_i$  对应的阈值， $\mu_j$  为  $B$  的第  $j$  元素  $b_j$  对应的阈值，假设神经元非线性函



数  $f$  为硬限幅函数特性，则描述 BAM 动态特性的方程式为：

正向传输（由  $B$  联想  $A$ ）

$$a_i(t+1) = \begin{cases} \text{sgn}[\sum_{j=1}^M w_{ij}b_j(t) - \theta_i] & \text{当} \sum_{j=1}^M w_{ij}b_j(t) - \theta_i \neq 0 \\ a_i(t) & \text{当} \sum_{j=1}^M w_{ij}b_j(t) - \theta_i = 0 \end{cases} \quad (4-16)$$

反向传输（由  $A$  联想  $B$ ）

$$b_j(t+2) = \begin{cases} \text{sgn}[\sum_{i=1}^N w_{ij}a_i(t+1) - \mu_j] & \text{当} \sum_{i=1}^N w_{ij}a_i(t+1) - \mu_j \neq 0 \\ b_j(t+1) & \text{当} \sum_{i=1}^N w_{ij}a_i(t+1) - \mu_j = 0 \end{cases} \quad (4-17)$$

由以上关系式可以证明，网络在演变过程中能量函数  $E$  递减，即  $\Delta E < 0$ 。网络稳定。现分析证明如下：

设  $a_i$  发生变化，由于  $a_i$  取值为 1 或 -1，所以  $\Delta a_i$  可能为 -2, 0, +2 三种情况。为书写方便，在以下的推证中省略时间序号  $t$ 。由式 (4-13) 可求得：

$$\Delta E = -(\Delta A^T)WB + (\Delta A^T)\theta = -(\Delta A^T)(WB - \theta) = -(\Delta a_i)(\sum_{j=1}^P w_{ij}b_j - \theta_i) \quad (4-18)$$

当  $\Delta a_i = -2$  时，则有  $\sum_{j=1}^P w_{ij}b_j - \theta_i < 0 \Rightarrow \Delta E < 0$ ；

当  $\Delta a_i = 2$  时，则有  $\sum_{j=1}^P w_{ij}b_j - \theta_i > 0 \Rightarrow \Delta E < 0$ ；

当  $\Delta a_i = 0$  时，则有  $\Delta E = 0$ ；可见，当状态改变时，总有  $\Delta E \leq 0$ ，网络稳定。

例 4-1 已知  $A \in \{-1, 1\}^N$ ， $B \in \{-1, 1\}^M$ ， $N = 15, M = 10$ ，样本对  $P = 4$ ，各个样本如下：

$$A_1 = [1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1]^T$$

$$A_2 = [1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1]^T$$

$$A_3 = [1, 1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, 1, 1, 1]^T$$

$$A_4 = [1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1]^T$$

$$B_1 = [1, 1, 1, 1, -1, -1, -1, -1, 1, 1]^T$$

$$B_2 = [1, 1, 1, -1, -1, -1, 1, 1, 1, -1]^T$$

$$B_3 = [1, 1, -1, -1, 1, 1, -1, -1, 1, 1]^T$$

$$B_4 = [1, -1, 1, -1, 1, -1, 1, -1, 1, -1]^T。请设计 BAM 权矩阵 W。$$

解：根据式 (4-8) 可以计算  $A_1 B_1^T, \dots, A_4 B_4^T$  并相加得到权矩阵  $W$  如下：

$$W = \begin{bmatrix} 4 & 2 & 2 & -2 & 0 & -2 & 0 & -2 & 4 & 0 \\ 2 & 0 & 0 & -4 & 2 & 0 & 2 & 0 & 2 & -2 \\ 2 & 0 & 0 & 0 & 2 & 0 & -2 & -4 & 2 & 2 \\ -2 & -4 & 0 & 0 & 2 & 0 & 2 & 0 & -2 & -2 \\ 0 & 2 & 2 & 2 & -4 & -2 & 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 & -2 & 0 & 2 & 4 & -2 & -2 \\ 0 & 2 & -2 & 2 & 0 & 2 & -4 & -2 & 0 & 4 \\ -2 & 0 & -4 & 0 & 2 & 4 & -2 & 0 & -2 & -2 \\ 4 & 2 & 2 & -2 & 0 & -2 & 0 & -2 & 4 & 0 \\ 0 & -2 & 2 & -2 & 0 & -2 & 4 & 2 & 0 & -4 \\ 0 & -2 & 2 & 2 & 0 & -2 & 0 & -2 & 0 & 0 \\ -2 & -4 & 0 & 0 & 2 & 0 & 2 & 0 & -2 & -2 \\ 2 & 4 & 0 & 0 & -2 & 0 & -2 & 0 & 2 & 2 \\ 0 & 0 & -2 & -2 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & -2 & 2 & 0 & 2 & -4 & -2 & 0 & 4 \end{bmatrix}$$

按此权矩阵参数作 BAM 网络即可存储给定的 4 对样本。它们是在系统可能形成的  $2^{15} \times 2^{10}$  个状态中取 4 个稳定状态。利用式 (4-12) 还可以计算出 4 个稳定状态的能量函数分别为：-56, -48, -60, -40。均小于零。

## 4.2 BAM 神经网络的应用

### 4.2.1 基于 BAM 网络的推理方法<sup>[3]</sup>

人工智能作为当前科学技术发展中的一门前沿学科，日益受到世界各国科学工作者的重视与关注。为使计算机具有智能，不仅要使它拥有知识，即将知识用某种模式表示出来存储到计算机中，还必须使它具有思维能力，即能运用知识进行推理，求解问题。因此，关于推理及其方法的研究就成为人工智能的一个重要研究课题。

神经网络是将信息的处理由神经元之间的相互作用来实现的一类网络。知识与信息的存储表现为网络元件互连间分布式的物理联系，网络的学习和识别决定于各神经元连接权的动态演化过程。它的主要特征为连续时间非线性动力学，网络具有全局作用、大规模并行分布处理及高度的鲁棒性和学习联想能力，同时还具有一般非线性动力学系统的共性，即不可预测性、吸引性、耗散性、非平衡性、不可逆性、高维性、广泛联结性与自适应性等<sup>[4]</sup>。双向联想记忆 (BAM) 神经网络是一类具有双向稳定性的反馈神经网络系统，具有简单、并可用大规模集成电路实现的特点。此处介绍一种 BAM 神经网络推理方法。

#### 1 常见推理策略及其特点分析

我们在推理求解一个问题时，经常考虑方法 (策略) 问题，是用正向推导好？还是用反

证法好，同样，进行推理机设计时，也考虑策略问题，以提高其效果或效率。一般来说，推理求解的基本策略<sup>[5]</sup>主要有：正向推理、反向推理、和双向推理。

正向推理也称由底向上、数据驱动、前向链、模式制导和前件推理。其基本思想是从已知的信息出发，选用合适的知识，逐步求解待解的问题。基于推理机的正向推理工作程序是：首先将与求解问题有关的信息存入数据库，再根据当前状态选用知识，如此反复，直到求出解为止。正向推理的主要优点是：允许用户主动提供有用的事实信息，而不必等到系统需要时才提供。它的特点使该控制策略适合于规划、预测、监控之类解空间很大的问题。它的主要缺点是：知识激活和执行的目性不强，有可能系统为达到某个目标采用了若干次无用动作。

反向推理也称由顶向下、目标驱动、方向链、目标制导和后件推理。其基本思想是选定一个目标，然后去推证此目标成立与否。基于推理机的反向推理程序是：首先提出要推证的目标（通常称为假设），判断此目标是否在知识库中，然后将所有结论部分含此目标的相关知识全部找出，检查应用这些相关知识的必要条件是否具备，若不具备则将该知识作为新的子目标再用上述方法继续寻找，直到问题解决或没有相关知识为止。反向推理的主要优点是不必使用与目标无关的知识，目的性强，特别适合于解空间小的问题，对于解空间大的问题，初始目标的提出比较困难。它的主要缺点是初始目标的选择盲目，没有充分利用用户提供的信息。

双向推理是将正向推理与反向推理同时进行。其基本思想是：根据数据库中原始数据进行正向推理，但不一直推理到目标；同时从目标出发进行反向推理，也不一直推理到原始数据，而是希望由正向推理所得出的中间结论，正好满足反向推理对证据的要求，即满足双向稳定性要求，这也是双向推理成功的终止条件。双向推理克服了正向推理的目标性不强和反向推理的选择目标盲目的缺点，同时又结合了两者的优点，但其实现却存在以下几个难点：

由于推理路径不唯一，接合点设计难。判定正向推理与反向推理是否接合比较困难。权衡正向推理与反向推理的比重难。

## 2 BAM 网络用于推理的理论依据

双向联想记忆（BAM）神经网络<sup>[2]</sup>（如图 4-2）是这样一个两层网络，第一层输出向量  $A$  通过权矩阵  $W$  作用到第二层得到相应的输出向量  $B$ ，第二层输出向量  $B$  又通过权矩阵

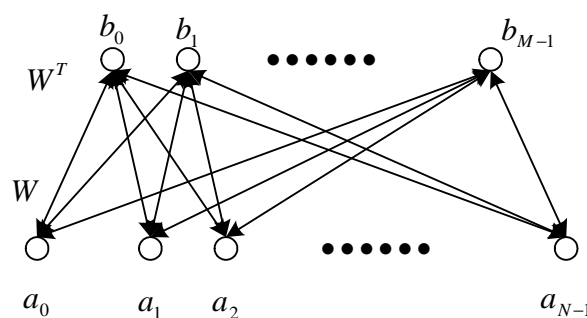


图 4-2 BAM 神经网络结构图

$W^T$  作用到第一层。这在前面图 4-1 中已有介绍。BAM 的基本关系式为：

$$B = F(AW) \quad (4-19)$$

$$A = F(BW^T) \quad (4-20)$$

在神经网络计算中，函数  $F$  通常被选为 LOGISTIC 型。GROSSBERG 指出该函数的中间高增益区适合解决处理小信号问题，而它的正负极增益递减的区域正适合处理大信号问题。常

用的 LOGISTIC 型函数有 SIGMOID 函数  $F(X) = \frac{1}{1 + \exp^{-\lambda \cdot x}}$  ,这里  $\lambda$  为决定曲线陡峭的常数 ,若  $\lambda$  被取得较大 ,产生的激励函数接近于一个简单的阈值函数。此时的 BAM 系统为二值 BAM ,它的输出遵循式 (4-16) 和式 (4-17) 所表示的规则 ,令  $A(a_1, a_2, \dots, a_m)$  为第一层神经单元的输出向量 ,  $\theta(\theta_1, \theta_2, \dots, \theta_m)$  为第一层神经单元的阈值 ,  $B(b_1, b_2, \dots, b_n)$  为第二层神经单元的输出向量 ,  $\mu(\mu_1, \mu_2, \dots, \mu_n)$  为第二层神经单元的阈值 , 则一个二值 BAM 网络可由如下传递函数定义 :

$$a_i(t+1) = \begin{cases} 1, & \text{如果 } \sum_{j=1}^n b_j(t) W_{i,j}^T > \theta_i \\ a_i(t), & \text{如果 } \sum_{j=1}^n b_j(t) W_{i,j}^T = \theta_i \\ -1, & \text{如果 } \sum_{j=1}^n b_j(t) W_{i,j}^T < \theta_i \end{cases} \quad (4-21)$$

$$b_j(t+1) = \begin{cases} 1, & \text{如果 } \sum_{i=1}^m b_i(t) W_{i,j} > \mu_j \\ b_j(t), & \text{如果 } \sum_{i=1}^m b_i(t) W_{i,j} = \mu_j \\ -1, & \text{如果 } \sum_{i=1}^m b_i(t) W_{i,j} < \mu_j \end{cases} \quad (4-22)$$

为将问题的描述表达成适合于推理的形式 , 我们采用了基于规则的产生式表示形式<sup>[4]</sup> :

IF 前件 THEN 后件

其中 前件 := 条件 | 条件 条件 | 条件 条件 ,

后件 := 结论 | 结论 后件

条件 := ( 断言 )

结论 := ( 断言 )

通过映射原理 , 可将基于规则的产生式表示形式转化为神经网络可以处理的形式<sup>[6]</sup>。下面给出规则集——BAM 网络映射原理 : 设规则集中无重复的前件总数为  $m$  , 无重复的后件总数为  $n$  , 可设置二个双极性向量  $A, B$  , 其中向量  $A$  含有  $m$  个元素 , 向量  $B$  含有  $n$  个元素 , 规则集中无重复的每个前件和后件分别对应向量  $A, B$  中的一个元素。若某规则第  $i$  个前件 , 第  $j$  个后件存在 , 则把向量中相应的元素  $a_i, b_j$  置 1 , 否则置 0。根据 Kosko 的研究<sup>[4]</sup> ,

在用 BAM 进行外积运算前将二值向量变成双极向量 , 能够取得较好的结果。

从图 4-2 结构上看 , BAM 是一个反馈系统。我们知道一个反馈系统极易出现不稳定 , 即振荡的情况。对于基于 BAM 的推理系统 , 若出现振荡则意味着得不到推理的结果 , 因为正向推理需要系统具有正向稳定性 , 反向推理需要系统具有反向稳定性 , 双向推理需要系统具有双向稳定性。当然 , 若一个系统是双向稳定的 , 则它必同时具有正、反向稳定性。对图 4-2 的 BAM 反馈系统 , 前已证明 : 对离散 BAM 网络 , 无论是同步还是异步更新 , 每个矩阵都是双

向稳定的。

必须指出矩阵  $W$  不一定为正定矩阵, Lyapunov 给出的判断动力系统平衡状态的稳定性条件只是充分条件不是必要条件。对于离散的神经网络系统, 如本文中离散 BAM 网络的稳定性证明可结合上述的反证法加以证明。对于连续的神经网络系统, 可结合如下连续自治动力系统的稳定性判据<sup>[4]</sup>。

对于有微分方程  $\dot{X} = f(x)$  所表示的系统:  $f: R^n \rightarrow R^n$  是连续的, 它是完全稳定的(即所有的轨迹都趋于平衡点集)条件为: 存在一标量函数  $V(x): R^n \rightarrow R^n$ ,  $V(x)$  具有连续的一阶偏导数且满足:

$$\dot{V}(x) = \left( \frac{\partial V}{\partial x} \right)^T \bullet f(x) \text{ 对于平衡点集之外的所有地方均小于零, 而在平衡点集处 } \dot{V}(x) = 0。$$

$\dot{X} = f(x)$  的所有解都是有界的。

### 3 BAM 推理系统的运行方式

在进行推理之前, 首先要用知识库中的规则集训练 BAM 推理系统。其具体训练方式为: 将知识库中的知识编码为向量对  $A_i, B_i$ , 一个样本即构成了一个向量对, 知识库中所有样本构成的向量对集合称为训练库。BAM 系统权值是通过计算被设计的, 即权矩阵是通过计算训练集中所有向量对的外积之和而得到的<sup>[7]</sup>。具体计算可参照式(4-8)和式(4-9)。

BAM 推理系统在训练好之后即可用于推理。在进行正向推理时, 将用户输入的数据作为前件, 然后把此前件编码为向量  $A$ 。再将向量  $A$  放到 BAM 系统第一层的输出上, 应用网络的权矩阵  $W$ , 通过式(4-19)计算相应的第二层输出向量  $B$ , 将  $B$  解码为相应的后件, 即为正向推理的结果。在进行反向推理时, 将目标(或假设)作为后件, 编码为双极性向量  $B$ , 将  $B$  放到 BAM 系统第二层的输出上, 应用网络的权矩阵  $W^T$ , 通过式(4-20)计算相应的第一层输出向量  $A$ , 再解码为相应的前件, 即为反向推理的结果。

在进行双向推理时, 首先将输入的数据和目标转化为相应的前件和后件, 再把此前、后件编码为两个向量  $A, B$ 。为了得到双向推理的结果, 完整的或不完整的向量  $A$  被迅速强迫地放到第一层的输出上, 向量  $B$  被迅速强迫地放到第二层的输出上, 然后  $A, B$  被传播且网络向着稳定的方向进行下去。反复应用  $B = F(AW), A = F(BW^T)$  公式, 即通过矩阵  $W$ , 在第二层的输出上产生一个与第一层输出向量相关的向量  $B'$ ,  $B'$  再次通过转置矩阵  $W^T$  从而在第一层的输出上产生一个与原输入向量相近的向量  $A'$ , 这样每一次循环使得第一层和第二层的输出向量逐渐逼近于一个稳定点。令此稳定点向量为  $(A_K, B_K)$ , 由于这个稳定点实质上是一种共振, 即向量在两层之间来回流动, 总保持目前的输出, 因此此稳定点值  $(A_K, B_K)$  即为双向推理的接合点, 具有双向稳定的特性。将  $(A_K, B_K)$  解码为基于规则的知识表达形式, 即为双向推理的结果。BAM 网络的共振点成功地解决了双向推理的结合点设计难、判断难等问题, 并且有实现容易、成本不高的优点。

#### 4 实例分析

给出的是由如下三条规则所组成的规则集，其中  $X_i (i = 1, \dots, 5)$  表示第  $i$  条条件存在， $Y_j (j = 1, \dots, 4)$  表示第  $j$  条条件成立。

规则一：IF  $X_1 \wedge X_2 \wedge X_4$  THEN  $Y_1 \wedge Y_4$

规则二：IF  $X_3 \wedge X_5$  THEN  $Y_2 \wedge Y_3$

规则三：IF  $X_1 \wedge X_3 \wedge X_5$  THEN  $Y_1 \wedge Y_2 \wedge Y_3 \wedge Y_4$

首先通过规则集—BAM 网络映射原理将规则集进行编码，具体做法是若规则的第  $i$  个前件，第  $j$  个后件存在，则把向量  $A$ ， $B$  中相应的元素  $A_i, B_j$  置 1，否则置 0。编码后得到相应的前件向量  $A$  和后件向量  $B$ 。编码顺序从右到左取  $i = 1, 2, 3, 4, 5; j = 1, 2, 3, 4$ 。则有：

前件向量  $A$ ： $a_1 = (0 \ 1 \ 0 \ 1 \ 1)$

$$a_2 = (1 \ 0 \ 1 \ 0 \ 0)$$

$$a_3 = (1 \ 0 \ 1 \ 0 \ 1)$$

后件向量  $B$ ： $b_1 = (1 \ 0 \ 0 \ 1)$

$$b_2 = (0 \ 1 \ 1 \ 0)$$

$$b_3 = (1 \ 1 \ 1 \ 1)$$

根据 KOSKO 的研究<sup>[7]</sup>，在进行外积运算  $W$  矩阵时将二维向量转变成双极向量，能够取得较好的结果。下面把前件向量  $A$ ，后件向量  $B$  转换为相应的双极  $A', B'$ 。

前件向量  $A'$ ： $a'_1 = (-1 \ 1 \ -1 \ 1 \ 1)$

$$a'_2 = (1 \ -1 \ 1 \ -1 \ -1)$$

$$a'_3 = (1 \ -1 \ 1 \ -1 \ 1)$$

后件向量  $B'$ ： $b'_1 = (1 \ -1 \ -1 \ 1)$

$$b'_2 = (-1 \ 1 \ 1 \ -1)$$

$$b'_3 = (1 \ 1 \ 1 \ 1)$$

$$W = a_1^T \cdot b_1' + a_2^T \cdot b_2' + a_3^T \cdot b_3' = \begin{bmatrix} -1 & 3 & 3 & -1 \\ 1 & -3 & -3 & 1 \\ -1 & 3 & 3 & -1 \\ 1 & -3 & -3 & 1 \\ 3 & -1 & -1 & 3 \end{bmatrix}$$

$$\text{从而: } a_1 W = (5 \quad -7 \quad -7 \quad 5) \xrightarrow{\text{Hebb}} (1 \quad 0 \quad 0 \quad 1) = b_1$$

$$a_2 W = (-2 \quad 6 \quad 6 \quad -2) \xrightarrow{\text{Hebb}} (0 \quad 1 \quad 1 \quad 0) = b_2$$

$$a_3 W = (1 \quad 5 \quad 5 \quad 1) \xrightarrow{\text{Hebb}} (1 \quad 1 \quad 1 \quad 1) = b_3$$

$$b_1 W^T = (-2 \quad 2 \quad -2 \quad 2 \quad 6) \xrightarrow{\text{Hebb}} (0 \quad 1 \quad 0 \quad 1 \quad 1) = a_1$$

$$b_2 W^T = (6 \quad -6 \quad 6 \quad -6 \quad 2) \xrightarrow{\text{Hebb}} (1 \quad 0 \quad 1 \quad 0 \quad 0) = a_2$$

$$b_3 W^T = (4 \quad -4 \quad 4 \quad -4 \quad 4) \xrightarrow{\text{Hebb}} (1 \quad 0 \quad 1 \quad 0 \quad 1) = a_3$$

并且若前件向量  $a_1$  的变化量为  $\delta$ ，它将收敛到最近的后件向量  $b_1$ 。例如：

$$\delta = (1 \quad 0 \quad 0 \quad 0 \quad 0) \quad , \quad a_1 + \delta = (1 \quad 1 \quad 0 \quad 1 \quad 1) \quad \text{则}$$

$$(a_1 + \delta) W = (4 \quad -4 \quad -4 \quad 4) \xrightarrow{\text{Hebb}} (1 \quad 0 \quad 0 \quad 1) = b_1$$

由以上分析可以看出，根据规则集训练好 BAM 网络的权值矩阵  $W$  后，把用户输入的数据编码为前件向量  $a_i$ ，通过式（4-19）即可计算出后件向量  $b_i$ ，解码后可得到正向推理的结果。进行反向推理时，将目标编码为后件向量  $b_j$ ，通过式（4-20）计算出相应的前件向量  $a_j$ ，解码后即可得到反向推理的结果。当输入的数据存在偏移量时，系统输出将收敛到最近的稳定状态。

## 5 BAM 推理系统的进一步讨论

上述的 BAM 推理系统具有简单，并可用大规模集成电路实现的优点。将它用于推理控制，特别是双向推理控制之中尤为适合。但该系统只适应于简单情况下的小容量推理，我们知道一个有  $N$  个神经元组成的网络共有  $2^N$  个状态，但是 BAM 网络的存储量远比这个值小。如果存储太多的记忆，网络可能产生不正确的输出。关于 BAM 网络所能存储的最大记忆数是目前一个流行的研究课题。Kosko 曾经估计过<sup>[7]</sup>，在一般情况下，齐次 BAM（即阈值全为 0）存储的最大记忆数不能超过较小层的神经元的数目。这个估计实际上假定了“偶数编码”，即每个双极性向量的 1 和 -1 的数目相等。Haines 和 Hecht-Nielsen 通过将每个神经元赋予一个适当的阈值  $\theta$ ，BAM 的稳定状态可上升到  $2^M$  个， $M$  为 BAM 中较小层的神经元的数目。这种结构称为非齐次 BAM，是对原齐次 BAM 的一种扩展。这里对每个神经元阈值的选择不是随

机的，因而如何选择阈值使得系统的存储容量达到满足需要是一个有待解决的问题。此外，在 BAM 推理系统的设计中，我们只考虑了可信度全为 1 的精确型推理情况，对可信度不全为 1 的不确定性推理和模糊推理情况须进行进一步的研究。

#### 4.2.2 基于 BAM 网络的控制系统故障诊断<sup>[9]</sup>

##### 1. 问题提出

如图 4-3 的控制系统包括被控对象，调节器，调节阀和检测仪表，任何一部分出现故障，都将造成控制系统的故障或异常。

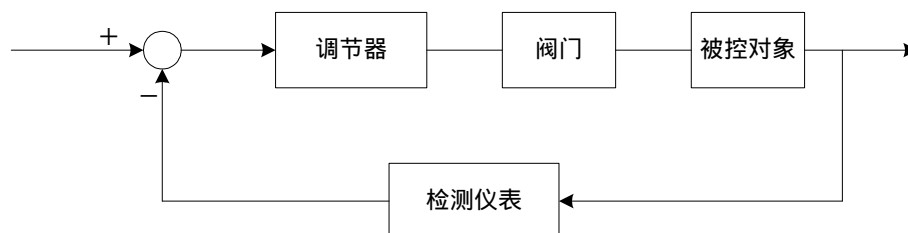


图 4-3 过程控制系统

为方便起见，这里仅诊断调节器和阀门两部分的故障。它们的故障类型（故障征兆）有：调节器误指令、调节器不工作、阀门误操作、阀门不工作；对应的解决故障的办法（故障原因）是：检查调节器的传动机构、检修调节器、检查阀门的传动机构、检修阀门。要求设计一 BAM 神经网络，用以存储几组故障模式。

##### 2. 故障诊断的 BAM 神经网络

设  $X_i = (x_{1i}, x_{2i}, x_{3i}, x_{4i})$  为输入向量，表示输入的故障现象信息； $Y_i = (y_{1i}, y_{2i}, y_{3i}, y_{4i})$  为输出向量，表示输出解决故障的办法。其中：

$x_{1i} = 1$  表示调节器故障； $x_{2i} = 1$  表示调节器误指令；

$x_{3i} = 1$  表示阀门不工作； $x_{4i} = 1$  表示阀门误操作；

$y_{1i} = 1$  表示检查调节器的传动机构； $y_{2i} = 1$  表示检修调节器；

$y_{3i} = 1$  表示检查阀门的传动机构； $y_{4i} = 1$  表示检修阀门；

权矩阵设计

样本用双极向量表示：

$$X_1 = (1, 1, -1, -1) \leftrightarrow Y_1 = (1, -1, -1, -1) ; X_2 = (1, -1, 1, -1) \leftrightarrow Y_2 = (-1, 1, -1, -1)$$

$$X_3 = (-1, 1, -1, 1) \leftrightarrow Y_3 = (-1, -1, 1, -1) ; X_4 = (-1, -1, 1, -1) \leftrightarrow Y_4 = (-1, -1, -1, 1)。$$

这里 -1 表示该位代表的操作或设备正常。 $(X_1 Y_1)$  指“调节器有误指令”，相应采取的措施为“检查调节器的传动机构”。BAM 的权矩阵为：

$$W = \sum_{i=1}^4 X_i^T Y_i = \begin{bmatrix} 2 & 2 & -2 & -2 \\ 2 & -2 & 2 & -2 \\ -2 & 2 & -2 & 2 \\ -2 & -2 & 2 & 2 \end{bmatrix} \quad (4-23)$$



### 联想记忆

对 (4-23) 其相应的正反向计算分别为：

$$X_1W = (4,0,0,-4) \rightarrow Y_1 = (1,-1,-1,-1) ; X_2W = (0,4,-4,0) \rightarrow Y_2 = (-1,1,-1,-1) ;$$

$$X_3W = (0,-4,4,0) \rightarrow Y_3 = (-1,-1,1,-1) ; X_4W = (-4,0,0,4) \rightarrow Y_4 = (-1,-1,-1,1) ;$$

$$Y_1W^T = (2,2,-2,-2) \rightarrow X_1 = (1,1,-1,-1) ; Y_2W^T = (2,-2,2,-2) \rightarrow X_2 = (1,-1,1,-1) ;$$

$$Y_3W^T = (-2,2,-2,2) \rightarrow X_3 = (-1,1,-1,1) ; Y_4W^T = (-2,-2,2,2) \rightarrow X_4 = (-1,-1,1,1) ;$$

以上说明记忆训练对  $X_i$  与  $Y_i$  是一一对应的。当输入任一新的  $X$  与  $X_i$  中某一个相同时，联想记忆会很快找到相应的  $Y_i$  作为结果；当输入任一新的  $X$  与  $X_i$  中每一个都不相同时，联想记忆仍会得到与  $X$  相近的  $X_i$  作为结果。

### 本章小结

在 Hopfield 神经网络自联想记忆的基础上，本章讨论异联想记忆神经网络，重点介绍了离散型双向联想记忆神经网络 BAM 的结构、工作原理、权矩阵设计方法及稳定性。最后讨论了 BAM 双向联想记忆神经网络在规则推理方面的具体应用。

### 参考文献

- [1] Kosko B., Adaptive bidirectional associative memories, Applied Optics, 1987, 26(23):4947-4960.
- [2] Kosko B., Bidirectional associative memories, IEEE Trans. On SMC, 1988, 18(1):49-60.
- [3] 李敏, 徐福缘, 黄延磊, 基于 BAM 神经网络的推理研究, 桂林电子工业学院学报, 1999, 19(2): 9-13。
- [4] 焦李成. 神经网络系统理论. 西安: 西安电子科技大学出版社, 1994.
- [5] 赵瑞清, 王晖, 邱涤虹. 知识表示与推理. 北京: 气象出版社, 1991, 40-57
- [6] Li Minfu, Li Chenfu. Mapping rule-based systems into neural architecture. Knowledgebased Systems, Butterworth&CoLtd 1990
- [7] Kosko B., Differential Hebbian Learning. PAIP, Neural Networks for Computing, 1986, (4): 277-282
- [8] 南京航空航天大学翻印, 人工神经网络, 1997.9。
- [9] 闻新, 张洪钺, 周露, 控制系统的故障诊断和容错控制, 机械工业出版社, 1998 年 2 月, 北京。

### 思考题

1. 何为双向联想记忆？
2. BAM 是如何实现双向联想记忆的？
3. 请简要介绍如何利用离散 BAM 实现规则推理。
4. 已知  $A \in \{-1,1\}^N$ ,  $B \in \{-1,1\}^M$ ,  $N=10, M=5$ , 样本对  $P=4$ , 各个样本如下：

$$A_1 = [1, -1, 1, -1, 1, -1, 1, -1]^T ; A_2 = [1, 1, -1, -1, 1, 1, -1, -1]^T ;$$

$$A_3 = [1, 1, 1, -1, -1, -1, 1, 1]^T ; A_4 = [1, 1, 1, 1, -1, -1, -1, -1]^T$$

$$B_1 = [1, 1, 1, 1]^T ; B_2 = [1, 1, 1, -1]^T ; B_3 = [1, 1, -1, -1]^T$$

$$B_4 = [1, -1, 1, -1]^T 。请设计 BAM 权矩阵  $W$  。$$

## 第 5 章 CMAC 小脑神经网络

前面几章介绍的 BP 神经网络、Hopfield 神经网络和 BAM 双向联想记忆神经网络分别属于前馈和反馈神经网络，这主要是从网络的结构来划分的。如果从神经网络的函数逼近功能这个角度来分，神经网络可以分为全局逼近网络和局部逼近网络。当神经网络的一个或多个可调参数（权值和阈值）在输入空间的每一点对任何一个输出都有影响，则称该神经网络为全局逼近网络，前面介绍的多层前馈 BP 网络是全局逼近网络的典型例子。对于每个输入输出数据对，网络的每一个连接权均需进行调整，从而导致全局逼近网络学习速度很慢，对于有实时性要求的应用来说常常是不可容忍的。如果对网络输入空间的某个局部区域只有少数几个连接权影响网络输出，则称网络为局部逼近网络。对于每个输入输出数据对，只有少量的连接权需要进行调整，从而使局部逼近网络具有学习速度快的优点，这一点对于有实时性要求的应用来说至关重要。目前常用的局部逼近神经网络有 CMAC 网络、径向基函数 RBF 网络和 B 样条网络等，其结构原理相似，本书主要介绍 CMAC 神经网络和 RBF 神经网络。

1975 年 J.S.Albus 提出一种模拟小脑功能的神经网络模型，称为 Cerebellar Model Articulation Controller<sup>[1-2]</sup>，简称 CMAC。CMAC 网络是仿照小脑控制肢体运动的原理而建立的神经网络模型。小脑指挥运动时具有不假思索地作出条件反射迅速响应的特点，这种条件反射式响应是一种迅速联想。CMAC 网络有三个特点：

作为一种具有联想功能的神经网络，它的联想具有局部推广（或称泛化）能力，因此相似的输入将产生相似的输出，远离的输入将产生独立的输出；

对于网络的每一个输出，只有很少的神经元所对应的权值对其有影响，哪些神经元对输出有影响则有输入决定；

CMAC 的每个神经元的输入输出是一种线性关系，但其总体上可看做一种表达非线性映射的表格系统。由于 CMAC 网络的学习只在线性映射部分，因此可采用简单的 算法，其收敛速度比 BP 算法快得多，且不存在局部极小问题。CMAC 最初主要用来求解机械手的关节运动，其后进一步用于机械人控制、模式识别、信号处理以及自适应控制等领域<sup>[3-4]</sup>。

### 5.1 CMAC 结构及工作原理<sup>[5]</sup>

#### 5.1.1 CMAC 结构

简单的 CMAC 结构如图 5 - 1 所示，图中  $X$  表示  $p$  维输入状态空间， $A$  为具有  $n$  个单元的存储区（亦称为相联空间或概念记忆空间）。设 CMAC 网络的输入向量用  $p$  维输入状态空间  $X$  中的点  $x^i = (x_1^i, x_2^i, \dots, x_p^i)^T$  表示，对应的输出向量用  $y_i = F(x_1^i, x_2^i, \dots, x_p^i)$  表示，图中  $i = 1, 2, 3$  输入空间的一个点  $X^i$  将同时激活  $A$  中的  $N_L$  个元素（图 5 - 1 中  $N_L = 4$ ），使其同时为 1，而其他大多数元素为 0，网络的输出  $y_i$  即为  $A$  中 4 个被激活单元的对应权值累加和。 $N_L$  称为泛化参数，反应网络泛化能力大小，也可将其看做信号检测单元的感受野大小。对 CMAC 来说，其工作过程一般包括两个方面：一是结果输出计算及误差生成阶段；二是权值调整阶段。

#### 5.1.2 CMAC 工作原理

### 1. CMAC 的结果输出计算及误差产生阶段

一般来说,实际应用时输入向量的各分量来自不同的传感器,其值多为模拟量,而  $A$  中每个元素只取 0 或 1 两种值。为使  $X$  空间的点映射为  $A$  空间的离散点,必须先将模拟量  $X^i$  量化,使其成为输入状态空间的离散点。设输入向量  $X$  的每一分量可量化为  $q$  个等级,则  $p$  个分量可组

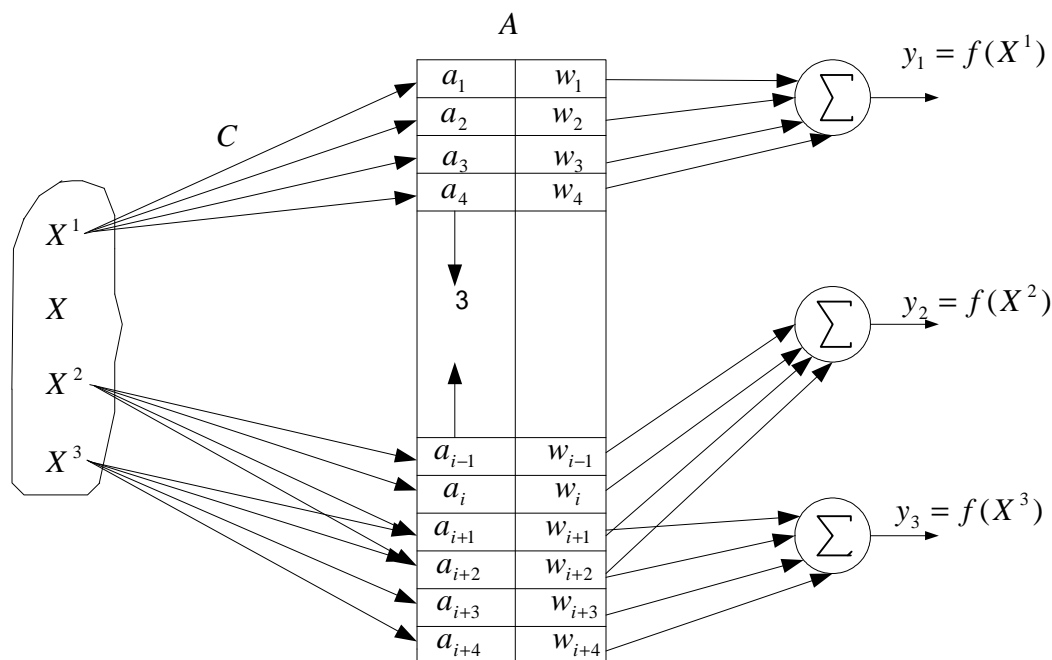


图 5 - 1 CMAC 网络的结构

合为输入状态空间  $q^p$  种可能的状态  $X^i$ ,  $i = 1, 2, \dots, q^p$ 。其中每一个状态  $X^i$  都要映射为  $A$  空间存储区的一个集合  $A^i$ ,  $A^i$  的  $N_L$  个元素均为 1。从图 5 - 1 可以看出,在  $X$  空间接近的样本  $X^2$  和  $X^3$  在  $A$  中的映射  $A^2$  和  $A^3$  出现了交集  $A^2 \cap A^3$ , 即它们对应的 4 个权值中有两个是相同的, 因此有权值累加和计算的两个输出也较接近, 从函数映射的角度看, 这一特点可起到泛化的作用。显然, 对相距很远的样本  $X^1$  和  $X^3$ , 映射到  $A$  中的  $A^1 \cap A^3$  为空集, 这种泛化不起作用, 因此是一种局部泛化。输入样本在输入空间距离越近, 映射到  $A$  存储区后对应交集的元素就越接近  $N_L$ , 其对应的输入样本在  $A$  中产生的交集起到了将相近样本聚类的作用。

为使对于  $X$  空间的每一个状态, 在  $A$  空间均存在惟一的映射。应使  $A$  存储区中单元的个数至少等于  $X$  空间的状态个数, 即  $n \geq q^p$ 。设将三维输入的每个分量量化为 10 个等级, 则  $n \geq 1000$ 。对于许多实际系统,  $q^p$  往往要比这个数字大得多, 但由于大多数学习问题不会包含所有可能的输入值, 实际上不需要  $q^p$  个存储单元来存放学习的权值。  $A$  相当于一种虚拟的内存地址, 每个虚拟地址和输入状态空间的一个样本点相对应。通过哈希编码 (Hash - coding) 可将

具有  $q^p$  个存储单元的地址空间  $A$  映射到一个小得多的物理地址连接  $A_i$  中。

对于每个输入， $A$  中只有  $N_L$  个单元为 1，而其余的均为 0，因此  $A$  是一个稀疏矩阵。哈希编码是压缩稀疏矩阵的常用技术，具体方法是通过一个产生随机数的程序来实现的。以  $A$  的地址作为随机数产生的程序的变量，产生的随机数作为  $A_i$  的地址。由于产生的随机数限制在一个较小的整数范围内，因此  $A_i$  远比  $A$  小得多。显然，从  $A$  到  $A_i$  的压缩是一种多对少的随机映射。在  $A_i$  中，对每一个样本有  $N_L$  个随机地址与之对应， $N_L$  个地址存放的权值通过学习得到，其累加和即作为 CMAC 的输出。其表达式为 (5-1)

$$y_i = \sum_{j=1}^{N_L} w_j a_j(x), \quad i = 1, \dots, m \quad (5-1)$$

式(5-1)中  $w_j$  为第  $j$  个存储单元的权值，若  $a_j(x)$  激活，则其值为 1，否则为 0，只有  $N_L$  个存储单元对输出有影响。相近的输入激活的存储单元有交叠，产生相近的输出，不相近的输入将产生不相近的输出。对应的误差表达式如 (5-2)：

$$\Delta E_i = \overline{y_s} - \sum_{j=1}^{N_L} w_j a_j(x) \quad i = 1, 2, \dots, m \quad (5-2)$$

## 2. CMAC 的权值调整阶段

CMAC 算法中结果输出阶段从 CMAC 存储单元产生一实际输出，学习过程根据期望输出与实际输出的误差大小来更新 CMAC 存储单元中的权值。在常规 CMAC 算法中误差被平均分配到所有被激活的存储单元。设  $s$  为某一状态， $w_j(t)$  是经过第  $t$  次迭代后存储在第  $j$  个存储单元中的

权值。常规 CMAC 更新  $w_j(t)$  算法为：

$$w_j(t) = w_j(t-1) + \frac{\alpha}{N_L} a_j(\overline{y_s} - \sum_{j=1}^{N_L} a_j(x) w_j(t-1)) \quad (5-3)$$

式(5-3)中  $\overline{y_s}$  为状态  $s$  的期望输出， $\sum_{j=1}^{N_L} a_j(x) w_j(t-1)$  为状态  $s$  的实际输出， $\alpha$  为学习常数。

## 5.2 CMAC 改进学习算法

在 CMAC 神经网络应用中，一般来说其实时性要求都较高。如非线性动态系统的在线辨识，不仅要求精度高，而且要求快速学习。但是，常规的 CMAC 仍然需要多个周期才能达到一定的收敛精度，也就是说，常规 CMAC 虽然其收敛速度快于 BP 网络，但作为在线学习来说，仍难满足其快速性的要求。为此，近些年来，一些学者提出了许多相关算法，大多时在常规 CMAC 算法上进行改进。下面介绍几种重要的改进算法。

### 5.2.1 模糊 CMAC 神经网络算法

为提高 CMAC 学习的实时性和准确性。Nie J. 和 Geng Z.J. 等人将模糊自组织竞争算法引入

CMAC 中, 来改造常规的 CMAC 神经网络。提出了一种模糊 CMAC 算法<sup>[6-7,10]</sup>。作如下定义:

**定义 1** 设 CMAC 中某个输入  $x$  激活的  $N_L$  个的存储单元可看作中心为  $z_j$ ,  $j = 1, 2, \dots, N_L$ , 宽度为  $2\delta$  的一个邻域  $\psi_j$ , 称  $\psi_j$  为联想域。对常规 CMAC 来说, 若  $a_j \in \psi_j$ , 则  $a_j = 1$ , 否则为 0。联想域有交叠, 使网络有局部泛化能力。

**定义 2** 设输入  $x \in R^n$ , 联想域  $\psi_j$   $j = 1, 2, \dots, N_L$  的中心为  $z_j$ , 半径为  $\delta$ , 将每个存储单元用一与输入同维的向量  $a_j$  表示, 则联想度为:

$$a_{\bar{j}} = \begin{cases} \frac{\delta - \|a_j - x\|}{\delta}, & \|a_j - x\| \leq \delta \\ 0 & \text{其它} \end{cases} \quad (5-4)$$

基于联想度的概念, 可获得一模糊化的联想向量  $a_{\bar{j}}(x) = (a_{\bar{j}1}, \dots, a_{\bar{j}N_L})^T$ , 进而得到 FCMAC 的输出:

$$y_i = \sum_{j=1}^{N_L} w_j a_{\bar{j}}(x), \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, N_L \quad (5-5)$$

若令  $a_{\bar{j}} = 1$ ,  $j = 1, \dots, N_L$ , 其它情况  $a_i = 0$ , 则  $a_{\bar{j}}(x)$  退化为二进制向量  $a_j(x)$ , 可见常规 CMAC 为 FCMAC 的特殊情况。

对网络权值 (存储的数据) 学习调整。文献[8,9]采用以下算法:

$$w_j(t) = w_j(t-1) + \eta(k) \left( \overline{y_s} - \sum_{j=1}^{N_L} a_{\bar{j}} w_j(t-1) \right) a_{\bar{j}} / \sum_{j=1}^{N_L} a_{\bar{j}} \quad j = 1, 2, \dots, N_L \quad (5-6)$$

$$\eta(t) = \begin{cases} 1.05\eta(t-1), & e(t-1) < e(t-2) \\ 0.7\eta(t-1), & e(t-1) > 1.04e(t-2) \\ \eta(t-1), & \text{其它} \end{cases} \quad (5-7)$$

由于联想度的引入, 也省去了 CMAC 的离散、量化、编码、HASHING 映射等复杂运算。对于联想域大小的确定, 采用自组织竞争算法来实现, 从而完成输入空间的自组织分割。使网络的学习速度和精度得到较大的提高。

### 5.2.2 基于信度分配的平衡学习 CMAC 神经网络算法

在常规 CMAC 及模糊 CMAC 学习算法的权值学习调整中, 误差被平均分配给每个被激活的存储单元, 而未考虑各个被激活存储单元对误差的贡献率, 也即在经过  $t$  次学习后, 对调整次数不同的激活存储单元其权值的可信度仍被看成完全相同的。这种权值更新算法完全违背了信度分配的概念, 这样的权值学习算法, 必然使那些权值不该调整或应该较少调整的存储单元 (其权值可信度高) 需反复学习调整; 而对误差贡献较大的存储单元 (其权值可信度低), 本应该使其权值得到较大调整, 但实际上权值学习调整量减少。为了达到预定的逼近精度, 网络必须多次反复学习, 从而使 CMAC 的学习效率降低, 学习时间延长。

为了提高 CMAC 学习速度, 文献[11]在分析常规 CMAC 权值调整规则的基础上, 考虑到已学习知识的可信度, 提出一种基于信度分配的 CA-CMAC 算法 (Credit assignment CMAC)。同时引入 CMAC 存储单元地址函数, 来解决信度的定义问题。文献[12]在此基础上进一步考虑到网络权值

调整时，新知识“学习”与旧知识“遗忘”的平衡问题，提出一种基于“平衡学习”的 CMAC 神经网络学习算法。下面以 2 维 CMAC 为例进行介绍。

### 1. 常规 CMAC 神经网络的分级量化方法

CMAC 的基本思想就是将学习的数据(知识)存储在交叠的存储单元(记忆空间)中，其输出为相应激活单元数据的累加和。以 2 维 CMAC 为例，定义输入矢量为变量  $x_1$  和  $x_2$ 。2 维 CMAC 结构如图 5 - 2 所示。在本例中每个变量分成 7 个等分区域，对每个状态变量有 3 级，每级由 3 块构成。对第 1 级，变量  $x_1$  被划为 3 块 A, B 和 C，变量  $x_2$  被划为 a, b 和 c。则 Aa, Ab, Ac, Ba, Bb, Bc, Ca, Cb, 和 Cc 为存储数据的地址或单元(hypercubes)；与此相似，对第 2 级，有存储单元 Dd, De, Df, Ed, Ee, Ef, Fd, Fe 和 Ff；对第 3 级，有存储单元 Gg, Gh, Gi, Hg, Hh, Hi, Ig, Ih 和 Ii。必须注意，只有同级的不同变量的分块组合才能形成存储单元。像组合 Ad, Db 等是不存在的。此处的 2 维 CMAC 中，用 27 存储单元来存储 49 个状态的数据。

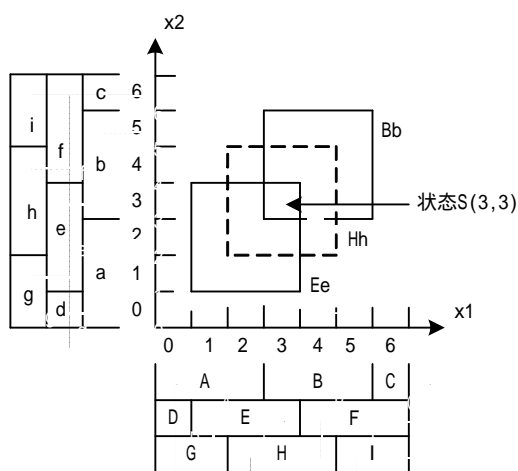


图 5 - 2 2 维 CMAC 结构

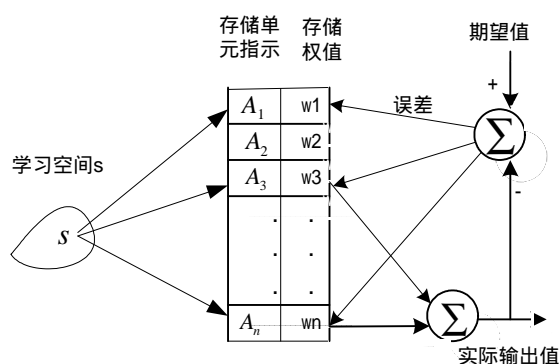


图 5 - 3. CMAC 的基本原理

CMAC 的基本思想如图 5 - 3 的表达。在 CMAC 算法实现过程中有两个阶段：结果输出阶段和权值学习阶段。在结果输出阶段，CMAC 将所有被输入状态激活的存储单元的权值(存储数据)相加。

$$y_s = \sum_{j=1}^N a_j w_j \quad (5 - 8)$$

$m$  为状态变量的级数，存储单元数为  $N$ ， $n$  为总状态数。 $y_s$  为状态  $s$  ( $s=1, \dots, n$ ) 的实际输出值。 $w_j$  为第  $j$  个存储单元的权值， $a_j$  是第  $j$  个存储单元是否被状态  $s$  激活的指示。由于每个状态仅被  $m$  个存储单元包含，只有这些存储单元被激活，其  $a_j$  为 1，而其他存储单元  $a_j$  全为 0。如图 5 - 2 中，存储单元 Bb, Ee 和 Hh 被状态  $s(3,3)$  激活，所以仅有这三个  $a_j$  为 1，而其他存储单元  $a_j$  全为 0。

CMAC 算法中结果输出阶段从 CMAC 存储单元产生一实际输出，学习过程根据期望输出与

实际输出的误差大小来更新 CMAC 存储单元中的权值。在常规 CMAC 算法中误差被平均分配到所有被激活的存储单元。设  $s$  为某一状态,  $w_j(t)$  是经过第  $t$  次迭代后存储在第  $j$  个存储单元中的权值。常规 CMAC 更新  $w_j(t)$  算法为:

$$w_j(t) = w_j(t-1) + \frac{\alpha}{m} a_j (\bar{y}_s - \sum_{j=1}^N a_j w_j(t-1)) \quad (5-9)$$

(5-9)式中  $\bar{y}_s$  为状态  $s$  的期望输出,  $\sum_{j=1}^N a_j w_j(t-1)$  为状态  $s$  的实际输出,  $\alpha$  为学习常数。

必须注意只有那些被激活的存储单元的权值才被更新。在上面的常规算法中误差被平均分配到所有被激活的存储单元,但是经过  $t-1$  次迭代后,最初的存储单元已经包含了一些先前学习的知识,不是每一个存储单元都有相同的学习历史,所以这些存储单元也不应有相同的可信度。无视这些差异,所有被激活的存储单元都获得相同的校正误差,那么那些由未学习状态产生的误差将对先前学习的信息产生“腐蚀”(corrupt),当然在经过多个训练周期后,这种腐蚀情形会逐渐消失,这也是许多常规 CMAC 算法应用成功的根据之所在。但是对在线动态系统的学习来说,其实时性要求非常高,有的情况下只允许在一到两个周期内就完成学习任务,也就没有足够的时间来消除这种腐蚀。因此其学习结果常常是无法满足在线学习的要求。

## 2. 基于信度分配的 CMAC(CA-CMAC)神经网络

为了避免“腐蚀”效应,校正误差必须根据存储单元的可信度进行分配。然而,在 CMAC 学习过程中,还没有一个好的方法来决定某一存储单元对目前的误差负更多的责任。换句话说,还没有一个好的方法来决定存储单元权值。唯一可用的信息是该存储单元权值目前更新的次数,文献[11]假设存储单元学习更新次数越多,其存储的数值越可靠。因此存储单元的学习次数被看成其可信度。可信度越高,其权值修正越小。由此(5-9)式改写为:

$$w_j(t) = w_j(t-1) + \alpha a_j \left\{ \frac{(f(j)+1)^{-1}}{\sum_{l=1}^m (f(l)+1)^{-1}} \right\} (\bar{y}_s - \sum_{j=1}^N a_j w_j(t-1)) \quad (5-10)$$

此处  $f(j)$  是第  $j$  个存储单元的学习次数,  $m$  是某状态激活的存储单元数。此处权值更新思

想是校正误差必须与激活单元的学习次数成反比。此处用  $\frac{(f(j)+1)^{-1}}{\sum_{l=1}^m (f(l)+1)^{-1}}$  代替了(5

-9)式的  $\frac{1}{m}$ , 它有效地改善了学习性能。但这并不是最好的结果,因为它没有进一步考虑已学习的先前知识与未学习或少学习的知识各自对网络输出误差的影响程度。即“学习”与“遗忘”的平衡问题。

## 3. 改进的基于信度分配 CMAC 神经网络 (ICA-CMAC)

根据以上分析,一种“平衡学习”的概念被提出,由此设计一种改进的基于信度分配的 CMAC 神经网络模型 ICA-CMAC(Improved Credit Assignment CMAC),此时(5-10)式被改写为:



$$w_j(t) = w_j(t-1) + \alpha \alpha_j \left\{ \frac{(f(j)+1)^{-k}}{\sum_{l=1}^m (f(l)+1)^{-k}} \right\} (\bar{y}_s - \sum_{j=1}^N a_j w_j(t-1)) \quad (5-11)$$

(5-11)式中 $k$ 是一平衡学习常数,很明显看出当 $k$ 为0或1时,ICA-CMAC分别为常规CMAC和文献[11]的CA-CMAC。也就是说,CMAC和CA-CMAC是ICA-CMAC的一种特殊情形。激活存储单元的学习次数 $f(j)$ 越大,则其存储的知识(先前学习的信息)越多。平衡学习常数 $k$ 越大,对学习次数 $f(j)$ 较大的存储单元,其权值改变较少。在 $k$ 很大时,对学习次数 $f(j)$ 较大的存储单元其权值基本不变。此时未学习的或者学习次数 $f(j)$ 较少的激活单元在权值修正时,将获得大多数误差校正。此种情形下,网络学习中“记忆”即“已学习知识的保持”占主导地位。

反之当 $k$ 值很小时,学习次数 $f(j)$ 对信度分配的影响也较小。当 $k=0$ 时,学习次数 $f(j)$ 对信度分配的影响为零。此时,误差被平均分配到所有被激活的存储单元。所有被激活的存储单元都有相同的信度分配,而不管学习次数 $f(j)$ 的大小。这时在网络学习中“遗忘”占主导地位。

可见 $k$ 是一平衡学习常数,它反映了在网络训练过程中,先前学习的信息与未学习或少学习信息对存储单元权值调整的影响程度。不同的 $k$ 将会有不同的学习结果。从后面的仿真结果可知当 $k$ 为某一数值时,其学习速度最快。说明此时网络的“记忆”与“遗忘”达到了最佳平衡。

#### 4. 地址函数设计

在常规CMAC中,一般采用HASHING技术来压缩存储空间,但HASHING映射会造成碰撞的发生,使得CMAC的逼近性能下降。文献[11]采用地址函数来产生所需要的存储单元的标记,它通过一定规则为所有可能存储单元编码,是一种简洁的地址方法,而且不存在数据碰撞问题。

以三维(3-D)CMAC为例,设 $m$ 是CMAC的级数, $nb$ 是每级包含的块数。则每维的等分块数为 $m*(nb-1)+1$ 。在本例中,每块包含 $m$ 个状态,仅用 $N=m*nb^3$ 个存储单元来映射 $(m*(nb-1)+1)^3$ 状态。考虑由 $(x_1, x_2, x_3)$ 表达的状态 $s$ ,由它激活的存储单元数为 $m$ ,各激活存储单元的地址函数为 $s(j)$ , for  $j=1, \dots, m$ , 则 $s(j) = F(x_1, x_2, x_3, j)$ , 定义:

if  $j=1$ , 则  $i=0$ , 其它  $i=m-j+1$ ;

$$ax = \text{int}((x_1 + i) / m);$$

$$ay = \text{int}((x_2 + i) / m);$$

$$az = \text{int}((x_3 + i) / m);$$

$$s(j) = F(x_1, x_2, x_3, j) = ax + ay + az * nb^2 + (j-1) * nb^3 + 1。$$

#### 5. 仿真算例及结果分析

为了进一步说明ICA-CMAC的在线学习效果,以下面的非线性函数为例,对不同平衡学习参数 $k$ 比较网络的学习速度。

$$z(x, y) = (x^2 - y^2) \sin 5x \quad -1 \leq x \leq 1, \quad -1 \leq y \leq 1 \quad (5-12)$$

本例中每个变量包含 64 个等分块。取级数  $m=9$ , 每级包含  $nb=8$  块。则总状态数  $(m * (nb - 1) + 1)^2 = 4096 = 64 * 64$ , 存储单元总数为:  $m * nb^2 = 9 * 8 * 8 = 576$  (仅为总状态数的 14%)。取  $\alpha = 1$ 。其训练数据数为 4096。

学习结果如图 5-4 所示, 在不同的  $k$  值下, 计算网络的绝对误差  $TAE$  (The total absolute error) 和方均根误差  $RMSE$  (root mean square error)。取  $k$  值为: 0.0, 0.7, 1.5, 2.0。其第 1 周期到第 6 周期和第 20 周期到第 25 周期的计算结果见表 5-1 和表 5-2。

$$TAE = \sum_{s=1}^n |\bar{y}_s - y_s| \quad (5-13)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{s=1}^n (\bar{y}_s - y_s)^2} \quad (5-14)$$

式中  $n$  为总状态数,  $\bar{y}_s$  是状态  $s$  的期望输出值,  $y_s$  是状态  $s$  的实际输出数值。

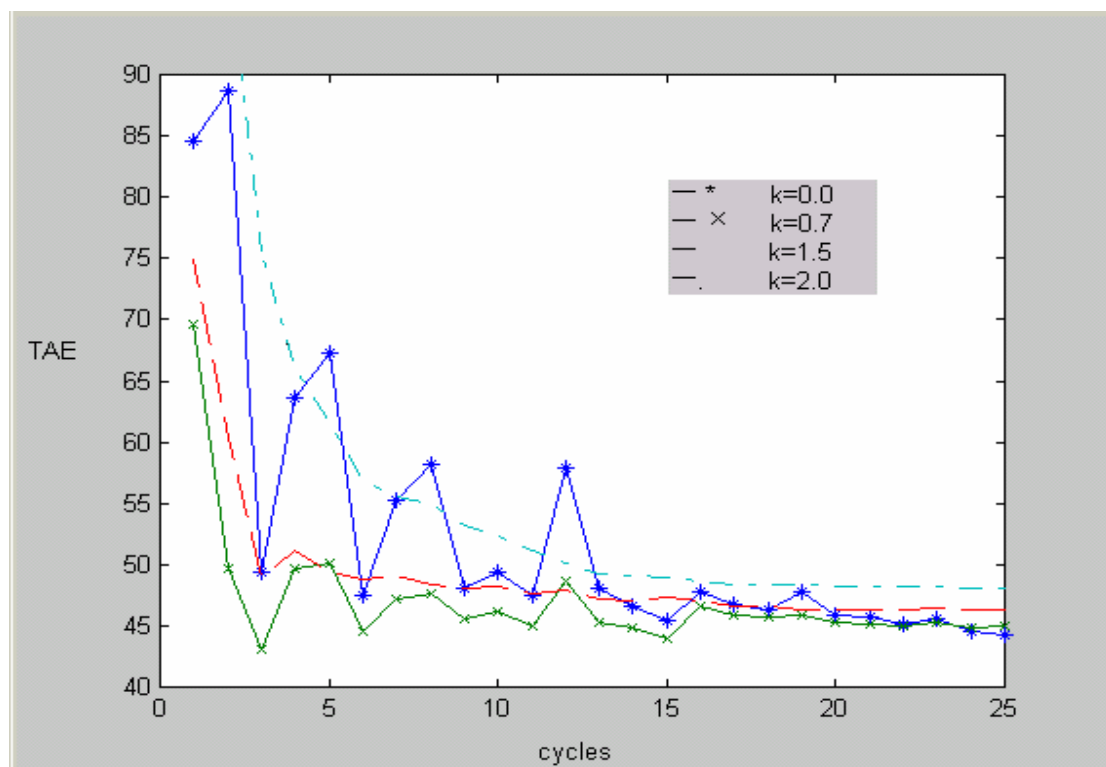


图 5-4 ICA-CMAC 的绝对误差

表 5-1 ICA-CMAC 的绝对误差

$k \setminus \text{cycle}$	1	2	3	4	5	6		20	21	22	23	24	25
0.0	84.53	88.70	49.39	63.56	67.27	47.51	$\downarrow$	45.89	45.64	45.13	45.62	44.58	44.22
0.7	69.67	49.74	43.01	49.60	50.17	44.47	$\downarrow$	45.24	45.16	45.09	45.31	44.79	44.95
1.5	74.83	60.33	48.87	51.17	49.42	48.72	$\downarrow$	46.34	46.37	46.34	46.38	46.27	46.30
2.0	193.9	100.1	75.42	66.08	61.52	56.86	$\downarrow$	48.25	48.28	48.26	48.18	48.05	48.01

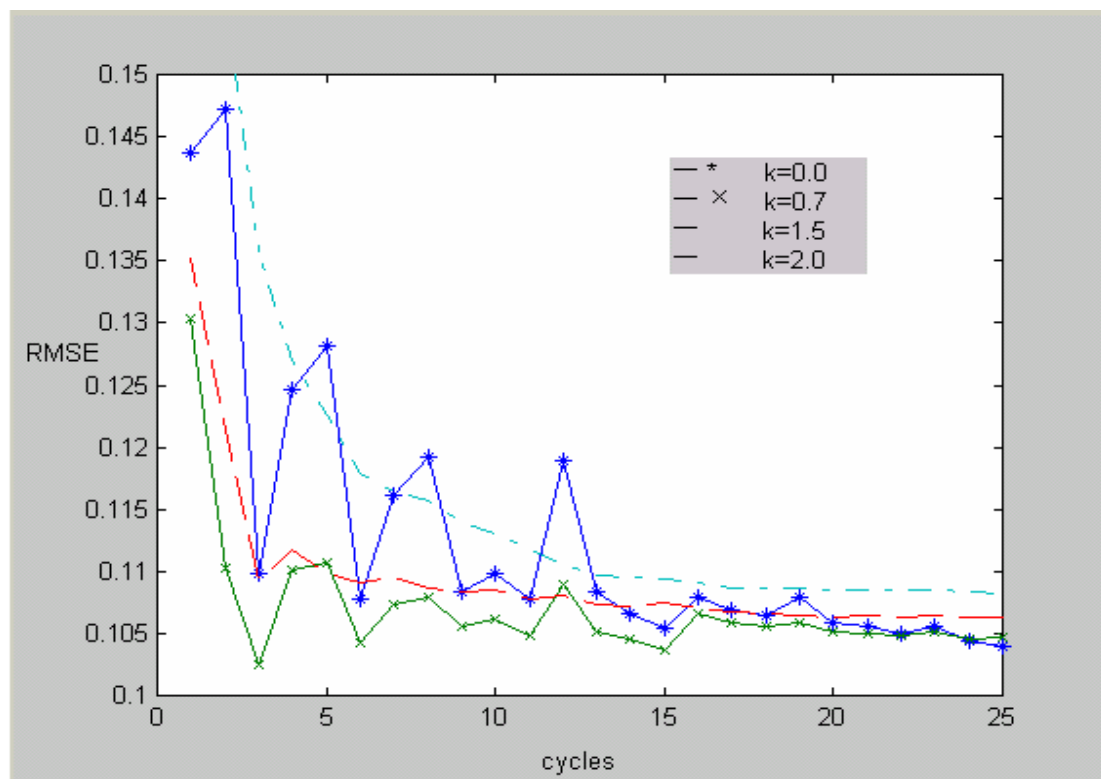


图 5 - 5 ICA-CMAC 方均根误差

表 5 - 2 ICA-CMAC 方均根误差

$k$ /cycle	1	2	3	4	5	6		20	21	22	23	24	25
0.0	0.14	0.14	0.11	0.12	0.12	0.11	$\downarrow$	0.106	0.106	0.105	0.105	0.104	0.104
0.7	0.13	0.11	0.10	0.11	0.11	0.10	$\downarrow$	0.105	0.105	0.104	0.105	0.104	0.104
1.5	0.13	0.12	0.11	0.11	0.10	0.11	$\downarrow$	0.106	0.106	0.106	0.106	0.106	0.106
2.0	0.21	0.15	0.13	0.13	0.12	0.12	$\downarrow$	0.108	0.108	0.108	0.108	0.108	0.108

从图 5 - 4 和图 5 - 5 可以看出, 在网络学习的初始阶段, 不同的  $k$  值, 误差下降的速度有很大的差别,  $k = 0$  时(常规 CMAC)收敛速度较慢, 随着  $k$  值的增加, 收敛速度变快, 在  $k = 0.7$  时收敛速度达到最快;  $k$  值继续增大, 网络学习速度反而变慢, 特别是  $k = 2.0$  其学习速度比常规 CMAC 还慢。可见  $k = 0.7$  为最佳平衡学习常数, 此时, 网络学习时的“记忆”与“遗忘”达到最佳平衡。文献[11]提出的 CA-CMAC ( $k = 1$ ) 的学习速度处于最佳平衡值附近, 但它未能分析网络学习时“记忆”与“遗忘”的平衡问题, 其学习结果虽然比常规 CMAC 好, 但并非是最好的结果。

从图 5 - 4 和图 5 - 5 及表 5 - 1, 表 5 - 2 还可以看出, 在网络学习的后续阶段(后 10 个周期), 尽管  $k$  值不同, 其学习误差区别很小, 不同  $k$  值的网络都学习得很好。这是因为网络学习过程中, 权值的调整依赖于误差的分配, 误差的分配又依赖于各存储单元的学习次数, 而学习次数在学习时间足够长时, 各存储单元将趋于相同。

对于非线性动态系统的在线学习来说, 最初的学习速度是极为重要的。因此, CMAC 神经网络学习算法的改进, 对提高非线性系统在线辨识的实时性有一定的指导意义。

### 5.2.3 基于信度分配的模糊 CMAC 神经网络学习算法<sup>[13]</sup>

#### 1. 算法设计

基于信度分配的模糊 CMAC 神经网络学习算法, 是从 CMAC 权值学习遗忘 (previous learned information be corrupted by unlearned state) 的角度出发, 将文献[11]的信度分配的思想引入模糊

CMAC (FCMAC) 的权值学习调整之中, 提出一种基于信度分配的模糊 CMAC (FCACMAC) 神经网络学习算法, 使网络的学习更加合理高效。仿真结果表明 FCACMAC 有较好的学习速度和较高的学习精度。

网络的存储单元的激活方式及输出计算方法与模糊 CMAC 完全相同; 其权值调整是将式(5 - 6)和 (5 - 10) 相结合, 得到:

$$w_j(t) = w_j(t-1) + \frac{a_{ff}}{\sum_{j=1}^{N_k} a_{ff}} \left\{ \frac{(f(j)+1)^{-1}}{\sum_{l=1}^{N_k} (f(l)+1)^{-1}} \right\} (y_s - \sum_{j=1}^{N_k} a_j w_j(t-1)) \quad (5-15)$$

此处按照信度分配的原则调整权值, 对先前学习次数较少, 误差贡献较大的存储单元, 其权值可信度低, 这时权值学习调整较多; 反之, 对学习次数较多, 对所产生的误差责任较少的存储单元, 权值可信度较高, 这时权值调整较少, 从而使 CMAC 的权值学习调整更加合理高效; 另外由于模糊联想度的引入, 省去了 CMAC 的离散、量化、编码、HASHING 映射等复杂运算。对于联想域大小的确定, 采用自组织竞争算法来实现, 从而完成输入空间的自组织分割。使网络的学习速度和精度得到较大的提高。

## 2. 仿真算例及结果分析

为了进一步说明 FCACMAC 的在线学习效果, 以下面的非线性函数为例, 研究常规 CMAC、模糊 CMAC (FCMAC) 及基于信度分配的模糊 CMAC (FCACMAC) 的学习效果。

### 一维非线性学习实例

假设有如下的非线性函数:

$$y(x) = \sin x + \cos x \quad -\pi \leq x \leq \pi \quad (5-16)$$

在学习过程中, 用网络的绝对误差 *TAE* (The total absolute error) 和方均根误差 *RMSE* (root mean square error) 来反映模型的学习速度和精度。其计算式同式 (5 - 13) 和 (5 - 14)

对 CMAC、FCMAC 及 FCACMAC, 其第 1 周期到第 20 周期的计算结果见表 5 - 3 和表 5 - 4。而图 5 - 6 和图 5 - 7 为相应的误差下降曲线。

表 5 - 3 一维 CMAC、FCMAC 及 FCACMAC 绝对误差

算法	周期									
	1	2	3	4	5	6	7	8	9	10
CMAC	38.13	27.98	14.15	7.09	4.08	2.53	2.07	1.76	1.65	1.58
FCMAC	13.66	10.46	5.62	2.98	1.56	0.82	0.43	0.25	0.14	0.086
FCACMAC	14.82	5.57	1.034	0.219	0.064	0.041	0.034	0.032	0.031	0.030
算法	周期									
	11	12	13	14	15	16	17	18	19	20
CMAC	1.57	1.53	1.51	1.49	1.48	1.46	1.45	1.44	1.43	1.43
FCMAC	0.057	0.043	0.036	0.033	0.031	0.031	0.030	0.029	0.029	0.028
FCACMAC	0.029	0.028	0.028	0.027	0.027	0.026	0.026	0.025	0.025	0.025

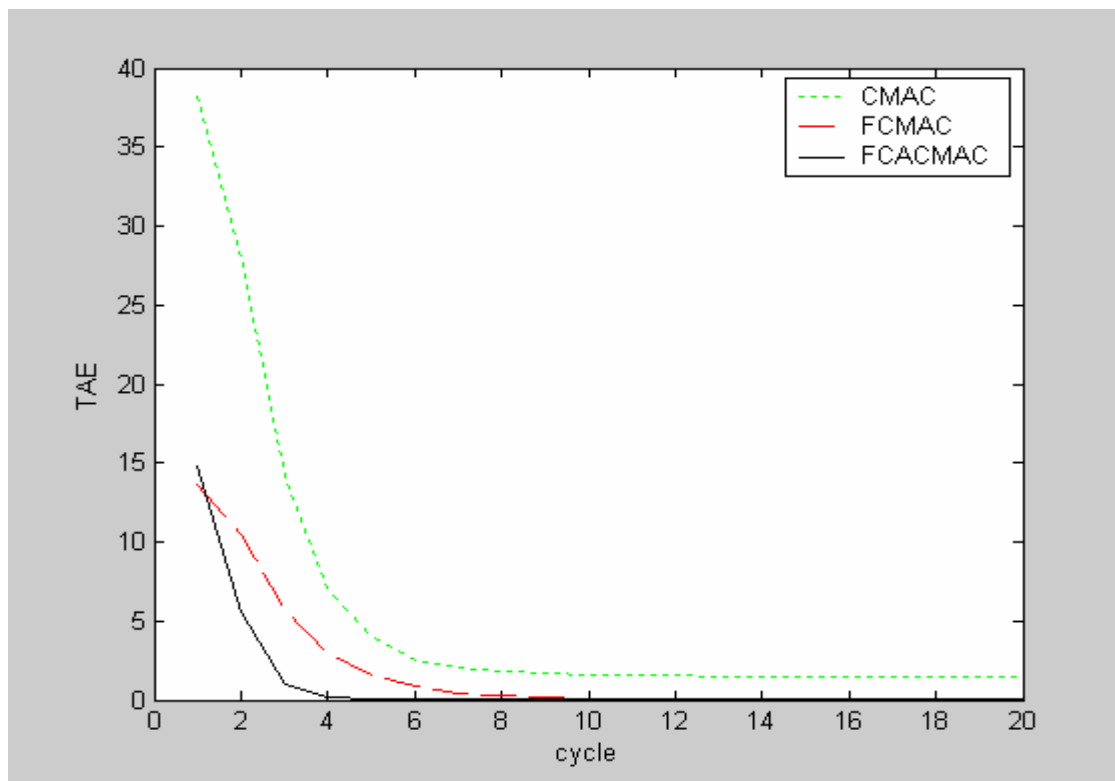


图 5 - 6 一维 CMAC、FCMAC 及 FCACMAC 绝对误差

表 5 - 4 一维 CMAC、FCMAC 及 FCACMAC 均方根误差

算法	周期									
	1	2	3	4	5	6	7	8	9	10
CMAC	0.772	0.661	0.470	0.333	0.252	0.199	0.180	0.166	0.161	0.157
FCMAC	0.462	0.404	0.296	0.216	0.156	0.113	0.082	0.062	0.046	0.037
FCACMAC	0.481	0.295	0.127	0.059	0.032	0.025	0.023	0.022	0.022	0.022
算法	周期									
	11	12	13	14	15	16	17	18	19	20
CMAC	0.157	0.156	0.154	0.153	0.152	0.151	0.151	0.150	0.150	0.149
FCMAC	0.030	0.026	0.024	0.023	0.022	0.022	0.022	0.021	0.021	0.021
FCACMAC	0.021	0.021	0.021	0.021	0.020	0.020	0.020	0.020	0.020	0.019

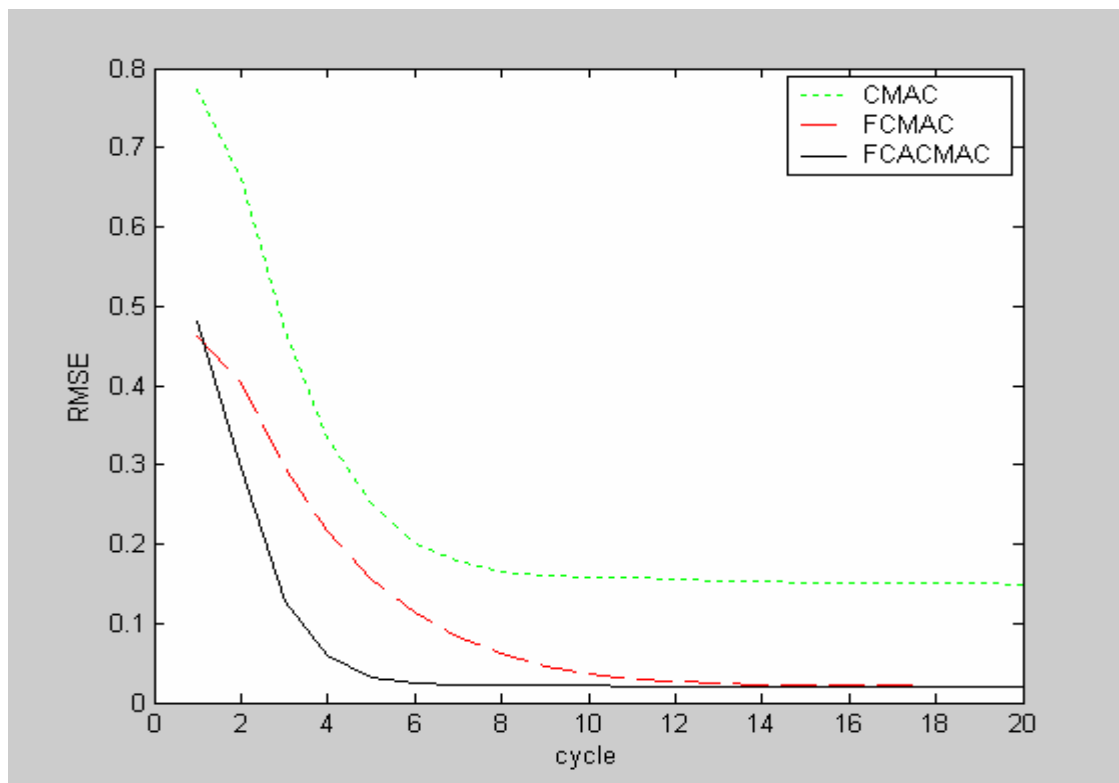


图 5 - 7 一维 CMAC、FCMAC 及 FCACMAC 均方根误差

## 二维非线性学习实例

假设有如下的非线性函数：

$$y(x_1, x_2) = \sin x_1 \cos x_2 \quad -\pi \leq x_1, x_2 \leq \pi \quad (5 - 17)$$

在网络学习过程中，绝对误差  $TAE$  (The total absolute error)和方均根误差  $RMSE$  (root mean square error)与式 (5 - 13) 和 (5 - 14) 完全相同。对 CMAC、FCMAC 及 FCACMAC，其第 1 周期到第 20 周期的计算结果见表 5 - 5 和表 5 - 6。其相应的误差下降曲线如图 5 - 8 和图 5 - 9 所示。

表 5 - 5 二维 CMAC、FCMAC 及 FCACMAC 绝对误差

算法	周期									
	1	2	3	4	5	6	7	8	9	10
CMAC	365.9	376.5	250.7	282.6	235.1	248.4	241.0	240.0	241.9	240.5
FCMAC	408.0	402.6	276.8	203.1	148.6	118.4	99.12	88.62	85.12	81.35
FCACMAC	344.9	254.2	153.4	109.5	83.75	71.26	60.69	57.96	53.71	53.76
算法	周期									
	11	12	13	14	15	16	17	18	19	20
CMAC	241.3	241.0	241.2	241.2	241.3	241.2	241.3	241.2	241.3	241.2
FCMAC	82.04	80.27	80.78	80.14	80.26	80.09	80.10	80.07	80.03	80.06
FCACMAC	52.63	52.74	52.50	52.40	52.40	52.30	52.31	52.27	52.26	52.25

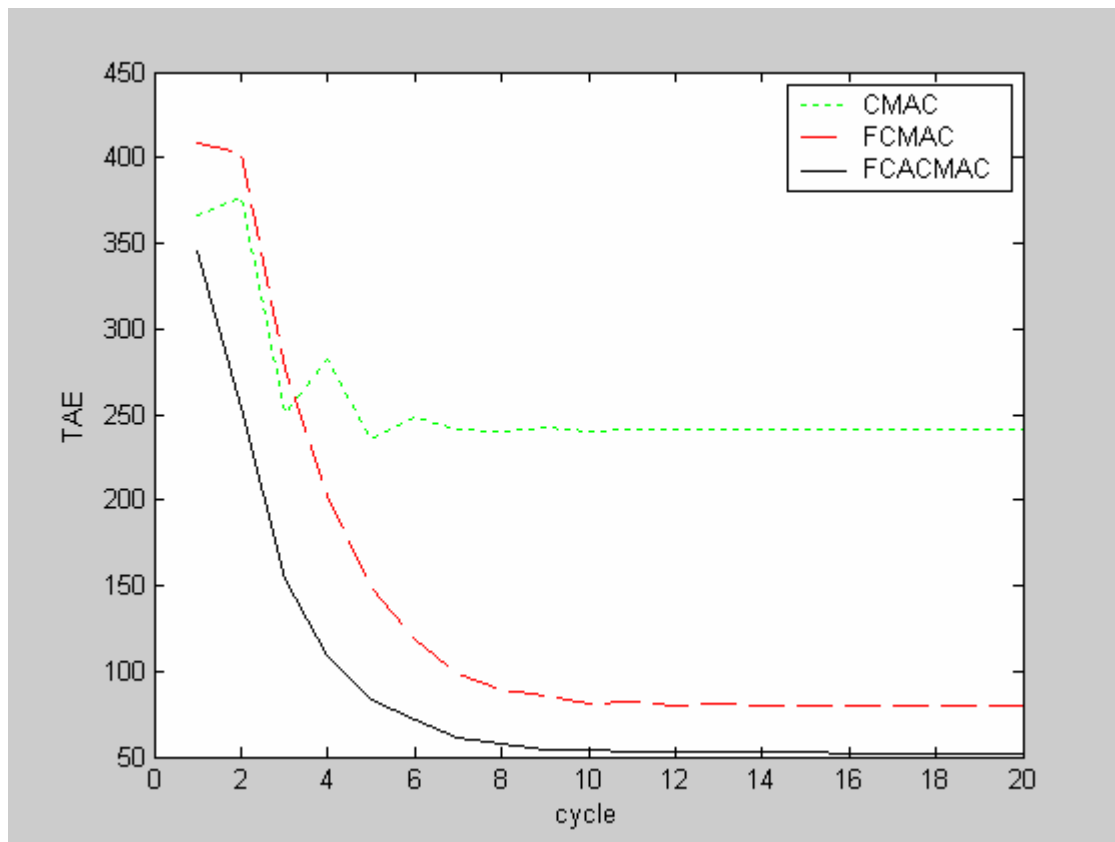


图 5 - 8 二维 CMAC、FCMAC 及 FCACMAC 绝对误差

表 5 - 6 二维 CMAC、FCMAC 及 FCACMAC 均方根误差

算法	周期									
	1	2	3	4	5	6	7	8	9	10
CMAC	0.2989	0.3032	0.2474	0.2627	0.2396	0.2463	0.2426	0.2421	0.2430	0.2423
FCMAC	0.3156	0.3135	0.2599	0.2227	0.1905	0.1700	0.1556	0.1471	0.1442	0.1409
FCACMAC	0.2902	0.2491	0.1935	0.1635	0.1430	0.1319	0.1217	0.1190	0.1145	0.1146
算法	周期									
	11	12	13	14	15	16	17	18	19	20
CMAC	0.2427	0.2426	0.2426	0.2427	0.2427	0.2427	0.2427	0.2427	0.2427	0.2427
FCMAC	0.1415	0.1399	0.1399	0.1398	0.1398	0.1398	0.1398	0.1398	0.1398	0.1398
FCACMAC	0.1134	0.1132	0.1131	0.1130	0.1130	0.1130	0.1130	0.1129	0.1129	0.1129

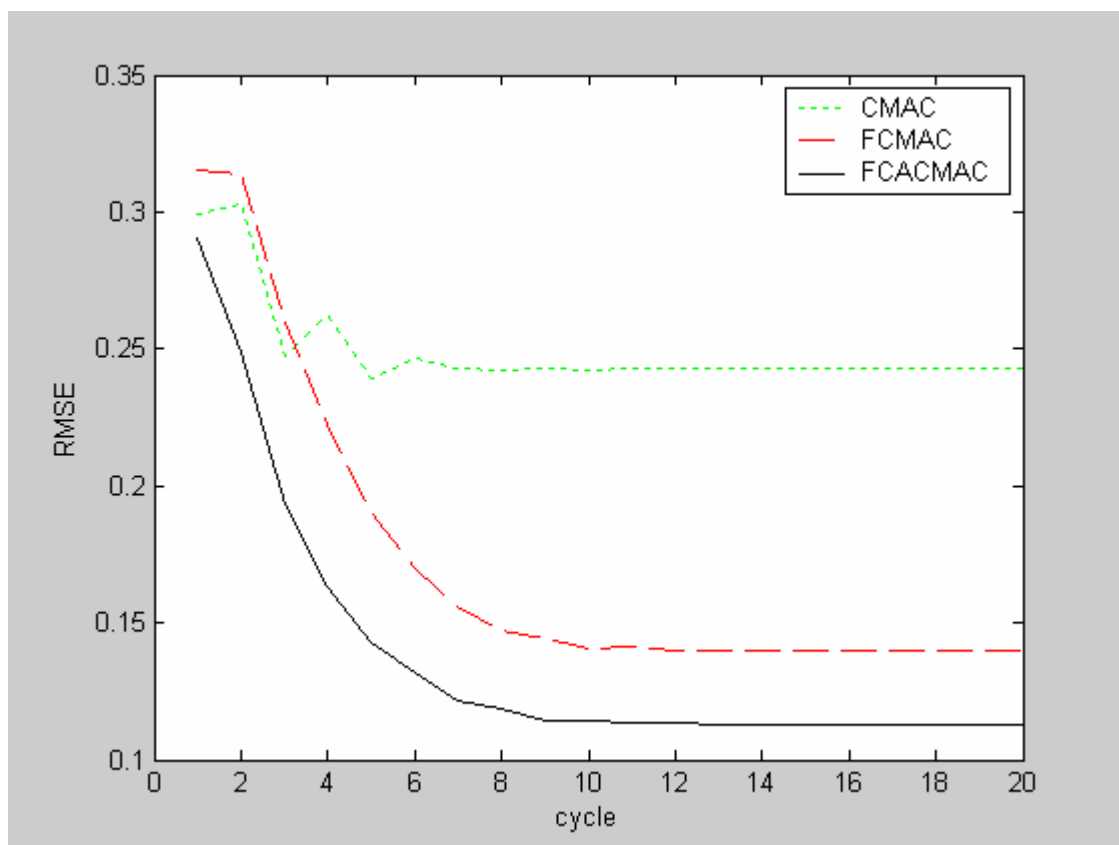


图 5 - 9 二维 CMAC、FCMAC 及 FCACMAC 均方根误差

### 结果讨论

从表 5 - 3 到表 5 - 6，图 5 - 6 到图 5 - 9 可以看出，无论是一维还是二维非线性目标函数，在网络学习过程中，不同的 CMAC 模型，误差下降的速度有很大的差别，其中常规 CMAC 收敛速度最慢，而基于信度分配的模糊 CMAC（FCACMAC）收敛速度最快，模糊 CMAC（FCMAC）学习速度处于二者之间。

同时还可看出 FCACMAC 的学习精度也比常规 CMAC 和模糊 FCMAC 高。随着目标函数维数的增加，其学习效果的差距更大，这是因为在低维情况下，计算量较小，误差的下降速度均较快，在 20 周期内都能达到了一定的逼近精度；随着维数的增加，计算量巨增，FCMAC、FCACMAC 算法的合理性将发挥更大的效果，使得三者之间的学习速度和学习精度的差距更明显，充分显示出 FCACMAC 在在线学习上的优越性。

由此可见，这种基于信度分配的模糊 CMAC 神经网络学习算法，由于将模糊 CMAC 的自组织竞争算法与基于信度分配的权值调整算法相结合，不仅在 CMAC 的存储单元激活、结果计算输出阶段，省去了 CMAC 的离散、量化、编码、HASHING 映射等复杂运算；而且在 CMAC 的权值学习调整阶段按各激活存储单元的权值可信度分配误差，使权值学习调整更加合理有效，极大地提高了 CMAC 神经网络的学习速度和学习精度。

## 5.3 CMAC 神经网络的应用

### ——基于CMAC的电液负载模拟器自学习控制<sup>[14]</sup>

#### 1. 问题提出

负载模拟器是模拟飞行器舵机所受空气动力负载或舰船舵机所受水动力负载的地面仿真设备，其功能是在实验室条件下复现舵面所受各种负载，考核操舵系统在实际负载条件下的性能指



标和可靠性，达到缩短研制周期，节约研制经费，提高可靠性和成功率的目的。

电液负载模拟器（简称加载系统）在加载的同时必须跟随舵机运动，这个运动是作用在加载系统上的干扰，使其产生多余力。通常在不加任何抑制或补偿的情况下，多余力所产生的误差非常大，严重影响加载系统的动态加载精度，因此如何抑制乃至消除多余力是电液负载模拟器必须解决的关键问题之一<sup>[15]</sup>。通常采用结构不变性原理的方法抑制多余力，但由于系统的结构刚度、时变特性、非线性等的影响，采用该方法抑制多余力达不到理想效果。

小脑模型关节控制器（CMAC）是一种模仿人的小脑功能的神经网络，它是基于局部学习的神经网络，即对于输入空间的某个局部区域，只有少数几个连接权影响网络的输出。CMAC模拟非线性连续函数逼近精度高，学习速度快，适合应用于实时控制。文献[16]已将CMAC神经网络成功应用于电液位置伺服系统的负载干扰抑制。考虑到用结构不变性原理的方法抑制多余力的特点及CMAC神经网络抑制负载干扰的作用，本文提出了采用结构不变性原理设计补偿环节的基于CMAC神经网络的电液负载模拟器新型控制结构，并将其应用于加载系统控制中。

## 2 CMAC结构原理与有关算法

$m$  维输入，一维输出的CMAC结构原理如图5 - 10所示。图中： $S(t)$ 为 $t$ 时刻 $m$  维离散输入状态空间， $A$ 为概念存储空间， $A'$  为物理存储空间， $W_i$  为物理存储空间地址对应的权值， $y(S(t))$ 为 $t$ 时刻CMAC的输出， $y_d(t)$ 为 $t$ 时刻的期望输出。

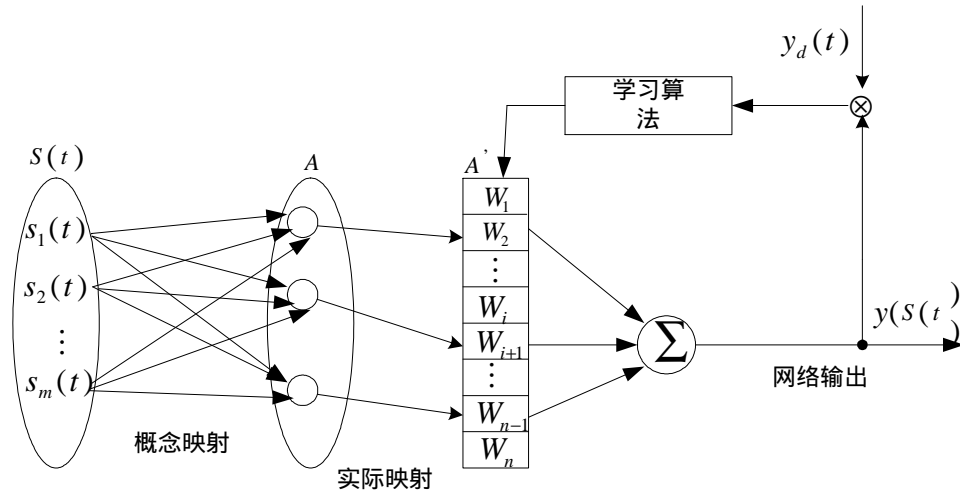


图5 - 10 CMAC结构原理

CMAC具体工作过程如下：

### 输入向量的量化

实际输入向量通常是模拟量，CMAC的输入是量化后的离散状态空间，因此需对输入向量进行量化。本文采用如

$$s_i = \text{round} \left( \frac{s_i' - s_{i\min}'}{s_{i\max}' - s_{i\min}'} R_i \right) \quad (5 - 18)$$

所示的量化算法。式中： $s_i$  为第 $i$  维输入的量化值； $s_i'$  为第 $i$  维输入； $s_{i\min}'$  为第 $i$  维输入的最小值； $s_{i\max}'$  为第 $i$  维输入的最大值； $R_i$  为第 $i$  维输入的量化级数； $\text{round}$  为取整函数。

### 概念映射

概念映射是将输入状态空间  $S$  中向量  $s_1 \sim s_m$  作为地址变量映射到空间  $A$  的映射过程。该映射通过滚动组合得到，具有局部泛化特性，即在输入空间中比较相近的向量，在输出空间中也比较相近。概念映射算法可由

$$A(S) = C \left( m_{s_1} + \sum_{i=2}^m \left( m_{s_i} \prod_{j=1}^{i-1} N_{s_j} \right) \right) + K \quad (5-19)$$

$$m_{s_i} = \text{floor} \left( \frac{s_i - K + C}{C} \right) + K \quad (5-20)$$

$$N_{s_j} = \text{floor} \left( \frac{R_j - K + C - 1}{C} \right) + 1 \quad (5-21)$$

表示。式中： $A(S)$  为输入向量在空间  $A$  的地址； $K = 1, 2, \dots, C$ ； $C$  为泛化参数，也称感受野；

$R_j$  为第  $j$  维输入的量化级数；函数  $\text{floor}(x)$  取不超过  $x$  的最大整数。

### 实际映射

实际映射是从空间  $A$  到空间  $A'$  的映射过程。如果输入矢量的维数较少，则空间  $A$  的虚地址可与空间  $A'$  的物理地址一一对应，如果输入矢量的维数较大，则空间  $A$  可能很大。由于绝大多数学习问题并不包括所有输入空间中的状态，故可采用散列编码的映射方法（即 Hash-coding 方法），将空间  $A$  映射到一个较小的空间  $A'$  中，对应的物理单元数仍为  $C$ 。

### CMAC 输出

CMAC 的输出算法为

$$y(S) = \sum_{l=1}^C W_l \quad (5-22)$$

式中： $y(S)$  是 CMAC 的输出； $W_l$  是空间  $A'$  中被激活地址所对应的权值。

### 学习算法

权值调整采用常规 CMAC 学习算法：

$$W_l(t) = W_l(t-1) + \beta \frac{(y_d(t) - y(S(t)))}{C} \quad (5-23)$$

式中： $W_l(t)$  为  $t$  时刻空间  $A'$  中被激活地址所对应的权值； $\beta$  ( $0 < \beta < 1$ ) 为学习系数。

## 3 电液负载模拟器数学模型

考虑外部干扰对系统的影响时，某飞行器舵机电液负载模拟器与操舵系统联动的动力机构方块图如图 5-11： $I_p$  为舵机系统伺服阀输入电流； $I_F$  为加载系统伺服阀输入电流； $A_p$  为舵机液压缸活塞有效面积； $A_F$  为加载液压缸活塞有效面积； $W_{svp}(s)$  为舵机伺服阀传递函数； $W_{svF}(s)$  为

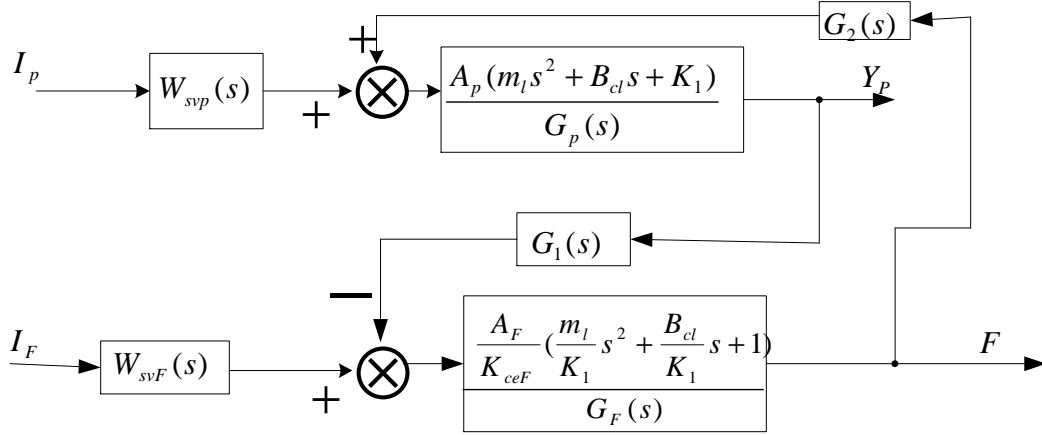


图5 - 11 电液负载模拟器与操舵系统联动的动力机构方块图

加载伺服阀传递函数； $B_{cl}$  为惯性负载的粘性阻尼系数； $K_{ceF}$  为加载系统总的流量——压力系数；

$K_1$  为等效舵机联接弹簧刚度； $m_l$  为惯性负载质量； $Y_p$  为舵机液压缸活塞位移； $F$  为加载系统液压缸输出力。此处，舵机伺服阀和加载伺服阀传递函数均按振荡环节考虑，其固有频率为 760rad/s，阻尼比为 0.7。

$$G_1(s) = \frac{G_{dF}(s)}{\frac{A_F}{K_{ceF}}(\frac{m_l}{K_1}s^2 + \frac{B_{cl}}{K_1}s + 1)} \quad (5 - 24)$$

$$G_2(s) = \frac{K_1(\frac{V_{tp}}{4\beta_e}s + K_{cep})}{A_p(m_l s^2 + B_{cl}s + K_1)} \quad (5 - 25)$$

$$G_p(s) = s\{\frac{m_l m_p V_{tp}}{4\beta_e}s^4 + [\frac{(m_l B_{cp} + m_p B_{cl})V_{tp}}{4\beta_e} + m_l m_p K_{cep}]s^3 + [\frac{(m_l + m_p)K_1 V_{tp} + B_{cl} B_{cp} V_{tp}}{4\beta_e} + (m_l B_{cp} + m_p B_{cl})K_{cep} + A_p^2 m_l]s^2 + [\frac{(B_{cl} + B_{cp})K_1 V_{tp}}{4\beta_e} + (m_l + m_p)K_1 V_{tp} + B_{cl} B_{cp} K_{cep} + A_p^2 B_{cl}]s + (B_{cl} + B_{cp})K_1 K_{cep} + A_p^2 K_1\}$$

(5 - 26)

$$G_F(s) = \frac{m_l m_F V_{tF}}{4\beta_e K_F K_1 K_{ceF}}s^5 + [\frac{(m_l B_{cF} + m_F B_{cl})V_{tF}}{4\beta_e K_F K_1 K_{ceF}} + \frac{m_l m_F}{K_F K_1}]s^4 + [\frac{(m_l + m_F)K_F V_{tF} + (m_F K_1 + B_{cl} B_{cF})V_{tF}}{4\beta_e K_F K_1 K_{ceF}} + \frac{(m_l B_{cF} + m_F B_{cl})K_{ceF} + A_F^2 m_l}{K_F K_1 K_{ceF}}]s^3 + [\frac{(B_{cl} + B_{cF})K_F V_{tF} + B_{cF} K_1 V_{tF}}{4\beta_e K_F K_1 K_{ceF}} + \frac{(m_l + m_F)K_F K_{ceF} + (m_F K_1 + B_{cl} B_{cF})K_{ceF} + A_F^2 B_{cl}}{K_F K_1 K_{ceF}}]s^2 + [\frac{V_{tF}}{4\beta_e K_{ceF}} + \frac{(B_{cl} + B_{cF})K_F K_{ceF} + B_{cF} K_1 K_{ceF} A_F^2 (K_F + K_1)}{K_F K_1 K_{ceF}}]s + 1$$

(5 - 27)

$$G_{dF}(s) = \frac{A_F^2 + K_{ceF}B_{cF}}{K_{ceF}} \left[ \frac{m_F V_{tF}}{4\beta_e (A_F^2 + K_{ceF}B_{cF})} s^2 + \left( \frac{B_{cF} V_{tF}}{4\beta_e (A_F^2 + K_{ceF}B_{cF})} + \frac{m_F K_{cF}}{A_F^2 + K_{ceF}B_{cF}} \right) s + 1 \right] s \quad (5 - 28)$$

上述各式中， $m_p$  为舵机液压缸活塞质量； $m_F$  为加载液压缸活塞质量； $K_F$  为传感器刚度； $B_{cp}$  为舵机液压缸活塞的粘性阻尼系数； $B_{cF}$  为加载液压缸活塞的粘性阻尼系数； $K_{cep}$  为舵机系统总的流量——压力系数； $V_{ip}$  为舵机液压缸两个油腔的总容积； $V_{tF}$  为加载液压缸两个油腔的总容积； $\beta_e$  为有效体积弹性数。

由上述分析可得加载系统动力机构传递函数为

$$F = \frac{\frac{A_F}{K_{ceF}} \left( \frac{m_l}{K_l} s^2 + \frac{B_l}{K_l} s + 1 \right) W_{svF}(s) I_F - G_{dF}(s) Y_p}{G_F(s)} \quad (5 - 29)$$

系统主要参数值如表5 - 7所示。

表5 - 7 系统主要参数值

参数	数值	单位	参数	数值	单位
$m_l$	14.5	kg	$m_p$	2.5	kg
$m_F$	2	kg	$B_{cl}$	100	N/(m/s)
$B_{cF}$	500	N/(m/s)	$B_{cp}$	500	N/(m/s)
$K_l$	$1.37 \times 10^7$	N/m	$K_F$	$5 \times 10^8$	N/m
$A_p$	$6.41 \times 10^{-4}$	$m^2$	$A_F$	$6.41 \times 10^{-4}$	$m^2$
$K_{cep}$	$4 \times 10^{-13}$	$m^5/(N.s)$	$K_{ceF}$	$4 \times 10^{-13}$	$m^5/(N.s)$
$V_{ip}$	$5.3 \times 10^{-5}$	$m^3$	$V_{tF}$	$5.3 \times 10^{-5}$	$m^3$
$\beta_e$	690	MPa			

#### 4. 基于CMAC的控制器结构

由于要求加载系统在随着舵机系统快速运动情况下高精度地跟踪各种指令加载函数，而干扰源的强度非常大，且加载系统的时变特性和非线性使得  $G_{dF}(s)$  是变化的，这就说明单独采用对消的方法来消除它比较困难。通常，电液负载模拟器采用结构不变性原理的方法抑制多余力，即通过确定图5-12的补偿环节  $G_e(s)$  来实现对消  $G_{dF}(s) Y_p$  项的影响。由于  $G_l(s)$  中的二阶微分环

节和振荡环节的频率很高，加载伺服阀的频带也较宽，在设计补偿环节时可以忽略，因此可得补偿环节  $G_e(s)$  的表达式为  $G_e(s) = \frac{U_e}{U_{pf}} = \frac{K_e s}{W_e(s)}$  (5-30)

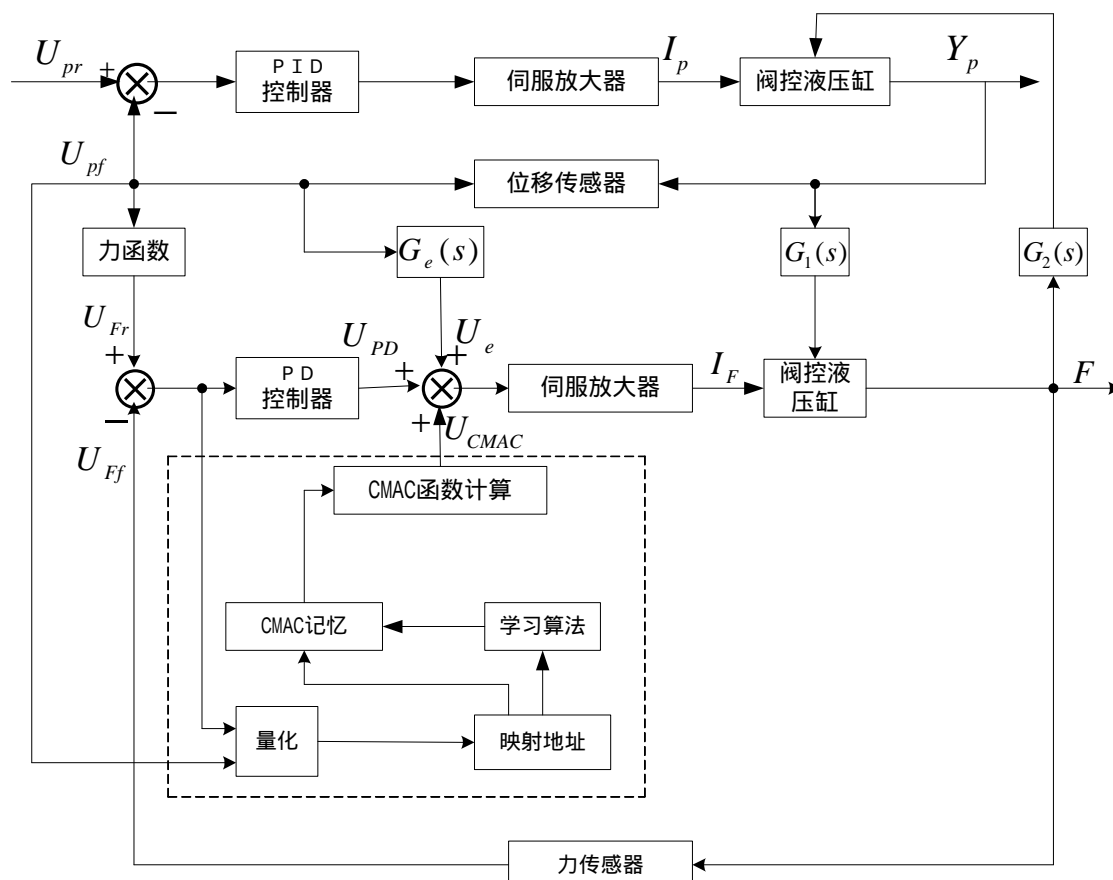


图5-12 基于CMAC的电液负载模拟器控制系统

式中： $U_{pf}$  为舵机位移传感器输出电压； $U_e$  为补偿环节输出电压； $K_e$  为补偿环节增益； $W_e(s)$  是考虑物理可实现因素而串联的一个滤波器。

基于CMAC的电液负载模拟器控制结构，如图5-12所示。其原理是：初始状态时，CMAC的所有权值为零；控制时，将加载系统的偏差信号和舵机位移信号量化后，作为地址输入到CMAC，激活空间  $A'$  中与之对应的  $C$  个地址单元，并将这  $C$  个地址中的权值相加，得到CMAC的输出。即

$$U_{CMAC} = \sum_{l=1}^C W_l \quad (5-31)$$

然后与PD控制器的输出  $U_{PD}$  和补偿环节的输出  $U_e$  相加，得到电液负载模拟器控制器的总输出。

即：

$$U = U_{CMAC} + U_{PD} + U_e \quad (5-32)$$

被激活单元的权值调整规则为

$$W_i(k) = W_i(k-1) + \beta \frac{U_{PD}}{C} \quad (5-33)$$

实际应用中，应设计一个允许误差极限  $\delta$ ，当  $|U_{PD}| \leq \delta$  时，权值不变，否则按式(5-33)修正权值。

## 5 仿真研究

为了验证所提出基于CMAC的自学习控制策略的有效性，需要进行对比仿真分析。仿真中，加载系统PID控制参数为： $k_p=1$ ， $k_i=0.005$ ， $k_d=0.001$ 。CMAC控制器参数为  $k_p=1$ ， $k_d=0.001$ ， $\beta=0.005$ ， $C=100$ ，加载系统的偏差信号和舵机位移信号量化级数分别为2000和1000，物理存储空间  $A'=10000$ ，加载系统力函数为  $F_r = 2.5 \times 10^5 y_p$ 。

被测舵机输入信号  $y_r=0.008\sin(\pi t)$  时的仿真结果比较发现，仅采用PID控制时，加载误差比较大；采用PID与结构不变性原理混合控制或采用基于CMAC的控制器，在舵机运动频率较低时，都可以有效地抑制多余力，系统加载精度较高。

被测舵机输入信号  $y_r=0.002\sin(20\pi t)$  时的仿真结果可知，仅采用PID控制时，加载误差非常大；采用PID与结构不变性原理混合控制，在舵机运动频率较高时，加载误差较大；采用基于CMAC的控制器，舵机运动频率较高时，由于干扰强度非常大，控制初始阶段系统误差也较大。这是因为CMAC控制器初始权值为零，控制起始阶段，PID与结构不变性原理混合控制器起主要作用。但经过13个周期的学习后，系统控制精度已较高。

综合仿真结果可知：电液负载模拟器仅采用PID控制时，加载误差很大；采用PID与结构不变性原理混合控制时，舵机运动频率较低时，加载精度很高，但舵机运动频率较高时，加载误差变得很大；采用基于CMAC的控制器，加载系统可在较宽的频率范围内满足对舵机系统的加载精度要求，加载系统的动态性能得到明显改善。

## 本章小结

本章讨论 CMAC 神经网络基本结构及原理的基础上，从网络输出计算阶段和网络权值调整阶段两个方面，重点介绍了模糊 CMAC 神经网络，基于信度分配的 CMAC 神经网络，基于平衡学习的 CMAC 神经网络及基于信度分配的模糊 CMAC 神经网络的结构、工作原理及权值调整学习算法，并给出了具体的仿真算例。最后讨论了 CMAC 神经网络在电液负载模拟器自学习控制中的应用。

## 参考文献：

- [1] Albus J.S., A new approach to manipulator control: The cerebellar model articulation controller(CMAC), ASME J. Dynamic Systems, Measurement, Control, pp.220-227, 1975.
- [2] Albus J.S., Data storage in cerebellar model articulation controller(CMAC), ASME J. Dynamic Systems, Measurement, Control, pp.228-233, 1975.
- [3] Wong Y.F. and Sideris A., Learning convergence in cerebellar model articulation controller, IEEE Trans. Neural Networks, vol.3, pp. 115-121, Jan. 1992.
- [4] Lin C.S. and Chiang C.T., Learning convergence of CMAC technique, IEEE Trans. Neural Networks, vol.8, no.6, pp. 1281-1292, 1997.

- [5] 韩力群, 人工神经网络的理论、设计及应用, 北京, 化学工业出版社, 2002.1。
- [6] Nie J. and Linkens D.A., FCMAC: A fuzzified cerebellar model articulation controller with self-organizing capacity, Automatica, vol.30, no.4, pp. 655-664, 1994.
- [7] Geng Z.J. and Mccullough C.L., Missile control using fuzzy cerebellar model arithmetic computer neural networks, J. Guid., Control, Dyn., vol. 20, no.3, pp. 557-565, 1997.
- [8] 王源, 胡寿松, 齐俊伟, 自组织模糊 CMAC 神经网络及其非线性系统辨识, 航空学报, 2001, 22 ( 6 ): 556-558。
- [9] 胡寿松, 王源, 基于自组织模糊 CMAC 网络的非线性系统鲁棒自适应跟踪控制, 自动化学报, 2002, 28 ( 6 ): 984-989。
- [10] Hana-Ming Lee., Chin-Ming Chen., A self-organizing HCMAC neural network classifier, IEEE Trans. On Neural Networks, 2003, 14(1):15-27.
- [11] Shun-Feng S., Ted T., and Hung T.H., Credit assigned CMAC and its application to online learning robust controllers, IEEE Trans. On Systems, Man, and Cybernetics----Part B: Cybernetics, vol. 33, no.2, pp. 202-213, 2003.
- [12] 朱大奇, 孔敏, 张伟, 基于平衡学习的 CMAC 神经网络动态非线性系统辨识算法, 控制与决策, 2004, 19 ( 12 ): 556-558。
- [13] 朱大奇, 孔敏, 张伟, 基于信度分配的模糊 CMAC 神经网络快速学习算法, 控制理论与应用, 审稿中。
- [14] 叶正茂, 李洪人, 王经普, 基于 CMAC 的电液负载模拟器自学习控制, 控制与决策, 2003, 18 ( 3 ): 343 - 347。
- [15] 李运华, 抑制电液力矩控制系统的多余力矩的新方法. 机械工程学报, 2000, 36(3):4-7.
- [16] 蒋志明, 林廷圻, 黄先祥. 基于 CMAC 的带有未知负载干扰电液位置伺服系统的自学习控制. 控制与决策, 2000, 15(3) :368-374.

### 思考题

1. CMAC 神经网络有哪些特点?
2. 请用自己的语言简要介绍 CMAC 神经网络的工作过程。
3. 请简要介绍基于信度分配的 CMAC 神经网络权值调整学习算法的理由。
4. 假设有如下的非线性函数:

$$y(x_1, x_2) = \sin x_1 \cos x_2 \exp(-x_1) \quad -\pi \leq x_1, x_2 \leq \pi$$

试比较 30 周期内常规 CMAC, 基于信度分配 CMAC 神经网络的学习效果。

5. 设有如下的非线性函数:

$$y(x_1, x_2) = \sin^2 x_1 \cos x_2 \quad -\pi \leq x_1, x_2 \leq \pi$$

试利用平衡学习算法, 求解最佳平衡学习参数。

## 第 6 章 RBF 径向基函数神经网络

对局部逼近神经网络，除第 5 章介绍的 CMAC 神经网络外，常用的还有径向基函数 RBF 网络和 B 样条网络等，本章介绍径向基函数 RBF 神经网络原理及应用。径向基函数 (RBF, Radial Basis Function) 神经网络，是由 J. Moody 和 C. Darken 于 20 世纪 80 年代末提出的一种神经网络<sup>[1]</sup>，径向基函数方法在某种程度上利用了多维空间中传统的严格插值法的研究成果。在神经网络的背景下，隐藏单元提供一个“函数”集，该函数集在输入模式向量扩展至隐层空间时为其构建了一个任意的“基”；这个函数集中的函数就被称为径向基函数<sup>[2]</sup>。径向基函数首先是在实多变量插值问题的解中引入的<sup>[3-4]</sup>。径向基函数是目前数值分析研究中的一个主要领域之一。

### 6.1 RBF 结构及工作原理<sup>[5]</sup>

#### 1. 径向基函数神经网络结构

最基本的径向基函数 (RBF) 神经网络的构成包括三层，其中每一层都有着完全不同的作用。输入层由一些感知单元组成，它们将网络与外界环境连接起来；第二层是网络中仅有的一个隐层，它的作用是从输入空间到隐层空间之间进行非线性变换，在大多数情况下，隐层空间有较高的维数；输出层是线性的，它为作用于输入层的激活模式提供响应。

基本的径向基函数 RBF 网络是具有单隐层的三层前馈网络，结构如图 6-1 所示。由于它模拟了人脑中局部调整、相互覆盖接受域 (或称感受域, Receptive Field) 的神经网络结构，因此，**RBF 网络是一种局部逼近网络，现已证明它能以任意精度逼近任一连续函数。**

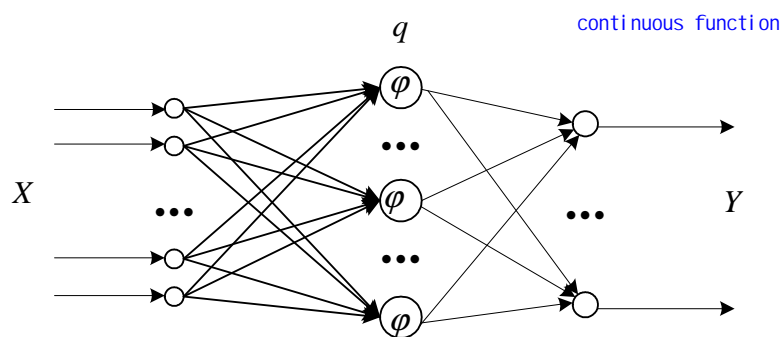


图 6-1 RBF 网络结构

#### 2. RBF 网络输出计算

设输入  $n$  维向量  $X$ ，输出  $m$  维向量  $Y$ ，输入/输出样本对长度为  $L$ ，则 RBF 网络隐层第  $i$  个节点的输出为：

$$q_i = R(\|X - c_i\|) \quad (6-1)$$

式 (6-1) 中， $X$ ： $n$  维输入向量； $c_i$ ：第  $i$  个隐节点的中心， $i=1, 2, \dots, h$ 。

$\|\bullet\|$ ：通常为欧氏范数； $R(\bullet)$  为 RBF 函数，具有局部感受的特性。它有多种形式，体现了 RBF 网络的非线性映射能力。

网络输出层第  $k$  个节点的输出为隐节点输出的线性组合：

$$y_k = \sum_i w_{ki} q_i - \theta_k \quad (6-2)$$



式 (6-2) 中,  $w_{ki} : q_i - y_k$  的连接权;  $\theta_k$  : 第  $k$  个输出节点的阈值。

## 6.2 RBF 学习算法

### 6.2.1 RBF 网络的常规学习算法

设有  $p$  组输入/输出样本  $x_p/d_p$ ,  $p=1, 2, \dots, L$ , 定义目标函数:

$$J = \frac{1}{2} \sum_p \|d_p - y_p\|^2 = \frac{1}{2} \sum_p \sum_k (d_{kp} - y_{kp})^2 \quad (6-3)$$

学习的目的是使  $J \rightarrow \varepsilon$ ; 式 (6-3) 中,  $y_p$  是在  $x_p$  输入下网络的输出向量。RBF 网络的学习算法

一般包括两个不同的阶段:

- \*隐层径向基函数的中心的确定阶段。常见方法有随机选取固定中心法; 中心的自组织选择法等。
- \*径向基函数权值学习调整阶段。常见方法有中心的监督选择法; 正则化严格插值法等。此处介绍中心的自组织选择法和中心的监督选择法两种算法, 其他见文献[6]。

#### 1. 中心的自组织选择法

它是一种无导师学习也称为非监督学习, 是对所有样本的输入进行聚类, 求得各隐层节点的 RBF 的中心  $c_i$ 。这里介绍常用  $k$ -均值聚类算法<sup>[7]</sup>, 算法步骤如下:

初始化: 给定各隐节点的初始中心  $c_i(0)$ 。

相似匹配: 计算距离 (欧氏空间) 并求出最小距离的节点:

$$d_i(t) = \|x(t) - c_i(t-1)\|, \quad 1 \leq i \leq h; \quad d_{\min}(t) = \min d_i(t) = d_r(t) \quad (6-4)$$

调整中心:

$$c_i(t) = c_i(t-1), \quad 1 \leq i \leq h, \quad i \neq r;$$

$$c_r(t) = c_r(t-1) + \beta(x(t) - c_r(t-1)), \quad i = r \quad (6-5)$$

式 (6-5) 中,  $\beta$  是学习速率,  $0 < \beta < 1$ 。

继续: 将  $t$  值加 1, 回到第二步, 重复上述过程, 直到中心  $c_r$  的改变量很小时为止。

#### 2. 中心的监督选择法

它是一种有导师学习也称为监督学习算法。径向基函数的中心以及网络的所有其他参数都将经历一个监督学习的过程。换句话说, RBF 网络将采用最一般的方式, 这个方法的自然后选是采用误差修正学习过程, 它可以很方便地使用梯度下降法。当  $c_i$  确定后, 训练由隐层至输出层之间权值, 由式 (6-2) 可知, 它是一个线性方程组, 则求权值就成为线性优化问题, 如 LMS 算法、最小二乘递推法等求得。

LMS 算法:

LMS 算法, 对于 RBF 网络, 权值调整算法为:

$$w_{ki}(t+1) = w_{ki}(t) + \alpha \frac{e_k(t)q_{ip}}{\|q_p\|^2} \quad (6-6)$$

式(6-6)中,  $\alpha$  是常值,  $0 < \alpha < 2$ 。当  $J(t) \leq \varepsilon$  时, 算法结束。

最小二乘递推法 (RLS)

为简单起见, 讨论单输出的情况。

定义目标函数:

$$J(t) = \sum_{p=1}^L E_p(t) = \frac{1}{2} \sum_{p=1}^L \Lambda(p) [d_p - y_p(t)]^2 \quad (6-7)$$

式(6-7)中,  $\Lambda(p)$  是加权因子。若第  $p$  个样本比第  $p-k$  ( $p > k, K > 1$ ) 个可靠, 则加权因子要大, 可取:

$$\Lambda(p) = \lambda^{L-p}, \quad 0 < \lambda < 1, \quad p = 1, 2, \dots, L \quad (6-8)$$

$L$  是样本长度。

使  $J$  值最小的  $W$  即为所求, 因此, 由

$$\frac{\partial J(t)}{\partial W} = 0 \quad (6-9)$$

可得最小二乘递推算法 (RLS):

$$\begin{aligned} W_p(t) &= W_p(t-1) + K(t)[d_p - q_p^T(t)W_p(t-1)] \\ K(t) &= P(t-1)q_p(t) \left[ q_p^T(t)P(t-1)q_p(t) + \frac{1}{\Lambda(p)} \right]^{-1} \\ P(t) &= [I - K(t)q_p^T(t)]P(t-1) \end{aligned} \quad (6-10)$$

式中,  $q_p(t) = [q_{1p}(t), q_{2p}(t), \dots, q_{hp}(t)]^T$ ,  $h$  是隐节点数。

### 6.2.2 有关的几个问题

#### 1. 径向基函数 RBF 特性分析

RBF 网与 BP 网的主要不同点是在非线性映射上采用了不同的作用函数, 分别为径向基函数和 S 型函数。前者的作用函数是局部的, 如高斯 RBF; 后者的作用函数是全局的。

已证明 RBF 网络具有唯一最佳逼近的特性, 且无局部极小。

求 RBF 网络隐节点的中心  $c_i$  和标准参数  $\sigma^2$  是个困难的问题。

径向基函数, 即径向对称函数有多种。对于一组样本, 如何选择适合的径向基函数, 如何确定隐节点数, 以使网络学习达到要求的精度, 是还没有解决的问题。当前, 用计算机选择、设计, 再检验是一种通用的手段。

RBF 网络用于非线性系统辨识与控制, 虽具有唯一最佳逼近的特性以及无局部极小的优点, 但隐节点的中心难求, 这是该网络难以广泛应用的原因。

#### 2. 径向基函数 RBF 与多层感知器 MLP 的比较

径向基函数网络 RBF 与多层感知器 MLP 都是非线性的层状前馈网络。它们都是通用逼近器, 所以, 对于一个特定的 MLP 总存在一个 RBF 网络能够精确的模仿它, 反之亦然。然而, 这两种网络在

几个重要方面存在着不同之处。

一个 RBF 网络只具有一个隐藏层，而一个 MLP 可以有一个或多个隐藏层。

一个 MLP 位于隐层或输出层的计算节点，其神经元模型是相同的；而 RBF 网络隐层中计算节点与网络输出层计算节点是相当不同的且作用也不一样。

RBF 网络隐层是非线性的，而输出层是线性的；但 MLP 作为模式分类器，其隐层和输出层都是非线性的。当用 MLP 解决非线性回归问题时，线性输出层通常是好的选择。

RBF 网络每一隐层单元的激活函数的自变量都要计算输入向量和该单元的中心之间的范数；而 MLP 隐层单元的激活函数只要计算输入向量和该隐层单元相关的权值向量的内积。

MLP 建立一个输入 - 输出映射的全局逼近；而 RBF 网络是以指数下降的局部非线性（如 Gauss 函数）来局部逼近一个输入 - 输出映射。这一结果意味着当逼近一个输入 - 输出映射时，在相同逼近精度要求下，MLP 所需要的时间要比 RBF 多。

### 6.3 改进的 RBF 学习算法

针对上节介绍的 RBF 网络诸多局限性，提出了不少改进的学习算法。此处介绍一种基于免疫算法的改进 RBF 学习算法<sup>[8]</sup>。

#### 6.3.1 RBF 神经网络分析

RBF 神经网络是由输入层、隐层和输出层神经元构成的前向型网络，其基本思想是用径向基函数作为隐层神经元的基，构成隐层空间，实现输入矢量到输出矢量的映射变换，其结构如图 6-2 所示。

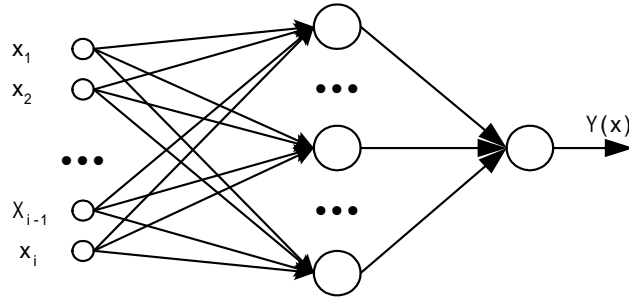


图 6-2 RBF 神经网络结构

设输入神经元数为  $I$ ，隐层神经元数为  $H$ ，输出层神经元数为  $O$ 。其输入矢量为：

$$X = [x_1, x_2, \dots, x_i, \dots, x_I]^T \quad (1 \leq i \leq I); \text{ 隐层输出矢量为: } Z = [z_1, z_2, \dots, z_j, \dots, z_H]^T \quad (1 \leq j \leq H);$$

输出矢量为：

$$Y = [y_1, y_2, \dots, y_k, \dots, y_O] \quad (1 \leq k \leq O)。 \text{ 假设隐层变化函数为高斯函数，则:}$$

$$z_j = \exp\left[-\sum_{i=1}^I \frac{(x_i - c_{ji})^2}{2\sigma_j^2}\right] \quad 1 \leq j \leq H \quad (6-11)$$

式中  $C_j = [C_{j1}, C_{j2}, \dots, C_{jI}]$  为隐层第  $j$  个神经元的中心矢量； $\sigma_j^2$  为隐层第  $j$  个神经元的宽度因子。

则网络输出为：

$$y_k = \sum_{j=1}^H w_{kj} z_j + \theta_k = W_k \cdot Z + \theta_k \quad 1 \leq k \leq O \quad (6-12)$$

式中  $W_k = [w_{k1}, w_{k2}, \dots, w_{kH}]$  为隐层神经元至输出层第  $k$  个神经元权值矢量， $\theta_k$  为初始阈值。

建立 RBF 神经网络模型的关键在于三个方面：一是 RBF 神经网络隐层数据中心个数的确定；二是 RBF 神经网络隐层数据中心  $C_j$  位置的确定；三是输出权值  $w_{kj}$  的学习调整。此处利用免疫算法来确定一、二两项；利用梯度下降法确定权值  $w_{kj}$ 。

### 6.3.2 基于免疫算法的 RBF 神经网络设计

#### 1. RBF 隐层数据中心

免疫系统是指具有免疫功能的组织机构。所谓免疫功能是机体对自身或外来的抗原性物质（细菌、病毒等）进行识别，通过产生免疫应答（如产生抗体）将其清除，以维持机体内环境相对稳定的一种生理反应。免疫系统具有记忆、学习和自组织调节功能，利用这些特性可以进行数据聚类分析<sup>[7]</sup>，此处文献[7]的基础上，将其应用于 RBF 神经网络隐层数据中心的选择，不仅计算效力高，而且无需预先指定 RBF 隐层数据中心的数量。

在免疫系统中，抗原-抗体相互作用的强度用它们的亲和力来表示，而抗体与抗体之间的相互作用由它们的相似度来描述：

定义 1 设第  $i$  个输入数据  $x_i$  与第  $j$  个数据中心  $C_j$  之间的亲和力为  $a_{ij}$ ：

$$a_{ij} = \frac{1}{1 + \|x_i - c_j\|} \quad i = 1, \dots, N, \quad j = 1, \dots, H \quad (6-13)$$

式中  $\|x_i - c_j\|$  为  $x_i$  与  $c_j$  之间的欧氏距离。当  $x_i = c_j$  时，它们之间的亲和力  $a_{ij} = 1$  为最大。

定义 2 设第  $i$  个数据中心  $C_i$  与第  $j$  个数据中心  $C_j$  之间的相似度为  $s_{ij}$ ：

$$s_{ij} = \frac{1}{1 + \|c_i - c_j\|} \quad i = 1, \dots, H, \quad j = 1, \dots, H \quad (6-14)$$

式中  $\|c_i - c_j\|$  为  $c_i$  与  $c_j$  之间的欧氏距离。当  $c_i = c_j$  时，它们之间的相似度  $s_{ij} = 1$  为最大。

假设  $N$  个输入数据  $X = \{x_1, x_2, \dots, x_N\}$ ，其中  $x_i = [x_{i1}, x_{i2}, \dots, x_{iH}]^T$ ， $i = 1, 2, \dots, N$ ，确定 RBF 的数据中心就是要寻找一个新的数据集  $C = \{c_1, c_2, \dots, c_H\}$ ， $c_j = [c_{1j}, c_{2j}, \dots, c_{Hj}]$ ， $j = 1, 2, \dots, H$ ， $H < N$ 。确定隐层数据中心的免疫算法的具体步骤如下：

(1) 对每一个输入数据  $x_i$ ：

以一个确定的数量进行随机初始化  $C$  数据集；

计算  $C$  中所有 RBF 数据中心与  $x_i$  的亲和力  $a_{ij}$ ；

选择亲和力最大的数据中心并复制  $N'$  个，产生一个 RBF 数据中心的复制集  $L$ ，再应用式 (6-15) 对  $P$  个相同的数据中心进行变异处理，形成变异集  $L'$ ， $\beta_k$  为变异率。这一过程实际上是在有最大亲和力的数据中心附近进行搜寻，以得到更有亲和力的数据中心；

$$c'_k = c_k - \beta_k (c_k - x_i); \quad k = 1, 2, \dots, P \quad (6-15)$$

重新计算  $L'$  中  $N'$  个经过变异的数据中心与输入数据  $x_i$  的亲合力，生成数据中心记忆集  $m$ ；

去除  $m$  中那些亲合力低于阈值  $\rho_a$  的中心，产生一个经过压缩的数据中心记忆集  $m'$ ；

计算  $m'$  中各记忆数据中心之间的相似度  $s_{ij}$  并除去那些相似度大于阈值  $\rho_s$  的数据中心，这一过程体现了免疫系统中的克隆抑制；

判断输入数据是否全部学习完（ $i = I$  否？），未完，转 继续；输入数据全部学习一遍后，将全部输入数据产生的数据中心记忆集  $m'$  合并为  $M$ ，转（2）；

- (2) 计算  $M$  中各个数据中心之间的相似度  $s_{ij}$ ，除去那些相似度大于阈值  $\rho_s$  的数据中心，这一处理过程体现了免疫系统中的免疫抑制；

- (3) 用  $r$  个新的数据中心替换  $C$  中亲合力较低的数据中心，这些新增加的数据中心可以随机选取，这一过程体现了免疫系统的自组织特性；判断数据中心  $C(t) = C(t+1)$  成立否？如成立，则整个确定中心的递推过程结束， $M$  即为所求的 RBF 数据中心集；如不成立，则转（4）。

- (4) 判断以上递推步数是否达到预定的数据  $Q$ ，如未达到，转 继续；如达到  $Q$ ，则整个确定中心的递推过程结束， $M$  即为所求的 RBF 数据中心集。

## 2. RBF 权值调整算法

选择方差为误差函数，取  $\rho$  为学习率，采用梯度下降法确定权值  $w_{jk}$ ：

$$E(t) = \frac{1}{2} \sum_{k=1}^O [y_{kd}(t) - y_k(t)]^2 \quad (6-16)$$

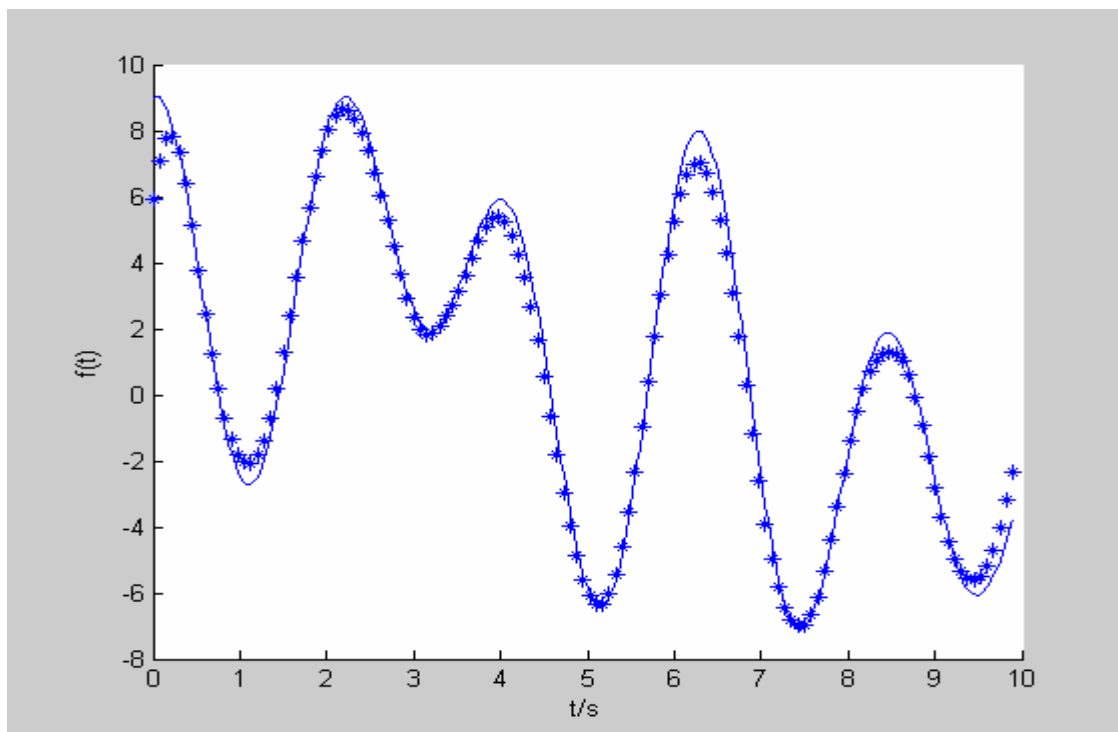
$$W(t+1) = W(t) - \rho \frac{\partial E(t)}{\partial W(t)} \quad (6-17)$$

$$\text{而 } \frac{\partial E(t)}{\partial w_{jk}(t)} = \frac{\partial E(t)}{\partial y_k(t)} \frac{\partial y_k(t)}{\partial w_{jk}(t)} = [y_{dk}(t) - y_k(t)] z_j(t) \quad j=1,2,\dots,H; k=1,2,\dots,O \quad (6-18)$$

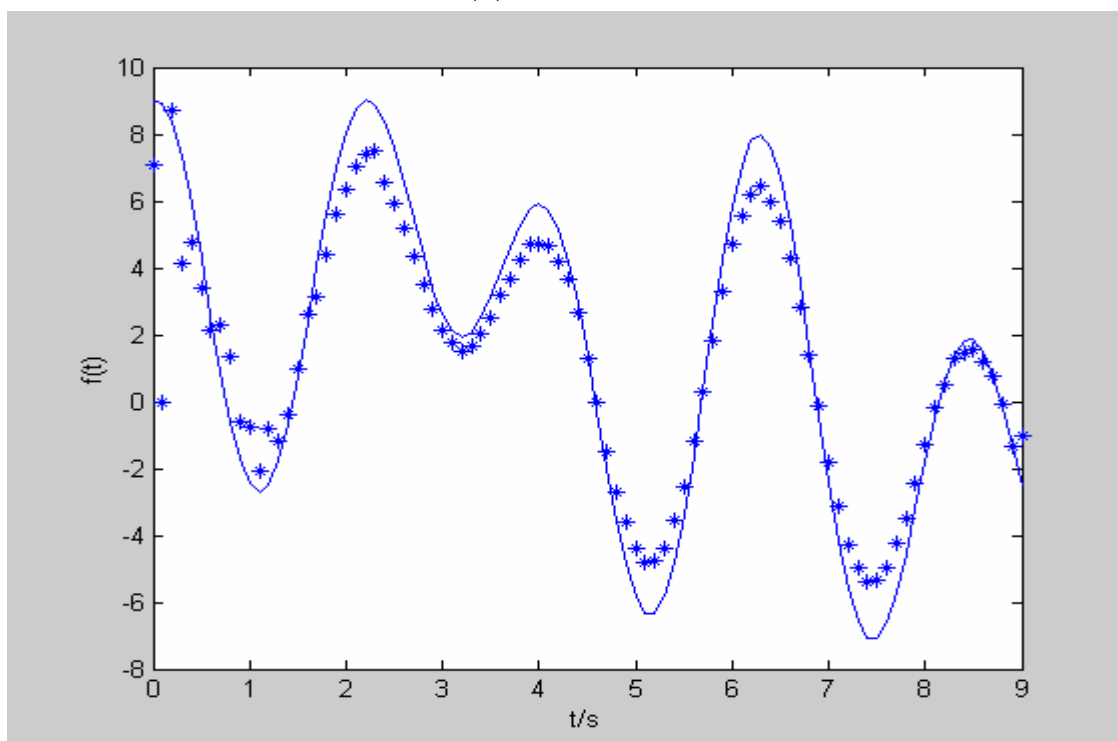
$$\text{权值迭代公式为：} w_{jk}(t+1) = w_{jk}(t) + \rho [y_{dk}(t) - y_k(t)] z_j(t) \quad (6-19)$$

## 3. 仿真算例及结果分析

仿真中选取非线性函数： $f(t) = \exp(-2t) + 2\sin t + 3\cos(2t) + 4\sin(0.5t) + 5\cos(3t)$ ，分别采用 IRBF 算法和 CMAC 算法<sup>[9]</sup>进行逼近，分析比较其逼近精度和学习速度。输入与输出单元均为 1，选取 25 个样本，初始隐层数据中心数为 30。变异率  $\beta_k = \exp(1-k)$ ，亲合力阈值  $\rho_a = 0.8$ ，相似度阈值  $\rho_s = 0.95$ ，学习率  $\rho = 0.01$ 。图 6-3(a) 和图 6-3(b) 分别为 IRBF 与 CMAC 学习 5 次后的逼近曲线；图 6-4(a) 和图 6-4(b) 分别为 IRBF 与 CMAC 学习 10 次后的逼近曲线。从图中可见在相同的训练步数下，IRBF 的逼近精度明显高于 CMAC。

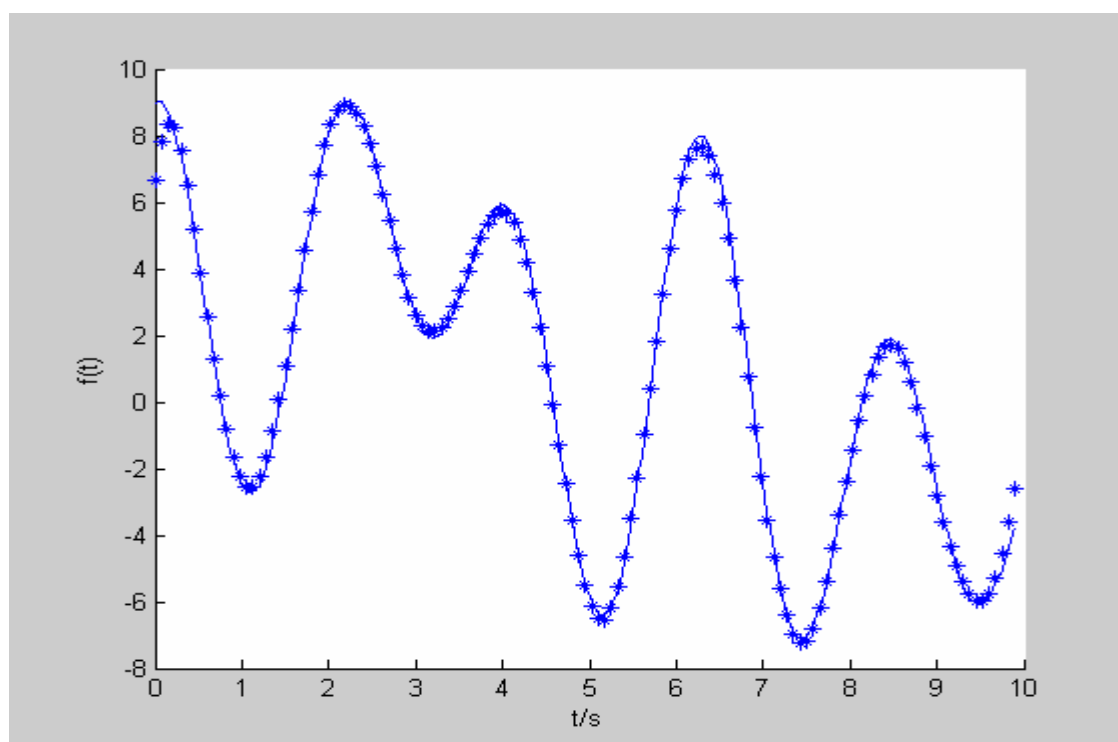


(a) IRBF 学习 5 次

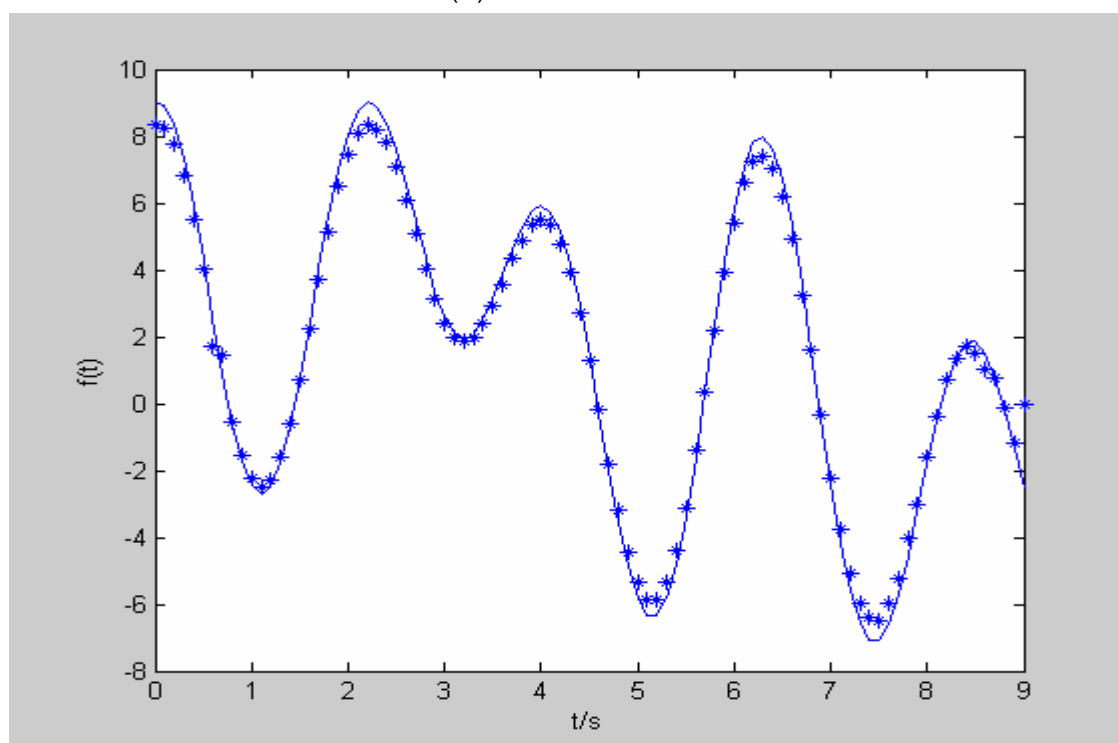


(b) CMAC 学习 5 次

图 6 - 3 IRBF 网络与 CMAC 逼近结果比较



(a) IRBF 学习 10 次



(b)CMAC 学习 10 次

图 6-4 IRBF 网络与 CMAC 逼近结果比较

表 6 - 1 为 IRBF 与 CMAC 学习结果的误差分析,其误差计算按式(6 - 16)进行,从中可见 IRBF 在学习 5 次、10 次、20 次及 30 次时,其逼近误差均小于 CMAC,特别是在初始训练阶段尤其明显。也就是说,基于免疫算法的 RBF 神经网络在非线性函数逼近中,不仅计算量小速度快、实时性好,而且精度高,误差小.对动态非线性系统的在线故障辨识来说,是一种实用而有效的辨识算法。

表 6-1 IRBF 与 CMAC 学习结果的误差比较

	误差			
学习步数	5	10	20	30
IRBF	0.0649	0.0181	0.0083	0.0071
CMAC	0.2108	0.0919	0.0276	0.0153

## 6.4 径向基函数神经网络 RBF 的应用

### ——利用RBF神经网络实现热工过程的在线辨识<sup>[10]</sup>

热工过程往往表现出非线性、慢时变、大迟延和不确定性，难以建立精确的数学模型，基于常规线性模型的控制系统的，在过程工况发生变化时，控制品质将会下降，甚至影响控制系统的正常运行。因此，建立热工过程的精确数学模型是提高控制系统性能的基础。近年来人们采用 BP 神经网络建立过程的非线性模型获得了满意的效果，但由于常用的多层神经网络计算量大，收敛速度慢，且容易陷入局部最小点，因而影响了它们的应用。

径向基函数神经网络具有可以逼近任意非线性映射的能力，且算法简单实用，因而成为人们的研究热点。RBF 神经网络在信号处理、系统建模、过程控制和故障诊断等领域得到了成功的应用<sup>[11]</sup>。基于 M-RAN 算法的 RBF 神经网络<sup>[12]</sup>是一种动态神经网络，其隐层单元的数量从零开始，并根据输入数据的“新颖性”来逐步增加，同时对算法的删除策略进行了改进，不仅删除那些连续对网络输出贡献较小的隐层单元，还将相似的隐层单元合并，使网络结构更加紧凑。将基于上述算法的 RBF 神经网络用于电厂热工过程非线性模型的在线辨识，仿真研究证明了这种在线建模方法的有效性，且建模过程计算量小，所得模型精度高，可以直接应用于基于模型的控制算法如模型预测控制和内模控制等。

#### 1. M-RAN 算法

##### RBF 神经网络

单输出 RBF 神经网络的结构如图 6-5 所示，它有输入层，隐含层和输出层三层。

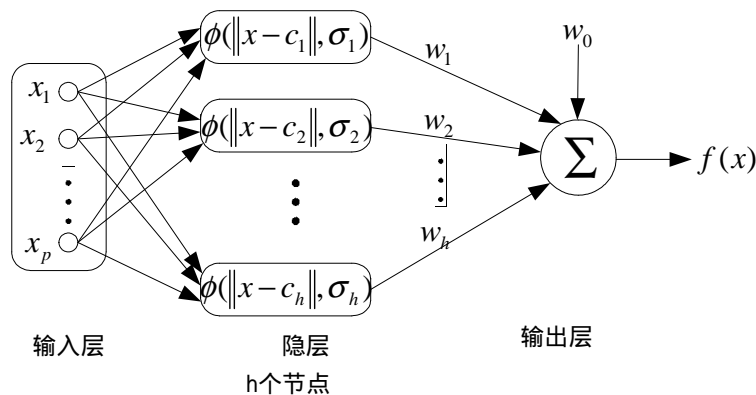


图 6-5 RBF 网络结构

设网络输入  $x$  为  $p$  维向量,  $x = [x_1, x_2, \dots, x_p]^T \in R^p$ , 隐层节点的输出为：

$$\phi_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{\sigma_i^2}\right) \quad (6-20)$$

网络输出为：

$$f(x) = w_0 + \sum_{i=1}^h w_i \phi_i(x) \quad (6-21)$$



其中  $\phi(\bullet)$  为径向基函数，一般取为高斯函数  $\phi(x) = \exp(-x^2/\sigma)$ ， $\|\bullet\|$  为欧几里德范数，

$c_i = [c_{i1}, c_{i2}, \dots, c_{ip}]^T \in R^P$  为隐层第  $i$  个径向基函数的数据中心， $\sigma_i$  为径向基函数的宽度， $w_0$  为偏

置项， $w_i$  为第  $i$  个基函数输出与输出节点的连接权值， $h$  为隐层节点的数目。

在 RBF 神经网络中，隐层 RBF 中心的数量和宽度直接影响着网络的逼近能力<sup>[11]</sup>，而且要求 RBF 中心应能覆盖整个输入空间，但如果 RBF 中心的数量过多，将使网络的计算量显著增加，而且也会导致网络泛化能力的降低，因此，建立 RBF 网络模型的关键在于正确地选择合适的 RBF 中心。目前，RBF 网络的学习算法主要有两大类：1)混合算法。首先，通过某种无监督学习方法如聚类算法，遗传算法等选择 RBF 的中心；第二阶段，采用最小二乘法计算网络的输出权值；2)将 RBF 中心的选择和输出权值的学习结合起来，典型的方法有正交最小二乘算法(OLS 算法)和剃度下降算法，在权值学习的同时确定隐层节点的数量和宽度。在混合算法中，可以通过两种方法来预先确定 RBF 中心的数量和宽度，一种是先将所有学习样本作为中心，然后采用聚类算法去除不必要的中心，另一种方法是开始时隐层节点的数量为零，然后逐步增加到预先确定的隐层节点数。总体来讲，这两类方法构造的 RBF 神经网络都属于静态结构，即一旦学习过程结束，则网络的结构即固定，如果训练数据集的代表性不强或未能包含系统的主要特征，则这样建立的 RBF 网络模型不适宜于用于在线控制。

1991 年 Platt<sup>[13]</sup>首次提出了资源分配网络 RAN(Resource Allocating Network)用于函数拟合。这是一种基于径向基函数的单隐层神经网络，在开始学习前隐层为空，没有任何节点。在学习过程中，随着输入数据的不断出现，网络根据“新颖性”条件选择某些输入数据作为隐层中心，隐层节点不断增加，在没有隐层节点增加时，网络参数采用最小二乘 LMS 算法进行调整，但网络存在收敛速度慢的缺点。Kadirkamanathan 和 Niranjan<sup>[14]</sup>1993 年从函数空间的角度对 Platt 提出的 RAN 网络进行了描述，并采用扩展卡尔曼滤波器 EKF(Extended Kalman Filter)算法代替 LMS 算法调整网络参数，改进了 RAN 的收敛速度，并可以降低网络的复杂性，其在函数逼近和时间序列预测方面的性能优于 RAN，所得到的网络称为 RAN-EKF。但在 RAN 和 RAN-EKF 网络学习过程中，一旦一个隐层单元产生，则不能被删除，因此 RAN 和 RAN-EKF 所产生的网络中，可能会有某些隐层单元，虽然在初始时活跃，但其后会对网络输出不产生任何贡献。如果在学习过程中能检测并删除这些不活跃的隐层单元，则可以实现更加紧凑的网络结构。基于这种思路，LU<sup>[12]</sup>等人将删除算法与 RAN-EKF 方法结合起来，提出了 M-RAN(Mimimal RAN)网络，并将其用于非线性系统的辨识<sup>[15]</sup>，研究表明，M-RAN 可以获得更加紧凑的网络结构，且具有自适应能力，能通过隐层神经元数量的增减和网络参数的调整跟踪系统变化的动态特性，因此，M-RAN 网络非常适合于实时在线应用。

#### M-RAN 算法

下面简要介绍 M-RAN 算法，有关算法的详细内容请参考文[12,15]。

网络开始时没有隐层单元，在学习过程中将根据下列“新颖性”条件来确定是否将某个输入  $x_n$  增加为新的隐层单元。

$$\|x_n - c_{nearest}\| > \varepsilon_n \quad (6-22)$$

$$|e_n| = |y(n) - f(x_n)| > e_{\min} \quad (6-23)$$

$$e_{rmsn} = \sqrt{\frac{\sum_{i=n-(M-1)}^n [y(i) - f(x_i)]^2}{M}} > e'_{\min} \quad (6-24)$$

其中  $C_{nearest}$  为所有隐层单元中与  $x_n$  距离最近的隐层单元的中心,  $\varepsilon_n, e_{min}, e'_{min}$  为阈值, 须合理选择, 在上面的"新颖性"条件中, 式(6-22)保证新加入的隐层单元与现有的隐层单元足够远, 式(6-23)确定现有隐层单元是否能满足输出误差的精度要求, 式(6-24)检查网络过去  $M$  个连续输出的均方差是否满足要求值, 当上述三个条件同时满足时, 则在网络中增加一个新的隐层单元, 与该隐层单元有关的参数指定如下:

$$w_{h+1} = e_n \quad (6-25)$$

$$c_{h+1} = x_n \quad (6-26)$$

$$\sigma_{h+1} = k \|x_n - c_{nearest}\| \quad (6-27)$$

其中  $k$  为重叠因子, 它决定了隐层单元的响应在输入空间的重叠程度。

当输入向量不满足增加新隐层单元的条件时, 将采用扩展卡尔曼滤波器来调整网络的参数, 同时, 该算法中增加了删除策略, 对于每一个输入值, 计算所有隐层单元的输出  $o_i^n (i = 1, \dots, h)$ , 找出隐层单元输出值绝对值中的最大值  $\|o_{\max}^n\|$ , 计算每个隐层单元输出的归一化值  $r_i^n = \|o_i^n / o_{\max}^n\|, (i = 1, \dots, h)$ , 如果对于  $M$  个连续的观测值  $r_i^n < \delta$ , 则将第  $i$  个隐层神经元删除。

### 改进的删除策略

在 M-RAN 学习过程中, 虽然增加的隐层单元可以通过最小距离的阈值来控制, 但由于网络参数的调整包括隐层单元的数据中心, 权值和宽度, 在经过一段的学习以后, 网络模型中往往会出现特别靠近的隐层单元, 而上述删除策略中只考虑了隐层单元, 则可以使网络的结构更加紧凑。这可以通过指定一个距离阈值  $s_d$  来进行控制, 当隐层单元之间的距离小于该阈值时, 则将这两个隐层单元合并。

## 2. 仿真研究

基于上述 RBF 神经网络的模型在线识别结构如图 6-6 所示, 选取 RBF 神经网络模型为下列串行并行结构<sup>[16]</sup>:

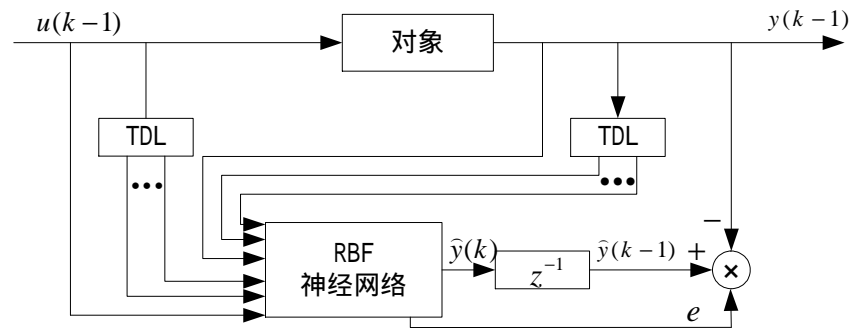


图 6-6 基于 RBF 神经网络的模型在线辨识结构

$$\hat{y}(k) = f[u(k-1), u(k-2), \dots, u(k-m), y(k-1), y(k-2), \dots, y(k-n)] \quad (6-28)$$

其中  $u(k), y(k)$  分别为系数的输入和输出,  $\hat{y}(k)$  为 RBF 神经网络模型的输出,  $f[\bullet]$  为系统的非

线性函数， $m$  和  $n$  分别为输入和输出阶段的阶数， $m \leq n$ ，TDL 为延时环节。RBF 神经网络采用上述 M-RAN 算法进行在线学习。利用图 6-6 所示在线辨识结构分别对以下实例进行仿真研究，为便于比较辨识精度，定义性能指标如下<sup>[15-16]</sup>：

$$PER = \frac{1}{286} \sum_{k=11}^{296} (y(k) - \hat{y}(k))^2 \quad (6-29)$$

式 (6-29) 中， $y(k)$  为  $k$  时刻的实际输出数据； $\hat{y}(k)$  为  $k$  时刻的 RBF 神经网络模型输出。

### 3. 在线辨识实例

燃烧炉建模是一个经典的系统辨识问题<sup>[17]</sup>，很多文献<sup>[18-19]</sup>都以之为例来比较各自的模拟实验结果。本文根据文献[17]中提供的 296 组实测输入/输出数据，用 RBF 神经网络建立燃烧炉的动态数学模型，并与其他方法进行比较。

选取模型的输入向量为  $[u(k-3), u(k-4), y(k-1), y(k-2)]^T$ ，模型的输出变量为  $y(k)$ ，其中  $u(k)$  是甲烷的流量， $y(k)$  是炉膛出口排烟中  $CO_2$  含量的百分比。

M-RAN 学习算法的参数为： $\varepsilon_{\max} = 4, \varepsilon_{\min} = 3, \gamma = 0.998, k = 0.9, e_{\min} = 0.01$ ，

$e'_{\min} = 0.5, P_0 = 1, R_n = 1, Q = 1, \delta = 0.001, S_d = 0.01, M = 30$ 。模型输出与实测输出数据的比较发现改进的 RBF 神经网络能较好地跟踪对象的实际输出数据。为了说明改进算法的有效性，将该算法与其他算法进行了精度比较，结果如表 6-2 所示。通过比较说明，本文的算法是有效的且具有较高的精度。

表 6-2 改进算法与其他文献算法的结果比较

文献	类型	注释	PER
[20]	模糊	修正的模型	0.469
[21]	模糊	r=5	0.776
		r = 9	0.320
[17]	普通线性模型	单输入单输出 6 参数	0.71
[18]	模糊	未修正的模型	0.4555
		修正的模型	0.328
[19]	模糊	线性模型	0.642
		5 条规则	0.248
改进 RBF 方法	神经网络	4 个隐层单元	0.212

从表 6-2 可以看出，改进的 RBF 神经网络热工过程在线辨识算法，具有较好的学习精度。

### 本章小结

RBF 神经网络的结构是不常见的，因为隐藏单元的结构与输出单元的结构完全不同。由于径向基函数是隐层单元设计的基础，所以 RBF 网络的理论与径向基函数理论有着密切的联系。另外，与采用反向传播算法的多层感知器不同，RBF 网络设计采用原理化的方法。

本章讨论径向基函数神经网络 RBF 的结构，工作原理，在此基础上，介绍了一种基于免疫算法的改进 RBF 神经网络的隐层数据中心个数的确定，隐层数据位置位置的确定和输出权值的学习调整算法。最后讨论了 RBF 神经网络在热工过程辨识方面的具体应用。

参考文献：

- [1] Moody J., Darken C., Fast learning in networks of locally-tuned processing units, *Neural Computation*, 1989, 1(2):281-294.
- [2] Powell M.J.D., Radial basis function for multivariable interpolation: A review, *IMA Conference on Algorithms for the Approximation of Functions and Data*, 1985, pp.143-167, Shrivenham, England.
- [3] Broomhead D.S., Lowe D., Multivariable functional interpolation and adaptive networks, *Complex Systems*, 1988, 2(2):321-355.
- [4] Poggio T., Girosi F., Regularization algorithms for learning that are equivalent to multilayer networks, *Science*, 1990, Vol.247, pp.978-982.
- [5] 徐丽娜, 神经网络控制, 电子工业出版社, 2003.2, 北京。
- [6] Simon Haykin 著, 叶世伟, 史忠植译, 神经网络原理, 机械工业出版社, 2004 年 5 月, 北京。
- [7] Duda R.O., Hart P.R., Pattern classification and scene analysis, New York: Wiley. 1973.
- [8] 朱大奇, 张伟, 一种改进的 RBF 神经网络在线非线性辨识算法, 第 14 届中国神经网络学术会议论文集, CNNC2004, 合肥工业大学出版社, 2004.8.
- [9] 王源, 胡寿松, 齐俊伟, 自组织模糊 CMAC 神经网络及其非线性系统辨识, 航空学报, 2001, 22(6): 556-558.
- [10] 刘志远等, 基于 RBF 神经网络的热工过程在线辨识, CIAC2003, 上册: 448 - 454.
- [11] Barry J., Ding Li Yu, Selecting radial basis function network centers with recursive orthogonal least squares training, *IEEE Trans Neural Networks*, 2000, 11(3):306-314.
- [12] Yingwei Lu, Sundararajan N., A sequential learning scheme for function approximation using minimal radial function neural networks, *Neural Computation*, 1997, 9(2):461-678.
- [13] Platt J., A resource-allocating network for function interpolation, *Neural Computation*, 1991, 3(2):213-225.
- [14] Kadiramanathan V., Niranjan M., A function estimation approach to sequential learning with neural networks, *Neural Computation*, 1993, 5(4):954-975.
- [15] Yingwei Lu, Sundararajan N., Performance evaluation of a sequential minimal radial function neural network learning algorithm, *IEEE Trans Neural Networks*, 1998, 9(2):308-318.
- [16] Narendra K.S., Parthasarathy K., Identification and control of dynamical systems using neural networks, *IEEE Trans Neural Networks*, 1990, 1(1):4-27.
- [17] Box G.E., Jenkins G.M., Time series analysis, forecasting and control, San Francisco: Holden Day, 1970.
- [18] Xu C.W., Zailu Y., Fuzzy model identification and self-learning for dynamic systems, *IEEE Trans on Syst. Man Cybern.*, 1987, 17(4):132-137.
- [19] 吕剑虹, 陈建勤, 刘志远, 基于模糊规则的热工过程非线性模型的研究, 中国机电工程学报, 2002, 22(11): 132-137.
- [20] Tong R.M., Synthesis of fuzzy models for industrial processes, *Int Gen. Syst.*, 1978, No.4, pp.143-162.
- [21] Pedrycz W., An identification of fuzzy relational systems, *Fuzzy Sets System*, 1984, No.13: 153-167.

### 思考题

1. 请介绍径向基函数 RBF 神经网络的基本构造及工作原理。
2. 和多层感知器 MLP 相比, 径向基函数 RBF 有何不同?
3. 请简要介绍  $k$ -均值聚类算法。
4. 介绍一例径向基函数 RBF 神经网络的具体应用

## 第 7 章 SOM 自组织特征映射神经网络

生物学研究表明,在人脑的感觉通道上,神经元的组织原理是有序排列的。当外界的特定时空信息输入时,大脑皮层的特定区域兴奋,而且类似的外界信息在对应的区域是连续映象的。生物视网膜中有许多特定的细胞对特定的图形比较敏感,当视网膜中有若干个接收单元同时受特定模式刺激时,就使大脑皮层中的特定神经元开始兴奋,输入模式接近,与之对应的兴奋神经元也接近;在听觉通道上,神经元在结构排列上与频率的关系十分密切,对于某个频率,特定的神经元具有最大的响应,位置相邻的神经元具有相近的频率特征,而远离的神经元具有的频率特征差别也较大。大脑皮层中神经元的这种响应特点不是先天安排好的,而是通过后天的学习自组织形成的。

据此芬兰 Helsinki 大学 Kohonen T. 教授提出一种自组织特征映射网络 SOM(Self-organizing feature Map), 又称 Kohonen 网络<sup>[1-5]</sup>。Kohonen 认为, 一个神经网络接受外界输入模式时, 将会分为不同的对应区域, 各区域对输入模式有不同的响应特征, 而这个过程是自动完成的。SOM 网络正是根据这一看法提出的, 其特点与人脑的自组织特性相类似。

### 7.1 竞争学习算法基础<sup>[6]</sup>

#### 7.1.1 自组织神经网络结构

1.定义:自组织神经网络是无导师学习网络。它通过自动寻找样本中的内在规律和本质属性, 自组织、自适应地改变网络参数与结构。

2.结构:层次型结构,具有竞争层。典型结构:输入层+竞争层。如图 7-1 所示。

输入层:接受外界信息,将输入模式向竞争层传递,起“观察”作用。

竞争层:负责对输入模式进行“分析比较”,寻找规律,并归类。

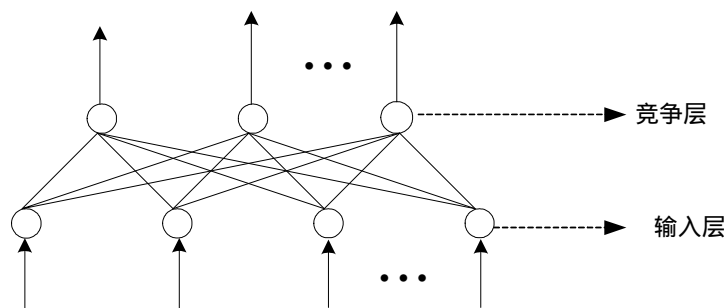


图 7-1 自组织神经网络结构

#### 7.1.2 自组织神经网络的原理

##### 1. 分类与输入模式的相似性

分类是在类别知识等导师信号的指导下,将待识别的输入模式分配到各自的模式类中,无导师指导的分类称为聚类,聚类的目的是将相似的模式样本划归一类,而将不相似的分离开来,实现模式样本的类内相似性和类间分离性。由于无导师学习的训练样本中不含期望输出,因此对于某一输入模式样本应属于哪一类并没有任何先验知识。对于一组输入模式,只能根据它们之间的相似程度来分为若干类,因此,相似性是输入模式的聚类依据。

##### 2. 相似性测量

神经网络的输入模式向量的相似性测量可用向量之间的距离来衡量。常用的方法有欧氏距离法和余弦法两种:

clustering

## 欧式距离法

设  $X, X_i$  为两向量，其间的欧式距离

$$d = \|X - X_i\| = \sqrt{(X - X_i)(X - X_i)^T} \quad (7-1)$$

$d$  越小， $X$  与  $X_i$  越接近，两者越相似，当  $d = 0$  时， $X = X_i$ ；以  $d = T$ （常数）为判据，可对输入向量模式进行聚类分析：

由于  $d_{12}, d_{23}, d_{31}$  均小于  $T$ ， $d_{45}, d_{56}, d_{46}$  均小于  $T$ ，而  $d_{1i} > T (i = 4, 5, 6)$ ， $d_{2i} > T (i = 4, 5, 6)$ ， $d_{3i} > T (i = 4, 5, 6)$ 。故将输入模式  $X_1, X_2, X_3, X_4, X_5, X_6$  分为类 1 和类 2 两大类。如图 7-2。

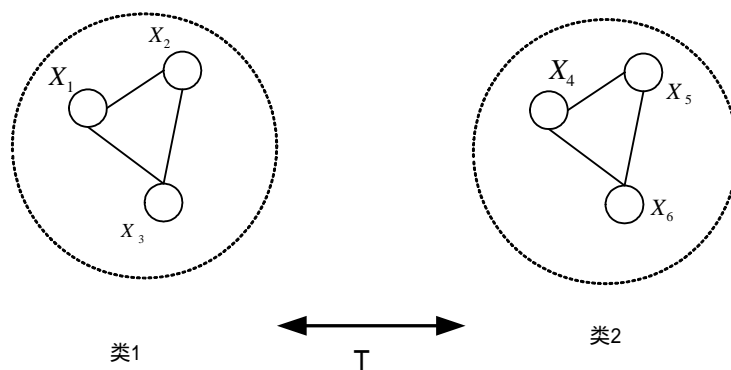


图 7-2 基于欧式距离法的模式分类

余弦法：设  $X, X_i$  为两向量，其间的夹角余弦

$$\cos \varphi = \frac{XX^T}{\|X\| \|X_i\|} \quad (7-2)$$

$\varphi$  越小， $X$  与  $X_i$  越接近，两者越相似；当  $\varphi = 0$  时， $\cos \varphi = 1$ ， $X = X_i$ ；同样以  $\varphi = \varphi_0$  为判据可进行聚类分析。

## 3. 竞争学习原理

竞争学习规则的生理学基础是神经细胞的侧抑制现象：当一个神经细胞兴奋后，会对其周围的神经细胞产生抑制作用。最强的抑制作用是竞争获胜的“唯我独兴”，这种做法称为“胜者为王”，(Winner-Take-All)。竞争学习规则就是从神经细胞的侧抑制现象获得的。它的学习步骤为：

向量归一化：对自组织网络中的当前输入模式向量  $X$ 、竞争层中各神经元对应的内星权向量  $W_j$  ( $j = 1, 2, \dots, m$ )，全部进行归一化处理，如图 7-3，得到  $\hat{X}$  和  $\hat{W}_j$ ：

$$\hat{X} = \frac{X}{\|X\|}, \quad \hat{W}_j = \frac{W_j}{\|W_j\|} \quad (7-3)$$

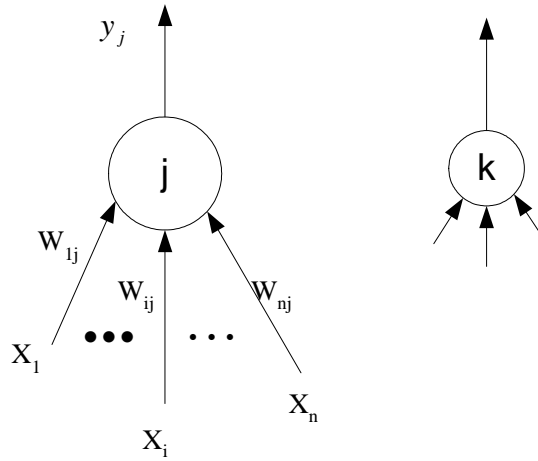


图 7 - 3 向量归一化

寻找获胜神经元：将  $\hat{X}$  与竞争层所有神经元对应的内量权向量  $\hat{W}_j (j = 1, 2, \dots, m)$  进行相

似性比较。其最相似的神经元获胜，其权向量为  $\hat{W}_{j^*}$ ：

$$\begin{aligned} \|\hat{X} - \hat{W}_{j^*}\| &= \min_{j \in \{1, 2, \dots, n\}} \{\|\hat{X} - \hat{W}_j\|\} \\ \Rightarrow \|\hat{X} - \hat{W}_{j^*}\| &= \sqrt{(\hat{X} - \hat{W}_{j^*})(\hat{X} - \hat{W}_{j^*})^T} = \sqrt{\hat{X}\hat{X}^T - 2\hat{W}_{j^*}\hat{X}^T + \hat{W}_{j^*}\hat{W}_{j^*}^T} \\ &= \sqrt{2(1 - \hat{W}_{j^*}\hat{X}^T)} \\ \Rightarrow \hat{W}_{j^*}\hat{X}^T &= \max_j (\hat{W}_j\hat{X}^T) \end{aligned} \quad (7-4)$$

网络输出与权调整：按 WTA 学习法则，获胜神经元输出为“1”，其余为 0。即：

$$y_j(t+1) = \begin{cases} 1 & j = j^* \\ 0 & j \neq j^* \end{cases} \quad (7-5)$$

只有获胜神经元才有权调整其权向量  $W_{j^*}$ 。其权向量学习调整如下：

$$\begin{cases} W_{j^*}(t+1) = \hat{W}_{j^*}(t) + \Delta W_{j^*} = \hat{W}_{j^*}(t) + \alpha(\hat{X} - \hat{W}_{j^*}) \\ W_j(t+1) = \hat{W}_j(t) & j \neq j^* \end{cases} \quad (7-6)$$

$0 < \alpha \leq 1$  学习率。 $\alpha$  一般随着学习的进展而减小，即调整的程度越来越小，趋于聚类中心。

重新归一化处理：归一化后的权向量经过调整后，得到的新向量不再是单位向量，因此要对学习调整后的向量重新进行归一化，循环运算，直到学习率  $\alpha$  衰减到 0。

为了更好地说明竞争学习算法的聚类分析效果，下面以一具体实例进行计算<sup>[6]</sup>。

例 7 - 1. 用竞争学习算法将下列各模式分为两类

$$X_1 = (0.8, 0.6), X_2 = (0.1736, -0.9848), X_3 = (0.707, 0.707),$$

$$X_4 = (0.342, -0.9397), X_5 = (0.6, 0.8), \quad \text{学习率 } \alpha = 0.5.$$

解：将上述输入模式转换为极坐标形式：

$X_1 = 1\angle 36.89^\circ$  ,  $X_2 = 1\angle -80^\circ$  ,  $X_3 = 1\angle 45^\circ$  ,  $X_4 = 1\angle -70^\circ$  ,  $X_5 = 1\angle 53.13^\circ$ 。如图 7 - 4 所示。

要求将各模式分为两类，则竞争层为 2 个神经元，设有两个权向量，随机初始化为单元向量： $W_1(0) = (1,0) = 1\angle 0^\circ$        $W_2(0) = (-1,0) = 1\angle -180^\circ$ ，其竞争学习过程如下：

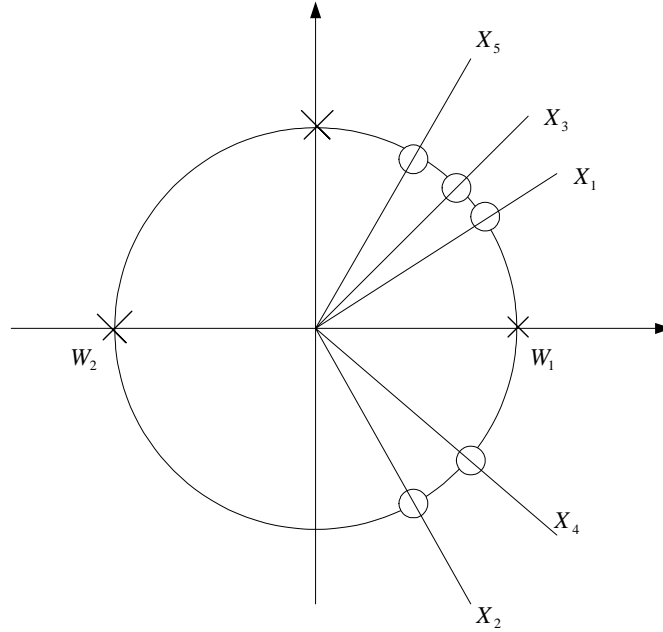


图 7 - 4 模式向量图

$X_1$  :

$$d_1 = \|X_1 - W_1(0)\| = 1\angle 36.89^\circ ; d_2 = \|X_1 - W_2(0)\| = 1\angle 216.89^\circ$$

$d_1 < d_2$ ，神经元 1 获胜， $W_1$  调整。

$$W_1(1) = W_1(0) + \alpha(X_1 - W_1(0)) = 0 + 0.5 \times 36.89 = 1\angle 18.43^\circ ;$$

$$W_2(1) = W_2(0) = 1\angle -180^\circ。$$

$X_2$  :

$$d_1 = \|X_2 - W_1(1)\| = 1\angle 98.43^\circ ; d_2 = \|X_2 - W_2(1)\| = 1\angle 100^\circ$$

$d_1 < d_2$ ，神经元 1 获胜， $W_1$  调整。

$$W_1(2) = W_1(1) + \alpha(X_2 - W_1(1)) = 18.43 + 0.5 \times (-80 - 18.43) = 1\angle -30.8^\circ ;$$

$$W_2(2) = W_2(1) = 1\angle -180^\circ。$$

$X_3$  :



$$d_1 = \|X_3 - W_1(2)\| = 1\angle 75.8^\circ ; d_2 = \|X_3 - W_2(2)\| = 1\angle 225^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(3) = W_1(2) + \alpha(X_3 - W_1(2)) = -30.8 + 0.5 \times (45 + 30.8) = 1\angle 7^\circ ;$$

$$W_2(3) = W_2(2) = 1\angle -180^\circ。$$

$X_4$

$$d_1 = \|X_4 - W_1(3)\| = 1\angle 77^\circ ; d_2 = \|X_4 - W_2(3)\| = 1\angle 110^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(4) = W_1(3) + \alpha(X_4 - W_1(3)) = 7 + 0.5 \times (-70 - 7) = 1\angle -31.5^\circ ;$$

$$W_2(4) = W_2(3) = 1\angle -180^\circ。$$

$X_5$  :

$$d_1 = \|X_5 - W_1(4)\| = 1\angle 84.63^\circ ; d_2 = \|X_5 - W_2(4)\| = 1\angle 126.87^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(5) = W_1(4) + \alpha(X_5 - W_1(4)) = -31.5 + 0.5 \times (53.13 + 31.5) \approx 1\angle 11^\circ ;$$

$$W_2(5) = W_2(4) = 1\angle -180^\circ。$$

$X_1$  :

$$d_1 = \|X_1 - W_1(5)\| = 1\angle 25.89^\circ ; d_2 = \|X_1 - W_2(5)\| = 1\angle 216.89^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(6) = W_1(5) + \alpha(X_1 - W_1(5)) = 11 + 0.5 \times 25.89 \approx 1\angle 24^\circ ;$$

$$W_2(6) = W_2(5) = 1\angle -180^\circ。$$

$X_2$  :

$$d_1 = \|X_2 - W_1(6)\| = 1\angle 104^\circ ; d_2 = \|X_2 - W_2(6)\| = 1\angle 100^\circ$$

$d_2 < d_1$ , 神经元 2 获胜,  $W_2$  调整。

$$W_2(7) = W_2(6) + \alpha(X_2 - W_2(6)) = -180 + 0.5 \times (-80 + 180) = 1\angle -130^\circ ;$$

$$W_1(7) = W_1(6) = 1\angle 24^\circ。$$

$X_3$  :

$$d_1 = \|X_3 - W_1(7)\| = 1\angle 21^\circ ; d_2 = \|X_3 - W_2(7)\| = 1\angle 175^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(8) = W_1(7) + \alpha(X_3 - W_1(7)) = 24 + 0.5 \times (45 - 24) \approx 1\angle 34^\circ ;$$

$$W_2(8) = W_2(7) = 1\angle -130^\circ。$$

$X_4$

$$d_1 = \|X_4 - W_1(8)\| = 1\angle 104^\circ ; d_2 = \|X_4 - W_2(8)\| = 1\angle 60^\circ$$

$d_2 < d_1$ , 神经元 2 获胜,  $W_2$  调整。

$$W_2(9) = W_2(8) + \alpha(X_4 - W_2(8)) = -130 + 0.5 \times (-70 + 130) = 1\angle -100^\circ ;$$

$$W_1(9) = W_1(8) = 1\angle 34^\circ。$$

$X_5$  :

$$d_1 = \|X_5 - W_1(9)\| = 1\angle 19.13^\circ ; d_2 = \|X_5 - W_2(9)\| = 1\angle 153.13^\circ$$

$d_1 < d_2$ , 神经元 1 获胜,  $W_1$  调整。

$$W_1(10) = W_1(9) + \alpha(X_5 - W_1(9)) = 34 + 0.5 \times (53.13 - 34) \approx 1\angle 44^\circ ;$$

$$W_2(10) = W_2(9) = 1\angle -100^\circ。$$

一直循环运算下去, 其前 20 次学习结果如表 7 - 1 所示。

表 7 - 1 竞争学习结果

学习次数	$W_1$	$W_2$	学习次数	$W_1$	$W_2$
1	18.43 °	-180 °	11	40.5 °	-100 °
2	-30.8 °	-180 °	12	40.5 °	-90 °
3	7 °	-180 °	13	43 °	-90 °
4	-32 °	-180 °	14	43 °	-81 °
5	11 °	-180 °	15	47.5 °	-81 °
6	24 °	-180 °	16	42 °	-81 °
7	24 °	-130 °	17	42 °	-80.5 °

8	34 °	-130 °	18	43.5 °	-80.5 °
9	34 °	-100 °	19	43.5 °	-75 °
10	44 °	-100 °	20	48.5 °	-75 °

从表 7 - 1 可见，在运算学习 20 次后，网络权值  $W_1$ ， $W_2$  趋于稳定：

$W_1 \rightarrow 45^0, W_2 \rightarrow -75^0$ 。同时我们还可以得到如下结论：

- (1) 当  $W_1$  调整时， $W_2$  不变，反之亦然，每次只有一个神经元获胜；
- (2)  $X_1, X_3, X_5$  属于同一模式，其中心向量为： $\frac{1}{3}(X_1 + X_2 + X_3) = 45^0$ ， $X_2$ 、 $X_4$  属于同一模式类： $\frac{1}{2}(X_2 + X_4) = 1 \angle -75^0$ 。
- (3) 若学习率  $\alpha$  为常数， $W_1$ 、 $W_2$  将在中心向量附近摆动，永远也不收敛。

## 7.2 SOM 神经网络模型与算法<sup>[7]</sup>

### 1. SOM 网络的拓扑结构

SOM 网络共有两层。即：输入层和输出层。

输入层：通过权向量将外界信息汇集到输出层各神经元。输入层的形式与 BP 网相同，节点数与样本维数相同。

输出层：输出层也是竞争层。其神经元的排列有多种形式。分为一维线阵、二维平面阵和三维栅格阵。最典型的结构是二维形式。它更具大脑皮层的形象。如图 7-5 所示。

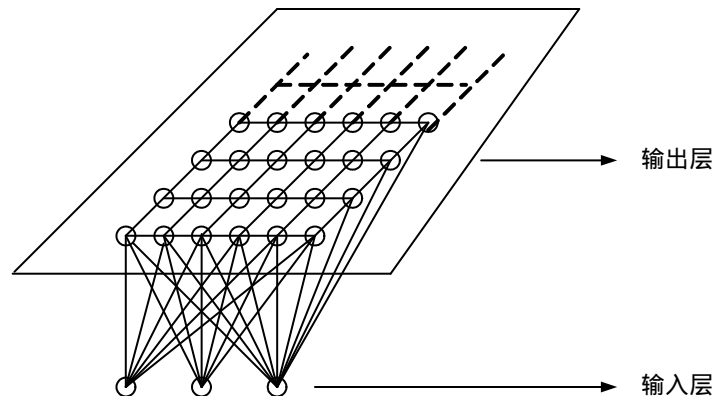


图 7-5 二维 SOM 平面阵列

输出层的每个神经元同它周围的其他神经元侧向连接，排列成棋盘状平面；输入层为单层神经元排列。

### 2. SOM 权值调整域

SOM 网采用的算法，称为 Kohonen 算法，它是在胜者为王 WTA (Winner-Take-All) 学习规则基础上加以改进的，主要区别是调整权向量与侧抑制的方式不同：

WTA：侧抑制是“封杀”式的。只有获胜神经元可以调整其权值，其他神经元都无权调整。Kohonen 算法：获胜神经元对其邻近神经元的影响是由近及远，由兴奋逐渐变为抑制。换句话说，不仅获胜神经元要调整权值，它周围神经元也要不同程度调整权向量。常见的调整方式有如下几种：

**墨西哥草帽函数**：获胜节点有最大的权值调整量，临近的节点有稍小的调整量，离获胜节点距离越大，权值调整量越小，直到某一距离  $d_0$  时，权值调整量为零；当距离再远一些时，权值调整量稍负，更远又回到零。如图 7-6 (a)；

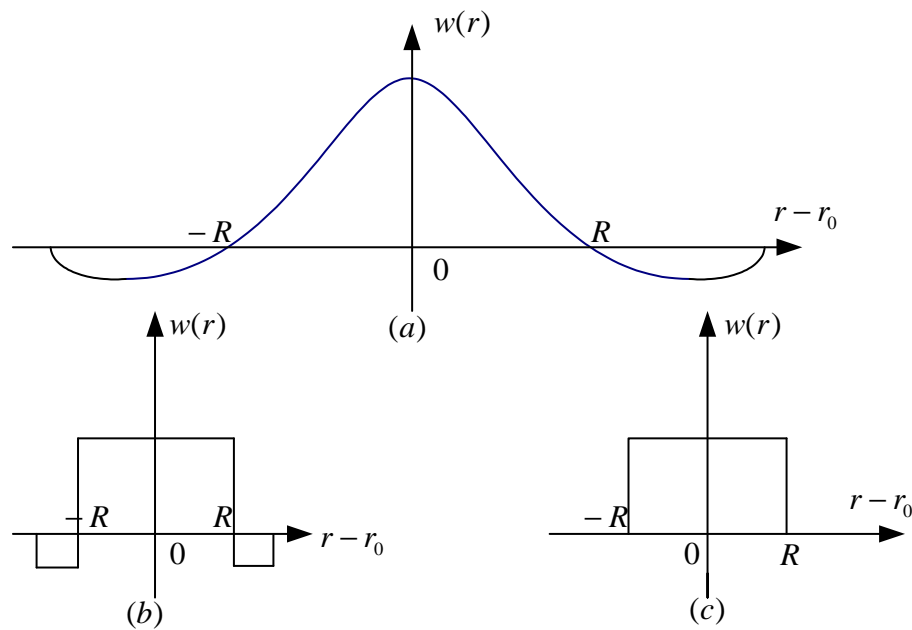


图 7-6 权值调整函数

**大礼帽函数**：它是墨西哥草帽函数的一种简化，如图 7-6 (b)；

**厨师帽函数**：它是大礼帽函数的一种简化，如图 7-6 (c)。

以获胜神经元为中心设定一个邻域半径  $R$ ，该半径固定的范围称为优胜邻域。在 SOM 网学习方法中，优胜邻域内的所有神经元均为按其离开获胜神经元的距离远近不同程度地调整权值。优胜邻域开始定的较大，但其大小随着训练次数的增加不断收缩，最终收缩到半径为零。

### 3. SOM 网络运行原理

SOM 网络的运行分训练和工作两个阶段。在训练阶段，网络随机输入训练集中的样本，对某个特定的输入模式，输出层会有某个节点产生最大响应而获胜，而在训练开始阶段，输出层哪个位置的节点将对哪类输入模式产生最大响应是不确定的。当输入模式的类别改变时，二维平面的获胜节点也会改变。获胜节点周围的节点因侧向相互兴奋作用也产生较大影响，于是获胜节点及其优胜邻域内的所有节点所连接的权向量均向输入方向作不同程度的调整，调整力度依邻域内各节点距离获胜节点的远近而逐渐减小。网络通过自组织方式，用大量训练样本调整网络权值，最后使输出层各节点成为对特定模式类敏感的神经元，对应的内星权向量成为各输入模式的中心向量。并且当两个模式类的特征接近时，代表这两类的节点在位置上也接近。从而在输出层形成能反应样本模式类分布情况的有序特征图。

### 4. 学习方法

对应于上述运行原理，SOM 网络采用的学习算法按如下步骤进行：

初始化：对输出层各权向量赋小随机数并进行归一化处理，得到  $\hat{W}_j$  ( $j = 1, 2, \dots, m$ ) 建立

初始优胜邻域  $N_j^*(0)$  和学习率  $\eta$  初值。 $m$  为输出层神经元数目。

接受输入：从训练集中随机取一个输入模式并进行归一化处理，得  $\hat{X}^p$  ( $p = 1, 2, \dots, n$ )， $n$

为输入层神经元数目。

寻找获胜节点:计算  $\hat{X}^P$  与  $\hat{W}_j$  的点积,从中找到点积最大的获胜节点  $j^*$  ;

定义优胜邻域  $N_{j^*}(t)$  :以  $j^*$  为中心确定  $t$  时刻的权值调整域 ,一般初始邻域  $N_{j^*}(0)$  较大(大约为总节点的 50~80%) , 训练过程中  $N_{j^*}(t)$  随训练时间收缩。如图 7-7。

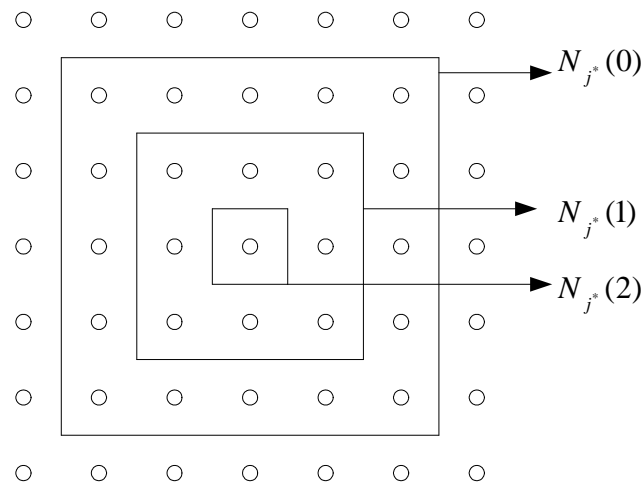


图 7-7 邻域  $N_{j^*}(t)$  的收缩

调整权值：对优胜邻域  $N_{j^*}(t)$  内的所有节点调整权值

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t, N)[x_i^P - w_{ij}(t)] \quad (7-7)$$

$$i = 1, 2, \dots, n, \quad j \in N_{j^*}(t)$$

式 (7-7) 中,  $\alpha(t, N)$  是训练时间  $t$  和邻域内第  $j$  个神经元与获胜神经元  $j^*$  之间的拓扑距离  $N$  的函数, 该函数一般有以下规律:

$t \uparrow \rightarrow \alpha \downarrow, N \uparrow \rightarrow \alpha \downarrow$ ; 如  $\alpha(t, N) = \alpha(t)e^{-N}$ ,  $\alpha(t)$  可采用  $t$  的单调下降函数也称退火函数。

结束判定：当学习率  $\alpha(t) \leq \alpha_{\min}$  时, 训练结束; 不满足结束条件时, 转到步骤 继续。

## 7.3 SOM 神经网络的应用

### 7.3.1 SOM 网络应用于字符排序<sup>[6]</sup>

#### 1. 问题分析

用 32 个字符作为 SOM 输入样本, 包括 26 个英文字母和 6 个数字 (1-6)。每个字符对应域一个 5 维向量, 各字符与向量  $X$  关系如表 7-2。由表可以看出, 代表 A、B、C、D、E 的各向量中有 4 个分量相同, 即  $x_i^A, x_i^B, x_i^C, x_i^D, x_i^E = 0, (i = 1, 2, 3, 4)$ , 因此, A、B、C、D、E 应归为一类; 代表 F、G、H、I、J 的向量中有 3 个分量相同, 同理也应归为一类;

其他类推。这样就可以由表 7-2 中输入向量的相似关系，将对应的字符标在图 7-8 所示的树形结构图中。用 SOM 网络对其他进行聚类分析。

表 7-2 字符与其对应向量

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	...
$X_0$	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3...
$X_1$	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3...
$X_2$	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3...
$X_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	3	3	3...
$X_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2...

## 2. 网络设计及学习结果

表格分析：A、B、C、D、E 的各向量有 4 个分量相同。 同类。

F、G、H、I、J 的各向量有 3 个分量相同。 同类。

.....

SOM 网络设计：

- (1) 输入层节点数  $n$ ：样本维数 = 5。
- (2) 输出层节点数：取 70 个神经元，二维平面阵。
- (3) 权值初始化：随机小数。
- (4)  $N_j(t)$  领域半径： $r(t) = 10(1 - t/t_m)$
- (5) 学习率  $\alpha(t) = C_2(1 - t/t_m)0.5(1 - t/t_m)$

训练：将训练集中代表各字符的输入向量  $X^P$  随机选取后训练，经 10000 步训练，各权向量趋于稳定。对网络输出，进行核准，即根据输出阵列神经元与训练集的已知的模式向量的对应关系的标号。结果是：70 各神经元中，有 32 个神经元有标号，另外 38 个为未用神经元。

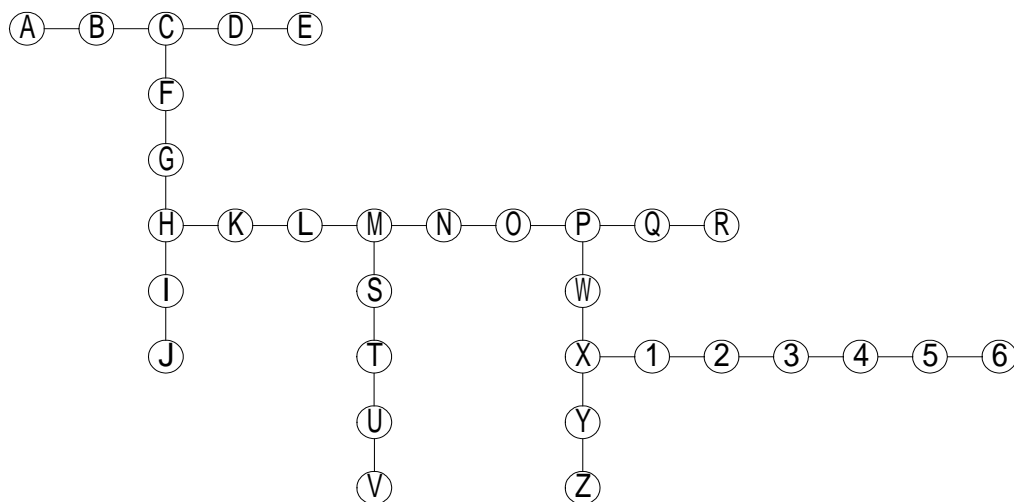


图 7-8 树形结构图

## 3. 结果输出

图 7-9 给出了自组织学习后的输出结果。SOM 网络完成学习训练后，对于每一个输入字符，输出平面中都有一个特定的神经元对其敏感，这种输入 - 输出的映射关系在输出特征平面中表现得非常清楚。SOM 网络经自组织学习后在输出层形成了有规则的拓扑结构，在神

神经元阵列中，各字符之间的相互位置关系与它们在树状结构中的相互位置关系类似，两者结构特征上的一致性是非常明显的。

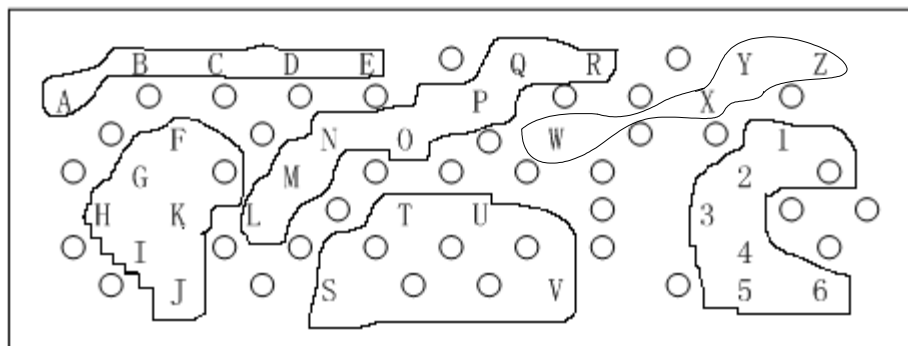


图 7 - 9 给出了自组织学习的输出结果

### 7.3.2 SOM 网络在复合材料损伤监测中的应用<sup>[8]</sup>

#### 1. 问题描述

复合材料的损伤现象及机理复杂，在使用时需要对其损伤累积进行监测并预警。一般的方法是应力波技术，即在复合材料中施加激励信号——应力波，然后在复合材料的另一点或多点接收应力波，由于应力波在复合材料中传输受损伤情况的影响，通过分析检测信号，来识别损伤情况。因此，关键问题是接收信号与损伤模式之间的函数关系，即要进行系统建模。但是一般来说，建模困难，大多是应用简化模型，检测精度较低。此处的解决途径是利用自组织神经网络 SOM 来识别。被检测的复合材料为碳纤维层合板。

#### 2. 损伤模式

$P_1$ ：碳纤维层合板健康；  $P_2$ ：材料表面冲击损伤；  $P_3$ ：材料托层损伤；

$P_4$ ：材料夹层健康；  $P_5$ ：材料夹层冲击损伤。

损伤特征向量  $X = [X_1, X_2, X_3, X_4, X_5, X_6]$  选择：

$X_1$ ：时域峰 - 峰值 (  $mv$  )；  $X_2$ ：频域第一峰值点频率 (  $KHz$  )；  $X_3$ ：频域第一峰值

点的幅值 (  $mv$  )；  $X_4$ ：频域第二峰值频率 (  $KHz$  )；  $X_5$ ：频率第二峰值点幅值 (  $mv$  )；

$X_6$ ：振铃应力波因子 ( 次 )

#### 3. 数据规格化处理

由于 6 种参数的量纲和数量级有差别，采用如下数据规格化处理，使每个分量  $X_K$  统一于某种共同的数据范围内：

$$X'_K = \frac{X_K - \bar{X}}{\sigma} \quad ; \text{其中 } \bar{X} = \frac{1}{6} \sum_{K=1}^6 X_K \quad \sigma = \sqrt{\frac{1}{6-1} \sum_{K=1}^6 (X_K - \bar{X})^2} \quad , \text{则得到损伤训练样}$$

本如表 7 - 3 所示。归格化处理后得到表 7 - 4。

表 7 - 3 损伤训练样本

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$
$P_1$	9.51	45.6	45.0	21.5	29.5	40
$P_2$	5.67	34.8	32.5	0.781	23.0	29
$P_3$	1.77	51.1	8.02	34.2	6.25	8
$P_4$	8.06	20.0	4.71	80.0	4.01	12
$P_5$	2.60	12.4	0.95	21.8	1.17	7

表 7 - 4 规格化处理

	$X'_1$	$X'_2$	$X'_3$	$X'_4$	$X'_5$	$X'_6$
$P_1$	-1.55	0.952	0.913	-0.718	-0.164	0.566
$P_2$	-1.06	0.964	0.799	-1.40	0.143	0.559
$P_3$	-0.831	1.66	-0.515	0.807	-0.605	-0.517
$P_4$	-0.458	-0.0499	-0.572	2.00	-0.597	-0.323
$P_5$	-0.617	0.579	-0.0818	1.71	-0.792	0.0797

#### 4. 网络设计与训练

从上面的分析,可作如下的网络设计:

输入层:6 个节点(神经元)

输出层:64 个 =  $8 \times 8$  (节点)

权值调整:  $w_{ij}^*(t+1) = w_{ij}^*(t) + \alpha(t)[x_i' - w_{ij}^*(t)]$

$t$  学习次数;  $\alpha(t) = \alpha(0)(1 - \frac{t}{T})$ ;  $T$  总学习次数

定义优胜领域  $N_{j^*}(t)$ :为简化起见,选取获胜神经元的相邻神经元优胜领域。

#### 5. 训练结果

图 7—10 为训练稳定后  $P_1$ 、 $P_2$ 、 $P_3$ 、 $P_4$ 、 $P_5$ ,对  $8 \times 8$  输出层的映射位置。SOM 网络完成学习训练后,对于每一个输入的损伤模式,输出平面中都有一个特定的神经元对其敏感,这种输入 - 输出的映射关系在输出特征平面中表现得非常清楚。



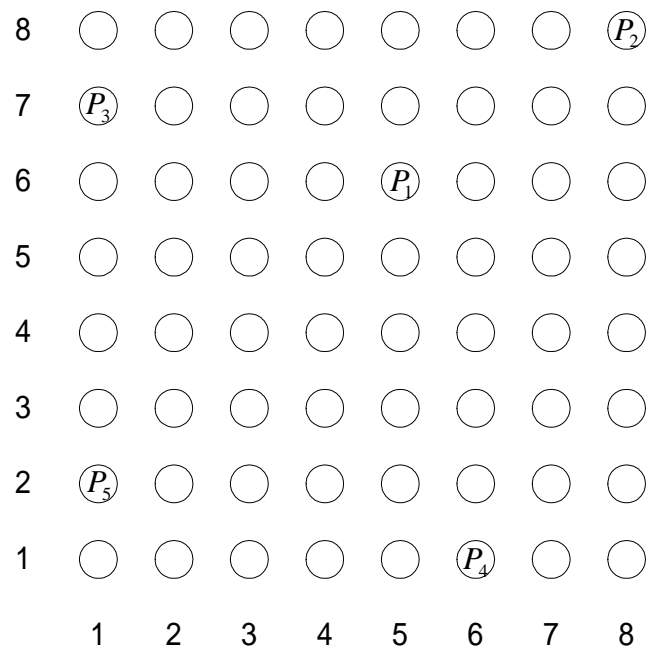


图 7—10 自组织学习的训练结果

6 . 损伤模式识别

表 7 - 5 待识别的损伤样本

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$
$F_1$	9.79	45.6	46.8	21.16	34.7	41
$F_2$	5.74	34.8	31.54	0.78	22.3	29
$F_3$	1.74	51.1	7.92	34.5	6.50	8
$F_4$	8.32	20	4.90	80	3.87	12
$F_5$	2.53	12.37	1.01	19.9	1.05	6

表 7 - 6 规格化处理表

	$X'_1$	$X'_2$	$X'_3$	$X'_4$	$X'_5$	$X'_6$
$F_1$	-1.58	0.839	0.922	-0.812	0.101	0.530
$F_2$	-1.05	1.01	0.781	-1.40	0.218	0.529
$F_3$	-0.835	1.65	-0.523	0.817	-0.595	-0.519
$F_4$	-0.445	-0.046	-0.561	2.00	-0.596	-0.353
$F_5$	-0.609	0.692	-0.809	1.68	-0.804	-0.151

在 SOM 训练结束后，可以对表 7—5 的检测数值进行复合材料的损伤模式识别。表 7 - 6 为归格化后的样本表格。将其输入训练过的 SOM 网络中，观察输出的敏感神经元，得到输出结果如图 7 - 11 所示。

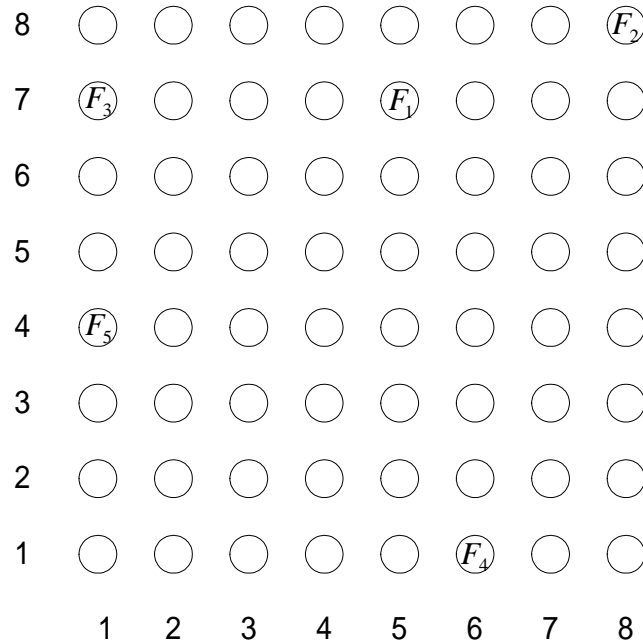


图 7 - 11 损伤模式识别输出结果

从输出结果很容易知道： $F_2$ 与 $P_2$ ：材料表面冲击损伤相对应； $F_3$ 与 $P_3$ ：材料托层损伤相对应； $F_4$ 与 $P_4$ ：材料夹层健康相对应；而 $F_5$ 接近 $P_5$ ：材料夹层冲击损伤； $F_1$ 接近 $P_1$ ：碳纤维层合板健康。

### 7.3.3 SOM 网络在电路故障诊断中应用<sup>[9]</sup>

模拟电路系统的故障诊断，是复杂系统故障诊断研究的一个难点。其中交流故障字典法是其中的一个重要方面。本节介绍的电路故障诊断的 SOM 神经网络方法，是利用自组织神经网络对传统的交流故障字典法进行了改进，以解决模拟电路故障诊断的容差问题。主要方法为：首先将模拟系统的激励响应，经小波变换得到特征值，然后应用 SOM 神经网络建立故障字典，进行故障的快速定位。采用此方法诊断，结果既准确又快捷。而且，由于神经网络系统具有良好的泛化特性，可以有效地克服容差因素对故障定位的影响。

#### 1. 故障字典的形成

在模拟电路中，故障可分为两大类：一类称为硬故障，指元件的开路和短路失效故障；另一类称为软故障，指元件的参数超出了预定的容差范围，一般它们均未使设备完全失效，例如，由于元件的老化、变质或使用环境的变化等，造成的元件参数变化。建立故障字典的过程，实际上是测试前的分析过程，它主要完成故障集及激励信号的选择，故障特征模糊域的分割，故障特征码的生成等工作。故障字典按以下步骤形成：

在确定故障集、激励信号后，通过现有的各种模拟电路的分析程序(如常用的 PSPICE 程序)，仿真每一个故障状态下模拟系统的冲激响应。

利用小波变换，将每一个故障状态下模拟系统的冲激响应进行处理，得到与正常情况的能量差，进行归一化处理，得到故障特征值。

用 SOM 网络建立故障字典。经过 SOM 训练后，形成具有容差且与故障特征值相匹配的故障字典。

## 2. SOM 网络工作特点

SOM 网络能对输入模式自动分类，即在无教师示教的情况下，通过输入模式的自组织学习，在竞争层将分类结果表示出来。与其他类型神经网络的区别在于：它不是以一个神经元或网络的状态矢量反映分类结果的，而是以若干神经元同时（并行）反映分类结果。这种特征映射神经网络，通过对输入模式的反复学习，使连接权矢量空间分布密度与输入模式的概率分布趋于一致，即连接权矢量的空间分布能反映输入模式的统计特性。

SOM 网络工作过程是这样的：如果输入样本用  $x^p$  表示，每个样本  $x^p$  的分量与输入第  $i$  个神经元相连，与输出第  $j$  个神经元之间的连接权为  $w_{ij}$ ，其输出为  $y_j$ ，则输入输出之间的关系为：

$$\frac{dy_j}{dt} = \sum_{i=1}^n w_{ij} x_i^p - \gamma(y_j) \quad (7-9)$$

其中： $\gamma(y_j)$  是非线性量，其效果是使输出的变化速度变慢。竞争就是比较  $x_i^p$  与  $w_{ij}$  之间的相近程度，计算  $x_i^p$  与  $w_{ij}$  的点积，从中找到点积最大的获胜节点  $j^*$ ；定义优胜邻域  $N_{j^*}(t)$ ；网络的训练过程即是  $w_{ij}$  的调整学习过程：

$$\frac{dW_j}{dt} = \begin{cases} \alpha(X^p - W_j) & j \in N_{j^*}(t) \\ 0 & j \notin N_{j^*}(t) \end{cases} \quad (7-10)$$

由上式可以看出，网络的功能是利用自组织的方法，通过对样本的训练，调整网络的权值，使得网络的输出能够反映样本数据的分布情况。

## 3. 具有容差的故障字典

如图 7 - 12 所示模拟电路为一无源线性网络，其元件参数如下： $R_1 = 1 M\Omega$ ， $R_2 = 10 M\Omega$ ， $R_3 = 2 M\Omega$ ，及  $R_4 = 1 M\Omega$ ， $C_1 = 0.01 \mu F$ ， $C_2 = 1000 pF$ ， $C_3 = 1000 pF$ 。

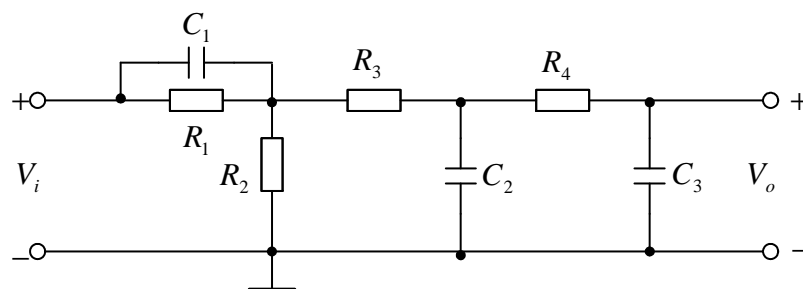


图 7 - 12 无源线性网络

利用 PSPICE 程序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，并进行数据处理，形成具有容差的故障集特征向量。如表 7 - 7 所示。

表 7 - 7 故障集特征向量

故障序号	故障内容	特征向量数据					
0	正常	0	0	0	0	0	0
1	$R_1$ 正偏 50% ( $R_1^+$ )	0.6529	0.5582	0.3707	0.3270	0.1310	0.0244

	容差	0.0119	0.0099	0.0024	-0.0081	-0.0025	-0.0006
2	R <sub>2</sub> 正偏 50% (R <sub>2</sub> <sup>+</sup> )	0.7302	0.5304	0.2162	0.3686	0.0453	0.0281
	容差	0.0007	0.0006	-0.0001	0.0005	0.0001	0.0002
3	C <sub>1</sub> 正偏 50% (C <sub>1</sub> <sup>+</sup> )	0.1220	0.3404	0.7553	0.5001	0.2180	0.0345
	容差	-0.0051	-0.0370	0.0173	-0.0056	0.0185	-0.0006
4	R <sub>4</sub> 负偏 50% (R <sub>4</sub> <sup>-</sup> )	0.4544	0.4967	0.6325	0.2779	0.2554	0.0647
	容差	0.0013	0.0002	0.0012	-0.0010	0.0031	0.0009
5	C <sub>2</sub> 正偏 50% (C <sub>2</sub> <sup>+</sup> )	0.2781	0.3350	0.6329	0.6027	0.2131	0.0343
	容差	-0.0044	0.0051	0.0164	-0.0218	0.0206	0.0086
6	C <sub>2</sub> 负偏 50% (C <sub>2</sub> <sup>-</sup> )	0.3042	0.3269	0.5615	0.6796	0.1220	0.0926
	容差	-0.0062	-0.0020	0.0106	-0.0012	-0.0109	-0.0117
7	R <sub>3</sub> 正偏 50% (R <sub>3</sub> <sup>+</sup> )	0.2631	0.3419	0.6799	0.5232	0.2705	0.0688
	容差	-0.0009	0.0027	0.0049	-0.0095	0.0035	0.0017
8	C <sub>3</sub> 正偏 50% (C <sub>3</sub> <sup>+</sup> )	0.2712	0.3446	0.6588	0.5573	0.2449	0.0561
	容差	-0.0033	0.0072	0.0151	-0.0030	0.0138	0.0110
9	R <sub>1</sub> 负偏 50% (R <sub>1</sub> <sup>-</sup> )	0.2209	0.2508	0.6427	0.6509	0.2240	0.0378
	容差	0.0031	0.0052	0.0019	-0.0045	-0.0011	-0.0004
10	R <sub>3</sub> 负偏 50% (R <sub>3</sub> <sup>-</sup> )	0.2718	0.2922	0.5504	0.6938	0.2293	0.0623
	容差	-0.0012	-0.0002	0.0039	-0.0043	0.0046	0.0032
11	C <sub>3</sub> C <sub>2</sub> 负偏 50%	0.3112	0.3254	0.5160	0.6791	0.2216	0.1437
	容差	-0.0057	0.0001	0.0068	-0.0029	-0.0083	0.0151
12	C <sub>1</sub> C <sub>2</sub> 负偏 50%	0.2665	0.2869	0.3301	0.5434	0.5877	0.3115
	容差	-0.0286	-0.0247	-0.0731	-0.0529	0.1061	0.0552
13	C <sub>3</sub> C <sub>1</sub> 负偏 50%	0.2989	0.3292	0.4403	0.5931	0.4305	0.2670
	容差	-0.0089	-0.0014	-0.0433	-0.0416	0.0393	0.0474

为了解决故障定位中的容差问题,此处采用的 SOM 网络为一维线阵,把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<sub>3</sub> 正偏 50% (C<sub>3</sub><sup>+</sup>),经过 SOM 网络训练后,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)和(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,其余两者为其相邻神经元。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。

#### 4. 故障特征量的再映射

经过 SOM 网络训练学习后,输出神经元兴奋的位置与输入样本的大小有关。即:竞争层可以在一维或二维处理单元阵列上,形成输入信号的分布拓扑图。为了使测试数据输入网络后,快速、准确定位,可进行故障特征量的再映射。具体方法为:故障数据标称值(0.2712,0.3446,0.6588,0.5573,0.2449,0.0561)经过 SOM 训练后,对于网络的最大输入,神经元 7 对应的输出为 1,其相关神经元为 6 和 8,用 7、6、8 代表同类故障,也就是说,对训练后的网络来说,如输入的测试数据能使 7、6、8 三个神经元中任一兴奋,就可将它归为此类故障模式。这样一来,用三个神经元的标号代表一种故障模式类,形成一个新的数

据阵列，如表 7 - 8 所示，故障诊断时可以直接对表 8-8 数据进行查找，准确快速分类故障。

表 7 - 8 故障数据分配结果

故障号	0	1	2	3	4	5	6	7	8	9	10	11	12	13
主神经元	51	59	64	2	47	9	14	4	7	11	15	19	32	29
相关神经元 1	50	58	63	3	46	8	13	3	6	10	14	18	31	28
相关神经元 2	52	60		1	48	10	15	5	8	12	16	20	33	30

本节介绍的基于自组织特征映射神经网络解决模拟电路故障诊断容差问题的方法。可以对单和多软、硬故障进行有效地识别，定位迅速；利用 SOM 神经网络的泛化能力，有效的克服了容差因素对故障定位的影响。是一种新颖可行的诊断方法。

## 本章小结

Kohonen 1982 年提出的 SOM 自组织映射神经网络是一个巧妙的神经元网络，它建立在一维、二维或三维的神经元网络上，用于捕获包含在输入模式中感兴趣的特征，描述在复杂系统中从完全混乱到最终出现整体有序的现象。

自组织映射也可以看成向量量化器，从而提供一个导出调整权值向量的更新规则的原理性方法。此方法明确地强调邻域函数作为概率密度函数地作用。

本章讨论竞争算法的学习过程，在此基础上进一步介绍了自组织 SOM 神经网络的结构，工作原理。最后介绍了 SOM 神经网络在聚类分析、复合材料的损伤检测及模拟电路故障识别中的具体应用。

## 参考文献：

- [1] Kohonen T., Content-addressable memories, Second Edition Spring Verlag, 1987.
- [2] Kohonen T., Self-Organization and associative memory, Third Edition Spring Verlag, 1989.
- [3] Kohonen T., An introduction to neural network, Neural Networks, 1988, 1(1):3-16.
- [4] Kohonen T., Learning vector quantization, Neural Networks, 1988, 1(4):290-303.
- [5] Kohonen T., The self-organizing map, Proc. IEEE, 1990, pp.1464-1480.
- [6] 韩力群，人工神经网络的理论、设计及应用，北京，化学工业出版社，2002.1。
- [7] 朱大奇，电子设备故障诊断的原理与实践，电子工业出版社，2004.1。
- [8] 王磊，袁慎芳，Kohonen 神经网络在复合材料损伤主动监测技术中的应用，材料科学与工程，2002，20（4）：513-516。
- [9] 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。

## 思考题

1. 请介绍 SOM 神经网络的基本构造及工作原理。
2. 自组织神经网络由输入层和竞争层组成，设初始权向量已归一化为：

$$\hat{W}_1 = [1 \quad 0], \quad \hat{W}_2 = [0 \quad -1], \quad \text{现有 4 个输入模式，均为单位向量：}$$

$$X_1 = 1\angle 45^\circ, \quad X_2 = 1\angle -135^\circ, \quad X_3 = 1\angle 90^\circ, \quad X_4 = 1\angle -180^\circ.$$

试用 WTA 学习算法调整权值，给出前 20 次的权值学习结果。

3. 给定 5 个 4 维输入模式如下：

$$X_1 = [1 \quad 0 \quad 0 \quad 0], \quad X_2 = [1 \quad 1 \quad 0 \quad 0], \quad X_3 = [1 \quad 1 \quad 1 \quad 0],$$

$X_4 = [0 \quad 1 \quad 0 \quad 0]$  ,  $X_5 = [1 \quad 1 \quad 1 \quad 1]$ 。试设计一个具有  $5 \times 5$  神经元的平

面 SOM 网络, 学习率  $\alpha(t)$  在前 1000 步训练中从 0.5 线性下降到 0.04, 然后在训练到 10000 步时减小到 0, 优胜邻域半径初始值设为相邻的 2 个节点, 1000 个训练步时降为 0, 即只含获胜神经元。每训练 200 步记录一次权值, 观察其在训练过程中的变化情况。给出训练结束后, 5 个输入模式在输出平面上的映射图。并观察下列输入向量映射区间。

$$F_1 = [1 \quad 0 \quad 0 \quad 1] \quad , \quad F_2 = [1 \quad 1 \quad 0 \quad 1] \quad , \quad F_3 = [0 \quad 1 \quad 1 \quad 0] \quad ,$$

$$X_4 = [0 \quad 1 \quad 0 \quad 1] \quad , \quad X_5 = [0 \quad 1 \quad 1 \quad 1]。$$

4. 介绍一例自组织 SOM 神经网络的具体应用。

## 第 8 章 CPN 对偶传播神经网络

1987 年美国学者 Robert Hecht—Nielson 提出了对偶传播神经网络<sup>[1]</sup>(Counter - propagation Networks, 简记为 CPN)。CPN 最早是用来实现样本选择匹配系统的。它能存储二进制或模拟值的模式对, 因此 CPN 网络可以用作联想存储、模式分类、函数逼近和数据压缩等用途。与 BP 网相比, CPN 的训练速度要快很多, 所需的时间大约是 BP 网所需时间的 1%。但是, 它的应用面却因网络的性能相对来说比较窄。

### 8.1 CPN 神经网络简介<sup>[2]</sup>

#### 1. CPN 的生物学基础

从网络的拓扑结构来看, CPN 与 BP 网类似, CPN 是一个三层的神经网络, 只不过这竞争层和输出层执行的训练算法是不同的。所以, 称 CPN 是一个异构网。与同构网相比, 网络的异构性使它更接近于人脑。在上一章 SOM 神经网络分析中已经看到, 在人脑中确实存在有各种特殊的模块, 它们使用完成不同的运算。例如, 在听觉通道的每一层, 其神经元与神经纤维在结构上的排列与频率的关系十分密切, 对某一些频率, 其中某些相应的神经元会获得最大的响应。这种听觉通道上的神经元的有序排列一直延续到听觉皮层。尽管许多低层次上的神经元是预先排列好的, 但高层次上的神经元的组织则是通过学习自组织形成的。

#### 2. CPN 的结构分析

在 Robert Hecht—Nielson 提出的 CPN 神经网络中, 其竞争层和输出层分别执行较早些时候出现的两个著名算法: 即 Kohonen 1981 年提出的自组织映射 SOM 算法和 Grossberg 1969 年提出的外星(Outstar)算法。人们将执行自组织映射的层称为 Kohonen 层, 执行外星算法的层则被称为 Grossberg 层。按这种方法将这两种算法组合在一起后所获得的网络, 不仅提供了一种设计多级网训练算法的思路, 解决了多级网络的训练问题, 突破了单极网的限制, 而且还使得网络具有了许多新的特点。如前所述, 多级网络的训练问题主要是在解决隐藏层神经元相应的联接权调整时, 需要通过隐藏层神经元的理想输出来实现相关误差的估计。然而, 它们对应的理想输出又是未知的。我们知道, 在无导师训练中是不需要知道理想输出的, 因此可以考虑让网络的隐藏层执行无导师学习。这是解决多级网络训练的另一个思路。实际上, CPN 就是将无导师训练算法与有导师训练算法结合在一起, 用无导师训练解决网络隐藏层的理想输出未知的问题, 用有导师训练解决输出层按系统的要求给出指定的输出结果的问题。

#### 3. CPN 的功能分析

Kohonen 提出的自组织映射由四部分组成, 包括一个神经元阵列(用它构成 CPN 的 Kohonen 层), 一种比较选择机制, 一个局部互联, 一个自适应过程。实际上, 这一层将实现对输入进行分类的功能。所以, 该层可以执行无导师的学习, 以完成对样本集中所含的分类信息的提取。

Grossberg 层主要用来实现类表示。由于相应的类应该是用户所要求的, 所以, 对应每一个输入向量, 用户明确地知道它对应的理想输出向量, 故该层将执行有导师的训练。

两层的有机结合, 就构成一个映射系统。所以, 有人将 CPN 看成一个有能力进行一定的推广的查找表(Look—up table)。它的训练过程就是将输入向量与相应的输出向量对应起来。这些向量可以是二值的, 也可以是连续的。一旦网络完成了训练, 对一个给定的输入就可以给出一个对应的输出。网络的推广能力表明, 当网络遇到一个不太完整的、或者是不完全正确的输入时, 只要该“噪音”是在有限的范围内, CPN 都可以产生一个正确的输出。

这是因为 Kohonen 层可以找到这个含有噪音的输入应该属于的分类，而对应的 Grossberg 层则可以给出该分类的表示。从整个网络来看，就表现出一种泛化能力。这使得网络在模式识别、模式完善、信号处理等方面可以有较好的应用。

另外，上述映射的逆映射如果存在的话，可以通过对此网的简单扩展，来实现相应的逆变换。这被称为全对传网。

## 8.2 CPN 神经网络结构及原理<sup>[2]</sup>

按照对传网所表达出来的意思，它应该是可以完成信号的双向变换和传递的。不过，为了叙述简洁、易懂起见，先介绍 CPN 的向前传递阶段，此为单向 CPN，而完整的 CPN(双向网)在原理上是一样的，将留在后面介绍。同时，为了区分起见，在这里简称完整的 CPN 为全对传网，并统一地将它们记为 CPN。

### 8.2.1 CPN 网络结构

图 8—1 给出了简化的单向 CPN 的拓扑结构。从表面上看，它和前面介绍的三层 BP 网络是相同的，但是在运行过程中，它们实现的策略却是不同的。因此，对一个网络，除了它的拓扑结构外，其运行机制也是确定网络结构(如：同构、异构)和性能的重要因素。

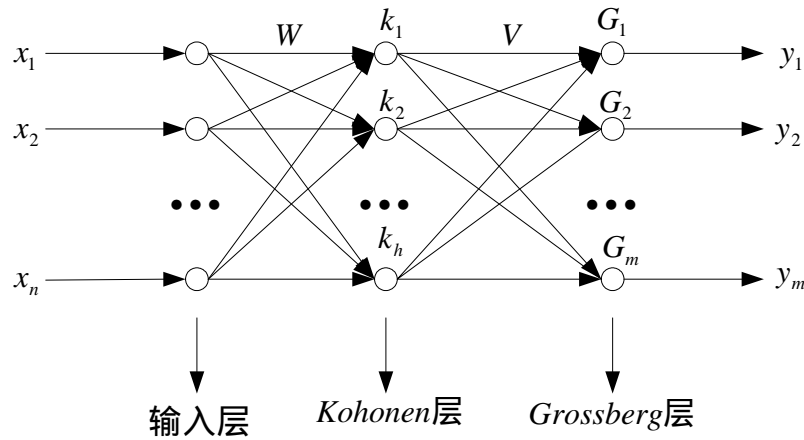


图 8-1 单向 CPN 结构

第 1 层(输入层)的神经元只起到将输入向量传送到 Kohonen 层的作用，使得输入向量的每一个分量可以被按照一定的联接权传递到第 2 层(Kohonen 层)的每个神经元。Kohonen 层的联接权矩阵用  $W$  表示，按照惯例， $w_{ij}$  表示输入向量的第  $i$  个分量到 Kohonen 层的第  $j$  个神经元的联接权。第 3 层叫做 Grossberg 层，Kohonen 层的第  $i$  个神经元到 Grossberg 层的第  $j$  个神经元的联接权用  $v_{ij}$  表示，它们放在一起构成 Grossberg 层的权矩阵。此处仍然用向量：

$$X = (x_1, x_2, \dots, x_n) \quad (8-1)$$

$$Y = (y_1, y_2, \dots, y_m) \quad (8-2)$$

$$W_1 = (w_{11}, w_{12}, \dots, w_{1h}) \quad (8-3)$$

$$W_2 = (w_{21}, w_{22}, \dots, w_{2h}) \quad (8-4)$$

$$W_n = (w_{n1}, w_{n2}, \dots, w_{nh}) \quad (8-5)$$

$$V_1 = (v_{11}, v_{12}, \dots, v_{1m}) \quad (8-6)$$



$$V_2 = (v_{21}, v_{22}, \dots, v_{2m}) \quad (8-7)$$

$$V_n = (v_{n1}, v_{n2}, \dots, v_{nm}) \quad (8-8)$$

$X$ ,  $Y$  分别表示输入向量和输出向量。其中,  $n$  为输入向量的维数,  $m$  为 Grossberg 层神经元的个数。  $h$  为 Kohonen 层神经元的个数。输出向量的维数为  $m$ 。

与其他网络一样, CPN 有两种工作模式: 训练模式和正常工作模式。在训练模式下, 对一个样本  $(X, Y)$ , Kohonen 层按照  $X$  的要求进行无导师学习(自组织映射要求的学习)。

对应地, Grossberg 层则按照理想输出  $Y$  的要求调整相应的联接权。在正常工作模式下, 给定的输入向量  $X$  被加在网络上, 通过 Kohonen 层的自组织映射而使该层的神经元处于相应的状态(在最简单的情况下, 该层的神经元将仅有一个处于激发态), 由这个状态表达的信息被 Grossberg 层转换成用户要求的形式输出。下面分别对它们进行讨论。

### 8.2.2 简单 CPN 网络运行原理

在 CPN 中, Kohonen 层和 Grossberg 层不仅执行不同的训练算法。

#### 1. Kohonen 竞争层

在最简单的情况下, Kohonen 层是以上一章介绍的“胜者为王”的方式工作的。这就是说, 对一个给定的输入向量, Kohonen 层的神经元处于激烈的竞争中, 最后, 谁获得的网络输入最大, 谁就处于激发态, 而其他的就处于抑制态。同时规定, 处于激发态的神经元输出为 1, 而处于抑制态的神经元输出为 0。在本书中, 不再讨论多个神经元可以同时处于激发状态的复杂情况的处理。实际上, 复杂情况的处理是类似的, 只是在处理细节上有一定的差别。

根据网络的特点, 讨论与 Kohonen 层的每个神经元相关联的权向量  $W_1, W_2, \dots, W_h$ 。

对 Kohonen 层的每个神经元所获得的网络输入, 即权向量与输入向量的点积:

$$W_{j^*} X^T = \max_j (W_j X^T) \quad (8-9)$$

从式 8—9 可知, 对每个输入向量  $X$ , Kohonen 层的每个神经元所获得的网络输入就是输入向量  $X$  与它相对应的联接权向量  $W_j$  的点积。点积最大的神经元为获胜神经元。由此, 可以给出 Kohonen 层的输出这样的几何解释: Kohonen 层的每个神经元对应一个分类, 同时, 它对应的联接权向量  $W_j$  与使神经元激发的这一组  $X$  的“几何距离”最近, 所以  $W_j$  就是这一组  $X$  的代表。

#### 2. Grossberg 层

按照人工神经网络的运行模型的定义, 对 Grossberg 层的每个神经元  $G_j$  ( $1 \leq j \leq m$ ), 它所获得的网络输入为:

$$G_{netj} = (WX^T) V_j^T \quad (8-10)$$

如上所述, 假定 Kohonen 层按照最简单的方式工作。这样, 对应每一个输入向量, 该层只有唯一的一个神经元处于激发态而输出 1, 其他的神经元均输出 0。假设这个唯一输出 1 的神经元为  $k_i$ , 由式 8—10, 此时, Grossberg 层的神经元  $G_j$  ( $1 \leq j \leq m$ ) 所获得的网络输入

为

$$g_{netj} = v_{ij} \quad (8-11)$$

此时有：

$$G_{NET} = (g_{net1}, g_{net2}, \dots, g_{neti}, \dots, g_{netm}) = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{im}) \quad (8-12)$$

该向量正好就是与  $k_i$  对应的向量  $V_i$ 。如果此时 Grossberg 层的神经元的激活函数为恒等映射函数，则 Grossberg 层的输出(也就是该 CPN 的输出)，就是  $V_i$ 。可以认为， $V_i$  为  $k_i$  对应的  $W_i$  的变换形式。由于  $V_i$  的各个分量都是从  $k_i$  到 Grossberg 层各神经元的联接权，所以此算法被称为外星(Outstar)法。

因此，如果将 CPN 用于模式的完善，则此时有  $n = m$ ，网络接受含有噪音的输入模式  $X = (x_1, x_2, \dots, x_n)$ ，而输出去掉噪音后的模式为  $(v_{i1}, v_{i2}, \dots, v_{im})$ 。

综上所述，对一个输入向量  $X$ ，CPN 首先检查它与  $W_1, W_2, \dots, W_h$  中的哪一个最为接近。最接近的  $W_j$  对应的神经元  $k_i$  被激发。一方面，这说明 CPN 将样本空间中的输入向量分成了  $h$  类，而且它们的代表分别是  $W_1, W_2, \dots, W_h$ 。所以，当面临的问题比较复杂，需要将输入向量分为更多的类的时候，可以适当地增加 Kohonen 层神经元的个数。另一方面，CPN 的分类实际上是在寻找与输入向量  $X$  最接近的  $W_i$ 。因此，它们的初始化是非常重要的。这不仅会影响到训练的速度，而且还会直接地影响到网络的精度。在 Kohonen 层完成模式的分类之后，对 Grossberg 层而言，它就是以适当的方式输出被激发  $k_i$  的所对应的向量  $V_i$ 。因此， $V_i$  实际上就是  $W_i$  的一个变换。所以，在训练过程中，对  $W_1, W_2, \dots, W_h$ ，将努力使它们代表各类  $X$  的共同特征。例如，可以将这一类  $X$  的平均值作为相应的  $W_i$  的训练目标；对  $v_1, v_2, \dots, v_h$ ，则是让它们对应地去表现  $W_1, W_2, \dots, W_h$  所代表的样本集中的  $X$  对应的理想输出  $Y$  的共同特征。

### 3. Kohonen 层的训练

前面介绍过，Kohonen 层按照输入向量  $X$  的相似度完成对它们的分类。分类的结果由 Grossberg 层转换成要求的输出形式。Kohonen 层使用自组织算法，按照无导师方式学习；因此，对一个给定的输入向量，Kohonen 层的哪一个神经元被激发，人们事先是不知道的，算法只能保证将不相似的向量区分开来，而使相似的向量能激发 Kohonen 层的同一个神经元。为了较好地实现这一思想，算法强烈希望将输入向量、以及 Kohonen 层各神经元对应的 Kohonen 层的权向量  $W_1, W_2, \dots, W_h$  进行规格化处理，使它们均为单位向量，以保证竞争

的“公平性”。

输入向量及权向量的预处理

输入向量的预处理就是对输入向量进行单位化处理。它是比较简单的数学运算，只要将向量的每一个分量除以该向量的模就可以了。得到单位向量  $\hat{X}$  和  $\hat{W}_j$ ：

$$\hat{X} = \frac{X}{\|X\|}, \quad \hat{W}_j = \frac{W_j}{\|W_j\|} \quad (8-13)$$

训练

为了使  $W_1, W_2, \dots, W_h$  能较好地代表网络所给出的样本集中的输入向量的类划分，对一个给定的输入向量  $X$ ，首先必须在  $W_1, W_2, \dots, W_h$  中找出目前谁最接近  $X$ 。Kohonen 层中的神经元的网络输入为  $X$  与它对应的权向量的点积，而点积最大者  $W_i$  拥有与  $X$  最大的相似度。所以，应该用它来代表  $X$  所在的类。而为了使它更好地代表  $X$ ，需进一步调整该  $W_i$ ，使它变的接近  $X$ ，以便更好地代表  $X$ 。但是，这种调整幅度不能太大，必须是有限度的，否则，算法将会出现强烈的抖动。 $W_i$  调整算法为：

$$W_i(t) = W_i(t-1) + \alpha(X - W_i(t-1)) \quad (8-14)$$

然后再对  $W_i(t)$  进行单位化处理。

由式 (8-14) 得到的  $W_i(t)$  变得比  $W_i(t-1)$  更接近  $X$ 。其中， $W_i(t)$  表示  $W_i$  在本次被调整后的值， $W_i(t-1)$  表示  $W_i$  在本次调整之前的值。 $\alpha$  为学习率： $\alpha \in (0,1)$ ，可以证明，使用式 (8-14) 对  $W_i$  进行变换后，确实会使得  $W_i(t)$  变得比  $W_i(t-1)$  更接近  $X$ 。证明如下：

$$W_i(t) = W_i(t-1) + \alpha(X - W_i(t-1)) = W_i(t-1) + \alpha X - \alpha W_i(t-1) \Rightarrow$$

$$X - W_i(t) = X - [W_i(t-1) + \alpha X - \alpha W_i(t-1)]$$

$$= X - W_i(t-1) - \alpha X + \alpha W_i(t-1) = (1-\alpha)X - W_i(t-1)(1-\alpha)$$

$$= (1-\alpha)(X - W_i(t-1)) \Rightarrow$$

$$\frac{X - W_i(t)}{X - W_i(t-1)} = \frac{(1-\alpha)(X - W_i(t-1))}{X - W_i(t-1)}$$

$$= 1 - \alpha$$

由于  $(1-\alpha) < 1$ ，所以， $W_i(t)$  变得比  $W_i(t-1)$  更接近  $X$ 。

图 8—2 为式 8 - 14 的几何表示，从图中可以看出， $W_i(t)$  确实变得比  $W_i(t-1)$  更接近  $X$ 。图中的圆为单位圆。所以，从原点到该圆上任一点的向量都是单位向量。

#### 4. Grossberg 层的训练

相对于 Kohonen 层的训练而言，Grossberg 层的训练要容易许多。在这里算法将依据 Kohonen 从被调整的权向量对应的神经元，来调整它所对应的 Grossberg 层的权向量。所以，与前面介绍的 Kohonen 层训练算法的一些调整相对应，这里介绍的 Grossberg 层的训练也应该随之有相应的调整。

输入一样本对  $(X, Y)$ ，计算净输入  $net = WX^T$ ，其中输入层到隐层的权值矩阵保持第一阶段的训练结果。

决定竞争获胜神经元

当  $X$  被输入到 CPN 中后，由  $net_i = \max(WX^T)$  决定 Kohonen 层的神经元  $k_i$  被激发，

此时  $k_i$  的输出为 1。其他的神经元输出为 0，用下式对外星权向量  $v_{ij}$  进行调整：

$$v_{ij}(t) = v_{ij}(t-1) + \beta(y_j - v_{ij}(t-1)) \quad (1 \leq j \leq m) \quad (8 - 15)$$

其中  $v_{ij}$  为 Kohonen 层的神经元  $k_i$  到 Grossberg 层的神经元  $G_j$  的联接权， $\beta$  为外星规则学习率，也是随时间下降的退火函数。写成向量形式，有

$$V_i(t) = V_i(t-1) + \beta(Y - V_i(t-1)) \quad (8 - 16)$$

为了方便，将式 8 - 14 重新写在下面：

$$W_i(t) = W_i(t-1) + \alpha(X - W_i(t-1)) \quad (8 - 17)$$

比较这两个式子，不难看出，Grossberg 层的训练与 Kohonen 层的训练是类似的：与 Kohonen 层的训练是将  $W_i$  训练成为这一类  $X$  的平均值相对应，Grossberg 层的训练则是将对应的  $X_i$  训练成为这一类  $X$  对应的  $Y$  平均值。

由此可见，Kohonen 层的训练可以说是以  $X$  为目标，而 Grossberg 层的训练则是以  $Y$  为目标的，这就是说，从单级网的角度来看，这两层的训练都是有导师训练。但是，对整个网络来说， $X$  是输入向量， $Y$  是输出向量。所以，又可以认为前者是无导师训练，后者是有导师训练。由以上规则可见，只有获胜神经元的外星向量得到调整，调整的目的是使外星权向量不断地靠近并等于期望输出，从而将该输出编码到外星权向量中。

重复步骤 至 直到学习率  $\beta$  下降为 0。

上面我们分别叙述了 CPN 的 Kohonen 层的训练和 Grossberg 层的训练。但实际上，CPN 这两层的训练既可以同时进行，也可以分开进行。当分开进行时，网络训练的实现算法可以被设计得效率要高一些，但它们执行的原理算法都是一样的。

#### 5. 几点讨论

### 规一化处理

算法在对  $W_i$  进行调整后, 要求对  $W_i(t)$  进行规范化处理, 这是为了保证后续的训练仍然能以相同的方式进行。当然, 包括对输入向量  $X$  的单位化处理在内, 都要求算法付出新的代价。但是, 对  $X, W$  进行了单位化处理之后, 会使本层的训练更有效地进行。因此, 为  $X, W$  的单位化所付出的代价是必须的。

在图 8—2 中, 被进行过单位化处理的  $W_i(t)$  将在未被单位化处理的  $W_i(t-1)$  的基础上, 继续延长到单位圆上, 也就是说, 它的方向不变, 只是长度延长。从上述讨论不难看出, 算法的第一步要求对  $W$  进行初始化处理是很有必要的。

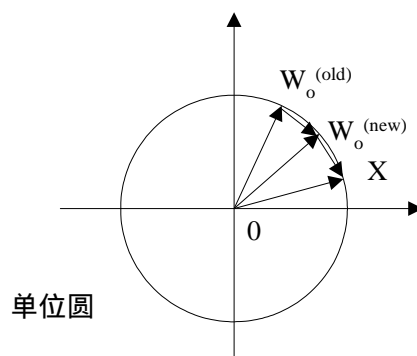


图 8—2 式 8—14 的几何意义

### 学习率的选择

学习率  $\alpha$  ( $\beta$ ) 在 0 和 1 之间取值。实践经验告诉我们, 在训练的初期,  $\alpha$  一般取  $0.5 < \alpha < 1$ , 多数情况下取  $\alpha = 0.7$  左右, 它将随着训练的进展不断变小。这是因为, 算法的运行表明, 到了后期,  $W_i$  已经对应一组  $X$ , 而不是一个。所以, 在这个阶段, 不能允许一个  $X$  对  $W_i$  的影响太大。因为, 此时如果再让  $X$  对  $W_i$  的影响过大, 它会过多地破坏  $W_i$  对该类中其他输入向量的表达, 等这些向量在下次迭代中重新出现时, 它又会对  $W_i$  进行较大的调整, 这样一来, 训练就会陷入抖动。当然, 在训练的初期, 由于各神经元正处于待选状态, 所以当遇到一个合适的输入向量时, 就尽量地去表示它。如果在开始训练时取  $\alpha$  为 1, 就相当于让  $W_i$  直接取  $X$ , 当样本顺序比较合适, 分布也比较恰当时, 这种做法是比较好的。

但是, 在大多数情况下, 它会导致  $W_i$  进入一个不太合适的“类”的表示中。也产造成分类的“人为”畸形, 使得有的类非常小, 有的类又太大, 甚至导致有的  $X$  被分到错误的类中。因此, 除非情况明显, 建议读者一般不要取  $\alpha$  为 1。

实际上, 如果在训练之前能用其他适当的方法事先给问题一个粗略的分类, 并从这个分类中提取一个较有代表性的向量构成样本集, 这时, Kohonen 层的每个  $W_i$  (对应一个神经元) 的值直接对应一个  $X$ , 则可取为 1, 使得获胜者的权向量直接取  $X$ 。这种做法又启发我

们采用训练和直接设定权向量的方式来完成该层的训练。即在进行初始化时,直接用初选的样本集中的样本来对  $W$  进行初始化,然后再适当地扩大样本集,用扩大后的样本集实施上述算法,完成对 Kohonen 层的训练。

一般来说,一个类含有许多向量。这个类对应的  $W_i$  应该是样本集中这一类向量(输入向量部分)的平均值。

### 8.2.3 改进的 CPN 网络运行原理<sup>[3]</sup>

改进 CPN 网络有各种形式,典型的有:双获胜神经元 CPN 神经网络和双向 CPN 神经网络。前者针对标准对偶传播网络中,竞争层上只允许有一个神经元获胜,作为一种改进,在完成训练后的运行期间允许隐层有两个神经元同时获胜,这两个获胜神经元均取值为 1,其他神经元取值为 0,于是两个神经元同时按式 8-16 影响网络地输出;后者是将 CPN 网络地输入层和输出层分为两组。这时网络结构变换为图 8-3 所示。该网络有两个输入向量  $X$

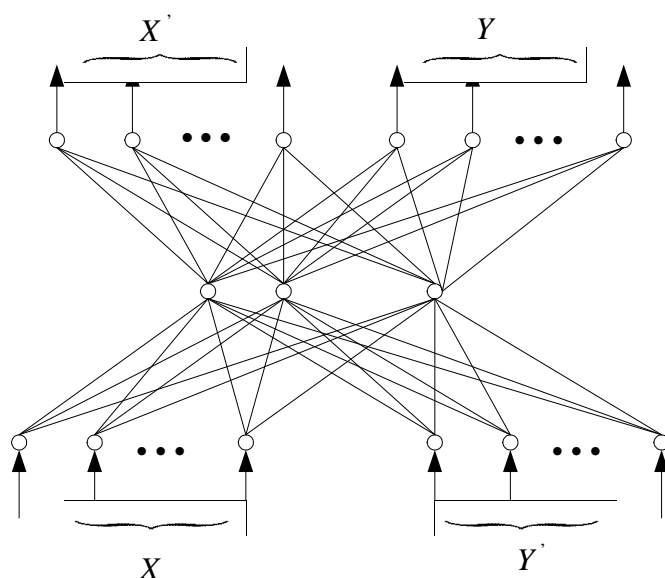


图 8-3 双向 CPN 神经网络拓扑结构

和  $Y'$ , 两个与之对应地输出向量  $Y$  和  $X'$ 。训练隐层内星权向量时,将两个输入向量作为一个输入向量处理;训练隐层外星权向量时,将两个输出向量作为一个输出向量来处理。两种权向量地调整规则与标准 CPN 网络完全相同。

双向 CPN 网络的优点是可以同时学习两个函数。如:

$$\begin{aligned} Y &= f(X) \\ X' &= f(Y') \end{aligned} \quad (8-18)$$

当向网络输入  $(X, 0)$  时,网络输出为  $(Y, 0)$ ;当向网络输入  $(0, Y')$  时,网络输出为  $(0, X')$ ;当向网络输入  $(X, Y')$  时,网络输出为  $(Y, X')$ 。

当两个函数互逆时,有  $X = X', Y = Y'$ 。双向 CPN 神经网络可以用于数据压缩与解压缩,可将其中一个函数  $f$  用作压缩函数,将其逆函数  $f^{-1}$  用作解压缩函数。应用时,双向 CPN

神经网络并不要求两个互逆函数是解析表达地，更一般地情况下两函数是互逆的映射关系，从而可利用双向 CPN 神经网络实现互联想。

### 8.3 CPN 神经网络的应用

#### ——CPN 网络在集成电路故障模式识别中的应用<sup>[4]</sup>。

在模拟电路故障诊断中，当电路中某元件出现故障后，不仅其本身输出信号失真，而且影响与之相连的其它正常元件的功能，也使它们的相关信号失真，也就是说电路前后元件相互影响，如果直接测试待诊断元件的电压或电流信号，难以准确判断是否有故障。常规方法是线路切割法<sup>[5]</sup>（也称网络撕裂法），对怀疑元件一般是割断其前后联系，通电测试其关键点电压，电流信号，判断是否故障，但这种方法不仅测试麻烦，而且由于无法猜准哪个元件故障，必须切割很多元件才能诊断出真正的故障元件。而在不少情况下是不允许进行这种破坏性诊断，特别是一些重要仪器电路或正在运行的机电设备故障搜寻，如文献[6]光电雷达电子部件故障诊断系统就不允许使用这种切割诊断法。在这方面，多传感器信息融合技术提供了一条崭新的途径，这是由信息融合独特的多维信息处理方式决定的。此处将基于 CPN 神经网络信息融合技术引入电子电路的故障搜寻之中，通过测试电子电路工作时电子元件的温度和关键点电压两方面的数据信息，利用 CPN 网络进行多信息融合，从而准确搜寻出故障元件。

#### 8.3.1 电路故障诊断的 CPN 神经网络信息融合方法

神经网络是仿效生物体信息处理系统获得柔性信息处理能力。它是从微观上模拟人脑功能，是一种分布式的微观数值模型，神经元网络通过大量经验样本的学习，将专家知识和诊断实例以权值和阈值的形式分布在网络的内部，并且利用神经网络的信息保持性来完成不确定性推理。更重要的是，神经网络有极强的自学习能力，对于新的故障模式和故障样本可以通过权值的改变进行学习、记忆和存储，进而在以后的运行中能够判断这些新的故障模式。神经网络作为一种新的智能化问题求解模型，它为故障诊断系统的建造提供了一种新的框架。

CPN 神经网络信息融合，是将 CPN 神经网络引入信息融合之中，同时结合模糊集合论进行故障判断。其具体过程为：通过多传感器测试被诊断对象，求出每一传感器对故障集中各类故障的隶属度值，将所有传感器的故障隶属度值矢量作为神经网络的输入，网络输出即为融合后该症状属于各类故障的隶属度值矢量，最后利用基于规则的判定原则进行故障决策，其融合诊断过程如图 8-4 所示。其中  $A_1, A_2, \dots, A_n$  为待诊断的故障模式。 $\mu_j(A_1) \dots \mu_j(A_n)$  为传感器  $j$  测得的各故障模式  $A_1, A_2, \dots, A_n$  所得的隶属度值， $\mu(A_1) \dots \mu(A_n)$  为融合的隶属度值。

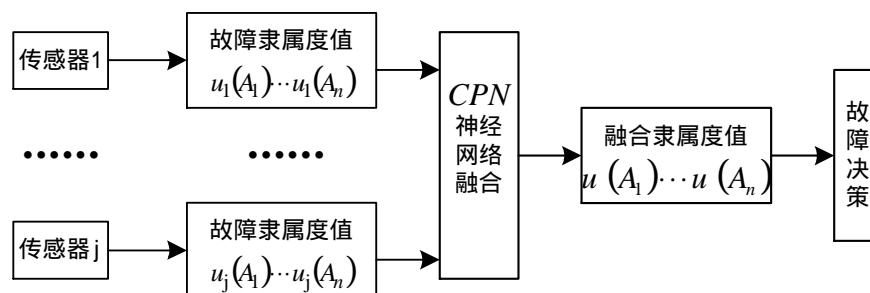


图 8-4 CPN 神经网络信息融合故障诊断

#### 1. 隶属度函数的形式

隶属度函数主要由传感器本身的工作特性及被测参数的特性而定，对电子电路系统某一特定元器件，当系统正常工作时，其关键点电压值应是稳定的，在环境温度一定时，其

芯片温度值也是一稳定数值，当元器件出现故障时，一般地说其电压值会偏离正常范围，温度信号也会发生变化（无论是升高还是降低），偏差越大相对来说元器件出现故障的可能性会变大，为讨论方便，同时结合文献[7]的隶属度函数待定系数法及具体的实验结果，

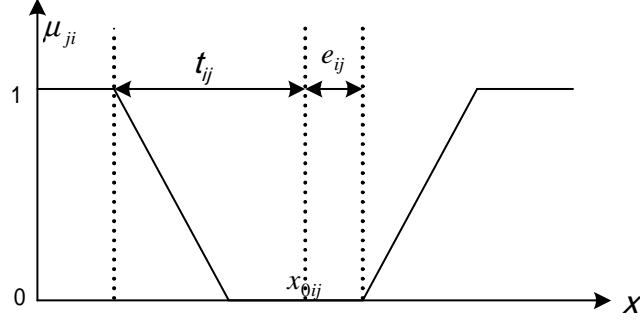


图 8-5 隶属函数分布

定义隶属度函数  $\mu_{ij}$  的分布如图 8-5 所示。其表达式如下：

$$\mu_{ij} = \begin{cases} 1 & x_i \leq x_{0ij} - t_{ij} \\ -\alpha' (x_i - x_{0ij} + e_{ij}) / (t_{ij} - e_{ij}) & x_{0ij} - t_{ij} < x_i \leq x_{0ij} - e_{ij} \\ 0 & x_{0ij} - e_{ij} < x_i \leq x_{0ij} + e_{ij} \\ \alpha' (x_i - x_{0ij} - e_{ij}) / (t_{ij} - e_{ij}) & x_{0ij} + e_{ij} < x_i \leq x_{0ij} + t_{ij} \\ 1 & x_i > x_{0ij} + t_{ij} \end{cases} \quad (8-19)$$

其中  $x_{0ij}$  为电路工作正常时被测元件的标准参数值； $e_{ij}$  为待诊断元件参数的正常变化范围，即容差； $t_{ij}$  为待诊断元件参数的极限偏差， $\alpha'$  为修正系数； $\mu_{ij}$  为传感器  $i$  测定被诊断

元件  $j$  属于故障的隶属度， $x_i$  为传感器  $i$  测定的实际数值。

## 2. CPN 神经网络模型

从表面上看，它和前面介绍的三层 BP 网络是相同的，但是在运行过程中，它们实现的策略却是不同的。图 8-1 为它的拓扑结构，为方便起见将其绘于图 8-6。

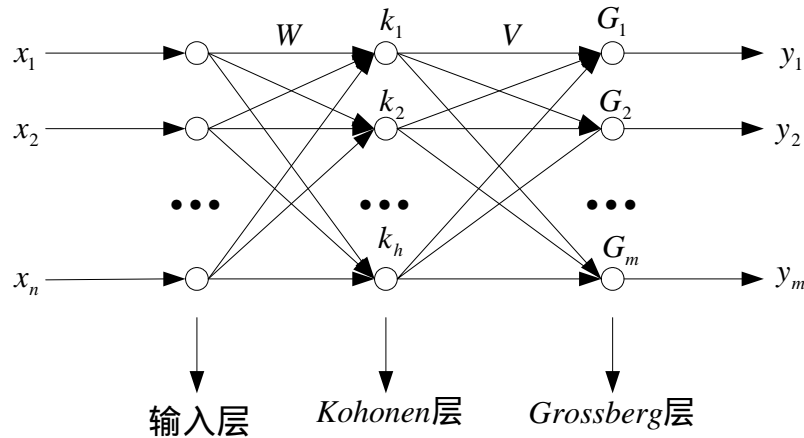


图 8-6 CPN 拓扑结构

第 1 层(输入层)的神经元只起到将输入向量传送到 Kohonen 层的作用，此处输入向量与



两种传感器测试的故障隶属度数值相关，神经元个数  $n = \text{传感器个数} \times \text{待诊断元件个数}$ ；在这里传感器个数为 2，待诊断元件数为 5，故  $n = 10$ 。输入向量的每一个分量可以被按照一定的联接权传递到第 2 层(Kohonen 层)的每个神经元。Kohonen 层的联接权矩阵用  $W$  表示，按照惯例， $w_{ij}$  表示输入向量的第  $i$  个分量到 Kohonen 层的第  $j$  个神经元的联接权，该层神经元个数为 12。第 3 层叫做 Grossberg 层，Kohonen 层的第  $i$  个神经元到 Grossberg 层的第  $j$  个神经元的联接权用  $v_{ij}$  表示，它们放在一起构成 Grossberg 层的权矩阵，Grossberg 层的输出与故障元件直接相关，其神经元个数  $m = \text{待诊断元件数}$ 。在这里  $m = 5$ 。相关训练算法前面已经介绍。

### 3.故障决策：

对网络融合后输出的各待诊断模式属于故障的隶属度值，在故障决策时，采用基于规则的方法，其基本原则有：

判定的故障模式应具有最大的隶属度值。

判定的故障模式的隶属度值要大于某一阈值。此阈值过小，易出现故障误判现象，而此阈值过大时又易出现故障漏判。一般来说此值至少要大于  $1/m$  ( $m$  为待诊断故障模式个数)，阈值越大故障判断的准确性越高，只是取得过大时测试得到的故障隶属度值无法满足要求，因此必须针对实际被诊断设备，根据实际诊断结果准确率的数据来选择一大于  $1/m$  的适当数值。此处为 0.7。

判定的目标模式和其它模式的隶属度值之差要大于某个门限。此门限的取值类似于 。此处为 0.50。

## 8.3.2 基于 CPN 神经网络信息融合的光电雷达电路故障识别<sup>[6]</sup>

### 1. 诊断处理

图 8-7 为实验用待诊断的光电雷达压-码转换电路板实际电路图片，它的主要功能是采用逐次比较的原理，将一模拟电压转换为 11 位数字信号。根据实际情况分析，其中有 5 个待诊断集成元件分别为元件  $A_1, A_2, A_3, A_4, A_5$ 。对应的电压测试点  $U_1, U_2, U_3, U_4, U_5$ 。下面进行实验测试诊断，并与单传感器测试诊断结果进行比较。

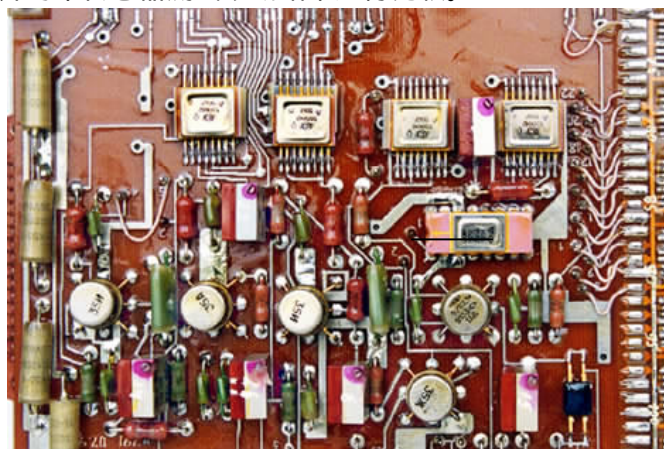


图8-7 光电雷达压-码转换电路板

首先利用探针测出各待诊断元件关键点的电压信号，利用热像仪 (Inframetrics 600) 测试出电路板待诊断元件的工作温度信号，对每一传感器来讲，被测元件属于故障的可能性可分别用一组隶属度值来表示，这样会得到两组共 10 个隶属度值。由于电路中前后元件的相互影响，同一传感器测得的不同元件的隶属度有的相互接近，如用一种传感器判别故障元件，往往会出现误判，解决的办法是：CPN 神经网络对两组隶属度值进行融合处理，得到两传感器融合后各待诊断元件属于故障的隶属度值，再根据一定的判定准则进行故障元件的判定。表 8 - 1 为模拟各个元件故障时，测试得到的电压，温度隶属度数值，为 CPN 网络的训

练样本。在此，为处理问题的方便结合实际测试数据取电压传感器  $\alpha' = 1/3$ ，温度传感器  $\alpha' = 2.8$ ； $e_{ij} = 0$ ， $t_{ij} = x_{0ij}$ ，学习率线性下降。

表 8 - 1 CPN 神经网络模式样本

序号	传感器	输入隶属度					输出值				
		$A_1$ ,	$A_2$ ,	$A_3$ ,	$A_4$ ,	$A_5$	$A_1$ ,	$A_2$ ,	$A_3$ ,	$A_4$ ,	$A_5$
1	电压	.2746	.0313	.3511	.0708	.2722	1	0	0	0	0
	温度	.5608	.2143	.0454	.0891	.0902					
2	电压	.0144	.5152	.2656	.0108	.1939	0	1	0	0	0
	温度	.1000	.6185	.1905	.0000	.0921					
3	电压	.0113	.0326	.6900	.2158	.0505	0	0	1	0	0
	温度	.0852	.0760	.6314	.1395	.0679					
4	电压	.0393	.3132	.2212	.3875	.0387	0	0	0	1	0
	温度	.1039	.0000	.0000	.7957	.1004					
5	电压	.0101	.2838	.2486	.0069	.4506	0	0	0	0	1
	温度	.1143	.0892	.0616	.1779	.5569					
6	电压	.3272	.0613	.2775	.0747	.2593	1	0	0	0	0
	温度	.6169	.1489	.0489	.0966	.0887					
7	电压	.0168	.5029	.2698	.0139	.1964	0	1	0	0	0
	温度	.0050	.7004	.1893	.0045	.1008					
8	电压	.0112	.0350	.7055	.1160	.1570	0	0	1	0	0
	温度	.0882	.0761	.6382	.1228	.0748					
9	电压	.0466	.3047	.2121	.3964	.0402	0	0	0	1	0
	温度	.1011	.0055	.0114	.7826	.0994					
10	电压	.0078	.2761	.2444	.0067	.4641	0	0	0	0	1
	温度	.0996	.0853	.0572	.1762	.5816					

## 2. 融合诊断结果讨论

表 8 - 2 融合诊断结果

故障 元件	传感器	故障隶属度值					诊断 结果
		$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	
$A_1$	电压	.3390	.0513	.3559	.0533	.2011	不定
	温度	.5677	.1378	.0579	.0971	.1394	不定
	融合	.7095	.0266	.1217	.0410	.1141	$A_1$ 故障
$A_2$	电压	.0718	.5012	.2138	.0196	.1936	不定
	温度	.0091	.6822	.1804	.1032	.0251	不定
	融合	.0340	.8021	.0508	.0334	.0813	$A_2$ 故障
$A_3$	电压	.0113	.0326	.6900	.2158	.0505	不定
	温度	.0852	.0760	.6314	.1395	.0679	不定
	融合	.0623	.0122	.7881	.0331	.1045	$A_3$ 故障
$A_4$	电压	.0393	.3132	.2212	.3875	.0387	不定
	温度	.1039	.0000	.0000	.7957	.1004	$A_4$ 故障
	融合	.0132	.0231	.0491	.8532	.0616	$A_4$ 故障

A <sub>5</sub>	电压	.0718	.2012	.2138	.0796	.4336	不定
	温度	.0901	.0782	.0804	.1932	.5581	不定
	融合	.0421	.0520	.1278	.0501	.7315	A <sub>5</sub> 故障

表 8-2 为利用热像仪和探针测试出的隶属度值分配表及单传感器和双传感器 CPN 神经网络融合故障识别结果。对表中每一种故障元件来说，其中第一行，第二行为从电压和温度所测得的各元件属于故障的隶属度值，第三行为融合后的隶属度值。从表中可以明显看出，在本实验电路被诊断的三种故障元件中，两种传感器所提取的故障隶属度值有的很相近，如果只用一种传感器的隶属度值来识别故障元件，会出现无法确定哪个元件故障的状况。如元件 1 故障时，两种传感器所提取的故障隶属度值十分接近，单独识别时无法判定故障元件，但融合后元件 1 的隶属度值大为提高，且与另两种元件的隶属度值相差很大，此时就能很准确地识别出故障元件。也就是说，融合后的隶属度值和单传感器的隶属度值相比，增加了实际故障元件的隶属度分配值，相对减少了其它元件的隶属度分配值，这必使待诊断对象的不确定性大幅度降低，消除了由于单传感器提供信息量少而产生的误诊断现象，在本例这种人为设置的简单电路故障诊断中，故障识别的准确率达到了 100%。

由此可见，基于 CPN 网络和模糊数学的多传感器信息融合故障诊断方法，使实际故障元件的隶属度值大为增加，待诊断对象的可分析性增强，故障元件定位的准确率大为提高。将 CPN 神经网络应用于信息融合模式识别是成功的。

## 本章小结

将执行自组织映射的 Kohonen 层和执行外星算法的 Grossberg 层相结合，得到的与 BP 神经网络结构相类似的 CPN 网络。不仅提供了一种设计多级网训练算法的思路，解决了多级网络的训练问题，突破了单极网的限制，而且还使得网络具有了许多新的特点。可以用作联想存储、模式分类、函数逼近和数据压缩等用途。而且训练速度远比 BP 网络快。

本章在讨论简单 CPN 神经网络拓扑结构及训练算法的基础上，进一步介绍了双向 CPN 神经网络的结构和工作原理。最后介绍了 CPN 神经网络在模拟电路故障识别中的具体应用。

## 参考文献：

- [1] Robert Hecht-Nielsen, Neurocomputing. Reading: Addison-Wesley Publishing Company, 1990.
- [2] 蒋宗礼, 人工神经网络导论, 高等教育出版社, 2001.8, 北京。
- [3] 韩力群, 人工神经网络的理论、设计及应用, 北京, 化学工业出版社, 2002.1。
- [4] Daqi Zhu, Yongqing Yang, Blind Fault Diagnosis Algorithm for Integrated Circuit Based on the CPN Neural Networks, Lecture Notes in Computer Science, 2004, 3174: 619 - 626, Springer-Verlag.
- [5] Chen Y Q., Experiment on fault location in large-scale analogue circuits, IEEE Trans. On IM. 1993,42(1):30-34
- [6] 朱大奇, 刘文波, 于盛林, 基于虚拟仪器的光电雷达电子部件性能检测及故障诊断系统。航空学报, 2001, 22 ( 5 ): 468-471。
- [7] 王浩, 庄钊文, 模糊可靠性分析中的隶属函数确定, 电子产品可靠性与环境试验, 2000, 10 ( 4 ): 2-7。
- [8] 朱大奇, 电子设备故障诊断的原理与实践, 电子工业出版社, 2004.1。

## 思考题

1. 请介绍 CPN 神经网络的基本构造及工作原理。

2. 请介绍 CPN 神经网络 Kohonen 层权值训练算法。
3. 请介绍 CPN 神经网络 Grossberg 层权值训练算法。
4. 试设计一个 CPN 神经网络实现将 4 维输入模式映射为 3 维输出模式。两个输入模式分别为： $X_1 = (1 \quad -1 \quad 1 \quad 1)$  和  $X_2 = (1 \quad 1 \quad -1 \quad -1)$ ，对应的输出模式分别为： $Y_1 = (1 \quad 1 \quad 0)$  和  $Y_2 = (1 \quad 0 \quad -1)$ 。请给出训练后的权值矩阵  $W$  和  $V$ 。
5. 给出一个 CPN 神经网络应用实例。

## 第 9 章 ART 自适应谐振理论

众所周知,在神经网络学习中,当网络规模给定后,由权矩阵所能记忆的模式类别信息量总是有限的,新输入的模式样本必然会对已经记忆的模式样本产生抵消或遗忘,从而使网络的分类能力受到影响。靠无限扩大网络规模解决上述问题是不现实的。

如何保证在适当增加网络规模的同时,在过去记忆的模式和新输入的训练模式之间作出某种折中,既能最大限度地接收新的模式信息,同时又能保证较少地影响过去的样本模式呢? ART 网络在一定程度上能较好解决此问题。

1976 年,美国 Boston 大学 Carpenter G.A.教授提出了自适应共振理论 ART(Adaptive Resonance Theory)。随后 Carpenter G.A.又与他的学生 Grossberg S.合作提出了 ART 神经网络<sup>[1-3]</sup>。经过多年的研究和发展,ART 网络已有几种基本形式:

ART1 型神经网络:处理双极性和二进制信号;

ART2 型神经网络<sup>[4]</sup>:它是 ART1 型的扩展,用于处理连续型模拟信号;

ART 综合系统<sup>[8]</sup>:将 ART1 和 ART2 综合在一起,系统具有识别、补充和撤消等综合功能。即所谓的 3R (Recognition、Reinforcement、Recall) 功能。

ART3 型神经网络<sup>[5]</sup>:它是一种分级搜索模型,兼容前两种结构的功能并将两层神经网络扩大为任意多层神经网络,由于 ART3 型在神经元的模型中纳入了生物神经元的生物电-化学反应机制,因而它具备了很强的功能和扩展能力。

本章主要介绍前两种 ART 神经网络和 ART 综合系统。

### 9.1 ART1 型神经网络结构及算法<sup>[6-7]</sup>

ART1 型神经网络和其他两种网络相比,其结构和算法都比较简单。

#### 9.1.1 ART1 型网络结构

图 9-1 (a)为 ART1 型网络结构示意图,它由两层神经元构成两个子系统,分别称为比较层 C (也称注意子系统)和识别层 R (也称取向子系统)。此外还有三个控制信号:复位信号 Reset;逻辑控制信号  $G_1$  和  $G_2$ 。

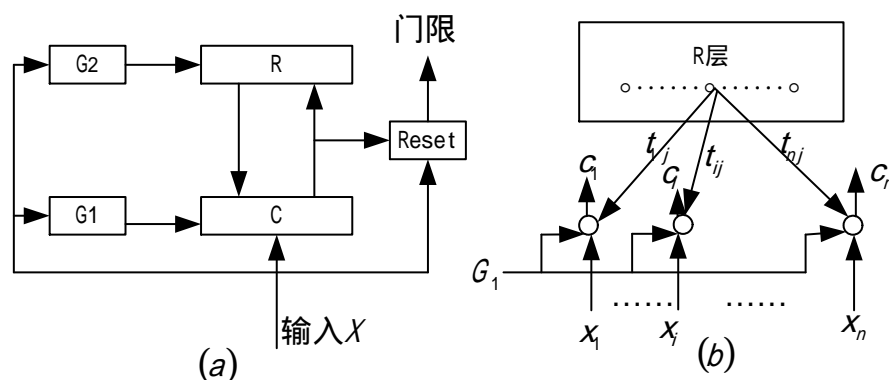


图 9-1 ART1 型网络结构图

#### 1. 比较层结构

$C$  层展开后的结构如图 9-1(b)所示,该层有  $n$  个节点,每个节点接受来自 3 个方面的信号:一个来自外界的输入信号  $x_i$ ,另一个来自  $R$  层获胜神经元的外星权向量  $T_{j^*}$  的返回信

号  $t_{ij}^*$ ，还有一个来自  $G_1$  的控制信号。 $C$  层节点的输出是根据 2/3 的“多数表决”原则产生的，即输出值  $c_i$  与  $x_i$ 、 $t_{ij}$ 、 $G_1$  3 个信号中的多数信号值相同。

网络开始运行时， $G_1=1$ ，识别层尚未产生竞争获胜神经元，因此反馈回送信号为 0。由 2/3 规则知， $C$  层输出由输入信号决定，有  $C = X$ 。当网络识别层出现反馈回送信号时， $G_1=0$ ，由 2/3 规则， $C$  层输出应取决于输入信号与反馈信号的比较情况，如果  $x_i = t_{ij}^*$ ，则  $c_i = x_i$ 。否则  $c_i = 0$ 。可以看出，控制信号  $G_1$  的作用是使比较层能够区分网络运行的不同阶段，网络开始运行阶段  $G_1$  的作用是使  $C$  层对输入信号直接输出；之后  $G_1$  的作用是使  $C$  层行使比较功能，此时  $c_i$  为对  $x_i$  和  $t_{ij}^*$  的比较信号，两者为 1 时  $c_i$  为 1，否则为 0，可以看出，从  $R$  层返回的信号  $t_{ij}^*$  对  $C$  层输出有调节作用。

## 2. 识别层结构

$R$  层展开后的结构如图 9-2 所示，其功能相当于一种前馈竞争网。设  $R$  层有  $m$  个节点，用以表示  $m$  个输入模式类。 $m$  可动态增长，以设立新模式类。由  $C$  层向上连接到  $R$  第  $j$  个节点的内星权向量用  $B_j = (b_{1j}, b_{2j}, \dots, b_{nj})$  表示。 $C$  层的输出向量  $C$  沿  $m$  个内星权向量  $B_j$  ( $j = 1, 2, \dots, m$ ) 向前传送，到达  $R$  层各个神经元节点后经过竞争产生获胜节点  $j^*$ ，指示本次输入模式的所属类别。获胜节点输出  $r_{j^*} = 1$ ，其余节点输出 0。 $R$  层的每个神经元都对应两个权向量，一个是将  $C$  层前馈信号汇聚到  $R$  层的内星权向量  $B_j$ ，另一个是将  $R$  层前馈信号散发到  $C$  层的外星权向量  $T_j$ ，该向量为对应于  $R$  层各模式类节点的典型向量。

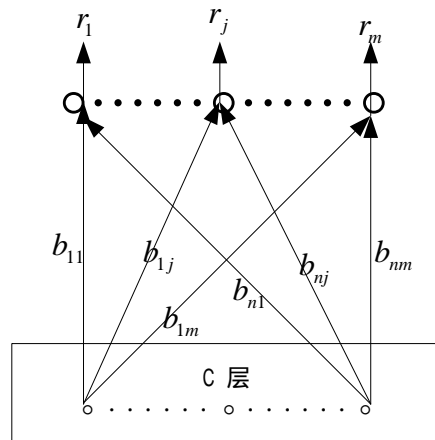


图 9-2 识别层结构示意图

### 3. 控制信号

3 个控制信号的作用分别是：信号  $G_2$  检测输入模式  $X$  是否为 0，它等于  $X$  各分量的逻辑“或”，如果  $x_i$  ( $i=1,2,\dots,n$ ) 为全 0，则  $G_2=0$ ，否则  $G_2=1$ 。设  $R$  层输出向量各分量的逻辑“或”用  $R_0$  表示，则信号  $G_1 = G_2 \bar{R}_0$ ，当  $R$  层输出向量的各分量为全 0 而输入向量  $X$  不是零向量时， $G_1=1$ ，否则  $G_1=0$ 。正如前面所指出的， $G_1$  的作用是在网络开始运行时为 1，以使  $C = X$ ，其后为 0 以使  $C$  值由输入模式和反馈模式的比较结果决定。Reset 信号的作用是使  $R$  层竞争获胜神经元无效，如果根据某种事先设定的测量标准， $T_{j^*}$  与  $X$  未达到预先设定的相似度  $\rho$ ，表明两者未充分接近，于是系统发出 Reset 信号使竞争获胜神经元无效。

#### 9.1.2 ART1 型网络工作原理

网络运行时接受来自环境的输入模式，并检测输入模式与  $R$  层所有模式类之间的匹配程度。对于匹配程度最高的模式类，网络要继续考察该模式的典型向量与当前输入模式的相似程度。相似程度按照预先设计的参考门限考察，可能出现的情况无非有下面两种：

如果相似程度超过参考门限，选该模式类作为当前输入模式的代表类。权值调整规则是，相似程度超过参考门限的模式类调整其相应的内外星权向量，以使其以后遇到与当前输入模式接近的样本时能得到更大的相似度；对其他权值向量则不做任何变动。

如果相似度不超过门限值，则对  $R$  层匹配程度次高的模式类进行相似程度考察，若超过参考门限网络的运行回到情况 1，否则仍然回到情况 2。可以想到，运行反复回到情况 2

意味着最终所有的模式类与当前输入模式的相似度都没有超过参考门限，此时需在网络输出端设立一个代表新模式类的节点，用以代表及存储该模式，以便于参加以后的匹配过程。网络对所接收的每个新输入样本，都进行上面的运行过程。对于每个输入模式，网络运行过程可归纳为四个阶段。

#### 1. 匹配阶段

网络在没有输入模式之前处于等待状态，此时输入端  $X=0$ ，因此信号  $G_2=0$ ， $R_0=0$ 。

当输入为不全为 0 的模式  $X$  时， $G_2=1$ ， $R_0=0$ ，使得  $G_1 = G_2 \bar{R}_0 = 1$ 。 $G_1$  为 1 时允许输入模式直接从  $C$  层输出，并向前传至  $R$  层，与  $R$  层节点对应的所有内星向量  $B_j$  进行匹配计算

$$net_j = B_j^T X = \sum_{i=1}^n b_{ij} x_i \quad j=1,2,\dots,m \quad (9-1)$$

选择具有最大匹配度（即具有最大点积）的竞争获胜节点： $net_{j^*} = \max_j \{net_j\}$ ，使获胜节点

点输出  $r_{j^*}=1$ ，其他节点输出为 0。

#### 2. 比较阶段

$R$  层输出信息通过外星向量返回到  $C$  层。 $r_{j^*}=1$ ，使  $R$  层获胜节点所连的外星向量  $T_{j^*}$

激活，从节点  $j^*$  发出的  $n$  个权值信号  $t_{ij^*}$  返回到  $C$  层的  $n$  个节点。此时， $R$  层输出不全为 0，

$R_0=1$  , 而  $G_1 = G_2 \bar{R}_0 = 1$  , 所以  $C$  层最新输出状态  $C'$  取决于由  $R$  层返回的外星向量  $T_{j^*}$  和网络输入模式  $X$  的比较结果, 即  $c_i = t_{ij^*} x_i$  ,  $i=1, 2, \dots, n$ 。由于外星向量  $T_{j^*}$  是  $R$  层模式类的典型向量, 该比较结果  $C'$  反映了在匹配阶段  $R$  层竞争排名第一的是模式类的典型向量  $T_{j^*}$  与当前输入模式  $X$  的相似程度。相似程度的大小用相似度  $N_0$  反映, 定义为

$$N_0 = X^T t_{j^*} = \sum_{i=1}^n t_{ij^*} x_i = \sum_{i=1}^n c_i \quad (9-2)$$

在 ART1 中, 因为输入  $x_i$  为二进制数 0 或 1,  $N_0$  实际上表示获胜节点的类别模式典型向量与输入模式样本之间相互重叠的非零分量数。设输入模式样本中的非零分量数为

$$N_1 = \sum_{i=1}^n x_i \quad (9-3)$$

用于比较的警戒门限为  $\rho$  ,  $\rho \in (0,1)$ 。检查输入模式与模式类典型向量之间的相似性是否低于警戒门限, 如果有:

$$\frac{N_0}{N_1} < \rho \quad (9-4)$$

则  $X$  与  $T_{j^*}$  的相似程度不满足要求, 网络发出 Reset 信号使第一阶段的匹配失败, 竞争获胜节点无效, 网络进入搜索阶段。如果有:

$$\frac{N_0}{N_1} \geq \rho \quad (9-5)$$

表明  $X$  与获胜节点对应的类别模式非常接近, 称  $X$  与  $T_{j^*}$  此时发生“共振”, 第一阶段的匹配结果有效, 网络进入学习阶段。

### 3. 搜索阶段

网络发出 Reset 重置信号后即进入搜索阶段, 重置信号的作用是使前面通过竞争获胜的神经元受到抑制, 并且在后续过程中受到持续的抑制, 直到输入一个新的模式为止。由于  $R$  层中的竞争获胜的神经元被抑制, 从而再度出现  $R_0=0$  ,  $G_1=1$  , 因此网络又重新回到起始的匹配状态。由于上次获胜的节点受到持续的抑制, 此次获胜的必然是上次匹配程度排在第二的节点。然后进入比较阶段, 将该节点对应的外星向量  $t_{j^*}$  与输入模式进行匹配计算。如果对  $R$  层所有的模式类, 在比较阶段的相似度检查中相似度都不能满足要求, 说明当前输入模式无类可归, 需要在网络输出层增加一个节点来代表并存储该模式类, 为此将其中内星向量  $B_{j^*}$  设计成当前输入模式向量, 外星向量  $T_{j^*}$  各分量全设为 1。以满足其相似度的要求。



#### 4. 学习阶段

在学习阶段要对发生“共振”的获胜节点对应的模式类加强学习，使以后出现与该模式相似的输入样本时能获得更大的“共振”。

权值学习调整是修改网络的内外星权向量。其外星向量的调整规则为：

$$t_{ij^*}(t+1) = t_{ij^*}(t)x_i (i=1,2,\dots,n; j^* \in J^*) ; \text{ 外星向量为对应模式类的典型向量或称聚类中心。上式可以保证相似的模式越来越相似，不同的模式越来越分离。其内星权向量的调整规则为：}$$

$$b_{ij^*}(t+1) = \frac{t_{ij^*}(t)x_i}{0.5 + \sum_{i=1}^n t_{ij^*}(t)x_i} = \frac{t_{ij^*}(t+1)}{0.5 + \sum_{i=1}^n t_{ij^*}(t+1)} (i=1,2,\dots,n)。$$

式如果不计分母中的常数 0.5，上式相当于对外星权向量进行归一化。

ART 网络运行中存在两种记忆方式， $C$  层和  $R$  层输出信号称为短期记忆，用 STM (Short time memory) 表示，短期记忆在运行过程中会不断发生变化；两层之间的内外星权向量称为长期记忆，用 LTM (Long time memory) 表示，长期记忆在运行过程中不会频繁变化。下面对两种记忆形式进行分析：

$C$  层输出信号是按照 2/3 原则取值的，在网络开始运行时， $C$  层输出与输入模式  $X$  相等，因此  $C$  层输出是对输入模式  $X$  的记忆。当  $R$  层返回信号  $T_{j^*}$  达到  $C$  层时，输出  $C$  立刻

失去对  $X$  的记忆而变成  $T_{j^*}$  和  $X$  的比较信号。 $R$  层输出信号是按照胜者为王原则取值的，

获胜神经元代表的模式类是输入模式的类别记忆。但当重置信号 Reset 作用于  $R$  层时，原获胜神经元无效，因此原记忆也消失。由此可见， $C$  和  $R$  对输入模式  $X$  的记忆时间非常短暂，因此称为短期记忆。

权向量  $T_{j^*}$  和  $B_{j^*}$  在运行过程中不会发生频繁变化，只在学习阶段进行调整以进一步加强记忆。经过学习后，对样本的记忆将留在两组权向量中，即使输入样本改变，权值依然存在，因此称为长期记忆。当以后输出的样本类似已经记忆的样本时，这两组长期记忆将  $R$  层输出回忆到记忆样本的状态。

## 9.2 ART2 型神经网络结构及算法<sup>[4,8]</sup>

### 9.2.1 ART2 型网络结构

ART2 神经网络不仅能对双极型或二进制输入模式分类，而且能够对模拟输入模式的任意序列进行自组织分类，其基本设计思路仍然是采用竞争学习策略和自稳机制。

ART2 结构如图 9-3 所示，图中给出第  $j$  个处理单元的拓扑连接。ART2 由注意子系统和取向子系统组成。注意子系统中包括短期记忆 STM 特征表示场  $F_1$  和短期记忆类别表示场

$F_2$ 。 $F_1$  相当于 ART1 中的比较层，包括几个处理级和增益控制系统。 $F_2$  相当于 ART1 的识别层，负责对当前输入模式进行竞争匹配。 $F_1$  和  $F_2$  共有  $N + M$  个神经元，其中  $F_1$  场有  $N$  个， $F_2$  场有  $M$  个，共同构成了  $M + N$  维状态向量，代表网络的短期记忆。 $F_1$  和  $F_2$  之间的内外星连接权向量构成了网络的自适应长期记忆 LTM，由下至上的权值用  $z_{ij}$  表示，由上

至下的权值用  $z_{ji}$  表示。取向子系统由图 9 - 4 左侧的复位系统组成。

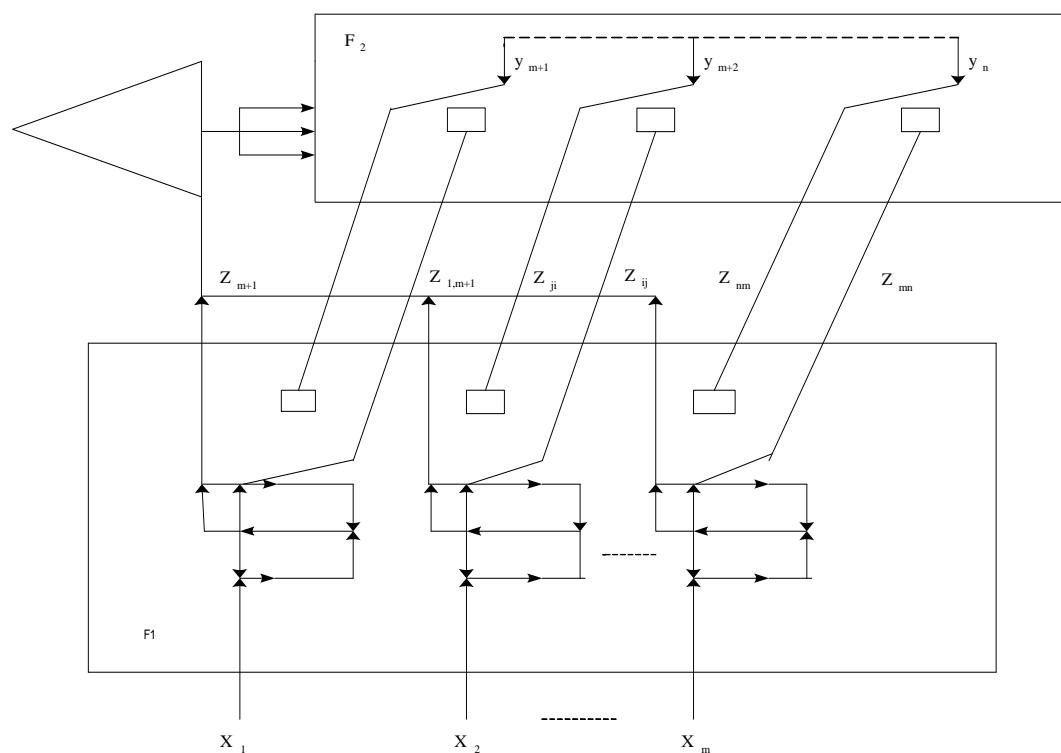


图 9-3 ART2 神经网络结构

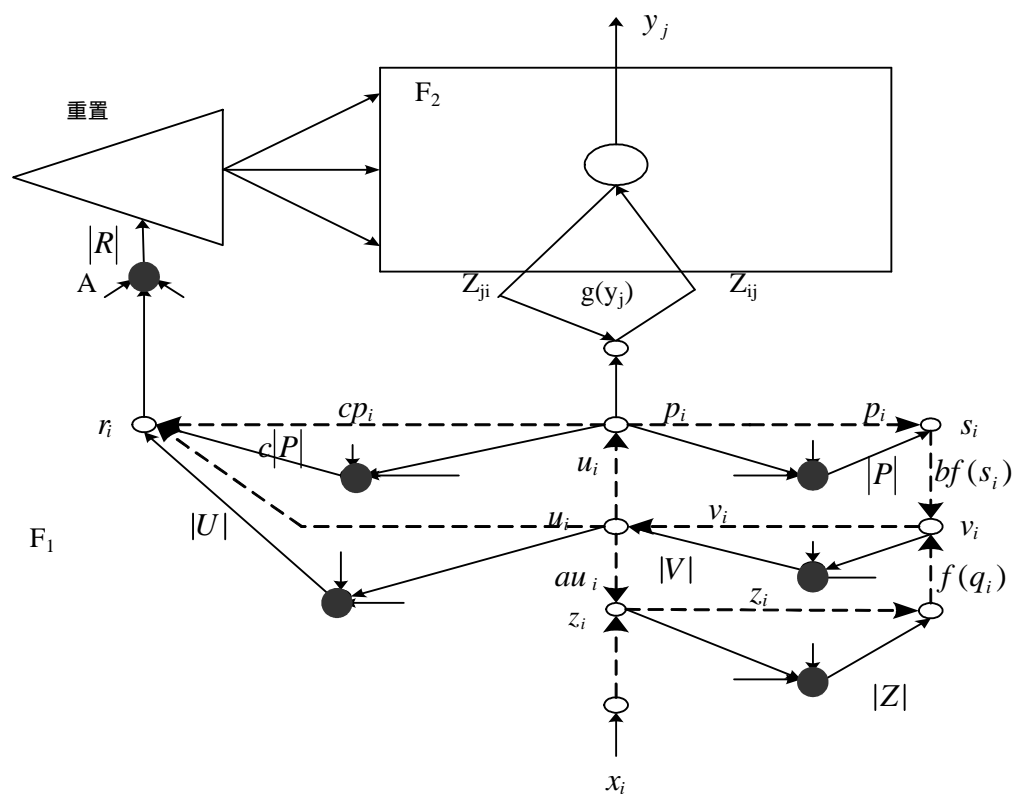


图 9 - 4 ART2 网络拓扑示意

$F_1$  场的  $N$  个神经元从外界接受输入模式  $X$ ，经场内的特征增强与噪声抑制等处理后通过由下至上的权值  $z_{ij}$  送到  $F_2$  场。 $F_2$  场的  $M$  个神经元接受  $F_1$  场上传的信号，经过竞争确定哪个神经元获胜，获胜神经元被激活，其他则均被抑制。与激活神经元相连的内外星权向量进行调整。增益控制系统负责比较模式与  $F_2$  场激活神经元的外星权之间的相似程度，当两向量的相似程度低于警戒门限时，复位子系统发出信号抑制  $F_2$  场的激活神经元。网络将在  $F_2$  场另选一个获胜神经元，直到相似程度满足要求。如果  $F_2$  场的神经元数  $M$  大于可能的输入模式类别数，总可以为所有新增的模式类分配一个代表神经元。

ART2 和 ART1 的原理类似，主要区别是 ART2 的比较层  $F_1$  场的结构与功能更为复杂一些。

### 9.2.2 ART2 型网络的数学模型与学习算法

#### 1. 特征表示场 $F_1$ 数学模型

特征表示场  $F_1$  由三层神经元构成，底层接收来自外界的输入，顶层接收来自  $F_2$  的外星反馈输入，在中间层对这两种输入进行相应的转换、比较并保存结果，将输出返回顶层节点。输入模式  $X$  是一个  $N$  维模拟向量，表示为  $X = (x_1, x_2, \dots, x_N)$ ，在  $F_1$  中有相应的  $N$  个处理单元，每个单元都包括上、中、下三层，每层都包含有两种不同功能的神经元，一种用小空心圆表示，另一种用大实心圆表示，它们的功能如下。

空心圆神经元：每个空心圆代表的神经元有两种输入激励，一种是兴奋激励，代表指向节点的特定模式；另一种是抑制激励，代表增益控制输入。设神经元  $i$  的输出用  $V_i$  表示，所

有兴奋激励的总和为  $J_i^+$ ，所有抑制激励总和为  $J_i^-$ ，则  $F_1$  的简化方程为：

$$V_i = \frac{J_i^+}{A + DJ_i^-} \quad (9-6)$$

由于  $D$  接近于 1， $A$  远小于 1，设  $e$  是远小于 1 的数，则：

$$V_i = \frac{J_i^+}{e + J_i^-} \quad (9-7)$$

实心点神经元：实心圆神经元的功能是求输入向量的模。在图 9-4 中， $F_1$  的底层和中层构成一个闭合的正反馈回路，其中标记为  $z_i$  的神经元接收输入信号  $x_i$ ，而标记为  $v_i$  的神经元接收上层送来的信号  $bf(s_i)$ 。这个回路中还包括两次规格化运算和一次非线性变换，其中底层输入方程和规格化运算为：

$$z_i = x_i + au_i \quad (9-8)$$

$$q_i = \frac{z_i}{e + \|Z\|} \quad (9-9)$$

式中  $e$  为很小的正实数，相对于  $\|Z\|$  可以忽略不计。

中层输入方程和规格化运算为：

$$v_i = f(q_i) + bf(s_i) \quad (9-10)$$

$$u_i = \frac{v_i}{e + \|V\|} \quad (9-11)$$

式中  $e$  为很小的正实数，相对于  $\|V\|$  可以忽略不计。

底层至中层至上层之间的非线性变换函数  $f(x)$  可以采用如下两种形式

$$f(x) = \begin{cases} 0, & 0 \leq x < \theta \\ x, & x \geq \theta \end{cases} \quad \text{或} \quad f(x) = \begin{cases} \frac{2\theta x^2}{x^2 + \theta^2} & 0 \leq x < \theta \\ x & x \geq \theta \end{cases} \quad \text{以上各式中, } a, b$$

和  $\theta$  由实验而定。

$F_1$  的中层和上层也构成一个闭合正反馈回路，其中标记为  $p_i$  的神经元接收来自中层的信号  $u_i$  和来自  $F_2$  场的信号，这个回路包括的运算是：

$$s_i = \frac{p_i}{e + \|P\|} \quad (9-12)$$

$$p_i = u_i + \sum_{j=0}^{M-1} g(y_j) z_{ji} \quad (9-13)$$

式中  $e$  为很小的正实数，相对于  $\|P\|$  可以忽略不计。式中第二项  $F_2$  场对神经元  $p_i$  的输入， $z_{ji}$  是自上而下的 LTM 系数。

## 2. 类别表示场 $F_2$ 的数学模型

类别表示场  $F_2$  的作用是增强自下而上（ $F_1 \rightarrow F_2$ ）的滤波输入模式的对比度，对比度的增强是通过  $F_2$  的竞争实现的。设  $F_2$  场中第  $j$  个节点的输入为：

$$T_j = \sum_{i=0}^{N-1} p_i z_{ij} \quad (9-14)$$

$F_2$  按下式进行选择： $T_{j^*} = \max\{T_j\} \quad j = M+1, \dots, N$

当选择节点  $j^*$  为最大激活时，其余节点处于抑制状态。选择 (9-13) 中的变换函数

$g(y_j) = dy_j$ ，则有：

$$g(y_j) = \begin{cases} d, & j = j^* \\ 0, & j \neq j^* \end{cases} \quad (9-15)$$

式中  $d$  为自上而下 ( $F_1 \rightarrow F_2$ ) 的反馈参数， $0 < d < 1$ 。则有：

$$p_i = \begin{cases} u_i + dz_{ji}, & j = j^* \\ u_i, & j \neq j^* \end{cases} \quad (9-16)$$

### 3. 权值调整规则

LTM 方程对长期记忆 LTM 权值的调整，按以下两个 LTM 方程进行。自上而下 ( $F_2$

$F_1$ ) LTM 方程为：

$$z_{ji}(t+1) = z_{ji}(t) + g[y_j(t)][p_i(t) - z_{ji}(t)] \quad (9-17)$$

自下而上 ( $F_2 \rightarrow F_1$ ) LTM 方程为：

$$z_{ij}(t+1) = z_{ij}(t) + g[y_j(t)][p_i(t) - z_{ij}(t)] \quad (9-18)$$

当  $F_2$  确定选择  $j^*$  节点后，对于  $j \neq j^*$ ， $g(y_j) = 0$ ，有：

$$z_{ji}(t+1) = z_{ji}(t) \quad (9-19)$$

当  $j = j^*$  时， $g(y_j) = d$ ，则有：

$$z_{ji}(t+1) = z_{ji}(t) + d[p_i(t) - z_{ji}(t)] = z_{ji}(t) + d(1-d)\left[\frac{u_i(t)}{1-d} - z_{ji}(t)\right] \quad (9-20)$$

$$z_{ij}(t+1) = z_{ij}(t) + d[p_i(t) - z_{ij}(t)] = z_{ij}(t) + d(1-d)\left[\frac{u_i(t)}{1-d} - z_{ij}(t)\right] \quad (9-21)$$

### 4. 取向子系统

图 9-4 中左侧为取向子系统，其功能是根据  $F_1$  的短期记忆模式与激活的长期记忆模式之间的匹配度决定  $F_2$  的重置。匹配度定义为

$$r_i = \frac{u_i + cp_i}{e + \|U\| + \|cP\|} \quad i=0, 1, 2, \dots, M-1 \quad (9-22)$$

式中  $e$  可以忽略。实心圆  $A$  的输出为匹配度的模, 用  $\|R\|$  表示。设警戒门限为  $\rho$ ,  $0 < \rho < 1$ ,

当  $\|R\| > \rho$  时, 选中该类, 否则, 取向子系统需对  $F_2$  重置。

### 9.2.3 ART 综合系统

ART 神经网络和人脑的某些功能类似, 它除了能够完成识别分类任务外, 还有其他功能, 如 " 补充 " 和 " 撤消 " 功能。这三种功能合称 ART 的 3R 功能, 即: Recognition(识别)、Reinforcement(补充)、Recall(撤消)功能。

#### 1. 补充功能

补充功能包括以下几个方面内容:

第一, 每当 ART 系统对输入矢量的类别作出一次判决, 即给出矢量所属类别的输出端编号, 根据此判决系统可以采取一种 " 行动 " 或作出某种 " 响应 "。这种情况和人相似, 人总是根据对外界情况的判断来决定自己的行动。凡是进行一项活动, 其结果不是使行动者 " 受益 " 就是使行动者 " 受损 ", 这两种结果也可称为受奖或受罚。对人来说, 所谓 " 受益 " 或 " 受损 " 总是与行动者内部的物质和精神需求相联系的, 诸如饥渴思饮食, 有痛苦时希望得到免除等等。又如, 在初步分类时, 我们把黄色和青色的香蕉归为一类, 但是在吃过青香蕉受了苦之后, 我们就知道应该将它们划分成两类。这就是说, 人们通过行动对外界作出某种响应, 这种响应造成的后果反馈回来对人形成奖或罚的作用。从而使人改变警戒参数来调整学习过程, 作出新的分类判决, 这使主体更适合于客观环境的实际情况, 这也是我们常说的 " 事实教育了我们 "。对于无生命的机器而言, 则可以用一个内容更广泛的 " 价值函数 " 来定义奖或罚。

第二, 人在识别时对于所有被识别的类并不是一视同仁的, 识别过程受到由顶向下预期模式的制约。这种制约使得人在某种情况下只关心几种类别, 而对其它类别则 " 听而不闻, 视而不见 "。这种集中注意力的本事可以使人在非常混乱的背景中发现目标, 在客体发生某种变形或缺损时仍能对其正确分类。例如, 当我们辨认一个模糊不清的字时, 如果这个字是孤立的就很难认出, 相反, 当它嵌在有意义的上下文之中时, 我们就可以根据构词规则、句法和语义知识将它辨认出来。这种现象在听懂话语、辨识图形和符号等许多场合普遍存在。ART 系统应该包括这方面的功能, 使分类和学习都受到由顶向下预期模式的制约, 这种制约是由系统与环境相互作用后形成的。换言之, 它是一种涉及范围更宽的 " 经验 "。

#### 2. 撤消功能

撤消功能与补充功能正好相反, 这是指某些不同的观察矢量在初步分类时被划分成不同的类别, 但在通过系统主体与客体相互作用, 又应判定它们属于一类。例如, 在初分类时, 同一个字母的印刷体和手写体被分为两类, 但它们实际上应属于一类。

#### 3. 综合 ART 系统

3R 功能可以组合在一起, 形成一个综合的 ART 系统, 其结构如图 9-5 所示。图中给出的框架可以用来模仿人的智能活动和心理特征, 诸如分类、集中注意力、转移注意力及归并分类结果等。还可以用此框架实现图象及语音识别系统<sup>[8]</sup>。框架包括 4 个部分: 预处理器、

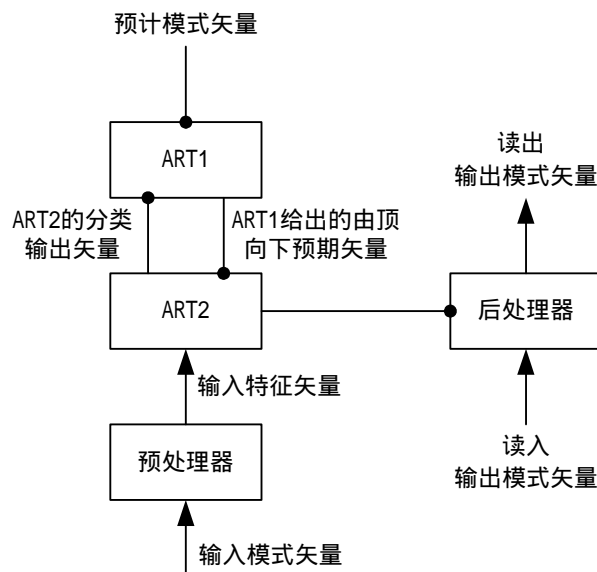


图 9-5 综合 ART 系统结构

ART2、ART1 及后处理器。它们各自的作用为：

预处理器：对输入模式矢量进行变换以产生一个更便于进行分类的特征矢量；

ART2：对输入特征矢量进行分类；ART1 的输入由两部分组成，一部分是 ART2 的分类输出，另一部分是来自框架以外的一个“预计模式”或称“预计矢量”。可以认为，预计矢量是由环境、背景或者上下文决定的，每当系统输入一个特定类别的矢量时，相应的预计矢量也是特定的。对于人来说，常出现类似的情况，即某种特定类别的刺激信号总是伴随着特定的外部和内部环境，后者也以特定的信号形式作用于生物体。

ART1：ART1 按照其全部输入产生它的分类输出，并形成由顶向下的预期矢量作用于 ART2。可以看出 ART1 的输出取决于 ART2 的分类输出矢量和外界作用于系统的“预计矢量”。通过学习后，在此两者之间建立了联想关系，即某种特定的 ART2 输出类别总是和某种特定的预计矢量伴随出现。这样一来，每当出现了某种特定预计矢量而 ART2 的分类结果尚未产生时，ART1 就会按联想产生与预计矢量相应的由顶向下预期矢量，这就限定了 ART2 进行识别的范围。

预计模式矢量：在系统中，预计模式矢量起两个作用：第一，预计模式矢量可以使系统“集中注意力”于某些特定的类别；第二，改变预计模式矢量可以改变或迁移系统的注意力集中点，从而满足系统的“内部需求”。例如，在某种特定预计模式矢量下得到的分类结果不能满足内部需要，则应改用其它预计模式矢量，再判断新得到的分类结果见诸行动后是否满足内部需要，甚至还可以建立新的类别。

后处理器：系统中后处理器的作用是完成撤消功能。后处理器的输入包括两部分，其一是 ART2 的分类输出；其二是由外界读入“输出模式矢量”，后处理器的输出即等于读入的输出模式矢量。如果 ART2 的不同分类结果具有一致的读入输出模式，那么后处理器经过学习后，对于这些不同的分类结果将给出完全相同的读出输出模式矢量，换句话说，将不同的类别归为一类。

## 9.3 ART 神经网络的应用

### 9.3.1 ART 神经网络在图像识别中应用<sup>[8-9]</sup>

我们知道，一个图象识别系统首先要解决的问题是当图象在平面上做任何平移、旋转、放大或缩小时，都能给出一致的识别结果。此处在对图象进行变换时，产生一种对图象的平移、旋转和尺度变化都不敏感的特征模式矢量，再将其赋予 ART 系统的输入矢量。按照这种

思路，需要用个预处理器对图象的输入进行变换。图 9-6 就是一预处理器系统。

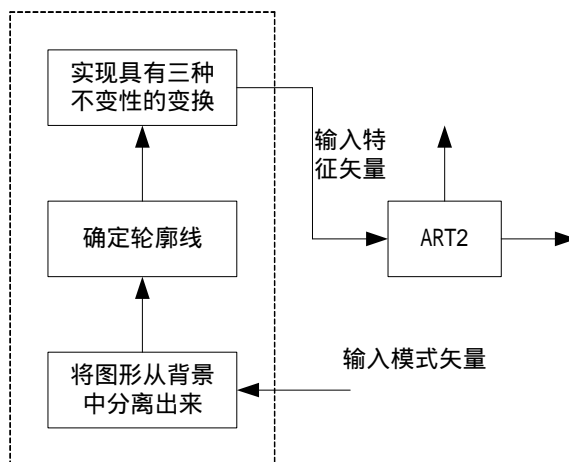


图 9-6 图象识别预处理系统

它由三级构成，其各部分的作用分别为：

第一级作用：将被识别的图象从背景中分离出来，为此可采用两个激光检测器，其中一个将焦点聚于被识别的物体上，另一个则能够把图象和背景分离开。例如，对于运动物体可以用多普勒原理将其与静止背景分离；对于静止不动的物体利用物体和背景的激光反射波的强度差异将两者分开。这样，前一个激光检测器所形成的图象是包括背景在内的被识别物体的清晰图象，而后一个激光检测器形成的是具有粗略轮廓的被识别物图象，但是它和背景之间的边界线并不是非常精确地确定的。

第二级作用：精确地确定物体图象和背景图象之间的边界线，也是物体图象的轮廓线。为了做到这一点，需要进行边缘检测和边缘锐化处理等，从而获得一个具有明确边界线的二维图象。此图象可以用一个二维平面的灰度函数  $f(x, y)$  来描述。在边界以外  $f(x, y) = 0$ ；

在边界以内，按照图象中各象素的不同灰度等级， $f(x, y)$  取不同的模拟数值，如：全白

为 0、全黑为 1， $f(x, y) \in (0, 1)$ 。

第三级作用：对  $f(x, y)$  进行变换，以获得一个具有三项不变性的特征矢量。其变换步骤为：

求  $f(x, y)$  的二维付里叶变换模值  $P(\omega_x, \omega_y)$ ： $f(x, y)$  的二维付里叶变换  $F(\omega_x, \omega_y)$  可

用式 (9-23) 计算：

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j(x\omega_x + y\omega_y)} dx dy \quad (9-23)$$

对  $F(\omega_x, \omega_y)$  取模即得到  $P(\omega_x, \omega_y)$ ，即：

$$P(\omega_x, \omega_y) = |F(\omega_x, \omega_y)| \quad (9-24)$$

不难证明，不管一个图形在二维平面上做何种移动，它的二维付立叶变换模值  $P(\omega_x, \omega_y)$  是不变的。



坐标变换：将直角坐标系  $(\omega_x, \omega_y)$  表示的  $P(\omega_x, \omega_y)$  改成极坐标系  $(r, \theta)$  表示的  $P(r, \theta)$ ，两坐标系之间的转换关系为：

$$r = (\omega_x^2 + \omega_y^2)^{\frac{1}{2}}, \quad \theta = \arctg\left(\frac{\omega_y}{\omega_x}\right) \quad (9-25)$$

如果一个灰度函数  $f(x, y)$  的图象放大  $k$  倍后，得到另一个图象，那么后者的灰度函数可以表示为  $f(x/k, y/k)$ ，与之相对应的直角坐标和极坐标表示的二维付里叶变换模值分别为： $P(k\omega_x, k\omega_y)$  和  $P(kr, \theta)$ ，可以看出，采用极坐标表示时，变量  $\theta$  将不随图象的尺度的变化而变化，而另一个变量仍受到图象尺度变化的影响。

对于  $P(r, \theta)$  的  $r$  维变量进行一维 Mellin 变换，得到  $M(\omega_\rho, \theta)$ ，其计算式为：

$$M(\omega_\rho, \theta) = \int_{-\infty}^{\infty} P(r, \theta) r^{-j\omega} \rho^{-1} dr \quad (9-26)$$

设  $P(kr, \theta)$  的 Mellin 变换是  $M_k(\omega_\rho, \theta)$ ，易于证明：

$$M_k(\omega_\rho, \theta) = k^{j\omega_\rho} M(\omega_\rho, \theta) \quad (9-27)$$

由于  $|k^{j\omega_\rho}| = 1$ ，所以  $|M_k(\omega_\rho, \theta)| = |M(\omega_\rho, \theta)|$ 。因此，可以知道上述 Mellin 变换的模函数  $|M(\omega_\rho, \theta)|$  具有不随尺度变化而变化的特性。如果设  $r = e^\rho$ ，那么式 (9-26) 可以改写为：

$$M(\omega_\rho, \theta) = \int_{-\infty}^{\infty} P(e^\rho, \theta) e^{-j\rho\omega} \rho d\rho \quad (9-28)$$

它表明只要将变量  $r$  转换为  $\rho$  ( $\rho = \ln r$ )，就可以用标准的付里叶变换来完成 Mellin 变换。

可以证明，当一个图象旋转了任意角度  $\theta_0$  后，它的极坐标可以表示为  $P(r, \theta - \theta_0)$ 。换言之，其二维付里叶变换模值也相应地旋转了同样的  $\theta_0$  角，相应的 Mellin 变换模函数可以表示为  $|M(\omega_\rho, \theta - \theta_0)|$ 。为了获得旋转不变性，只需要对变量  $\theta$  通过适当取样（离散化）后，再做离散付里叶变换即可。对后者取模后就可以得到一个与  $\theta_0$  无关的函数<sup>[11]</sup>。

综上所述，通过预处理器的三级处理后，能够向 ART 系统提供一个具有位移、尺度及旋转变化三方面皆具有不变性的特征矢量。如果调整警戒参数，甚至还允许图形有适度的“变形”，在图象有上述三种变化及适度变形时，仍能给出正确的分类结果。

### 9.3.2 ART 神经网络在故障诊断中应用

为了提高电路板故障诊断定位精度和故障隔离率，此处介绍一种基于 FTFART (Field Theory based Fuzzy Adaptive Resonance Theory) 模糊神经网络模型。

模糊逻辑理论和神经网络技术在知识表示、知识存贮、推理速度及克服知识窄台阶效应等方面起到了很大的作用。模糊理论主要模仿人脑的逻辑思维，具有较强的结构性知识表达能力；神经元网络模仿人脑的神经元功能，具有较强的自学能力和数据直接处理能力，二者结合为模糊神经网络可以优势互补，大大提高了故障信息的处理能力。

目前的模糊神经网络模型依据其中的神经网络模型大致可分为联想型、前馈型和竞争型三种。联想型网络的容量小，所存储的信息之间容易互相干扰或者发生退化现象。前馈型网络特别是其中的 BP 网络一直是研究的热点，并在实际工程中广泛应用。但是 BP 网络所固有的缺陷包括难以确定隐层节点数目、训练学习时间长、容易陷入局部极小点等并没有得到彻底的解决。竞争型网络多用于分类器系统，训练学习时间短，收敛性好。并且，故障诊断本身也就是一个故障模式的分类与匹配问题，与竞争型网络可以很好的结合。

本系统采用了 FTFART (Field Theory based Fuzzy Adaptive Resonance Theory) 模糊神经网络模型。该模型能够处理模糊概念，知识易于提取与解释，无需人为设定隐层神经元数目，能够自适应的调整网络的拓扑结构，学习速度快，分类效率高、准确性好。较适合电路板的故障诊断。以下为模糊神经网络的一个用例。

神经网络用于对电路板不十分了解，但通过仿真测试能够得到足够多的诊断经验数据，训练收敛后的网络用于诊断能够得到较好的结果。神经网络诊断具有操作简单，不需要用户太多神经网络理论知识。缺点是需要足够多的诊断经验知识。诊断经验知识越丰富，诊断效率就越高。

#### 1. 故障诊断模型

本系统采用的神经网络模型为 FTART2 (基于域理论的自适应谐振神经网络模型，Field Theory based Adaptive Resonance Theory 2)。该模型结合了自适应谐振理论和域理论的优点，学习速度快，归纳能力强，效率高，可以根据输入样本自适应地调整拓扑结构，克服了前馈型网络需要人为设置隐层神经元的缺点。通过基准测试表明，FTART2 在学习精度和速度上都远远优于标准 BP 算法，是一个适用于故障分类的神经网络模型，而且其网络结构本身描述了记录在网络中的知识的意义，网络知识易于解释。因此，该诊断模型也较适于电路板的故障诊断。

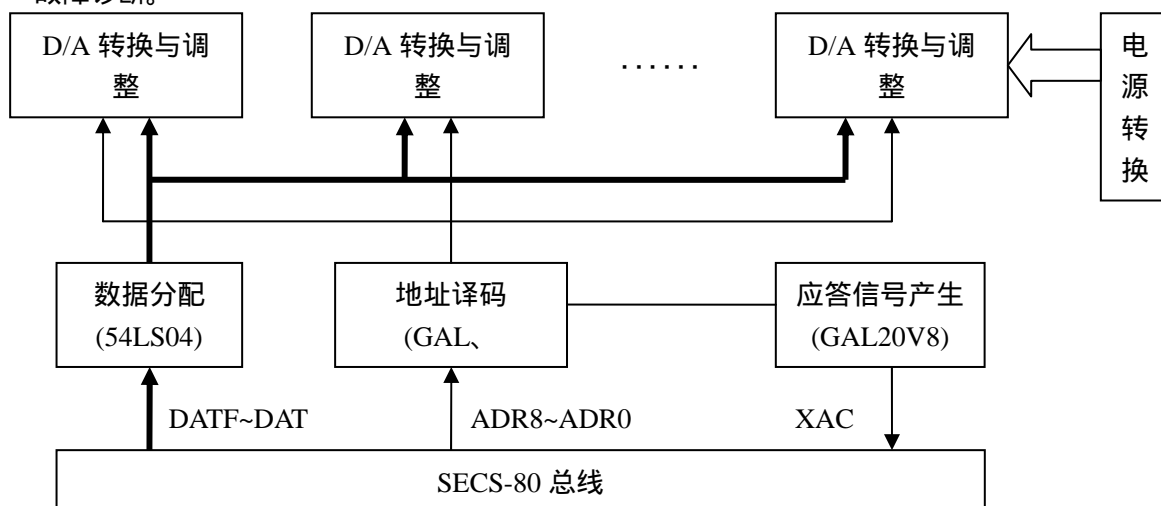


图 9-7 HTZ/DAC8 电路功能框图

图 9-7 为待诊断的电路板原理图。在被测板中有一块 8 路数字/直流电压转换板

HTZ/DAC8 用于 ZJ-26 导弹指挥仪，主要由 D/A 转换、数据传输、地址译码和电源 4 大部分组成。

以 D/A 转换芯片 A1 为例进行测试诊断，包括 5 个元器件，L1-L5。其中 L1：电源 Aa；L2：GAL 译码器；L3：反相器 A11；L4：反相器 A12；L5：转换芯片 A1。进行功能测试时，加一组激励 DAC1 得到 10 个响应数据 U1-U10，加上测试得到的 Load 值 Ld、芯片供电正负电压值 Up1、Up2，可以对元器件出现的故障进行诊断。诊断知识来自仿真测试和故障树分析，如表 9-1 所示。

表 9-1 诊断知识表

U1	U2	U3	U4	U5	U6	U7
-10.0000	-9.9951	-9.9902	-7.5000	-5.0000	0.0000	5.0000
0.0000	0.0049	0.0098	2.5000	5.0000	0.0000	5.0000
-5.0000	-4.9951	-4.9902	-2.5000	-5.0000	5.0000	5.0000
-7.5000	-7.4951	-7.4902	-7.5000	-2.5000	2.5000	7.5000
-8.7500	-8.7451	-8.7402	-6.2500	-3.7500	1.2500	6.2500
-9.3728	-9.3679	-9.3630	-6.8728	-4.3728	0.6272	5.6272
-9.6864	-9.6815	-9.6766	-7.1864	-4.6864	0.3136	5.3136
-9.8432	-9.8383	-9.8334	-7.3432	-4.8432	0.1568	5.1568
-9.9216	-9.9167	-9.9118	-7.4216	-4.9216	0.0784	5.0784
-9.9608	-9.9559	-9.9510	-7.4608	-4.9608	0.0392	5.0392
-9.9804	-9.9755	-9.9706	-7.4804	-4.9804	0.0196	5.0196
-9.9902	-9.9853	-9.9804	-7.4902	-4.9902	0.0098	5.0098
-9.9951	-9.9902	-9.9853	-7.4951	-4.9951	0.0049	5.0049
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

U8	U9	U10	Up1	Up2
7.5000	9.9902	9.9951	15.0000	-15.0000
7.5000	9.9902	9.9951	15.0000	-15.0000
7.5000	9.9902	9.9951	15.0000	-15.0000
7.5000	9.9902	9.9951	15.0000	-15.0000
8.7500	9.9902	9.9951	15.0000	-15.0000

8.1272	9.9902	9.9951	15.0000	-15.0000
7.8136	9.9902	9.9951	15.0000	-15.0000
7.6568	9.9902	9.9951	15.0000	-15.0000
7.5784	9.9902	9.9951	15.0000	-15.0000
7.5392	9.9902	9.9951	15.0000	-15.0000
7.5196	9.9902	9.9951	15.0000	-15.0000
7.5098	9.9902	9.9951	15.0000	-15.0000
7.5049	9.9951	9.9951	15.0000	-15.0000
0.0000	0.0000	0.0000	15.0000	0.0000
0.0000	0.0000	0.0000	0.0000	-15.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000

Ld	L1	L2	L3	L4	L5
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	0	0	1
1	0	1	0	0	0

0	1	0	0	0	0
---	---	---	---	---	---

由诊断知识可以看出，由于测试响应数据 U1 - U10、Up1 以及 Up2 具有容差，且 U1 - U10 的容差值和响应值有时出现覆盖和冲突，测试数据与元器件故障模式的对应关系较复杂。如果应用规则推理，由于数据容差的影响而出现征兆组合爆炸，规则推理繁杂，有时甚至推不出结果；若应用模糊 BP 神经网络推理，由于测试数据模糊子集众多，设置模糊函数比较困难，且万一隶属函数设置错误则导致错误的推理结果甚至推导不出。因此对专家的经验要求很高，采用模糊神经网络推理也不太适合。而采用 FTART2 神经网络模型来解决这个问题，神经网络推理对操作者技术水平要求不高，只要了解诊断模型的输入和输出，经训练收敛后的神经网络诊断模型即可用于诊断。其具体训练算法见本章第 2 节。

样本管理器												
样本输入值											期望输出值	
-10.0000	-9.9951	-9.9902	-7.5000	-5.0000	0.00...	0.0000	0.0049	0.0098	2.5000	5.0000	0.00...	0,0,0,0,0
-5.0000	-4.9951	-4.9902	-2.5000	-5.0000	5.00...	0.0000	-4.9951	-4.9902	-2.5000	-5.0000	5.00...	0,0,1,0,0
-7.5000	-7.4951	-7.4902	-7.5000	-2.5000	2.50...	0.0000	-7.4951	-7.4902	-7.5000	-2.5000	2.50...	0,0,1,0,0
-8.7500	-8.7451	-8.7402	-6.2500	-3.7500	1.25...	0.0000	-8.7451	-8.7402	-6.2500	-3.7500	1.25...	0,0,1,0,0
-9.3728	-9.3679	-9.3630	-6.8728	-4.3728	0.62...	0.0000	-9.3679	-9.3630	-6.8728	-4.3728	0.62...	0,0,1,0,0
-9.6864	-9.6815	-9.6766	-7.1864	-4.6864	0.31...	0.0000	-9.6815	-9.6766	-7.1864	-4.6864	0.31...	0,0,1,0,0
-9.8432	-9.8383	-9.8334	-7.3432	-4.8432	0.15...	0.0000	-9.8383	-9.8334	-7.3432	-4.8432	0.15...	0,0,0,1,0
-9.9216	-9.9167	-9.9118	-7.4216	-4.9216	0.07...	0.0000	-9.9167	-9.9118	-7.4216	-4.9216	0.07...	0,0,0,1,0
-9.9608	-9.9559	-9.9510	-7.4608	-4.9608	0.03...	0.0000	-9.9559	-9.9510	-7.4608	-4.9608	0.03...	0,0,0,1,0
-9.9804	-9.9755	-9.9706	-7.4804	-4.9804	0.01...	0.0000	-9.9755	-9.9706	-7.4804	-4.9804	0.01...	0,0,0,1,0
-9.9902	-9.9853	-9.9804	-7.4902	-4.9902	0.00...	0.0000	-9.9853	-9.9804	-7.4902	-4.9902	0.00...	0,0,0,1,0

图 9-8 诊断知识表

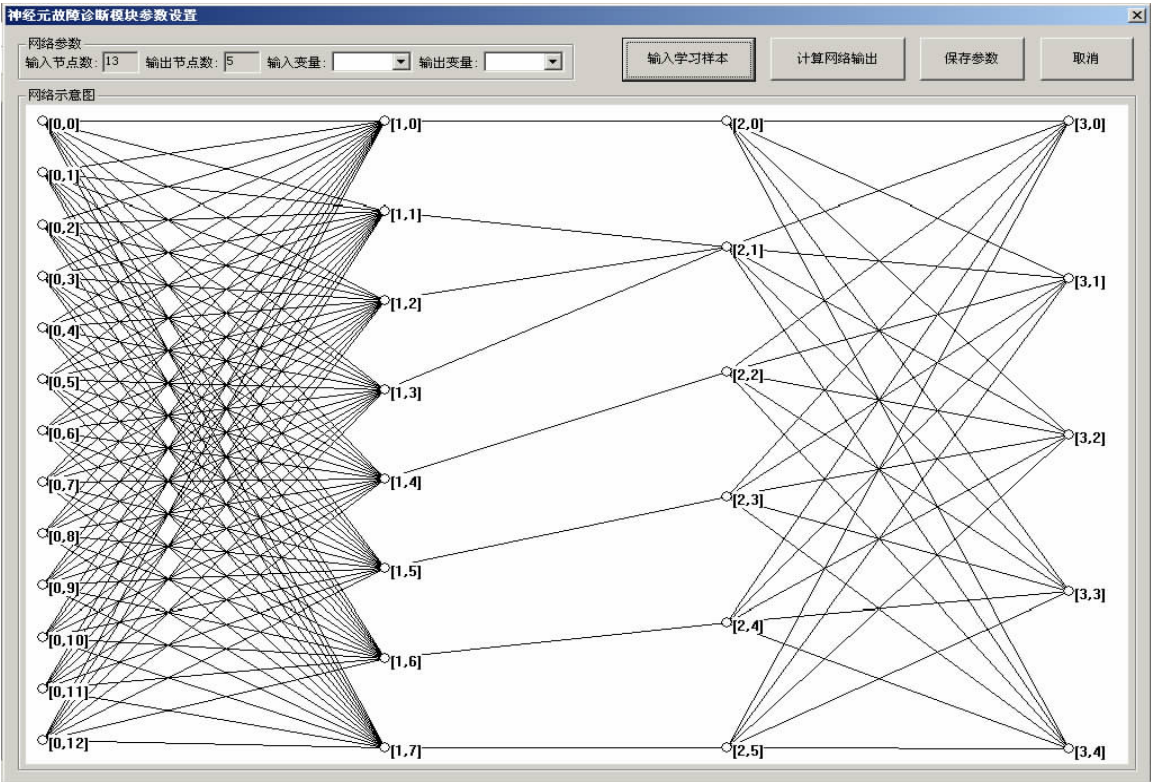


图 9-9 训练后收敛的网络拓扑结构

用神经网络法进行推理诊断，确定输入数据数为 13，输出数据数为 5。首先输入诊断知

识对网络进行训练，本系统诊断数据输入界面如图 9-8 所示。训练后收敛的网络拓扑结构界面图如图 9-9 所示。

## 2. 诊断结果分析

设实际测试得到响应数据向量如下：

$(-9.9945, -9.9899, -9.9804, -7.5001, -4.9972, 0.0055, 5.0032, 7.5102, 9.9949, 9.9949, 15.0001, -15.0003, 0)$ ，将其作为神经网络的输入，经过神经网络推理可得到对应元器件故障模式  $(0, 0, 0, 1, 0)$ ，即反向器 A12 故障，经过实际测试得出实际结果与神经网络诊断结论相符。

## 参考文献：

- [1] Carpenter G.A., Grossberg S., The ART of adaptive pattern recognition by a self-organizing neural network, IEEE Trans on Computer, 1988, No.3, pp.77-88.
- [2] Carpenter G.A., Grossberg S., A massively parallel architecture for a self-organizing neural pattern recognition machine, IEEE Trans on Computer Vision, Graphics and Image Processing, 1987, Vol.37, pp.54-115.
- [3] Grossberg S., Neural networks and native intelligence, Cambridge, MA: MIT Press, 1988.
- [4] Carpenter G.A., Grossberg S., ART2: self-organization of stable category recognition codes for analog input patterns, Applied Optics, 1987, Vol.26, pp.4919-4930.
- [5] Carpenter G.A., Grossberg S., ART3: Hierarchical searching using chemical transmitters in self-organizing pattern recognition architectures, Neural Networks, 1990, Vol.3, pp.129-152.
- [6] 韩力群，人工神经网络的理论、设计及应用，北京，化学工业出版社，2002.1。
- [7] 朱大奇，电子设备故障诊断的原理与实践，电子工业出版社，2004.1。
- [8] 南京航空航天大学翻印，人工神经网络，1997.9。
- [9] Cohen M.A., Grossberg S., Masking fields: a massively parallel neural architecture for learning, recognizing and predicting multiple grouping of patterned data, Applied Optics, 1987, 26(10): 1866-1891.
- [10] 闻新等，控制系统的故障诊断和容错控制，北京，机械工业出版社，1998.2。
- [11] Oppenheim A.V., Schaffer R.W., Digital signal processing prentice-hall, Inc., Englewood Cliffs, N.J. 1975.

## 思考题

1. 请用自己的语言介绍 ART1 型神经网络的基本构造及工作原理。
2. 请用自己的语言介绍 ART2 型神经网络的基本构造及工作原理。
3. 在 ART 神经网络中，何为短期记忆？何为长期记忆？  
试设计一个 ART1 神经网络，对下面 3 个输入模式进行分类。设计一个合适的警戒门限，使得 ART1 网络能将 3 个输入模式分为 3 类。三个输入模式为：

$$X_1 = (1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1) ;$$

$$X_2 = (1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1) ;$$

$$X_3 = (1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1)。$$

5. 给出一个 ART 神经网络应用实例。

## 第10章 量子神经网络

人工神经网络由于其大规模并行处理、容错性、自组织和自适应能力和联想功能等特点,已成为解决许多问题的有力工具,对突破现有科学技术的瓶颈,更深入的探索非线性等复杂现象起到了重要的作用,广泛应用于众多科学领域<sup>[1-4]</sup>。然而,随着问题的复杂程度提高,网络结构相对单一的传统神经网络模型表现出了种种缺陷和不足,例如固有的学习速度慢、灾变性失忆、硬件实现难度大等等,一定程度上制约了它的应用<sup>[5]</sup>。

### 10.1 量子神经网络的发展<sup>[5]</sup>

量子神经网络的概念出现于上个世纪 90 年代后期,一经提出后便引起了不同领域的科学家的关注,人们在这个崭新的领域进行了不同方向的探索,提出了很多想法和初步的模型,充分体现了量子神经网络研究的巨大潜力。主要研究方向可以概括为:

量子神经网络采用神经网络的连接思想来构造量子计算机,通过神经网络模型来研究量子计算中的问题;

量子神经网络在量子计算机或量子器件的基础上构造神经网络,充分利用量子计算超高速、超并行、指数级容量的特点,来改进神经网络的结构和性能<sup>[6-10]</sup>;

量子神经网络作为一种混合的智能优化算法在传统的计算机上的实现,通过引入量子理论中的思想对传统神经网络改进,利用量子理论中的概念、方法(如态叠加、“多宇宙”观点等),建立新的网络模型;改善传统神经网络的结构和性能<sup>[11-22]</sup>;

基于脑科学、认知科学的研究。

以上不同的研究方向之间并没有明显的界限,它们之间是紧密联系相互交叉的。目前国际上对于量子神经网络的研究才刚刚开始,对它还无确切定义。其中美国的 Kak 是最早提出量子神经网络的学者之一,通过对认知和意识的研究将量子理论引入到人工智能领域,认为神经元的信息处理应该是有量子特性,并在随后的研究中,从大脑意识的角度对量子神经网络进行研究<sup>[23-26]</sup>;美国的物理学家 Berhman 等人在量子神经网络方面做了大量的工作,他们在量子点分子模型上构造了时间和空间的量子神经网络,通过神经网络调节量子器件的各个物理参数,影响微观粒子的量子态,改变波函数,使其最终演变到需要的量子态<sup>[27-35]</sup>;美国的 Dan 博士在攻读博士期间发表了一系列量子神经网络的文章,研究了量子计算的学习问题提出了量子联想存储的算法,相对传统的存储方法,量子联想存储有指数级的存储容量<sup>[6-10]</sup>;英国的 Tammy 等人也研究了量子神经网络的结构和模型,从多宇宙的量子力学观点出发,提出了构建叠加的多宇宙量子神经网络模型的思想<sup>[36]</sup>;巴西的华人 Li Weigang 提出了纠缠的量子神经网络(Entanglement Neural Network),采用了量子的隐形传态<sup>[18]</sup>;俄罗斯 A.A.Ezhov 等<sup>[37]</sup>进行了基于光学元器件的量子神经网络的研究,采用光学元件构成神经网络;日本的 Matsui 等人在 2000 年提出了用量子态作为神经元的状态,在传统神经网络拓扑结构的基础上,采用了基于量子比特的信息表示方式和处理方式<sup>[19-21]</sup>;N.B.Karayiannis 等人 1997 年提出了基于多层激励函数的量子神经网络<sup>[11, 12, 38]</sup>,三层的网络结构中,隐层量子神经元的激励函数采用多个传统激励函数的叠加,使网络有了一种固有的模糊性,并引起了模糊识别领域研究者的注意,在气象预测<sup>[13-15]</sup>、手写字识别<sup>[16]</sup>、语音识别<sup>[17-18]</sup>等方面得到了成功的应用。

### 10.2 量子神经网络模型

这里重点介绍传统计算机上能实现的量子神经网络,例如基于多层激励函数的量子神经网络,日本学者 Matsui 等人提出的神经网络模型等等,同时介绍其它量子神经网络模型。

### 10.2.1 多层激励函数的量子神经网络

N.B.Karayiannis 等人 1997 年提出多层激励函数的量子神经网络，并且从理论分析和实验验证了这种神经网络在模式分类时具有的内在模糊性，它能检测到数据中固有的模糊性和不确定性，对两类交叉的数据样本，这种神经网络能以一定的隶属度将其同时分在两类中。

基于多层激励函数的量子神经网络是 3 层的网状结构，输入层，隐层，输出层，其中输入层和输出层与传统的前向神经网络无异，而隐层的量子神经元借鉴了量子理论中量子态叠加的思想，激励函数表示为多个 sigmoid 函数的线性叠加，称之为多层激励函数。这样一个隐层神经元就能表示更多的状态或量级，而一个传统的 sigmoid 函数只能表示两个状态或量级，叠加的每一个 sigmoid 函数有多个不同的量子间隔，可以调整量子间隔，能使不同的数据映射到不同的量级或阶梯上，从而使分类有更多的自由度。如图 10-1 为量子神经网络和传统神经网络激励函数示意图。

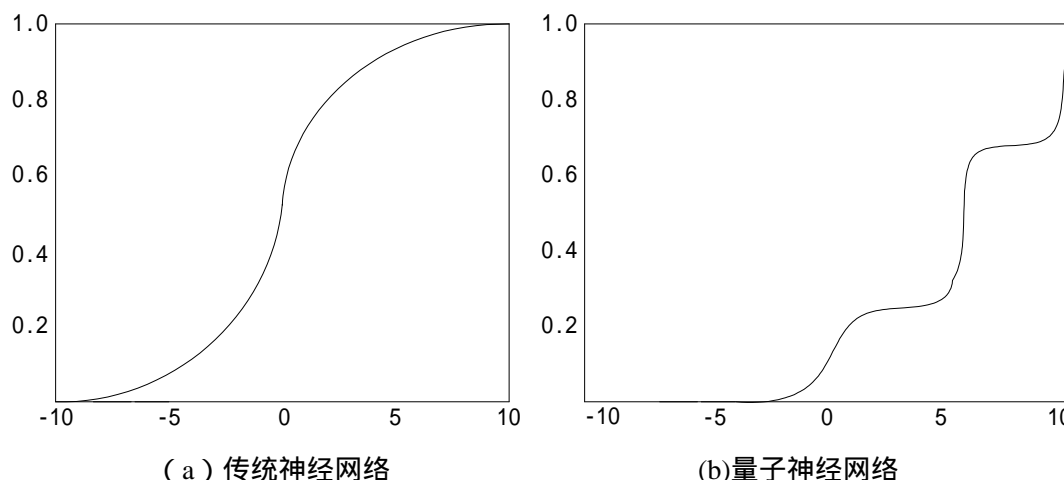


图 10-1 量子神经网络和传统神经网络激励函数

量子神经网络的学习分两步，一是对权值的调整，使输入数据对应到不同的类空间中，二是对隐层的量子神经元的量子间隔进行调整，体现数据的不确定性<sup>[12]</sup>。由于这种量子神经网络固有的模糊特性，它已经成功地应用于模式分类方面，特别是在处理两类有交叉的数据样本时，更显示出它的优越性。

### 10.2.2 Qubit 神经元模型

日本的 Nobuyuki Matsui 等人在 2000 年提出了用量子比特表示神经元状态的网络模型<sup>[19-21]</sup>。在这种网络模型中，网络的拓扑结构和传统的神经网络相同，图 10-2 是一个三层的量子神经网络，看起来和传统的神经网络无异。但是，它的神经元信息表示、权值、激励函数等都与传统神经网络不同。

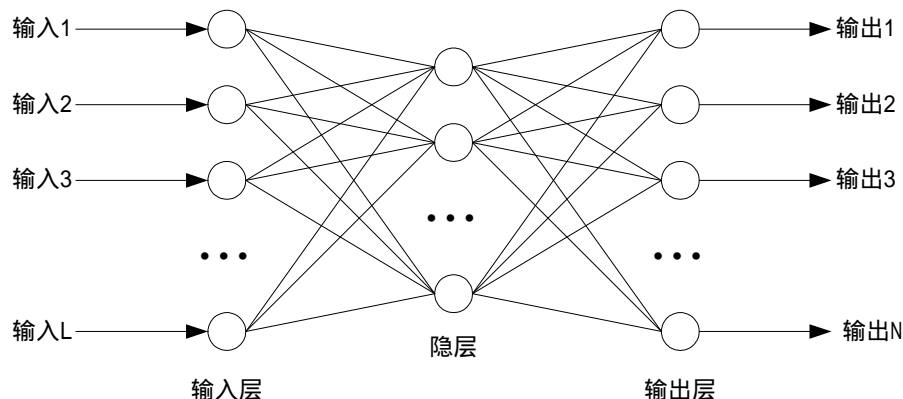


图 10-2 Qubit 神经元模型



这种量子神经网络的主要特点是用量子态相位表示信息,网络权值的作用是对相位进行旋转变化,激励函数对相位可控非门操作,通过改变量子态达到运算的目的。这种结构改变了传统的网络模式,但是它的输入只能是 0、1 值,输出是概率幅值,限制了它的应用。目前的实验仅限于逻辑门操作、奇偶校验<sup>[19-21]</sup>等。

### 10.2.3 多宇宙的量子神经网络模型

Tammy 等人从光学的双缝干涉实验和量子力学中多宇宙的观点得到启发,提出了多宇宙的量子神经网络模型<sup>[36]</sup>。多宇宙的量子神经网络是多个相似网络组件的叠加。在训练时不同的输入模式对不同的网络组件进行训练,测试时不同的输入会有不同的网络组件来对输入信息处理。也就是量子神经网络的网络组件的线性叠加,当有输入时,就会破坏网络组件的叠加,使其塌陷到某个网络组件上,然后神经网络就通过这一组件对输入数据进行处理。

这种模型的特点是:

训练时间短:由于对每一种输入模式,量子神经网络中都有与之对应的网络组件,这样仅需要训练对应的网络组件,与其他的网络组件无关,各个网络组件的学习可以并行进行,减少了神经网络的学习时间;

可消除灾变性失忆现象:传统的神经网络在对样本训练时,不同模式的样本将导致权值朝向不同的方向变化,网络学习新的样本信息时,会“忘掉”以前学到的信息,使权值调节产生振荡(灾变性失忆现象)。而量子神经网络对于不同的样本数据,会有不同的网络组件与之对应,网络的调节只限于输入模式对应的网络组件,这样不同的模式的学习之间互不相干,避免了灾变性失忆现象的发生。

目前这种神经网络在学习算法、坍塌算法、具体的网络结构方面还没有突破性的研究。首先在学习算法方面,要解决的问题是采用什么方法训练网络,使不同的训练数据对应到不同的网络组件上。其次,网络在处理数据时应该如何根据输入数据使量子神经网络坍塌到对应的组件上。

### 10.2.4 其它模型研究

#### 1. 其它量子神经网络模型

美国的 Berhman 等人在量子神经网络方面的工作,主要集中于量子计算机的研究,他们在 SQID 物理设备上(Superconducting quantum interference devices)通过神经网络算法控制培养基上量子点分子外界的势场、核声子等,使量子状态不断变化,直到达到所需的量子状态,提出了时间的量子点分子神经网络<sup>[27-35]</sup>。

美国的 Dan 提出了量子联想记忆模型,改进了 Grover 搜索算法<sup>[39]</sup>,表明量子存储器具有指数级的存储容量。这种神经网络模型不具有传统的网络结构,信息的载体是量子器件,

其主要特点是量子联想存储器具有指数级的存储容量, $n$  位的量子寄存器同时存储  $2^n$  个不同状态的信息。实现的思想是,通过学习算法,使量子存储器存储特定信息体现在波函数的变化上,然后,联想记忆过程就是通过搜索算法中的么正变换作用于波函数,并最终使要查询的信息的概率增大,使观测者能以较大的概率观测到需要的信息。

巴西的 Li Weigang 利用量子的隐形传态提出了纠缠神经网络<sup>[18]</sup>。

#### 2. 量子神经网络模型比较

目前提出的量子神经网络有着不同的特点,不同程度的利用了神经网络和量子计算的特性,有的研究推动了量子计算的发展,有的研究改进传统神经网络的性能。为了明确量子神经网络的研究方向,此处对各项研究作一比较,见表 10-1:

表 10-1 量子神经网络的比较

	网络主要特点	突出性能	与量子计算关系	目前应用前景
多层激励函数	传统网络拓扑结	固有的模糊	函数的迁移称为	模式识别,在传

量子神经网络	构。隐层神经元使用多层激励函数	特性,学习周期减少	量子迁移(量子间隔)。只是概念上使用量子计算	统计算机上实现
Qubit 神经元模型	传统网络拓扑结构。信息用 qubit 表示权值和激励函数,是对量子态波函数的改变	学习速度较传统神经网络快	采用量子计算中的信息表示、运算门,网络通过传统计算机上模拟量子计算的过程,达到比特位映射的目的	无实际应用,且目前的研究只限于对 0、1 比特信息的处理
多宇宙的量子神经网络模型	网络是多种传统神经网络或网络组件的叠加	消除了灾变失忆现象,学习速度快	利用量子理论量子态叠加的重要思想(多宇宙的量子神经网络)	无实际应用,只是提出一种实现思想
Dan 等人的量子神经网络	不具有传统神经网络的拓扑结构。采用量子寄存器存储量子信息通过么正变换改变寄存器状态	具有指数级的存储容量。是量子计算智能化理论的进步,具有学习能力,联想记忆功能	属于量子计算领域的研究,与量子器件的实现密切相关,实际上是对 grover 算法的改进	无实际应用,量子计算的应用需要量子计算机的物理实现,目前只处于理论研究阶段
Behman 等人的研究	不具有传统结构,侧重量子计算机的研究。	通过量子点分子等物理器件实现量子计算	通过神经网络来调节物理参数,控制量子器件中量子态的演化,达到量子计算的目的	物理实验室的研究
Kak 等关于脑科学的研究	从生物智能的角度对神经网络和量子计算进行了研究,提出大脑中的认识,决策等具有量子特性的观点			

### 10.2.5 量子神经网络的发展

尽管量子神经网络的研究仅处于初步的阶段,但是从目前的研究成果来看,该技术还是很有潜力的。研究的方向可能集中于:

- 1.神经网络集成的研究。**神经网络集成通过训练多个神经网络并将其结论进行合成,可以显著提高学习系统的泛化能力。量子理论中的叠加思想与神经网络集成有相似性。可以在神经网络集成的研究基础上开展量子神经网络的研究,借鉴神经网络集成的研究成果研究量子神经网络。
- 2.Tammy 多宇宙神经网络思想的实现。**Tammy 的思想可以解决传统神经网络灾变性失忆的问题,能有效提高网络的学习速度;然而具体怎么实现这种思想是需要解决的问题,例如网络的训练算法,塌陷算法等。
- 3.复数神经网络。**Matsui 等人提出的量子神经网络与复数神经网络有着某种对应关系。通过对量子神经网络的进一步研究,可有效解决复数域的计算问题。
- 4.打破当前的神经网络结构,提出新的神经元计算模型。**
- 5.量子计算在传统计算机上就失去它的优越性。**因此,很可能真正的量子神经网络只有在量子计算机上才能实现,当前量子计算的研究是物理学家和计算机学家的研究热点,并已经有人初步提出了基于量子器件的神经网络模型。

## 10.3 量子神经网络的应用

### 10.3.1 基于量子双缝干涉实验的神经网络模型<sup>[40]</sup>

量子计算(Quantum Computation)成功地利用了量子理论中有关量子态的叠加(Superposition)、纠缠(Entanglement)和干涉(Interference)等性质,其计算能力与传统的计算模式相比有了质的突破。它的出现使人们对于计算产生了新的认识:计算实质上是一个物理系统随时间演化的过程,它具有非常丰富的物理学基础。由于量子计算的独特性质,已有越来越多的学科与之结合,产生了诸如量子通讯、量子计算机、量子信息处理等新的交叉学科。

1995 年, Kak 首次提出了量子神经计算(Quantum Neural Computation)的概念<sup>[12]</sup>,将人工神经网络理论与量子计算结合起来,它在某些方面会改善传统神经网络的性能,此举丰富了神经网络理论的研究内涵,并开创了新的发展方向。

此处所提出的 QNN 模型就是在量子理论中著名的双缝干涉实验的基础上构建起来的。由于从信息处理的角度考察量子双缝干涉实验,它的结构十分类似于前馈的神经网络,因此引起特别的关注。人们试图通过某种定义将二者结合起来,其中有代表性的是 Chrisley 提出的非叠加态量子神经计算模型,但是它纯粹是一个理论模型,无法实现仿真计算。鉴于此,此处提出一种全新的量子神经网络模型,它克服了上述缺点,能够实际计算。最突出的是,它可以用两层网络结构实现类似 XOR 逻辑的学习,突破了传统神经网络的限制,体现了量子计算的优越性。为探索神经网络与量子计算的结合提供了一个新的思路。

#### 1. 双缝干涉实验简介

双缝干涉实验是量子理论中一个著名的实验,它能直观地展现出微观粒子的波粒二象性,同时它也是量子力学中最为离奇、在概念上最难理解的现象之一。该实验的结构如图 10-3 所示,入射源是微观粒子(如电子、光子等)。这里以光子为例,控制左侧的入射光子枪,使光子几乎一个一个地通过双缝(即狭缝 1 和狭缝 2),然后打在探测底板上。起初底板上出现一些点的分布,随着时间的延长点会越来越多,最后在底板上会形成一个有规律的干涉花纹(见图 10-3 的最右侧)。若在实验中依次开启其中一个狭缝,则会相应得到图 10-3 中间所示的两个花纹。但奇怪的是,当双缝同时开启时,最终所得的干涉花纹并非是前两者的简单叠加,这就体现出光子不仅具有粒子性,而且同时还具有波动性的事实。

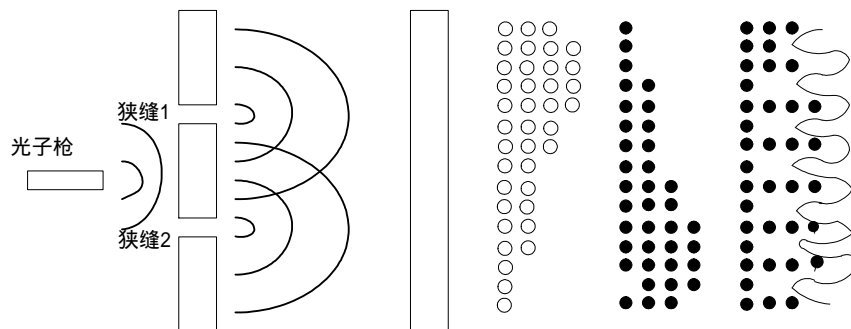


图 10-3 光子双缝干涉实验示意图

依据量子理论,可以得到底板上点的强度分布为

$$I(x, y) = |\psi(x, y)|^2 = \psi^*(x, y)\psi(x, y) = 4A^2 \cos^2\left(\frac{kd}{2D}x\right) \quad (10-1)$$

其中,几率幅波函数为:

$$\psi(x, y) = 2A \exp\left(ik\left(D + \frac{(d/2)^2 + x^2 + y^2}{2D}\right)\right) \cos\left(\frac{kd}{2D}x\right) \quad (10-2)$$

(10-2)式中,  $A$  是光子源在底板上产生的振幅,  $D$  是栅板到底板的距离,  $\lambda$  为波矢,  $d$  为双缝间距,  $x$ 、 $y$  是底板上的直角坐标轴。

### 1. Chrisley 量子神经计算模型

鉴于上述实验装置的结构与前馈神经网络模型十分相似, Chrisley 提出了基于非叠加态的量子神经计算模型, 并给出了相应的学习算法。Chrisley 模型如图 10-4 所示, 粒子束  $G$  入射栅板  $B$  之后是一个光敏接受盘  $P$ , 在栅板  $B$  上有许多狭缝, 其中上部分组成输入区, 下部分组成权值区, 调节狭缝的结构便可以在接受盘  $P$  上得到不同的干涉模式, 这样便实现了两组映射。即: 一个是由输入(如特征串、图像、查询等)到栅板  $B$  上狭缝结构的映射(1); 另一个是由干涉模式到输出(如分类、存储数据等)的映射(0)。

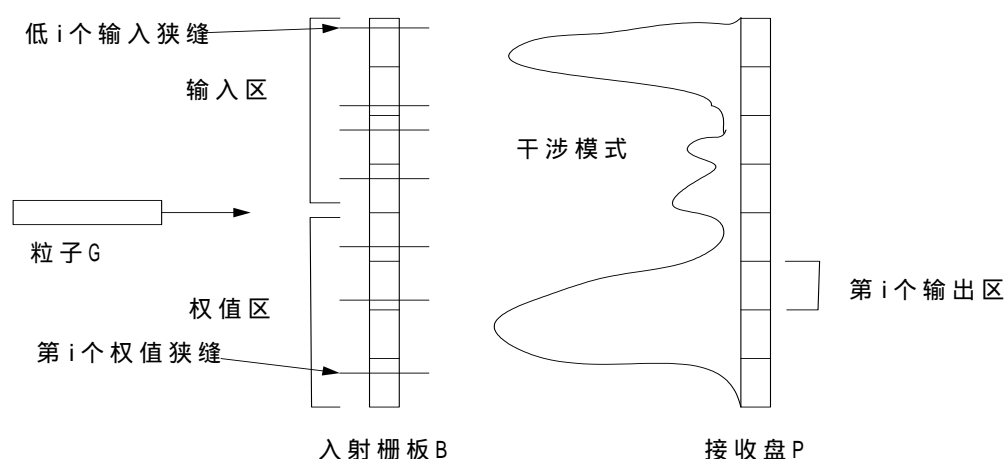


图 10-4 Chrisley 非叠加态量子神经计算模型

假设输入为  $\vec{x}$ , 输出为  $\vec{a}$ , 权值为  $\vec{\omega}$ , 干涉模式为  $\vec{p}$ , 这样可以得到  $S(\vec{x}, \vec{\omega}) = \vec{p}$ , 此处  $S(\cdot)$  为 sigmoid 函数, 而  $\vec{a} = O(S(\vec{x}, \vec{\omega}))$ 。再定义系统的误差函数为  $E = \sum_i (\vec{a}' - \vec{a})^2$ ,

其中  $\vec{a}'$  为期望输出, 然后利用类似于经典 BP 算法进行训练, 理论上就能够学习任意的函数映射关系。于是, 利用上述神经网络结构便实现了量子学习(Quantum Learning)。

Chrisley 模型巧妙地将量子实验的结构与神经计算的方式结合起来, 改造了传统的神经网络模型, 并从中产生一些新的特点。主要表现在: (1) 权值更新。它不同于传统神经网络中每个权值仅调节与之相联的节点的连接关系。在这里, 每改变权值区中的任何一个权值(即权值区狭缝的结构), 将会影响到所有输入节点的连接; (2) 输出。该模型的输出是几率幅的叠加, 本质上是复值的, 它增加了网络的信息容量; (3) 物理实现。它具有很直观的物理意义, 易于物理系统实现。

但是, 该模型也存在着以下几点问题: (1) 其前端未考虑量子态的叠加, 本意是为了避免相干态的维持问题, 但现实中量子态的叠加是客观存在的, 仅仅利用上述理论框架, 将会失去由量子态叠加而产生的计算优势, 如量子超并行、量子纠缠等等; (2) 虽然 Chrisley 给出了学习算法的大致框架, 但很难实现, 因为在该模型中很难设计出神经网络的量子对照物, 比如说输入量的表示, 由于粒子源是单一的, 它对于每个狭缝应该是等同的, 仅靠狭缝的不同位置不宜表征不同的输入量, 若采取粒子的入射角度或相位来表示, 就会导致计算复杂度的极大提高甚至无解。正是上述两点严重制约了 Chrisley 模型的应用。

### 2. 基于双缝干涉实验的量子神经网络模型

此处所提出的新模型也是建立在双缝干涉实验基础之上,并将它推广到多狭缝多栅板结构。考虑到 Chrisley 模型的缺陷,我们从网络的输入量着手,通过设计量子实验来寻找合适的神经网络对照物。由量子理论可知,当微观粒子进入不同介质后其运动轨迹会发生改变,我们就选取不同介质的介电常数矢量  $\vec{n} = (n_1, n_2, \dots, n_m)$  作为网络的输入量(其中  $m-1$  为栅板的个数),以粒子穿越栅板到达探测器的实际运动轨迹  $Tr(S \rightarrow s_{b_1}^1 \rightarrow \dots \rightarrow s_{b_j}^j \rightarrow \dots \rightarrow s_{b_{m-1}}^{m-1} \rightarrow D)$  作为网络的权值(实际上它最终由狭缝的位置  $s_{b_j}^j$  和栅板间距  $h_j$  所决定),取探测器所测得点的强度(即几率)  $P(\vec{n})$  作为网络的输出,以此来构造一个全新的量子神经网络模型,参见图 10-5 为了准确理解此模型的意义,我们必须将其几何构形与神经网络上的含义区分开来。

下面利用有关的量子理论来推导此模型的动力学表达式:

令  $l(s_{b_j}^j, s_{b_{j+1}}^{j+1})$  为连接第  $j$  个栅板上狭缝  $s_{b_j}^j$  与第  $j+1$  个栅板上狭缝  $s_{b_{j+1}}^{j+1}$  之间的路径长度,

其中  $b_j = 1, 2, \dots, N_j$ , 是第  $j$  个栅板上的狭缝,共有  $N_j$  个,  $r_{b_j}^j$  是狭缝  $s_{b_j}^j$  距离中心轴的距离。可以得到:

$$l(s_{b_j}^j, s_{b_{j+1}}^{j+1}) = \sqrt{h_j^2 + (r_{b_j}^j - r_{b_{j+1}}^{j+1})^2} \quad (10-3)$$

这里考虑了微观粒子的粒子性。

那么粒子沿轨迹  $l$  到达底板所测得的几率幅波函数为:

$$\psi(Tr_1) = \frac{1}{\prod_{j=1}^m l(s_{b_{j-1}}^{j-1}, s_{b_j}^j)} \times \exp(ik \sum_{j=1}^m n_j l(s_{b_{j-1}}^{j-1}, s_{b_j}^j)), \quad (10-4)$$

此处有  $s_{b_0}^0 \equiv S, s_{b_m}^m \equiv D$ 。这里考虑了微观粒子的波动性。

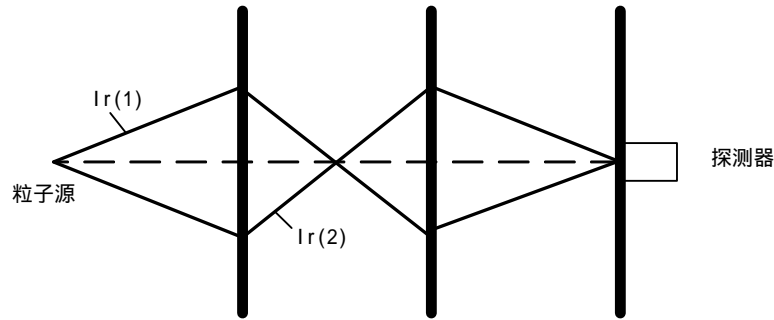


图 10-5 基于双缝干涉实验的 QNN 模型

再定义:

$$\theta = \ln\left(\prod_{j=1}^m l(s_{b_{j-1}}^{j-1}, s_{b_j}^j)\right) \quad (10-5)$$

$$\omega_j = kl(s_{b_{j-1}}^{j-1}, s_{b_j}^j) \quad (10-6)$$

则(10-4)式便可改写为:

$$\psi(Tr_1) = \exp(i \sum_{j=1}^m h_j \omega_j - \theta) = \exp(i \vec{W} \vec{n} - \theta) \quad (10-7)$$

最终可得神经网络的输出(即强度)为:

$$P(\vec{n}) = |\psi(Tr_1) + \psi(Tr_2) + \dots|^2 \quad (10-8)$$

可见方程(10-7)具有与经典神经元极其相似的数学表示形式,但又有所区别。鉴于其广义上均为复值形式,我们利用改进的模拟退火算法(MSA)来训练网络的权值,具体算法表述如下:

Begin

initialize  $\vec{\omega} = \vec{\omega}_0, T(0) = T_0, \tau_{\max}, \varepsilon$

set  $\tau = 0$

while  $\Delta E > \varepsilon$

begin

while  $\tau < \tau_{\max}$

begin

$\tau = \tau + 1, \omega_i = \omega_i + \Delta \omega_i, \Delta E = E(\vec{\omega} + \Delta \vec{\omega}) - E(\vec{\omega})$

prob =  $\min(1, e^{-\frac{\Delta E}{T}})$

if prob > random(0,1)

then updata( $\omega_i$ )

end

$T(t) = \alpha T(t)$

end

output  $\omega_i$

End

考虑到复值函数的计算复杂度,仅以 3 狭缝 1 栅板结构为例(即  $N_j = 3, m = 2$ ),训练

上述 QJNN 学习二值逻辑函数 AND( $n_1, n_2$ ) 和 XOR( $n_1, n_2$ )。其中,输入值范围取

$n_1, n_2 \in [1, 5/3]$ , 输出强度阈值取 0.6。

特别地,对于二值 XOR 函数,考察其最简单的 QNN 模型(即  $N_j = 2, m = 2$ )。我们可以直接从物理公式出发进行推导,由于微观粒子运动的两条轨迹分别为:

$$\psi(Tr_1) = \frac{1}{l_1^1 l_1^2} \exp(ik(n_1 l_1^1 + n_2 l_1^2)),$$

$$\psi(Tr_2) = \frac{1}{l_2^1 l_2^2} \exp(ik(n_1 l_2^1 + n_2 l_2^2)) . \quad (10-9)$$

设  $r = r_1 + r_2, h_1 = h_2 = h$  , 则可得量子神经网络的输出为 :

$$P(n_1, n_2) = |\psi(Tr_1) + \psi(Tr_2)|^2 \cong \frac{1}{(r_2 + h)^2} |1 + \cos k(n_1 + n_2)(Tr_1 - Tr_2)|^2 \quad (10-10)$$

令  $A = \frac{1}{(r_2 + h_2)^2}, Tr_1 = Tr_2 + \frac{3}{4}\lambda, n_1 = 1, n_2 = 5/3$  , 于是有

$$P(n_1, n_1) = A|1 + \cos 3\pi|^2 = 0 ;$$

$$P(n_1, n_2) = A|1 + \cos 4\pi|^2 = 4A ;$$

$$P(n_2, n_1) = A|1 + \cos 4\pi|^2 = 4A ;$$

$$P(n_2, n_2) = A|1 + \cos 5\pi|^2 = 0 . \quad (10-11)$$

再令  $A = 1/4$  , 就可以得到表 10-2 所示的结果 :

表 10-2 类似 XOR 逻辑真值表

输出	输入 1	输入 2
0	1	1
1	1	5/3
1	5/3	1
0	5/3	5/3

可见, 两层结构的量子神经网络可以完成类似于 XOR 的逻辑运算, 这在传统神经网络中是无法实现的。另外, 从上述推导过程中也可以看到, 正是由于 QNN 中包含了量子态波函数的几率幅叠加, 在输出层出现了二次交叉乘积项, 提高了网络的信息容量, 使它具备了量子计算的优越性。

实际上, 该网络还可以看作是利用神经网络来调节量子实验装置的参数, 从而实现量子学习的功能。对于本文所提出的量子神经网络结构, 为调整网络权值, 需要改变狭缝的位置和栅板的间距, 这在光学实验设计中可以通过调节相应的透镜及偏振片的参数来完成, 因此具有易于物理实现的潜力。它能够作为量子系统实现神经计算的一个范例, 这为探索神经网络与量子理论的结合提供了一个新的途径。

### 10.3.2 多层激励函数的量子神经网络模型及其在故障诊断中应用

1997 年 N.B.Karayiannis 等人<sup>[8-12]</sup> 借用了量子理论中量子态叠加的思想, 提出了基于多层激励函数的量子神经网络模型, 在三层的神经网络结构中, 隐层量子神经元的激励函数采用多个传统激励函数的叠加, 使网络有了一种固有的模糊性。理论和实验均证明多层激励函数的量子神经网络模型对具有不确定性、两类模式之间存在交叉数据的模式识别问题, 有极好的分类效果。目前已成功地应用于手写字识别、气象预测及语音识别等领域<sup>[11-12]</sup>。此处将其应用于多传感器数据融合之中, 利用多层激励函数的量子神经网络模型的特有辨识属性和多传感器融合思想, 来减少多数模混合电路故障识别的模糊性和不确定性, 提高故障定位的准确性。

### 1. 常规 BP 神经网络数据融合故障诊断算法

为便于对比分析,这里再介绍一下 BP 神经网络数据融合故障诊断算法。典型的 BP 网络是三层前馈网络,即输入层,隐含层和输出层。如图 10-6 所示,其中输入层对应于故障现象,输出层对应于故障原因,设输入层 LA 有  $m$  个节点 ( $m$  等于传感器数目与待诊断元件数目  $n$  之积),输出层 LC 有  $n$  个节点 (等于待诊断元件数目),隐含层 LB 的节点数目为  $u$ 。隐含层中的节点输出函数为:

$$b_r = f(W^T X - \theta) \quad (r=1, \dots, u) \quad (10-12)$$

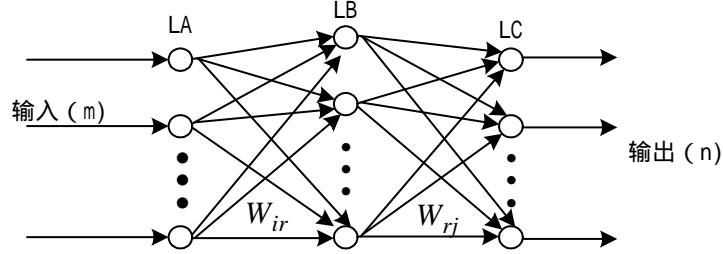


图 10-6 三层 BP 神经网络结构

输出层中节点的输出函数为

$$c_j = f(V^T B - \varphi) \quad (j=1, \dots, n) \quad (10-13)$$

上式中  $f$  采用 S 型函数即  $f(x) = (1 + e^{-x})^{-1}$ ,  $W_{ir}$  为输入层神经元  $a_i$  到隐含层神经元

$b_r$  间的连接权,  $V_{rj}$  为隐含层神经元  $b_r$  到输出层神经元  $c_j$  间的连接权,  $\theta_r$  为隐含层的阈

值,  $\varphi_j$  为输出层单元的阈值。

神经网络信息融合,是将神经网络(如 BP 网络,量子神经网络)引入信息融合之中,同时结合模糊集合论进行故障判断。其具体过程为:通过多传感器测试被诊断对象,求出每一传感器对故障集中各类故障的隶属度值,将所有传感器的故障隶属度值矢量作为神经网络的输入,网络输出即为融合后该症状属于各类故障的隶属度值矢量,最后利用基于规则的判定原则进行故障决策,其融合诊断过程如图 10-7 所示。其中  $A_1, A_2, \dots, A_n$  为待诊断的故障模式。

$\mu_j(A_1), \mu_j(A_2), \dots, \mu_j(A_n)$  为传感器  $j$  测得的各故障模式  $A_1, A_2, \dots, A_n$  所得的隶属度值,

$\mu(A_1), \mu(A_2), \dots, \mu(A_n)$  为融合的隶属度值。

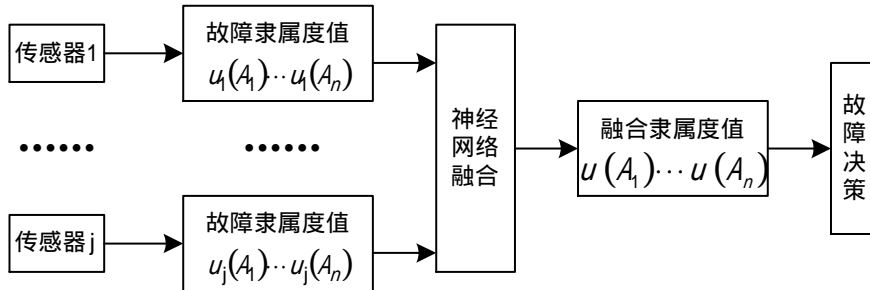


图 10-7 神经网络数据融合故障诊断



## 2. 基于多层激励函数的量子神经网络

多层激励函数的量子神经网络,主要是隐层量子神经元的激励函数采用多个传统激励函数的叠加,使网络有了一种固有的模糊性,它能将决策的不确定性数据合理地分配到各故障模式中,从而减少模式识别的不确定度,提高模式识别的准确性。此处(10-12)式变为:

$$b_r = \frac{1}{ns} \sum_{s=1}^{ns} f[\beta(W^T X - \theta_s)] \quad (10-14)$$

式中  $\theta_s$  为量子间隔,  $s$  为量子间隔数目,其大小的选择与待诊断的故障模式数目相同,  $\beta$  为陡度因子。多层激励函数的量子神经网络模型对具有不确定性、两类模式之间存在交叉数据的模式识别问题,有极好的分类效果。将其与多传感器数据融合相结合,应用于数模混合电路的故障识别,必将能消除由于单传感器提供信息不完全及相邻元件相互影响而出现误判断现象。提高故障诊断的准确率。

## 3. 网络训练算法

多层激励函数的量子神经网络模型的训练算法仍采用梯度下降法。每个训练周期中,训练算法不仅更新不同层神经元之间的连接权,而且更新隐层各神经元的量子间隔。前者与常规 BP 网络更新算法完全相同,后者隐层神经元的量子间隔更新算法如下<sup>[12,41]</sup>:

对模式类矢量  $C_m$  ( $m$  为模式类数目), 隐层第  $i$  个神经元的输出变量有:

$$\sigma_{i,m}^2 = \sum_{x_k \in C_m} (<O_{i,m}> - O_{i,k})^2 \quad (10-15)$$

上式中  $O_{i,k}$  为在输入矢量为  $x_k$  时, 隐层第  $i$  个神经元的输出。 $<O_{i,k}>$  为:

$$<O_{i,m}> = \frac{1}{|C_m|} \sum_{x_k \in C_m} O_{i,k}, \quad (10-16)$$

通过最小化  $\sigma_{i,m}^2$ , 我们能够得到隐层量子间隔  $\theta_{i,s}$  的更新方程, 隐层第  $i$  个神经元的  $s$  个量子间隔为:

$$\Delta\theta_{i,s} = \eta \frac{\beta}{ns} \sum_{m=1}^n \sum_{x_k \in C_m} (<O_{i,m}> - O_{i,k}) * (<v_{i,m,s}> - v_{i,k,s}) \quad (10-17)$$

(6)式中  $\eta$  是学习率,  $\beta$  为陡度因子,  $n$  是输出神经元数目, 如故障模式数目。

$$<v_{i,m,s}> = \frac{1}{|C_m|} \sum_{x_k \in C_m} v_{i,k,s}; \quad v_{i,k,s} = O_{i,k,s} (1 - O_{i,k,s}) \quad (10-18)$$

$O_{i,k,s}$  为在输入矢量为  $x_k$ , 第  $s$  个量子间隔时, 隐层第  $i$  个神经元的输出。

## 4. 旋转机械故障诊断实例分析

大型旋转机械是电力、冶金、化工生产中的常见而重要的设备,对它们进行实时状态监测和故障准确诊断一直是工业生产的重要内容。在此以实验室中旋转机械振动实验台为仿真对象设计实现了一套基于多层激励函数的量子神经网络故障诊断实验系统,此实验平台是通过变频器控制电机使转子可以在不同转速下转动,同时可以在转子上固定不同质量

的吸盘人为设置故障模式，如：转子零件松动、转子不平衡、转子不对中等。在故障诊断中，利用两种传感器即速度传感器和加速度传感器得到转子的振动信号，经分析处理后得到每种传感器对各种故障模式的隶属函数值分配，用多层激励函数的量子神经网络进行融合，由融合隶属函数值进行故障模式判定。其诊断系统结构如图 10-7 所示。

### 故障隶属函数的形式

隶属度函数形式的确定主要应考虑两个因素：a)传感器本身的工作性能；b)传感器工作时的各种干扰情况如机械噪声、电磁波的影响。若仅考虑第二种因素，则传感器对同一种物理量的测定值的概率密度函数形式一般可确定为高斯分布。然而，一般的传感器，由于机械、温度、压力等原因使其输出产生线性漂移，也就是说，某传感器在不同时间对同一物理量进行测定时，其平均值也在一定范围内产生漂移，这一现象使得各个传感器对同一物理量测定结果  $f(x)$  符合图 10-8 的分布。

显然，要用数学表达式来精确地表达这一模型，并把它作为隶属度函数，其形式就显得过于繁琐，而且也无必要。此处采用梯形函数对这一模型进行理想化如图 10-9 所示，并把它作为各模式特征的隶属度函数，用数学公式可表示如下：

$$\mu_{ij} = \begin{cases} 0 & x_j \leq x_0 - t_{ij} \\ (x_j - x_0 + t_{ij}) / (t_{ij} - e_{ij}) & x_0 - t_{ij} < x_j \leq x_0 - e_{ij} \\ 1 & x_0 - e_{ij} < x_j \leq x_0 + e_{ij} \\ -(x_j - x_0 - t_{ij}) / (t_{ij} - e_{ij}) & x_0 + e_{ij} < x_j \leq x_0 + t_{ij} \\ 0 & x_j > x_0 + t_{ij} \end{cases} \quad (10-19)$$

式中  $\mu_{ij}$  为传感器 j 测定被测对象属于 i 模式的隶属度， $x_j$  表示传感器 j 测定的实际特征值，

而  $x_0, e_{ij}, t_{ij}$  则分别表示为传感器 j 测定被测目标隶属于 i 模式的标准特征值，特征允许误差和特征最大偏差。 $x_0$  的值可用传感器对某一类模式测定值的数学期望来代替； $e_{ij}$  的值可

通过求图 10-8 中概率分布函数的双侧百分点来实现。 $t_{ij}$  的值则可通过求样本极差得到。

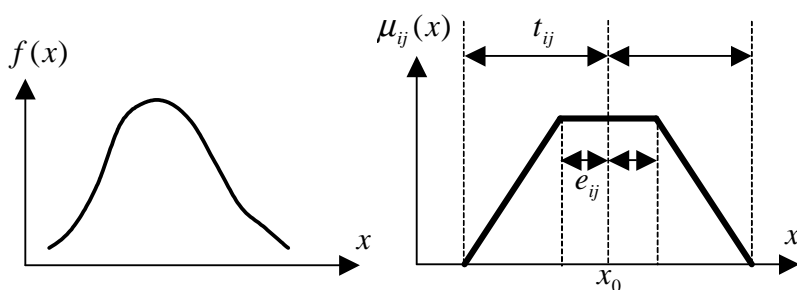


图 10-8 传感器测定值分布

图 10-9 特征值隶属度分布函数

### 目标模式判定规则

对神经网络融合后输出的各模式属于故障的隶属度值，在故障模式决策时，其基本原则有：

- (1) 判定的目标模式应具有最大的隶属度值。
- (2) 判定的目标模式的隶属度值要大于某一阈值。一般来说此值至少要大于  $1/n$  ( $n$  为待诊断模式个数)，阈值越大故障判断的准确性越高，只是取得过大时测试得到的故障隶属

度值无法满足要求，因此必须针对实际被诊断对象，选择一大于  $\frac{1}{n}$  的适当数值。此处

取此值为 0.7。

(3) 判定的目标模式和其它模式的隶属度值之差要大于某个门限。此门限的取值类似于条件(2)。对后面的诊断实例取其值为 0.5。

### 实验设计

实验时，人为设置了 3 种不同类型故障进行模拟实验，它们是：转子不平衡、转子不对中、转子零件松动，再加入设备无故障情形，共 4 种模式。通过速度和加速度传感器分别测试转子在各种状况下的参数值，利用梯形函数，得到各故障模式特征值的隶属函数，进一步计算出每种传感器对各故障模式的隶属函数值分配。

本实例中有 4 个目标模式：转子不平衡、转子不对中、转子零件松动、无故障，再加一不确定度数值；2 种传感器，因此网络输入神经元个数为  $2 \times 5 = 10$ ，分别对应 2 传感器测得的各情况的隶属函数值，输出神经元个数为 5，分别代表 4 种目标模式和 1 个不确定数值，不确定度大小为 1 减去其它模式隶属函数之和。隐层神经元个数取 12，相应隐层神经元的量

子间隔数  $ns = 5$ ，其初始量子间隔  $\theta_s$  为一随机数，陡度因子  $\beta = 1$ 。在不同的转速下，通过

人为设置不同故障模式，可以得到不同故障模式时的训练样本。样本中输入为传感器测试的各个模式隶属函数值（包括无故障模式和不确定度项），输出代表待诊断设备的是否属于故障的状态，样本中对应实际目标模式的神经元输出为 1，其他神经元输出为 0。

### 结果讨论

表 10-3 为分别利用 DS 证据理论<sup>[42]</sup>、常规 BP 神经网络、量子神经网络进行信息融合的诊断结果及单传感器诊断结果。其中第一行，第二行为速度和加速度传感器所测得的各目标模式隶属函数值，第三、第四、第五分别为 DS 证据理论融合、BP 融合和量子神经网络融合后属于各目标模式的隶属函数值。显而易见本设备被诊断的 3 个故障模式中，两种传感器单独提取的隶属函数值有的很相近，且其不确定度数值均较大，如果只用一种传感器的隶属函数分配来识别故障模式，就会出现无法确定故障模式的状况。

表 10-3 单传感器及信息融合诊断诊断结果

故障模式	传感器	信度函数				不确定度	诊断结果
		不平衡	不对中	零件松动	无故障		
不平衡	速度	0.2060	0.1519	0.1462	0.1326	0.3633	不定
	加速度	0.3709	0.0907	0.2378	0.1024	0.1983	不定
	DS 融合	0.4098	0.1225	0.2394	0.1138	0.1144	不定
	BP 融合	0.6478	0.0986	0.1107	0.0589	0.0860	不定
	量子融合	0.7208	0.0890	0.0976	0.0556	0.0370	不平衡
不对中	速度	0.2928	0.4680	0.2269	0.0092	0.0031	不定
	加速度	0.1630	0.1679	0.1645	0.1243	0.3803	不定
	DS 融合	0.2917	0.4670	0.2269	0.0091	0.0022	不定
	BP 融合	0.1232	0.7108	0.1023	0.0062	0.0575	不对中
	量子融合	0.0897	0.7936	0.1106	0.0051	0.0012	不对中
零件松动	速度	0.1352	0.1411	0.3115	0.1103	0.302	不定
	加速度	0.1827	0.1769	0.2240	0.1035	0.3111	不定
	DS 融合	0.1873	0.1878	0.3599	0.1182	0.1443	不定
	BP 融合	0.1289	0.1345	0.5497	0.0846	0.1024	不定
	量子融合	0.1023	0.0873	0.7036	0.0402	0.0676	零件松动

从表 10-3 可以明显看出,无论是 DS 证据理论融合、BP 融合还是量子神经网络融合,其融合后隶属函数值和单传感器的隶属函数值相比,增加了实际故障模式的隶属函数值,相对减少了其它模式的隶属函数值,同时使设备故障诊断的不确定性大幅度降低。也就是说,基于多传感器信息融合增加了设备的可分析性,有效地提高故障模式的识别能力,使故障诊断的准确率得以提高。

从实验结果还可以看出,多层激励函数的量子神经网络信息融合结果要比 DS 证据理论融合、常规 BP 神经网络融合结果好。当故障判定阈值取 0.35,在不考虑“目标模式和其它模式的隶属度值之差门限”这一条件时,无论是 DS 证据理论融合、BP 融合还是量子神经网络融合,其结果都能准确判定故障模式;当故障判定条件阈值提高时,如取 0.7 时,量子神经网络融合仍能准确识别故障模式,BP 融合只能部分识别故障模式(如转子不对中故障),而 DS 证据理论融合则无法进行故障诊断。这充分反应了多层激励函数的量子神经网络模型对具有不确定性、两类模式之间存在交叉数据的模式识别问题,有极好的分类效果。

### 本章小结

量子神经网络大致可以分为三种类型:参照传统神经网络的学习算法(如 BP、SOM 算法等),构造其在量子领域中的对照物;利用量子理论中的概念、方法(如态叠加、“多宇宙”观点等),建立新的网络模型;借鉴量子理论中的物理实验或假想的理论模型(如 EPR 隐形传态等)来构造计算模型。

本章在综述量子神经网络研究状况的基础上,进一步介绍了几种量子神经网络模型。最后给出了两个具体应用实例。

### 参考文献

- [1] 边肇祺,张学工等。模式识别。北京:清华大学出版社
- [2] 阎平凡,张长水。人工神经网络与模拟进化算法。北京:清华大学出版社
- [3] 孙增圻等。智能控制理论与技术。北京:清华大学出版社,广西科学技术出版社
- [4] 王凌。智能优化算法及其应用。北京:清华大学出版社,施普林格出版社
- [5] 周树德,王岩,孙增圻,孙富春,量子神经网络,2003 年,中国智能自动化会议论文集, CIAC2003, PP: 163-168, 中国香港。
- [6] Ventura Dan and Tony Martinez, An Artificial Neuron with Quantum Mechanical Properties. Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, Norwich, England, 1997.
- [7] Ventura Dan and Tony Martinez, Quantum Associative Memory with Exponential Capacity. Proceedings of the International Joint Conference on Neural Networks, Anchorage, Alaska, 1998, PP.509-513.
- [8] Ventura Dan and Tony Martinez, A Quantum Associative Memory Based on Grover's Algorithm. International Joint Conference on Artificial Neural Networks and Genetic Algorithms, Portoroz, Slovenia, April 1999.
- [9] Ventura Dan and Tony Martinez, Quantum Associative Memory, Information Science. 2000, 124:273-296,
- [10] Andre cut M and Ali M K, A Quantum Neural Network model, International Journal of Modern Physics, 2002, 13(1):75-88.
- [11] Karayiannis N B, Purushothaman G, Fuzzy pattern classification using feedforward neural networks with multilevel hidden neurons, IEEE International Conference on Neural Networks. Vol.3, 27 jun-2 Jul 1994.

- [12]Gopathy P ,Nicolaos B., Karayiannis ,Quantum Neural Networks( QNN's ) :Inherently Fuzzy Feedforward Neural Networks. IEEE Transactions on Neural Networks. 1997, 8 ( 3 ): 679-693.
- [13]Kretzschmar R , Bueler R , Karayiannis N B , Eggimann F., Quantum neural networks versus conventional feedforward neural Networks: an experimental study. Neural Networks for Signal Proceedings of the 2000 IEEE Signal Processing Society Workshop, 2000, Vol.1.
- [14]Kretzschmar R; Karayiannis N B;Richner H., NEURO-BRA:a bird removal approach for wind profiler data based on quantum neural networks. Proceedings of the IEEE-INNS-ENNS International Joint Conference, 2000, Vol.4.
- [15] Kretzschmar R; Karayiannis N B;Richner H., A comparison of feature sets and neural networks classifiers on a bird removal approach for wind profiler data. Proceedings of the IEEE-INNS-ENNS International Joint Conference , 2000,Vol.2.
- [16]Zhou J., Qian G., Adam Krzyzak,, Ching Y S., Recognition of handwritten numbers by Quantum Neural Networks with fuzzy features., 1999, IJDAR.2:30-36.
- [17]Zhou J. Automatic, Detection of premature ventricular contraction using quantum Neural Networks, Bioinformatics and Bioengineering, Proceedings of Third IEEE Symposium on Neural Networks, 2003.
- [18]Li F. Zhao S G.Zheng B Y., Quantum neural networks in speech recognition.2002 6th International Conference on Neural Networks., 2000, Vol.2.
- [19]Nobuyuki Matsui. Masato Takai,Haruhiko Nishimura., A network model based on qubit like neuron corresponding to quantum circuit., Electronics and communications in japan, 2000, Part 3,83(10).
- [20] Nobuyuki Matsui., Masato Takai, Haruhiko Nishimura., Learning performance of neural model based on quantum superposition., Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication Osaka; Japan, 2000, September 27-29.
- [21]Nobuyuki Matsui., Masato Takai,, Haruhiko Nishimura., Neural network based on QBP and its performance., IEEE International Workshop on Robot and Human Interactive Communication Osaka; Japan, 2000
- [22]Noriaki Kouda, Nobuyuki Matsui., Masato Takai, Image Compression by Layered Quantum Neural Networks, Neural Processing Letters. 2002, 16:67-80.
- [23]Kak S., Quantum neural computing. Advances in Imaging and Electron Physics. 1995, 94:259-313.
- [24]Tad Hogg, Subhash Kak, Dan Ventura,et.al., A quantum leap for AI.IEEE Intelligent Systems 1999.
- [25]Subhash Kak., From Many to One: On Starting a Computation., Department of Electrical and Computer Engineering Louisiana State University, 2001, December 12.
- [26]Subhash Kak., Active Agents, Intelligence and Quantum Computing. Department of Electrical and Computing Engineering Louisiana State University, 2000, January 5.
- [27]Behrman E C, Chandrashekar VG, Wang Z, Belur C K. and Skinner S R., A Quantum Neural Network Computer Entanglement., Physical Review Letters, 2002.
- [28]Allauddin R, Gaddam K, Behrman E C, Steck J E. and Skinner S R., Advantages of Quantum Recurrent Networks: An Examination of Stable States, Proceedings of the International Joint Conference on Artificial Neural Networks, July 2002.
- [29]Chandrashekar V G, Behrman E C, and Steck J E., Physical Realizations of a Temporal Quantum Neural Computer(Invited Paper), Proceedings of the 2001 International Joint

- Conference on Neural Networks, Washionton,DC, July 15-19,2001.
- [30] Behrman E C, Nash L R, Steck J E, Chandrshekar V, Skinner S R., Simulations of Quantum Neural Networks, Information Sciences, 2000, 128-257.
- [31]Cruz-Cabrera A, Steck J E, Behrman E C, Skinner S R. Backpropagation Training for an Optical Neural Networks Using Self-lensing Effects. IEEE Transactions on Neural Networks, 2000, Vol.11.
- [32]Steck J E, Skinner S R, Cruz-Cabrera A, Yang M., Behrman E C., Spatially Continuous Learning Systems: Artificial Neural Networks in a Bulk Material Continuum. Ninth International Conference on Artificial Neural Networks(ICANN99).
- [33]Behrman E C, Steck J E, Skinner S R., A Spatial Quantum Neural Computer. Proceedings of the 1999 International Joint Conference on Artifical Neural Networks.
- [34] Skinner S R, Steck J E, Behrman E C, An optical Neural Network using krr-type nonlinear materials. IEEE Micro., 1995, Vol.15.
- [35]<http://webs.wichita.edu/physics/behrr.htm>
- [36]Ajit Narayanan and Tammy Menneer., Quantum Artificial Neural Network architectures and components., Information Science.,2000, PP.231-255.
- [37]Ezhov A., Analog quantum neuron for functions approximation, Neural Networks Proceedings. IJCNN'01 , 2001
- [38]<http://www.isi.ee.ethz.ch/~kretzsch>
- [39]Grover L., A Fast Quantum Mechanical Algorithm for Database Search., Proc. ACM Symp., Theory of Computing, ACM Press, New York,212-219,1996
- [40] 解光军, 杨俊安等, 基于量子双缝干涉实验的神经网络模型, 模式识别与人工智能, 2003, 16 ( 1 ): 28-32。
- [41] Duda R.O., Hart P.E., Pattern classification and scene analysis, John Wiley & Sons, New York, 1973.
- [42] 朱大奇, 纪志成, 基于多智能体的复杂工程系统故障诊断研究, 计算机集成制造系统—CIMS, 2004, 10(6):699-703.

### 思考题

- 1 . 何为神经网络的灾变性失忆现象？
- 2 . 多层激励函数的量子神经网络的主要特点是什么？
- 3 . 多宇宙的量子神经网络模型有何特点？