

人工神经网络原理 ——入门与应用

王 伟 编著

北京航空航天大学出版社

内 容 提 要

本书是人工神经网络理论的入门书籍。全书共分十章。第一章主要阐述人工神经网络理论的产生及发展历史、理论特点和研究方向；第二章至第九章介绍人工神经网络理论中比较成熟且常用的几种主要网络结构、算法和应用途径；第十章用较多篇幅介绍了人工神经网络理论在各个领域的应用实例。

本书可作为理工科大学及农林医科大学计算机、自动控制、信号与信息处理、电路与系统、系统工程、经济管理等有关专业的博士生、硕士生、高年级大学生的教材，同时也可以作为有关学科领域研究人员及工程技术人员学习人工神经网络理论的入门参考书。

前 言

人工神经网络(Artificial Neural Networks)理论是 80 年代中后期世界范围内迅速发展起来的一个前沿研究领域,其发展已对计算机科学、人工智能、认知科学等领域产生了重要影响。近年来,在欧美和日本等国掀起了人工神经网络研究、开发应用的热潮,有关人工神经网络理论的新的研究成果不断涌现。值得注意的是,目前我国人工智能及其它相关学科领域的专家、学者、工程技术人员在人工神经网络理论和应用研究方面,也做出了许多可喜的成绩。

人工神经网络理论是在怎样的科学背景下产生的呢?要回答这个问题,首先要明确什么是“智能”和“智能理论”?虽然到目前为止对“智能”还没有一个统一、确切的定义,但简单说来,智能是指人们认识客观事物并运用知识解决实际问题的能力。它表现为运用知识认识新情况、解决新问题、学习新方法、预见新趋势、创造新思维的能力。智能的高低反映在对客观事物认识的深刻、正确、全面的程度以及运用知识解决实际问题的速度和质量上。有了对什么是“智能”的解释,就不难推而得知什么是“智能理论”。简言之,“智能理论”包括两个基本问题:探索人类智能的奥秘(研究人类的认识过程)和运用人工手段模仿人类的智能行为。在对后一问题的研究上又有两种主导思想:结构主义——从硬件结构上模拟人脑的构成;功能主义——撇开人脑的具体结构,仅从输出输入关系上构造出与人脑功能相一致的人工智能系统。功能主义成了传统人工智能理论的研究基础。这种理论的核心就是以冯·诺依曼

型数字计算机为支持,编制出本领超凡的软件程序,以此达到模拟人脑智能的目的。传统人工智能理论在求解高精度计算问题以及过程模拟、过程控制等方面取得了巨大的成功。但是,这种理论也遇到了许多它无法克服的问题,对此读者可从本书第一章结论中详细了解到。与传统的人工智能理论相对应,以结构主义为主导思想,从分析人脑神经网络的微观结构上入手,抓住人脑结构的主要特征,即大量相对简单的非线性神经元之间复杂而又灵活的联接关系,深刻揭示人脑认识过程的结构作用的研究——人工神经网络理论研究,成为人工智能理论研究的一大分支。正如本书始终贯穿的一个思想,人工智能理论研究的这两大分支,不是相互替代、相互对立的关系,而是相辅相成、相互补充的关系。它们有着共同的研究基础,这就是对信息的处理;所不同的只是处理方法上的差别。

对于人工神经网络理论的研究,从 80 年代开始获得蓬勃发展,但人类对这一理论的探索可追溯到 1943 年,法国心理学家 McCulloch 和数学家 Pitts 合作提出的神经元数学模型,即著名的 M-P 模型。这一模型的提出,开创了人工神经网络这一新兴边缘学科研究的先河。在这之后,有关研究异常活跃,一系列关于神经元及神经网络的模型、计算方法被提出来,并引起学术界的极大兴趣。

但是,到了 70 年代后期,由于当时冯·诺依曼数字计算机出色的性能表现和各种以数字计算为研究基础的人工智能理论的日臻完善,使人们渐渐地对人工神经网络理论的研究失去了以往的热情。特别是 1969 年,人工智能理论界具有权威地位的 Minsky 和 Papert 在《感知机》一书中指出了在当时看来是无法克服的神经元模型的致命缺陷——感知机只能对线性可分模式进行识别,而对象“异或”(XOR)这样简单的非线性逻辑问题却都束手无策。在这之后的 10 多年的时间里,人工神经网络理论的研究一直处于低潮。但是,进入 80 年代以后,由少数坚韧不拔的学者经过 10 多

年的理论研究积蓄,终于在人工神经网络理论的一系列问题上取得了重大突破,给这一理论的研究重新带来了生机。其中尤其值得一提的是美国生物物理学家 Hopfield 于 1982 年提出的一种被人们称为 Hopfield 的模型。他在其建立的网络模型中引入“能量”函数,在理论上解决了网络稳定性的判据问题,并使网络具有联想记忆、自动分类和自动校正记忆误差的能力。他的研究成果开拓了人工神经网络理论研究的新途径,令人耳目一新。当 1984 年他用模拟电子电路实现了这一网络,并用网络计算方法成功地解决了“巡回推销员问题”之后,人工神经网络理论的研究进入了一个崭新的、突飞猛进的时代。当今,比较成熟的网络模型及相应算法有近百种,各种修正和演变的模型、算法就更多。美、日等国还推出了各种类型的神经元集成电路芯片和神经网络计算机。虽然目前还很难预料人工神经网络理论究竟能对世界科学的发展起到何种推动作用,但是,这种以人脑本身组织结构,思维原理为研究基础,并对其高度抽象和合理简化、概括的探索,无疑是一个富有挑战性,并将深刻影响对人类认识行为理解的伟大研究工程。

本书是在综合国内外有关理论书籍和文献并结合作者的实际研究工作的基础上编写的。其主要目的是把人工神经网络理论这一新兴的科学理论,以尽可能简明、通俗的语言及尽可能少的预备知识介绍给广大读者。作为一本入门参考书,作者没有过于追求数学推导的严谨和理论阐述的深度,而是以由浅入深、形象生动、通俗易懂的方式,使读者在没有太多理论障碍的情况下,准确地掌握人工神经网络的基本原理、应用方法及发展趋势,并能学以致用,将这一理论与读者所从事的研究工作有机地结合起来,发挥其应有的作用。

北京科技大学博士生导师,中国人工智能学会理事长涂序彦教授对本书给予了热情关怀和悉心指导。国家教委 1994 年度留学回国人员科研基金资助的研究课题,对本书中有关应用实例的写作,提供了很大帮助。在此谨致谢意。

由于作者学术水平所限,加之人工神经网络理论本身还处于不断发展、完善的阶段,书中的错误和不妥之处在所难免。恳切欢迎读者提出批评、指正。

作 者

1995 年 4 月

目 录

第一章 绪 论

1.1	一则现代童话的启示	1
1.2	大脑——一个神秘的世界	3
1.3	神经网络研究、发展的历史	6
1.4	神经网络研究的主要方向	9

第二章 神经网络理论基础

2.1	大脑与生物神经系统	10
2.2	生物神经系统的模型化	16
2.3	M-P 神经元模型与人工神经网络的构成	20
2.4	人工神经网络的学习机理与 Hebb 学习规则	22
2.5	教师示教学习与无教师示教学习	24
2.6	生物神经网络与人工神经网络的比较	25
2.7	模式识别的基本定义与方法	28
2.8	线性分类器	31

第三章 神经网络的初期模型与基本算法

3.1	感知机模型与感知机学习规则	33
3.2	感知机的局限性	41
3.3	自适应线性神经网络	44
3.4	自适应线性神经网络的应用举例	50

第四章 多阶层神经网络与误差逆传播算法

4.1	多阶层网络与误差逆传播算法的提出	52
4.2	误差逆传播神经网络结构与学习规则	53
4.3	误差逆传播学习规则的数学推导	62
4.4	隐含层——特征抽取器的作用	67
4.5	BP 网络应用举例	70
4.6	BP 网络小结	73
4.7	几种改进方案	74

第五章 Hopfield 神经网络

5.1	Hopfield 网络的基本思想	77
5.2	Hopfield 网络的结构与算法	78
5.3	Hopfield 网络运行规则	80
5.4	网络计算能量函数与网络收敛	81
5.5	联想记忆	86
5.6	Hopfield 网络联想记忆的设计方法	89
5.7	Hopfield 网络联想记忆的缺陷	97
5.8	连续时间型 Hopfield 神经网络	100
5.9	Hopfield 网络在优化组合问题中的应用	103
5.10	网络应用与网络能量函数	110

第六章 随机型神经网络

6.1	随机型神经网络的基本思想	111
6.2	模拟退火算法	112
6.3	Boltzmann 机与 Boltzmann 机工作规则	115
6.4	Boltzmann 机学习规则	121
6.5	网络小结	130

第七章 竞争型神经网络

7.1	竞争型神经网络的基本思想	132
-----	--------------------	-----

7.2	基本竞争型神经网络及学习规则	133
7.3	抑制竞争型神经网络及学习规则	138
7.4	自适应共振理论网络的提出及特点	141
7.5	自适应共振理论网络结构及学习、工作规则	142
7.6	自适应共振理论网络特性分析	149
7.7	应用举例	154

第八章 自组织特征映射神经网络

8.1	自组织特征映射神经网络的基本思想	157
8.2	自组织特征映射网络学习、工作规则	159
8.3	SOM 网络的自组织特性	165
8.4	网络的分类精度分析	168
8.5	SOM 网络的局限性	170
8.6	SOM 网络的有教师示教学习规则	171
8.7	SOM 网络应用举例——语音打字机	171

第九章 对向传播神经网络

9.1	对向传播神经网络的基本思想	175
9.2	CP 网络学习、工作规则	176
9.3	网络应用举例	183
9.4	网络的改进与完善	185

第十章 人工神经网络应用

10.1	文字识别	189
10.1.1	文字识别的生理特点	189
10.1.2	印刷体文字识别	190
10.1.3	手写体文字识别	196
10.2	图象处理	199
10.2.1	图象处理概述	199
10.2.2	图象数据压缩	201
10.2.3	图象边缘检测	205

10.2.4	图象自动分类	206
10.2.5	医学自动诊断	207
10.2.6	目标自动识别	207
10.2.7	图象补正	208
10.2.8	工业产品检查	208
10.3	优化组合应用——有价证券的选择	209
10.4	神经网络专家系统	215
10.4.1	神经网络专家系统的基本原理与结构	215
10.4.2	高炉炉温预测专家系统	219
10.4.3	财务评价专家系统	221
10.4.4	医疗诊断专家系统	223
10.5	神经网络在智能控制中的应用	226
10.5.1	神经网络控制系统的特点	226
10.5.2	神经网络控制系统的几种结构形式	227
10.5.3	控制系统中神经网络的学习结构与方法	229
10.5.4	应用神经网络进行系统辨识	234
10.6	神经网络计算机简介	235
10.6.1	神经网络计算机的基本特征	235
10.6.2	神经网络计算机的分类与研究现状	237
10.6.3	神经网络计算机的有关概念	239
参考文献		241

第一章 绪 论

1.1 一则现代童话的启示

“X 队和 Y 队是两支势均力敌的足球劲旅,在新赛季决战之前,两队都摩拳擦掌试图问鼎冠军宝座。其中 Y 队为了做到万无一失,煞费苦心准备了一件秘密武器:一个装配有高速大容量计算机的机器人——门将 W。然而 W 在绿茵场上的表现却令人大失所望,面对飞速而来的足球,尽管他使尽浑身解数,以每秒百亿次的运算速度,对由视觉传感器获得的足球方位、速度等数据进行计算,以求作出扑球的行动决策。但是屡屡不等他作出正确反应,足球早已破门而入。与此相反,X 队经验丰富的守门员 Q,反应灵敏,左右逢源,频频将来自各个方向、风驰电掣般的足球拒之门外,比赛结果可想而知。Y 队机关算尽却大败而归,恼羞成怒的 Y 队教练怒斥 W:你的本事跑哪儿去了?难道凭你脑壳里那台性能高超的计算机还比不上一个四肢发达的 Q 吗? W 无言可对,从此他失业了。”这是一则似乎有些荒唐的现代童话。然而它却包含着一个不可否认的事实——尽管现代数字计算机发展到今天,其运算速度可达每秒数百亿次以上,存贮量和计算精度成倍地扩大和提高,并在数值计算、符号逻辑推理方面显示出巨大的优势,然而面对连续的、模糊的和随机的信息处理问题如:模式识别、图象处理、决策判断等却显得非常迟钝和笨拙。虽然人们习惯地将数字计算机称为电脑,而实际上应该说它只是一个不健全的“大脑”,一个只

有逻辑思维而缺乏形象思维的“大脑”。一个不知道一加一等于二的两、三岁的幼儿却可以一眼从众多的人群中确认出自己的父母，但若让电脑来完成这个对人类来说轻而易举的任务却要花九牛二虎之力，且成功率只有百分之十几。

模糊数学的创始人 L. Zadeh 教授曾举过一个十分有趣的停车例子来说明在寻找问题满意解方面，精确的理论计算和擅长精确数值运算的计算机显得多么死板和低能：把一辆汽车停在停车场中两辆车之间的一个空位上。对于这样一个看起来并不复杂的问题，若让一台计算机来指挥完成，则要大费一番周折。首先要建立问题的数学模型，以便编制计算机程序，求出问题的数值解。其理论计算的步骤是：设 ω 代表车 C 上一个固定的参考点位置， θ 代表 C 的方向，则汽车的状态为 $X = (\omega, \theta)$ ，车的微分方程为 $\dot{X} = f(X, u)$ ，其中 u 是一个有约束的控制矢量，它有两个分量 u_1 和 u_2 ， u_1 是前轮角度， u_2 为车速。两辆停着的车之间的空隙定义为许可的终局状态的集合 Γ ，邻近两辆车定义为 X 执行中的约束，记为集合 Ω 。现在的任务是：寻找一个控制 $u(t)$ ，使在满足各种约束条件下把初始状态转移到 Γ 中去。对于这样一个复杂的运动模型，即使大型计算机也很难胜任。因此，从实用角度来看，这里所谈的问题没有精确解存在。面对同样的问题，即使是一名刚刚掌握驾驶技术的司机，经过前、后、左、右几次移车，就能将车停到两车的空隙之间。这就是说，司机通过一些不精确的观察，执行一些不精确的控制，就能轻易地达到目的。此时，人们不是先去推导汽车运动的数学模型，然后再根据模型加以控制，而是利用诸如直觉推理、经验和知识技巧等智能行为，再加上学习能力来对汽车进行控制的。从这个例子我们可以进一步看出：建立在图灵算法基础上的 Von Neumann 计算机，把对任何性质的问题求解都当作一个计算过程，问题只有转换成算法才能进入计算机，从而求得问题的精确解。从计算机的内部计算过程来看，又是一个串行的信息处理过程，即计算机 CPU 每次从存储器中取出一条数据信息进行计算

和逻辑判断,然后根据事先编排的程序及中间结果决定下一步应该继续执行的指令,同时将本次的计算结果送回到存贮器某个确定的位置存贮起来。这种串行的、被动的信息处理过程,完全集中在 CPU 中进行,使得无法进一步提高计算机的运算速度以满足信息处理量迅猛增长的需要,同时也很难适应大量复杂的计算问题。特别是随着世界科技水平的不断提高,计算机所面临的任务已由单纯的数值计算和一般的信息处理发展到知识处理、模糊识别等高层次的智能处理。人们曾把希望寄托于提高数字计算机的运算速度和增大硬件的集成度,然而无论是速度还是集成度终将达到一定的极限。数字计算机的前途受到了严重的挑战。这不得不使人们用审视的眼光,从信息处理体系的角度重新评价 Von Neumann 计算机,并寻找计算机发展的新途径。

1.2 大脑——一个神秘的世界

著名哲学家罗素曾经说过:没有什么比研究人本身更重要的了。人类在认识自然、改造自然的过程中逐渐体会到,如果没有对人之灵魂的大脑的深刻认识,就很难在更高的层次上揭示客观世界的奥秘。因此,整个人类文明历史的发展,都伴随人类对自身大脑的物质结构、意识活动和生物特征的不懈探索。大脑这一神秘世界的面纱被一层层地揭开。虽然到目前为止人们还不能完全解释大脑思维、意识和精神活动,但已能从神经结构、细胞体构成的水平上初步探明了大脑的组织特征,并已可以通过生理实验证明许多大脑的认知机理。这些对于大脑的认识成果,为研究和设计与脑的智能接近的人工智能机器奠定了理论基础。各种实验与研究表明:大脑中存在着由巨量神经细胞(约 100 亿个)结合而成的神经网络,正是大脑的神经网络系统,构成了大脑信息处理的主体,神

神经网络的活动决定了大脑的功能。

神经网络信息处理的基本特性主要有以下几点：

(1) 分布存贮与容错性

信息在神经网络内的存贮是按内容分布于大量的神经细胞之中,而且每个神经细胞实际上存贮着多种不同信息的部分内容。信息的记忆,主要反映在神经元之间的连接强度上。网络的各个部分对学习和记忆具有等势作用,即使个别单元出错,部分信息丢失,但大部分信息仍存在,因而可以使完整的信息得到恢复。事实上,人的大脑平均每天有 10~40 万个神经细胞死亡,但这丝毫不会影响其记忆与处理功能。

(2) 并行处理性

神经元的响应速度很慢,每次约 1ms 左右,比一般的电子元件要慢几个数量级,而且每个神经的处理功能也很有限,但却可以在几百微秒的时间内对一个复杂的过程作出判断和决策。而对于一个工作速度可达纳秒级的数字计算机来说,由于其信息是按串行方式处理的,若要在几百个串行步骤内完成复杂的推理或计算是不可能的。在神经网络中,巨量的神经元可同时进行同样的处理,因而是大规模的并行处理,而且这种并行处理决不是简单地“以空间的复杂性代替时间复杂性”,而是反映了根本不同的操作机理。

(3) 信息处理与贮存的合二而一性

大脑对信息的分布存贮与并行处理并不是相互独立的,而是相互融合的,即每个神经元都兼有信息处理和存贮功能。神经元之间连接强度的变化,既反映了对信息的记忆,同时又与神经元对激励的响应一起反映了对信息的处理。大脑对信息的提取过程,也不象数字计算机那样先找到信息的存贮地址,再由地址提取到信息,而是将信息地址的寻找过程融会于信息的提取过程之中。

(4) 可塑性与自组织性

人脑发展、进化的本身是在由遗传信息先天确定的结构和特

性条件下的自组织过程。人的大脑在人出生后的有限时间内就形成了良好的可塑性。而现代生育医学的研究又证明：胎教同样可以促进出生后婴儿智力的发育，这说明在大脑形成之日起可塑性也就开始形成了。一个咿呀学语的幼儿，经过半年左右的语言训练，其词汇量的增长速度是十分惊人的，这正说明幼儿的大脑具有良好的可塑性。从生理学的角度看，可塑性主要反映在神经元之间连接强度的可变性。大脑可通过其内部的自组织、自学习能力不断适应外界环境的变化。俗话说“熟能生巧”，正是反映了大脑的自组织能力。当人们反复从事某一项工作时，这件工作的各个技术特征及要领就会“熔化”在大脑的若干群神经元中，而且随着工作的反复进行，使得这些神经元之间的连接强度不断增加，从而提高了神经元对这些技术特征及操作要领反映的灵敏度，也就发生了由“熟”到“巧”的变化。这种自组织、自学习能力，是只能被动地执行程序的数字计算机所无法做到的。

(5) 层次性与系统性

神经网络是个十分复杂、规模庞大的系统。由结构和功能都十分简单、有限的神经元组成的网络系统，通过系统的组合功能效应，可完成各种复杂的信息处理任务。神经生理学研究还表明，大脑对信息的处理过程是分层次进行的。从初级皮层的处理到顶区皮层、最后达到脑前额叶的处理过程，是对信息分层加工的过程。

脑神经的思维活动特征和认识机理，无疑代表了信息处理的一种高层次的模式。归纳起来，数字计算机的信息处理机制是：以串行的、离散的、逻辑符号推理为处理手段，以求得问题的精确解为处理目标；而神经网络的信息处理机制是：以分布存贮、并行处理为处理手段，以求问题的满意解为处理目标。实际上早在世界上第一台数字电子计算机诞生之前，人类对于脑神经细胞、神经网络以及对脑神经元模型的研究就已开始，并取得了积极成果。从那时起人们就开始了通过对人脑存贮、处理和搜索信息的机制的研

究来探索和实现名副其实的智能机器的不懈努力。

1.3 神经网络研究、发展的历史

从 1943 年心理学家 W. S. McCulloch 和数学家 W. Pitts 研究并提出 M-P 神经元模型^[1]起到今天,人类对神经网络的研究走过了半个世纪的历程。这种用人工神经网络(Artificial Neural Network)模拟、仿真和近似生物神经网络(Biological Neural Networks)的探索,经历了曲折的发展道路。40 年代初期 McCulloch 和 Pitts 用数理逻辑的方法研究生物神经网络的创举,大大地开阔了人们的思路。在这之后,1949 年 D. O. Hebb 从心理学的角度提出了至今仍对神经网络理论有着重要影响的 Hebb 学习法则^[2]。50 年代末 60 年初,神经网络系统已开始作为人工智能的一条途径受到人们的重视。1961 年 E. Rosenblatt 提出了著名的感知机(perceptron)模型^[3]。这个模型由简单的阈值性神经元构成,初步具备了诸如学习性、并行处理、分布存贮等神经网络的一些基本特征,从而确立了从系统角度进行人工神经网络研究的基础。1962 年 Widrow 提出了主要适用于自适应系统的自适应线性元件(Adaline)网络。神经网络的研究进入了一个高潮。

1969 年,美国麻省理工学院著名的人工智能专家 M. Minsky 经过数年潜心研究,以感知机为代表,对简单神经网络的功能和局限性从数学上作了深入的剖析,他和 Papert 共同出版了影响很大的《perceptron》(感知机)一书,指出:简单神经网络只能运用于线性问题的求解,而对象 XOR(异或)这样简单的非线性问题却无法求解。他们同时指出:能够求解非线性问题的网络,应该是具有隐层的多层神经网络,而将感知机模型扩展到多层网络是否有意义,还不能从理论上得到有力的证明。由于 Minsky 的悲观结论,以及

当时世界上以逻辑推理为研究基础的人工智能理论和数字计算机的辉煌成就,大大降低了许多人对神经网络研究的热情及克服理论障碍的勇气。在这之后近十年中,神经网络的研究进入了一个缓慢发展的低潮期。

虽然在整个 70 年代中,对神经网络理论的研究进展缓慢,但并没有完全停顿下来。世界上一些对神经网络抱有坚定信心和严肃科学态度的学者,没有放弃他们的努力。在这期间,Grossberg 提出了自适应共振理论;芬兰的 Kohonen 提出了自组织映射理论;日本的福岛邦彦提出了认知机(Cognitron)模型,随后又提出了改进型新认知机(Neocognitron)模型;另一位日本学者中野馨提出了联想记忆模型——联想机(Associatron);东京大学的甘利俊一博士则一直致力于用数学理论解释神经网络。与此同时,有关神经网络的生理、心理基础研究也取得了一些进展:Marr 提出了视觉计算理论,Marr 的学生 Poggio 又在 Marr 理论框架的基础上,提出了初级视觉正则化理论,在计算理论与实现机制间架起了一座桥梁,大大推动了视觉研究的进程。

进入 80 年代,特别是 80 年代末期,神经网络的研究从复兴很快转入了新的热潮。这主要是因为:一方面经过十几年迅速发展起来的以逻辑符号处理为主的人工智能理论和 Von Neumann 计算机在处理诸如视觉、听觉、形象思维、联想记忆等智能信息处理问题上受到了挫折;另一方面,80 年并行分布处理模式的神经网络本身的研究成果,使人们转而看到了新的希望。这一时期首先要提到的是美国加州理工学院的物理学家 Hopfield 的开拓性工作。1982 年他提出了一个新的神经网络模型——Hopfield 网络模型,并首次引入了网络能量函数概念,使网络稳定性研究有了明确的判据。特别是当 Hopfield 用这一网络模型成功地解决了 N-P 类典型问题的 TSP(巡回售货商问题)后,引起了学术界的广泛重视。另外 Hopfield 网络可以用电子线路近似地实现,为建造新型的电子神经计算机(Electronic Neural Computer)奠定了基础。1984 年

Hinton 和 Sejnowski 提出的 Boltzmann 机模型借用统计物理学的概念和方法来研究神经网络,首次采用了多层网络的学习算法,并将模拟退火概念移植到网络的学习机理之中,以保证网络能收敛到全局最小值。如果说 Hopfield 的研究成果打破了神经网络理论十年徘徊的局面,那么 1986 年 Rumelhart 和 McClelland 及其研究小组提出的 PDP (Parallel Distributed Processing) 网络思想,则为神经网络研究新高潮的到来起到了推波助澜的作用。尤其是他们提出的误差逆传播学习算法,成为至今影响最大的一种网络学习方法,据统计有近 90% 的网络应用使用的是这一算法。与此同时,美国、日本等国在神经网络计算机的硬件实现方面也取得了一些实实在在地成绩。有关神经网络的国际学术组织、学术会议、学术刊物也纷纷成立、召开和创刊。神经网络的应用,已渗透到模式识别、图象处理、非线性优化、语音处理、自然语言理解、自动目标识别、机器人、专家系统等各个领域,并取得了令人瞩目的成果。神经网络理论也已成为涉及神经生理科学、认识科学、数理科学、心理学、信息科学、计算机科学、微电子学、光学、生物电子学等多学科的新兴的、综合性的前沿学科。当然,必须指出的是,对神经网络及神经计算机的研究,决不意味着曾为人类文明发展作出巨大贡献的数字计算机将要退出历史舞台,而是要在智能的、模糊的、随机的信息处理方面,发挥神经网络及神经计算机的优势;而在对符号逻辑推理、数值精确计算等方面,数字计算机仍将发挥其不可替代的作用。两种计算机互为补充,共同发展。从真正的“电脑”意义上来说,未来智能型计算机应该是兼备逻辑思维和形象思维,具有模糊处理和精确计算功能的崭新计算机。

最后还应指出,人类对大脑的研究还有很长的路要走。目前出现的各种人工神经网络决不是生物神经网络的生理学模型,而是经过大大简化、易于工程实现的、反映生物神经网络某些特征的数理表示。人工神经网络理论的潜在作用,是否能够达到人们所期望的那样,在理论发展过程中是否还会暴露出新的难以克服的困难,

目前还很难下定论。当然,这些疑虑丝毫不会影响对神经网络的深入研究,神经网络的应用也决不应停留在目前的水平上。

1.4 神经网络研究的主要方向

神经网络的研究可以分为理论研究和应用研究两大方面。

理论研究可分为以下两类:

① 利用神经生理与认知科学研究人类思维及智能机理。

② 利用神经基础理论的研究成果,用数理方法探索功能更加完善、性能更加优越的神经网络模型,深入研究网络算法和性能,如:稳定性、收敛性、容错性、鲁棒性等;开发新的网络数理理论,如:神经网络动力学、非线性神经场等。

应用研究可分为以下两类:

① 神经网络的软件模拟和硬件实现的研究。

② 神经网络在各个领域中应用的研究。这些领域主要包括:模式识别、信号处理、知识工程、专家系统、优化组合、机器人控制等。

随着神经网络理论本身以及相关理论、相关技术的不断发展,神经网络的应用定将更加深入和广泛。

第二章 神经网络理论基础

2.1 大脑与生物神经系统

对于整个生物系统来说,恐怕没有比大脑神经系统更复杂、更深奥,从而也更具有魅力的了。脑是生物的神经中枢,是人的灵魂所在。大脑神经系统的构造究竟怎样?其工作机理到底如何?人的思维过程是怎样通过它实现的?这些正是人们长期为之不懈探索的问题。虽然到目前为止,大脑神经系统的奥密还没有全部被揭开,但是通过大量的生理学、解剖学实验,人们已经基本掌握了大脑神经系统的结构特征,初步探明了大脑神经活动机理以及与神经活动有关的一些生物化学反应。

图 2.1 是人脑的典型神经网络。如图所示,大脑是由约 100 亿 (10^{10}) 个生物神经元(简称神经元)构成,而其中的每一个神经元又与约 1 万~10 万个其它神经元相连接,如此构成一个庞大的三维空间的神经网络。神经元不但是组成大脑的基本单元,而且也是大脑进行信息处理的基础元件。

神经元的结构如图 2.2 所示。它是以细胞体为主体,由许多向四面延伸的不规则树枝状纤维构成的神经细胞。其形状很象一棵独立寒冬的枯树枝杆。除细胞体外,神经元的其它主要部分有:树突(dendrite,也称为树状突起);神经键(Synapse,也称为突触)等。细胞体是由细胞核、细胞质和细胞膜构成的;树突是由细胞体向外

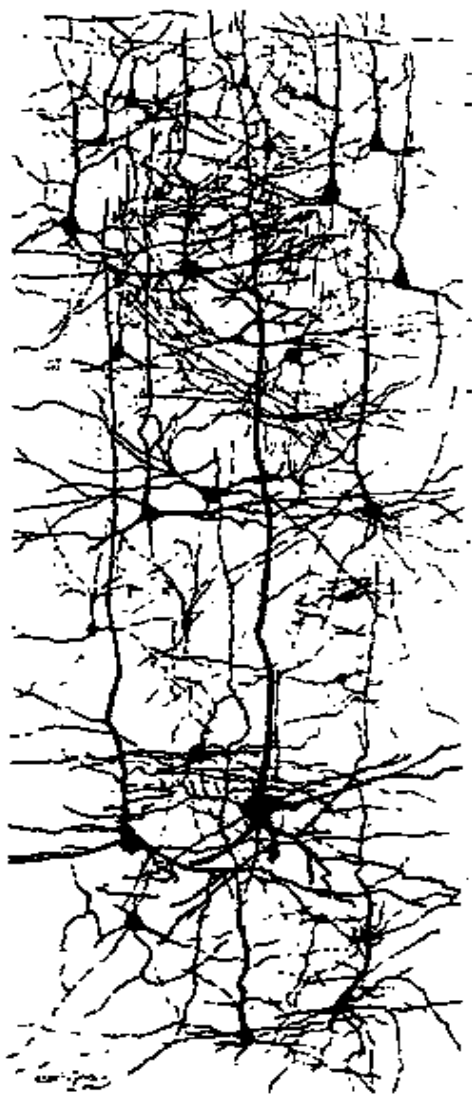


图 2.1 人脑的神经网络

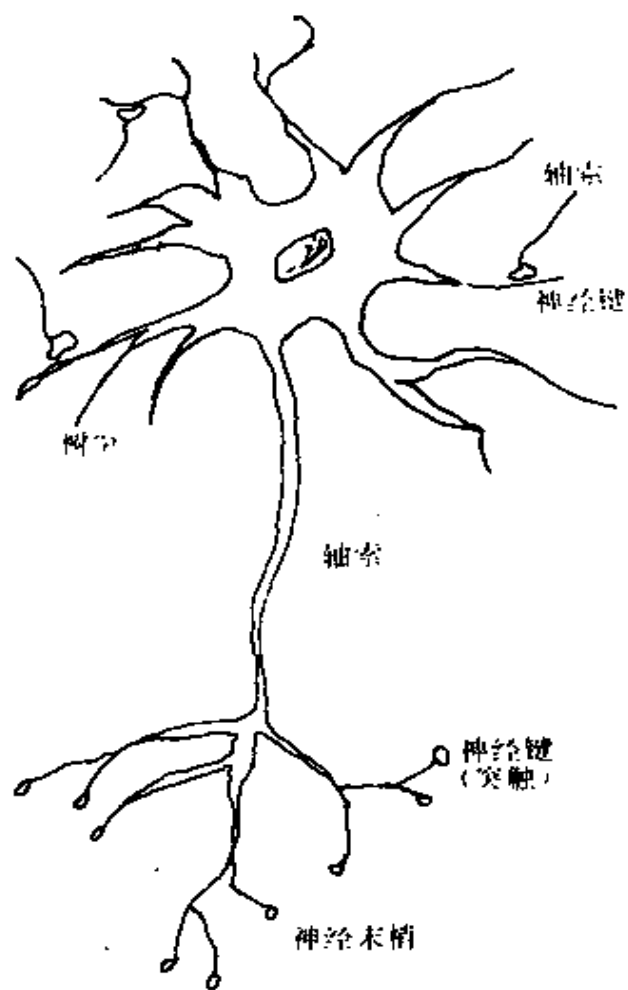


图 2.2 生物神经元

延伸的树枝状纤维体，它是神经接受输入信息的通道。与其连接的所有神经元通过树突传入神经激发脉冲；轴索是由细胞体向外延伸的最长、最粗的一条树枝纤维体，即神经纤维，它相当于细胞的输出通道。神经元的输出激发脉冲，通过轴索由神经键传给与其相连的其它神经元；由细胞体向外延伸的最长神经键位于轴索的末

端(即神经末梢处),是轴索与树突的连接部分。它相当于神经元的输入/输出接口。激发脉冲通过轴索(输出通道)经神经键传给树突(输入通道)。如前所述,每个神经元大约有 $10^3 \sim 10^4$ 个神经键。

从神经元各组成部分的功能来看,信息的传递与处理主要发生在轴索与树突的连接部,即神经键的附近。神经键的结构如图 2.3 所示。在神经键中含有大量的神经键微细胞(Synaptic Vesicle),在这些微细胞中含有信息传递媒介。在神经键与树突之

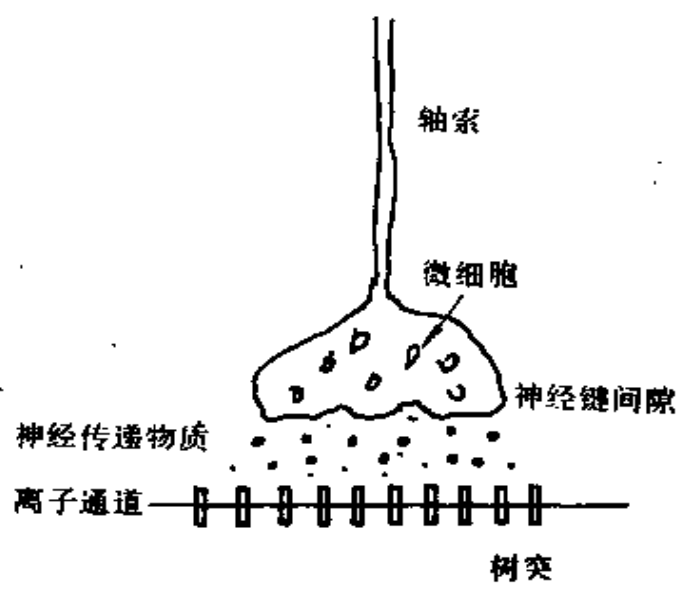


图 2.3 神经键结构

间有大约 200\AA 的间隙,称为神经键间隙(Synaptic Cleft),在神经键和树突的表面包着一层薄膜,称为神经膜。其中神经键一侧的称为键前神经膜;树突一侧的称为键后神经膜,在神经膜内外之间存有电位差,称为膜电位,也叫静息电位差,约为 $20 \sim 100\text{mV}$,膜外为正,膜内为负。当由神经元细胞体通过轴索传到神经键的动作电位(action potential)达到一定强度,即超过其阈值电位后,神经键将向神经键间隙释放出神经传递化学物质。由于这种化学物质的

扩散,位于神经键后膜的离子通道(ion Channel)被打开,流过离子电流,神经键后由于膜电阻的存在,将产生一个与静息电位方向相反的电压,从而引起膜电位的变化。一个神经元的各个树突往往和大量的其它神经元的神经键相连接,这些神经键键后膜电位的变化,将对这一神经元产生综合作用。神经键分为两类,即兴奋性神经键和抑制性神经键,前者使树突的电位上升,后者使树突的电位下降。当这些神经键活动电位的综合作用结果超过轴索的阈值电位时,此神经元的激活电位将发生变化,并沿轴索向前传播,从而实现信息的传递。十分有趣的是,神经元中动作电位脉冲的传递,与日常生活中导线中电脉冲的传递有惊人的相似之处。我们知道,导线的直径越大,即导线越粗,其电阻越小,导线中通过的电流就越大,同时单位时间传输的电脉冲越多,即传输速率越快。在神经元中,有同样的现象:如果轴索越粗,其动作电位脉冲的传输速率也越快。传输速率 V 与轴索直径 d 的关系为:

$$V \propto \sqrt{d} \quad (2.1)$$

另外我们还知道,处于复杂环境中的导线,为保证其安全、有效地传输电脉冲,往往在导线外表加绝缘层。神经元也与其类似,作为神经元信息传递通路的轴索有两种结构形式:髓鞘纤维和无髓纤维,并以此分为两种神经元,即:有髓神经元和无髓神经元。其中有髓神经元相当于表面有绝缘物的导线,其动作电位脉冲的传输速率一般高出无髓神经元的 10 倍左右。

值得注意的是,神经元电脉冲的传输速率是很低的,其中有髓神经元最大脉冲频率约为 500 次/秒。电脉冲每经过一个神经键大约需要 0.5~1ms,这段时间称为突触延时。在两个相邻脉冲之间,神经元阈值电位突然增高,阻止下一个脉冲的通过,这段时间称为不应期(refractory period),大约需要 3~5ms。为什么如此慢的传输速率竟能使大脑产生非常敏捷的信息处理速度呢?这就是由于大脑处理信息存在三个主要特点:巨量并行性、处理与存贮的合二而一性及自组织、自学习性。前面曾介绍过一个神经元与 1 万~10

万个其它神经元相连接,如:大脑视觉通道内的 10^8 个视觉神经细胞与大约 10^6 个神经细胞相关连。一个神经细胞电位的变化,将影响与之相关的一大群神经细胞的电位,真可谓“牵一发而动万军”。尽管一个神经元信息处理速度较慢,但面对如此超量、并行的神经元信息处理大军,大脑对信息处理的快速性也就不足为奇了。

神经元中信息传递的另一大特点,就是神经键的非线性响应特性,如图 2.4 所示。只有当输入神经元的激励信号超过神经膜的阈值电位后,神经元才产生输出动作电位脉冲(宽度约为 1ms),即输出为“1”状态,否则神经元将没有电位输出,即处于“0”状态。神经元的这种阈值特性称为 all-or-none 法则,即符号“0-1”规律。

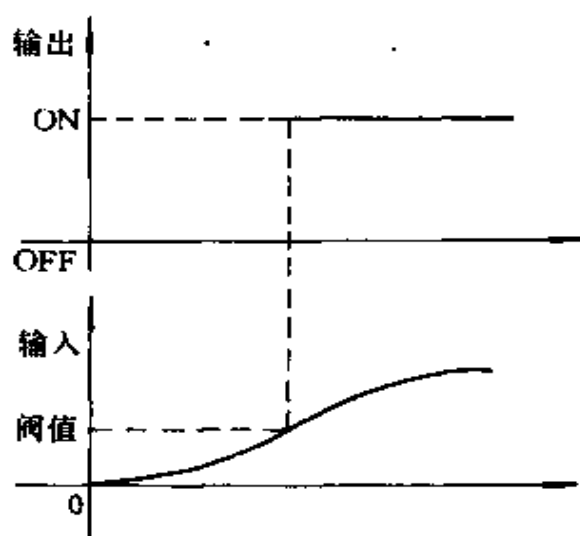


图 2.4 神经元输入/输出状态变化关系

在分析了作为大脑神经系统基本单元——神经元的结构与动作机理之后,再来看一下大脑的结构及整个神经系统的组成。

大脑的神经系统是由庞大的神经元构成的有序结构网络。大脑神经系统各主要部分的划分如图 2.5 所示。大脑具有阶层构造,其中最复杂的是处于大脑最外层的大脑皮层,在大脑皮层中密布着由大量神经元构成的神经网络。人们为弄清大脑皮层的功用,曾做过以下的动物试验:摘除狗的大脑皮层,观察摘除后狗的动作反应。结果发现,失去大脑皮层的狗,仍然可以进食、睡眠,但是其动作反应却处于一种失常状态,完全失去了视觉与嗅觉能力,特别是完全失去了对环境的适应能力。由此可推知:一旦人失去大脑皮层,则将失去智能本性,如感觉、视觉、兴趣、对环境的适应和学习

能力。因此可以认为,大脑皮层是神经系统的核心部分。

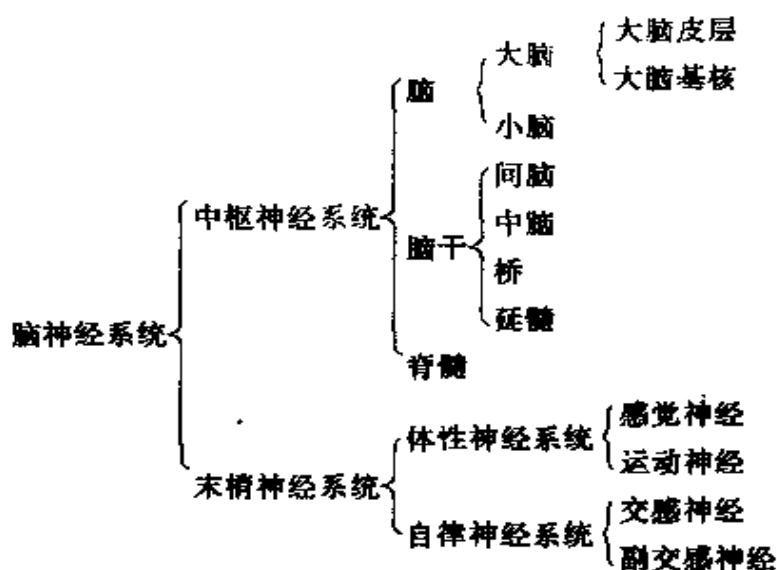


图 2.5 脑神经系统的构成

人们通过长期研究还进一步探明了大脑皮层的组织结构。大脑皮层是由具有不同功能的若干区域所构成。有的区域专门负责运动控制,有的区域专门负责听觉,还有的区域专门负责视觉等等。在同一区域,同样存在着负责某一具体功能的神经元群。如在视觉神经区域,存在着只对光线的方向性产生反应的神经元。更进一步细分,某一层神经元仅对水平光线产生输出响应,而另一层神经元只对垂直光线产生输出响应。特别需要指出的是,大脑皮层的这种区域性结构,虽然是由人的遗传特性所决定的,即是天生的,但各个区域所具有的功能 大部分是人在后天通过对环境的适应和学习而获得的。有人曾做过这样的试验:在一个只有水平光线的环境中饲养一只刚出生的小猴,结果经过一段时间后,这只猴只能判别出水平方向的光线,而对垂直方向的光线却毫无反应。这说明,神经元对外界刺激的反应不是天生的,而是后天通过学习而获得的。神经元的这种特性称为自组织(self-organization)特性。之所以称为自组织,是因为神经元的學習过程,完全是一种自我学习的过程,并不存在所谓教师的示教过程。如对水平光线产生反应的

神经元,并没有“教师”告诉它这是水平光线,而是靠神经元本身对这种光线的不断适应,把这种光线的特征“记忆”下来,并能在记忆之后将这种方向的光线再识别出来。应该指出的是,神经元的这种自组织特性来自其神经网络结构的可塑性,即神经元中的神经键随着动作电位脉冲激励方式与强度的变化,其电位传递的作用可增强或减弱,所以神经之间的连接是柔性的、可塑的。

2.2 生物神经系统的模型化

前节已经指出,大脑信息处理的一个主要特点是具有自组织学习能力。从宏观上可将大脑的学习过程分为三个阶段:进化阶段、健全阶段和日常阶段的学习,如图 2.6 所示。

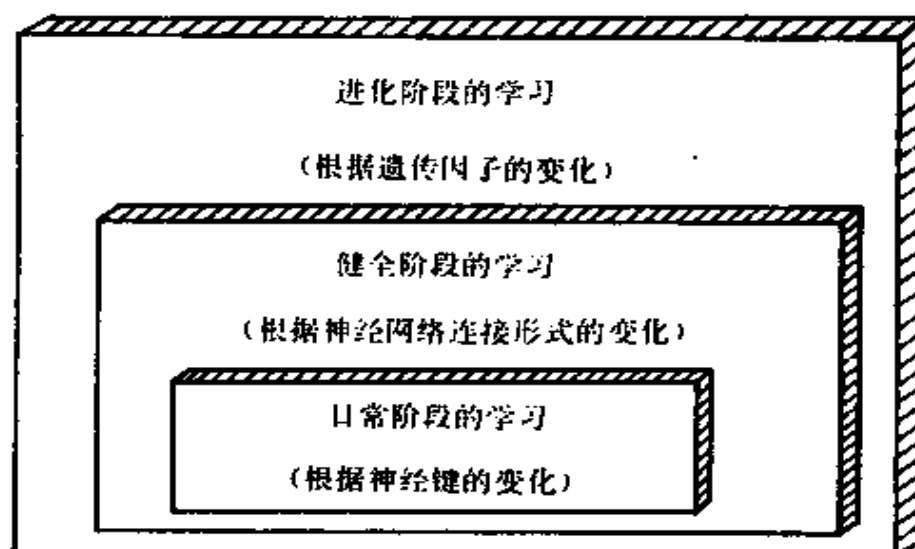


图 2.6 大脑学习的三个阶段

婴儿一出生,其进化阶段的学习已经完成。这时人的大脑的“硬件”结构已基本形成且具有了一些天生的能力,也就是所谓“本

能”。在之后的大脑健全学习阶段,通过对环境的适应,逐渐获得语言及文字表达等基本的人类活动能力,这一阶段的学习进展在幼儿时期最为显著。而且在这一阶段获得的记忆是最牢固、最难忘的。日常阶段的学习,在大脑获得了一些基本技能之后随时都在进行。人们根据这种随时发生的学习,进行正常的日常生活与活动。从微观上分析,如前节所述,大脑的学习与各神经元之间的连接方式、各神经元的阈值电位、神经键的类型(兴奋或抑制)、活动电位脉冲的传递方式、神经键释放的神经传递化学物质的变化等等许多因素有关。总之,神经系统是一个有层次、分区域、含有巨量神经元的动态信息处理系统。它接受生物体内、外环境的输入信息加以综合分析处理,然后调节控制机体对环境作出适当的反应。人类对于自身大脑的研究目的,归纳起来主要有两点:一是通过对大脑神经系统的解剖,进一步揭示脑的生命活动特征,掌握脑的活动规律,从而更有效地发挥和保护大脑的机能,造福于人类。其二是通过对大脑神经网络系统结构、机理和功能的探索,设法模拟、构造出与大脑智能相接近的人工神经网络。然而,要达到这一目的,最有效、最现实的方法就是首先建立大脑的数学模型。

我们知道针对任何一个问题建立数学模型时,如果对这一问题的特性了解的越多、越透彻,则其模型的合理性也就越高。虽然到目前为止,还不能说人类对大脑的结构及各种活动机理已完全明了,但人们对大脑基本结构和机理的理解水平已具备了建立大脑数学模型的条件。应该指出的是,我们所建立的数学模型及所构造出的人工神经网络,并不是人脑的真实描写,而只是对其结构和功能进行大大简化之后的某种抽象与模拟。

正如生物神经元是大脑神经系统的基本单元一样,建立大脑的数学模型,也必须首先从神经元入手。通过前节的分析与介绍,对生物神经元可以总结出以下几点特性:

① 神经元是一个多输入(一个神经元的多个树突与多个其它神经元的神经键相联系)、单输出(一个神经元只有一个轴索作为

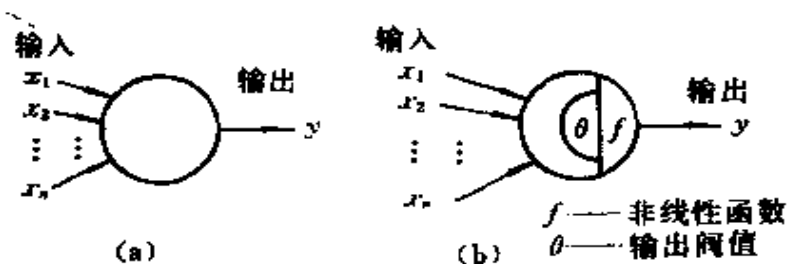
输出通道)元件。

② 神经元是一个具有非线性输入/输出特性的元件。表现在只有当来自各个神经键的活动电位脉冲达到一定强度之后,该神经元的神经键才能被激活,释放出神经传递化学物质,发出本身的活动电位脉冲。

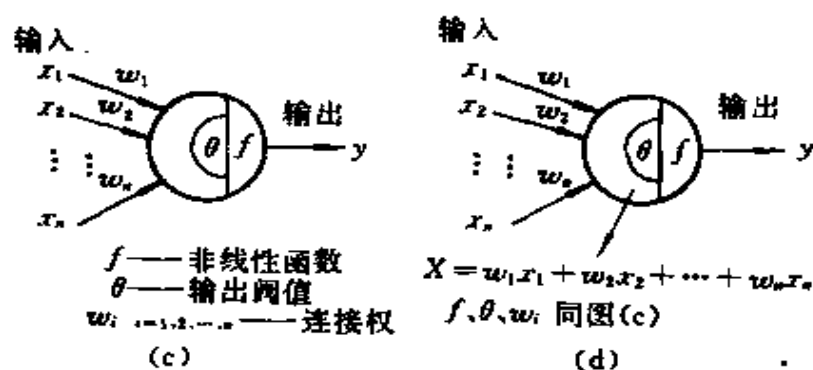
③ 神经元具有可塑性,表现在其活动电位脉冲的传递强度依靠神经传递化学物质的释放量及神经键间隙的变化是可调节的。

④ 神经元的输出响应是各个输入的综合作用的结果,即所有输入的累加作用。输入分为兴奋型(正值)和抑制型(负值)两种。

根据这四个特性我们得到如图 2.7 所示的生物神经元模型的演化过程。



(a) 根据特性①得到的模型 (b) 根据特性①+②得到的模型



(c) 根据特性①+②+③得到的模型 (d) 根据特性①+②+③+④得到的模型

图 2.7 生物神经元模型的演化过程

图 2.7(d)就是最终得到的生物神经元模型。其数学表达式

为:

$$y = f(X) \quad (2.2)$$

$$X = \sum_{i=1}^n w_i x_i - \theta \quad (2.3)$$

表达式(2.3)还可以有另一种形式,即把阈值 θ 视为神经元的第 0 个输入,而 w_0 为常数 -1 ($w_0 = -1$) 则有:

$$X = \sum_{i=0}^n w_i x_i \quad (2.4)$$

关于输入/输出函数,即神经元的响应函数根据要求和特点的不同,可以有各种形式,其中基本的有两种:阶跃响应函数和 S 型 (Sigmoid) 响应函数。其响应特性如图 2.8 所示。

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2.5)$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.6)$$

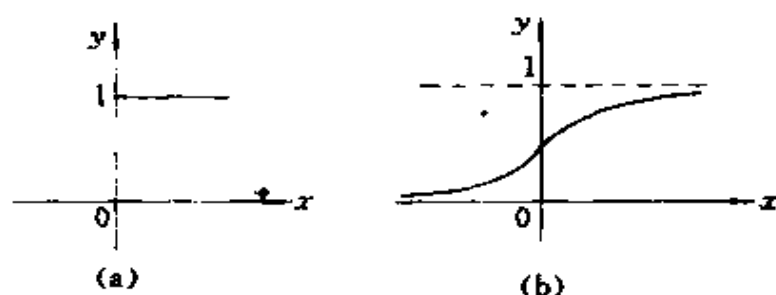


图 2.8 两种基本类型的输出响应函数特性

2.3 M-P 神经元模型与人工神经网络的构成

早在 1943 年, McCulloch 和 Pitts 就定义了一种简单的人工神经元模型, 称为 M-P 模型。这一模型与上节介绍的神经元模型基本相同, 其响应函数为阶跃函数, 模型结构如图 2.9 所示。数学表达式如下:

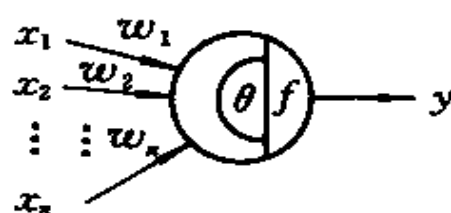


图 2.9 M-P 模型结构

设神经元的一组输入用向量表示为:

$$X = (x_1, x_2, \dots, x_n)$$

其相应权值为:

$$W = (w_1, w_2, \dots, w_n)$$

神经元的阈值为 θ , 输出为 y , 则

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (2.7)$$

其中:

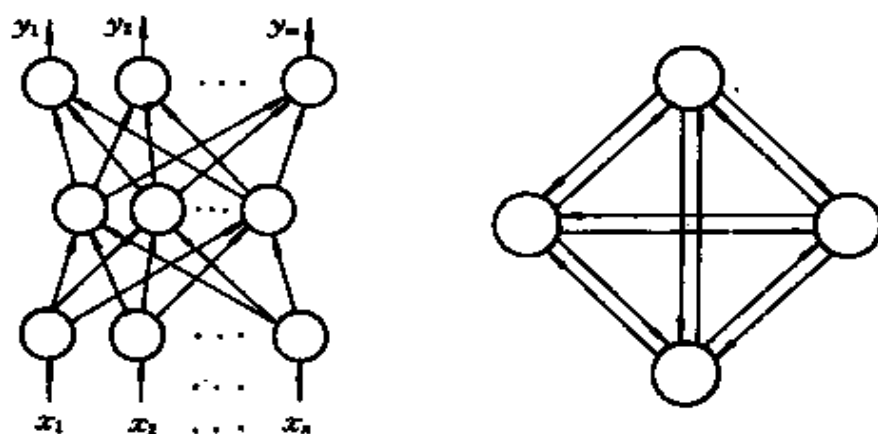
$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

另外, 其响应函数也可以采用符号函数, 即

$$S_{gn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (2.8)$$

M-P 模型虽然非常简单, 但它完全反映了上节介绍的生物神经元的四个主要特性。可以说是对生物神经元一个完整的数学描述。

由大脑神经网络的活动机理可知,仅由单个神经元是不可能完成对输入信息的处理的,只有当大量的神经元组成庞大的网络,通过网络中各神经元之间的相互作用,才能实现对信息的处理与存贮。同样道理,只有把人工神经元按一定规则连接成网络,并让网络中各神经元的连接权按一定的规则变化,才能实现对输入模式的学习与识别。生物神经网络与人工神经网络的不同之处是,前者是由上亿个以上的生物神经元连接而成、仅具有统计性规律的庞大网络;而后者,限于物理实现的困难和为了计算的简便,是由数量远少于前者的、完全按一定规律构成的网络。人工神经网络中每一个神经元具有完全相同的结构,在没有特别规定的情况下,所有神经元的动作无论在时间上还是空间上都是同步的。大脑神经网络往往具有层状结构,如大脑皮层的六层结构和小脑的三层结构。人工神经网络的连接形式,尽管其拓扑结构有一些差别,但总的来说主要是如图 2.10 所示的两种形式:阶层型和全互连接型。



(a) 阶层型

(b) 全互连接型

图 2.10 人工神经网络的连接形式

阶层型神经网络的层数,以及各层的神经元的个数根据要求

可以变化;全互连接型神经网络中神经元的个数也可根据要求有所不同。但无论哪种形式的神经网络都有一个共同的特点:网络的学习和运行取决于各种神经元连接权的动态演化过程。某些拓扑结构相同而却具有各种不同功能和特性的神经网络,是因为其具有各种不同的工作和学习规则,即不同的连接权的动态演化规律。可见,决定一个网络性质的主要因素有两点:一是网络的拓扑结构;一是网络的学习、工作规则。二者结合起来构成了一个网络的主要特征。

2.4 人工神经网络的学习机理 与 Hebb 学习规则

一个神经网络仅仅具有拓扑结构,还不能具有任何智能特性,必须有一套完整的学习、工作规则与之配合。其实,对于大脑神经网络来说,完成不同功能的网络区域都具有各自的学习规则,这些完整和巧妙的学习规则是大脑在进化学习阶段获得的。人工神经网络的学习规则,说到底就是网络连接权的调整规则。我们可以从日常生活中一个简单的例子了解网络连接权的调整机理。例如,家长往往对按时、准确地完成家庭作业的孩子大加赞扬,甚至给一些物质奖励;而对于不走人行横道,随意过马路的孩子狠狠地批评。这其中包含着这样一个规则:对于正确的行为给予加强(表扬),不正确的行为给予抑制(批评)。把这一规则运用到神经网络的学习中,就成为网络的学习准则。如图 2.11 所示,让某一网络识别 A、B 两个手写字母,规定当 A 输入网络时,输出应为“1”;而当 B 输入网络时,输出应为“0”。网络学习的总原则是:如果网络作出错误的判断,通过网络的学习,使网络减少下次犯同样错误的可能性。首先,给网络的各连接权赋予(0,1)区间内的随机值,将 A 所对应

的图象模式输入给网络,网络将输入模式按式 2.3 求加权和,再按

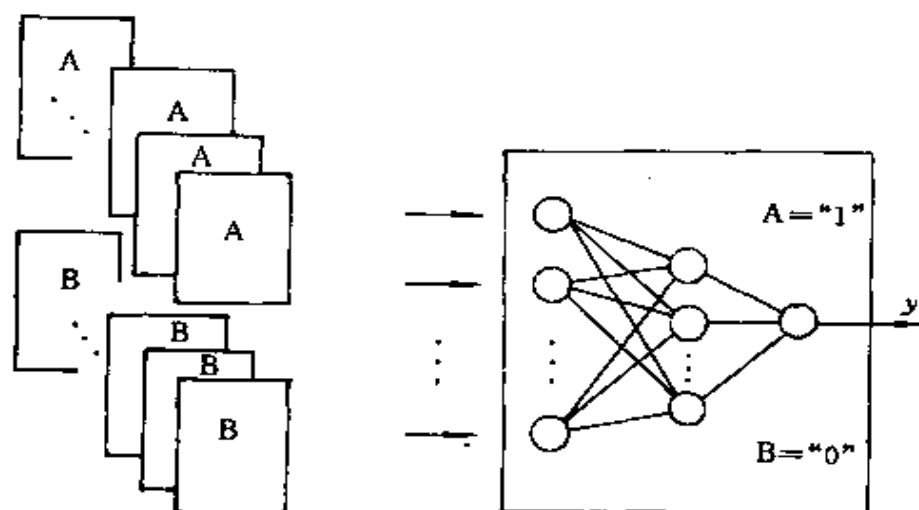


图 2.11 网络学习举例

式 2.2 得到网络的输出。在这种情况下,网络输出为“1”和“0”的概率各为 50%,也就是说完全随机的。这时如果输出为“1”(结果正确),则把网络连接权向着进一步增大综合输入加权值的方向调整,即使连接权值加大,目的是使网络再遇到 A 模式输入时,仍然能作出正确的判断。如果输出为“0”(即结果错误),则把网络连接权朝着减小综合输入加权值的方向调整,即使连接权值减小,其目的是使网络下次再遇到 A 模式输入时,减少犯同样错误的可能性。如此操作调整,当给网络轮番输入若干个手写字母 A、B 后,经过网络按以上学习方法进行若干次学习之后,网络判断的正确率将大大提高。这说明网络对这两个模式的学习已获得了成功,它已将这两个模式分布地记忆在网络的各个连接权上。当网络再次遇到其中任何一个模式时,能够按式 2.2、2.3 作出迅速、准确地判断。可将网络的学习过程表示为如图 2.12 所示那样。

值得一提的是,在上例中,当给学习之后的网络提供一个介于 A、B 两个字母之间的、很难辨认的一个字母时,网络按式

2.2、2.3 仍可得到一个结果。如果为“1”，说明这字母偏向于 A；如果为“0”则偏向于 B。也就是说网络具有了联想判断功能，也就是后面将要介绍的具有对最邻近模式分类的能力。

对于人工神经网络的初期学习规则，最著名的是 Donall Hebb 根据心理学中条件反射机理，于 1949 年提出的神经细胞间连接强度变化的规则，即所谓 Hebb 学习规则。内容为：如果两个神经元同时兴奋（即同时为“1”），则它们之间的神经键（突触）联系得以增强。以 a_i 表示神经元 i 的激活值（输出）， a_j 表示神经元 j 的激活值， w_{ij} 表示两个神经元之间的连接权，则 Hebb 学习规则的数学表达式为：

$$\Delta w_{ij} = a_i a_j \quad (2.9)$$

Hebb 学习规则还有许多变形和改进形式，式 2.9 是其最基本形式。

2.5 教师示教学习与无教师示教学习

在网络的学习过程中，对于网络的学习结果，即网络输出的正确性必须有一个评价标准。如前节例子中的 A 应输出“1”，B 应输出“0”。网络根据实际输出与评价标准的比较，决定连接权的调整方式。在上例中，这个评价标准是人为由外界提示给网络的，即相当于有一位知晓正确结果的教师示教给网络。这种学习方式称为

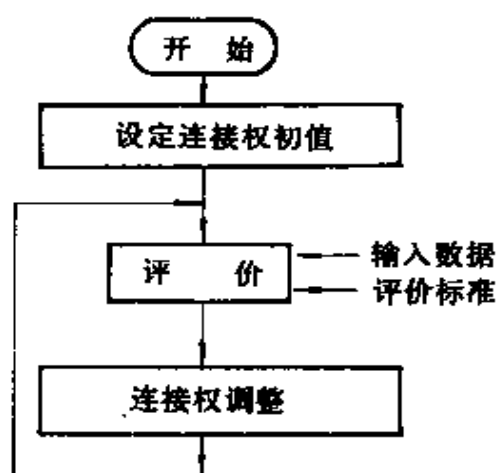


图 2.12 学习过程描述

教师示教学习方式。其原理如图 2.13 所示。

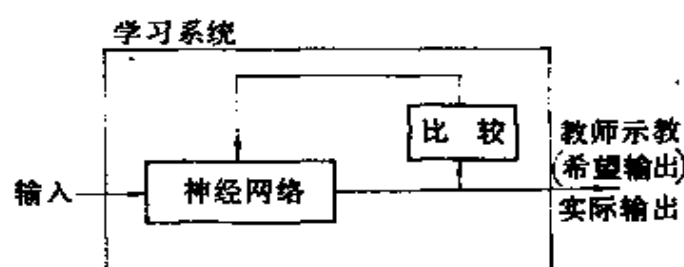


图 2.13 教师示教学习方式

另外还有一类重要的学习方式,在网络外部没有教师示教。网络能够根据其特有的网络结构和学习规则,对属于同一类的模式进行自动分类。可以认为,这种网络的学习评价标准,隐含于网络的内部。因此称为**无教师示教学习方式**。如图 2.14 所示。

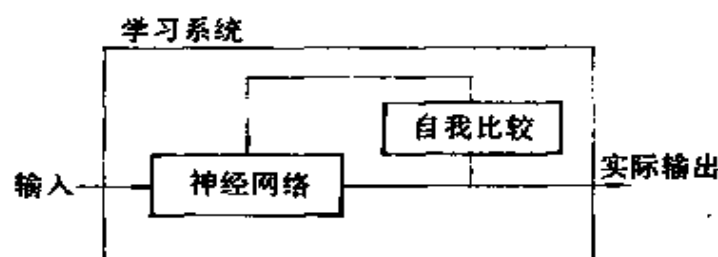


图 2.14 无教师示教学习方式

2.6 生物神经网络与人工神经网络的比较

前几节简单介绍了大脑生物神经系统的主要结构及智能机理,并阐述了由向生物神经系统的模拟到人工神经网络构成的演变方法,得到了生物神经元的数学模型、人工神经网络的学习规则,建立了有关人工神经网络的一些基本概念和方法。从中可以看出,人工神经网络与生物神经网络无论在结构还是在性能上有许多相似之处,但是正如前面曾指出的那样,二者之间决不是一一对

应的写真关系,而是存在着很大的差别。本节将对它们之间的差异做一总结,以便更加清楚地了解人工神经网络的构成原理,把握住其主要原理特征。

首先,生物神经网络与人工神经网络最基本的差异在于生物神经元与人工神经元特性的复杂程度。对于生物神经元来说,影响其神经键激活电位的因素很多、很复杂。如神经键微细胞的大小与多少、神经传递化学物质含量的多少、神经传递化学物质释放速度的快慢、神经键间隙的变化、树突的位置与大小等等诸多因素都会直接影响某个生物神经元对输入活动电位脉冲的响应。而对于人工神经元来说,对输入信号的响应过程十分简单:以对神经元所有输入所求的加权和,作为非线性响应函数的输入,即可得到神经元的输出结果。

生物神经元之间通过活动电位脉冲传递信息;而在人工神经网络中,各神经元之间是以标量数值形式传递信息的。生物神经元之间电位脉冲的传递是非同步的,具有很大的随机性;而人工神经网络中,神经元之间的信息传递与连接权的更新是同步进行的。

生物神经网络中各个神经元的连接方式,从主体上说人与人基本上是相似的。但是,在细微的结构上,如树突的走向、粗细、长短却因人而异,即使具有同一遗传因子的动物或人,也很难找出完全相同的两个树突来。因此,人与人、动物与动物之间存在着智力上的差别。而人工神经网络具有十分整齐、对称的拓扑结构,网络中每个神经元的结构完全相同。因此,对于同一类型的人工神经网络来说,一旦其结构与学习方法、学习参数确定下来,其性能也就随之确定了。

生物神经网络由大约 10^{10} 个生物神经元构成;而人工神经网络中的神经元数量远远少于这个数量,到目前为止,一般人工神经网络中神经元的数量限于 10000 以下。

生物神经网络与人工神经网络另一个主要差别是:生物神经网络的智能性远远超过人工神经网络,人工神经网络所实现的智

能特性仅是生物神经网络中的一小部分。例如,生物神经网络往往只需对某一模式学习一次,就可将其记忆下来。如从电影广告上见到某一电影明星的剧照,则在看电影时,一眼就能将其辨认出来。而对于人工神经网络来说,对某一模式必须经过数百次乃至上千次的学习,才能将其正确地记忆下来。

以上介绍了生物神经网络与人工神经网络之间的主要差异。它们之间在一些重要的特性上也存在着一致性,否则就不能称之为人工神经网络了。

首先,二者都具有对信息处理的巨量并行性和信息存贮的分布性,这也正是人工神经网络对生物神经网络模拟的重要理论根据——连接主义(connectionism)原理。生物神经网络通过各神经元之间神经键的电位变化,对信息进行传递、处理、贮存,不同的信息可以互不影响地分布存贮于整个网络之中;人工神经网络通过网络中各连接权值的改变,实现信息的处理与贮存。每个神经元既是信息的存贮单元,又是信息的处理单元,信息的处理与贮存合二为一,由这些神经元构成的网络,在每个神经元的共同作用下,完成对输入模式的识别与记忆。

与上一共同点相关的是,生物神经元和人工神经元都具有可塑性,这是他们具有自学习机能的基础。生物神经元通过相互之间神经键的连接强度、人工神经元通过相互之间连接权大小的改变,使其自身具有了这种可塑性。

生物神经网络与人工神经网络另一个重要的共同点是,都具有冗余性(redundancy)。生物神经网络不会因个别神经细胞的损失而失去对原有模式的记忆。最有力的证明是,当一个人大脑因意外事故受轻微损伤之后,并不会失去对原有事物的全部记忆。人工神经网络也有类似的情况,因某些原因,无论是网络的硬件实现,还是软件实现中某个神经元失效,不会影响整个网络的工作性能(某些人工神经网络不具有这一特性,如第七章介绍的 ART1 网络)。

还有一点需要指出的是,神经网络的稳定性,即对后一事物的

记忆不会影响对前一事物的记忆。而某些人工神经网络也具有这一特性,如自适应共振理论网络。

总之,人工神经网络与生物神经网络虽然存在着许多差别,但在一些重要特性上却是一致的。这也正是人工神经网络具有很大发展潜力并日益受到人们重视的原因所在。而且,随着研究工作的不断深入,人工神经网络与生物神经网络之间的差异将会逐渐缩小。

2.7 模式识别的基本定义与方法

模式识别是人工神经网络应用最多的一个重要领域,因此这里对模式识别的有关基本定义与方法分别作一简单介绍。

什么是模式?什么是模式识别?我们通过日常生活中一个简单的例子来加以说明。读书是人们经常从事的一项活动,实际上,读书的过程就是一个模式识别的过程。人们首先通过视觉系统将书的文字输入大脑的视觉神经系统,由视觉神经系统对代表输入文字的信息信号进行分析,提取出这一文字的信息特征,传给大脑皮层,由大脑皮层根据以往的学习记忆,辨认出这一文字。在这一过程中,代表文字的信息信号就是模式,而大脑皮层对这一文字信号的辨认就是模式识别。因此通俗地讲,模式就是我们所面临的一类对象或事物。具体地,往往把代表这一类对象或事物的信息信号表示成一个向量,向量的维数代表模式空间的维数,向量中的每个元素,代表对象或事物中某一部位的信息强度。模式识别就是对模式进行正确的分类。

这里特别需要指出的是模式的特征抽取,简单地讲,一个模式的特征,就是一个模式区别于另一个模式的主要差别。例如,两个英文字母 F 和 E。为了区别这两个字母,可以取两字母中的水平线

和垂直线的个数作为依据。这里水平线和垂直线的个数就是这两个字母的模式特征。对于一些复杂的模式,往往仅以一、两个特征很难区分出它们之间的差别,因此根据需要可取 n 个参量代表模式的 n 个特征。由这 n 个特征量构成的向量,称为 n 维特征向量。由具有相同维数和代表意义的特征向量构成的集合,称为特征向量空间。例如,如果把身長和体重作为区别体操运动员和篮球运动员的两个模式特征,则其模式空间可形象地表示为如图 2.15 所示的状况。

图中的斜线为模式特征空间的分类边界线。可根据得到的数据是处于分类边界线的左、右边,判断出这组数据所代表的模式所属的类别。如果把分类边界用数学形式表达出来,就形成了所谓模式判别函数。实际上,模式判别函数可以有各种形式,如曲线、平面或超平面、超曲面等,其中最简单的就是如图 2.15 所示的直线形式。对于能够用一条直线进行分类的模式空间,称为线性模式空间或线性分类系统。

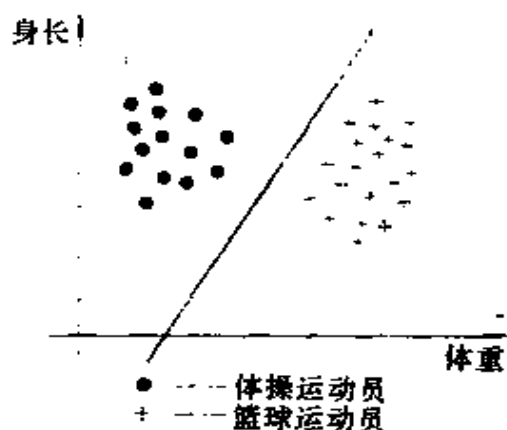


图 2.15 模式空间举例

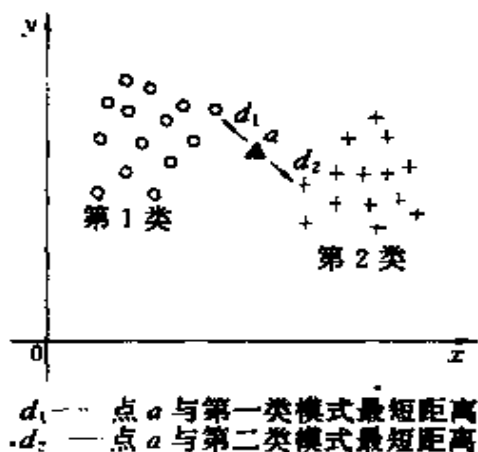


图 2.16 最邻近分类法

对于模式分类有许多数学上的方法。比较常用的有 K 最邻近 (K nearest neighbour classification) 分类方法。先看一个简单的例子,如图 2.16 所示。为判别图中模式点 a 属于哪一类,也就是说点

a 最邻近哪一类,可按式 2.10 进行计算:

$$f(x) = d_1 - d_2 \quad (2.10)$$

如果 $f(x) > 0$, 则 a 属于第一类; 如果 $f(x) < 0$, 则 a 属于第二类。但这种分类方法在使用上往往受到许多限制, 特别是当被分类模式点附近存在一些孤立模式点时, 用这一方法往往会导致分类的失败。因此, 实际应用 K 最邻近分类法往往采用分类模式点与模式空间中各组模式类别点的平均距离代替式 2.10 中的某个单点距离来进行分类。这里 K 表示计算平均距离的平均点数。下面介绍三种常用的计算模式距离的方法:

(1) 汉明距离(Hamming distance)

设有两个模式向量:

$$X = (x_1, x_2, \dots, x_n)$$

$$Y = (y_1, y_2, \dots, y_n)$$

则两个模式向量的汉明距离 d_h 表示为

$$d_h = \sum_{i=1}^n x_i \oplus y_i \quad (2.11)$$

(2) Euclid 距离(Euclidean distance)

$$d_{\text{euc}} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.12)$$

d_{euc} ——Euclid 距离;

Euclid 距离的二维几何表示如图 2.17 所示。这一计算距离的方法计算简单, 因而被广泛采用。

(3) 矩形距离(Square distance)

$$d_{\text{sq}} = \text{MAX}_{i=1}^n |x_i - y_i|$$

(2.13)

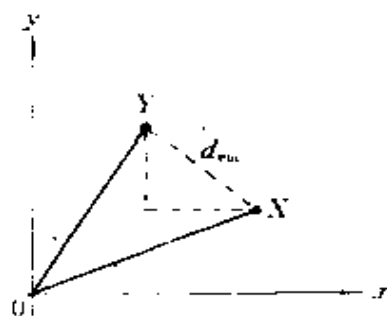


图 2.17 Euclid 距离的二维几何表示

2.8 线性分类器

线性分类法和线性分类器的概念在后面几章介绍人工神经网络各种算法时经常会遇到,为了便于以后的分析与解理,这里对线性分类器的概念做一简单介绍。

如图 2.18 所示,有 A_1 、 A_2 两组模式,现需要对这两组模式进行分类,求出这两组模式在模式空间中的分类边界线。为此,可用式 2.14 来定义这一分类界线:

$$f(A) = \sum_{i=1}^n w_i a_i \quad (2.14)$$

式中: a_i 为输入模式 A 中的第 i 个元素; w_i 为模式空间权向量 (weight vector) W 的第 i 个元素; n 为向量的维数。

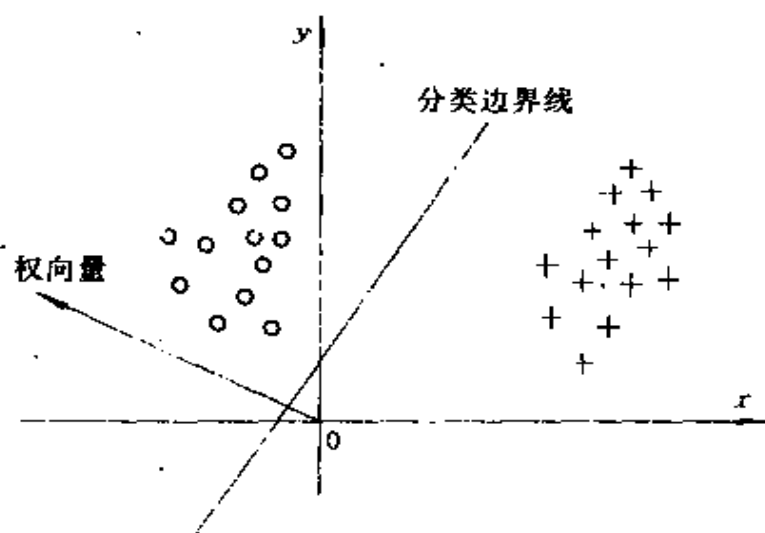


图 2.18 线性分类器举例

不难看出,式 2.14 取正或取负取决于输入模式向量 A 和权

向量 W 的值,而且当 $n=2$ 时,式 2.14 实际上是一条直线方程。假设对任意一组输入模式 A , 定义为:

$$\begin{aligned} &\text{当 } f(A) > 0 \text{ 时, } A \text{ 属于 } A_1 \text{ 模式组} \\ &\text{当 } f(A) < 0 \text{ 时, } A \text{ 属于 } A_2 \text{ 模式组} \end{aligned} \quad (2.15)$$

现在的问题是,能否找到一个合适的权向量 W ,使得对于分别属于 A_1 和 A_2 模式组中的所有输入模式都满足定义式 2.15。

现将式 2.14 用式 2.16 的形式表示为:

$$f(A) = |W| \cdot |A| \cdot \cos\varphi \quad (2.16)$$

式中: $|W| \cdot |A|$ 分别是向量 W 和 A 的模; φ 是两个向量之间的夹角, $\cos\varphi$ 的取值范围是 $[-1, +1]$, 所对应的角度为 $0 \sim 180$ 度。显然,两个模式组的分界线应在 $\varphi = \pm 90^\circ$ 处。如果对于一个模式空间可以找到这样一个权向量 W , 则可称这个模式空间为线性模式空间,式 2.14、式 2.16 称为线性分类器。如果在模式空间中找不到这样一个权向量 W , 则称这个模式空间为非线性模式空间。对于非线性模式空间,必须使用非线性分类器进行分类。在第三章中将介绍,历史上非线性分类问题曾经给人工神经网络的研究工作带来很大的困扰,出现了近 20 年的研究工作“冰河期”。

第三章 神经网络的初期模型与基本算法

3.1 感知机模型与感知机学习规则

如前章所述,生物神经元对信息的传递与处理是通过各神经元之间神经键的兴奋或抑制作用来实现的。根据这一事实,美国学者 F. Rosenblatt 于 1957 年在 M-P 模型和 Hebb 学习规则的基础上提出了具有自学习能力的感知机(perceptron)模型。

感知机模型结构如图 3.1 所示。

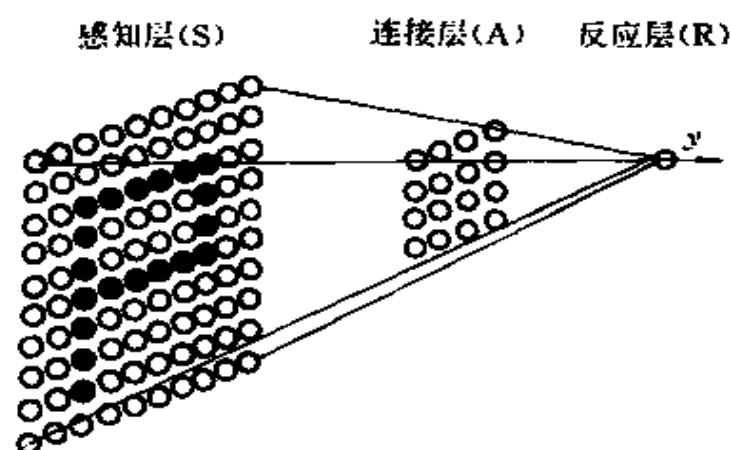


图 3.1 感知机模型结构

如图所示,感知机模型由感知层 S (Sensory)、连接层 A (Association)、反应层 R (Response) 三层构成。其中,S 层如同眼睛内的视网膜细胞,对应图形的黑、白圆点由 0、1 信号组成输入模

式;S 层至 A 层的连接权固定为 1,输入信号由感知层 S 传递到连接层 A 的过程中始终保持不变;连接层 A 至反应层 R 的连接权,预先随机确定为 $[-1, +1]$ 区间内的任意值,并根据学习规则在信号的传递过程中不断调整,从而使网络具有自学习功能。A 层至 R 层的输出由 M-P 模型算法决定。原始感知机模型只有一个输出节点,它相当于单个神经元。网络的学习目的是对输入模式(如图 3.1 所示的图象)进行正确的分类,即通过对模式样本的学习,能够对输入模式进行“0”、“1”分类。

单层感知机网络的一般拓扑结构如图 3.2 所示。

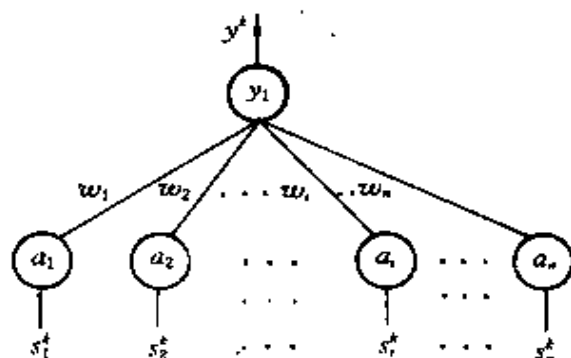


图 3.2 单层感知机网络的一般拓扑结构

设网络输入模式向量为:

$$S_k = (s_1^k, s_2^k, \dots, s_n^k)$$

对应的输出为: $y^k, k=1, 2, \dots, m$, 由 A 层至输出层 R 的连接权向量为:

$$W = (w_1, w_2, \dots, w_n)$$

网络按如下规则进行学习。

① 初始化 将 A 层至 R 层的连接权向量及输出单元的阈值 θ 赋予 $(-1, +1)$ 区间内的随机值。

② 连接权的修正 每个输入模式对 $(S_k, Y_k), k=1, 2, \dots, m$ 完成如下计算。

(a) 按式(3.1) 计算网络输出:

$$y = f\left(\sum_{i=1}^n w_i s_i^k - \theta\right) \quad (3.1)$$

式中, f 为双极值阶跃函数, 且

$$f(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases} \quad (3.2)$$

(b) 计算输出层单元希望输出 y^k 与实际输出 y 之间的误差:

$$d^k = y^k - y \quad (3.3)$$

(c) 修正 A 层各单元与输出层 R 之间的连接权与阈值:

$$w_i(N+1) = w_i(N) + \Delta w_i(N) \quad (3.4)$$

$$\Delta w_i(N) = \alpha \cdot s_i^k \cdot d^k \quad (3.5)$$

$$\theta(N+1) = \theta(N) + \Delta \theta_i(N) \quad (3.6)$$

$$\Delta \theta(N) = \beta \cdot d^k \quad (3.7)$$

式中, $i=1, 2, \dots, n$; N 为学习回数; α, β 为正常数, 称为学习率 ($0 < \alpha < 1, 0 < \beta < 1$)。

③ 对 m 个输入模式重复步骤②, 直到误差 d^k ($k=1, 2, \dots, m$) 趋于零或小于预先给定的误差限 ε 。

学习结束后的网络将学习样本模式以连接权的形式分布记忆下来。当给网络提供一输入模式时, 网络将按式 3.1 计算出输出值 y , 并可根据 y 为 +1 或 -1 判断出这一输入模式属于记忆中的哪一种模式或接近于哪一种模式。这一过程相当于人们根据一个人的肤色、身材、语言等特征, 运用大脑中已有的记忆, 联想判断出这个人属于哪一个国家、地区或民族。这个过程被称为回想过程。

为简单起见, 这里分析一个只有三个输入单元的网络例子, 以进一步理解感知机模型网络是如何进行模式分类与记忆的。由三个 0、1 元素组成的输入模式 $X=(x_1, x_2, x_3)$ 恰好构成了一个正方体的八个顶点, 如图 3.3 所示。把图中五个黑色顶点作为一种模式集合, 另外三个白色顶点作为另一种模式集合, 它们分别对应网络的八种输入模式, 当把它们反复地提供给网络进行学习时, 可按式 3.8 计算网络的实际输出。

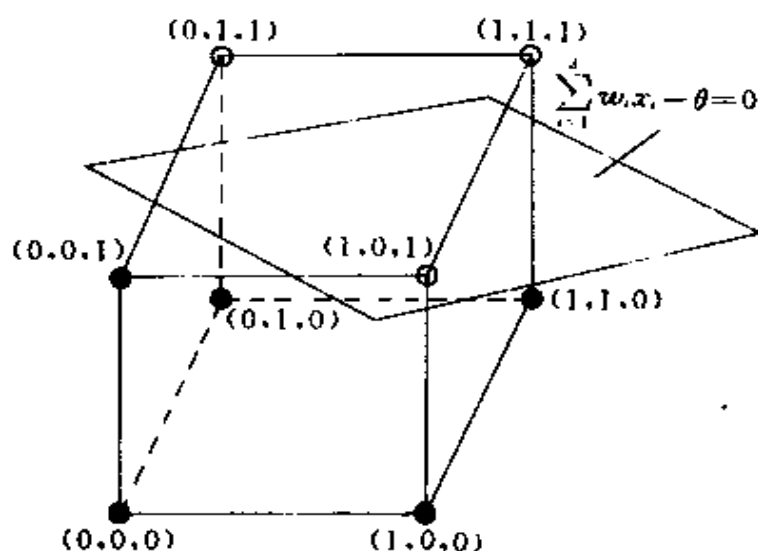


图 3.3 三维线性输入模式及截割平面

$$y = f\left(\sum_{i=1}^3 w_i x_i^k - \theta\right) \quad (3.8)$$

$$k = 1, 2, \dots, 8$$

当实际输出与希望输出一致时,误差为零,连接权无需修改,当实际输出与希望输出不相同,连接权需要增加或减少 2 倍的 $\alpha \cdot d^k$, 阈值也要产生相应的变化。实际输出取 +1 还是 -1 取决于式 $\sum_{i=1}^3 w_i x_i^k - \theta$ 大于零还是小于零,而式 $\sum_{i=1}^3 w_i x_i^k - \theta = 0$ 是由参数 $W = (w_1, w_2, w_3)$ 和 θ 值决定的一个三维平面方程。只有当这一截割平面的空间位置如图 3.3 所示时,才能正确地划分输入模式集合。因此,整个学习和记忆过程,就是根据实际输出与希望输出之间的误差调整参数 W 和 θ ,即调整截割平面的空间位置使之不断移动直到它能将两类模式恰当划分的过程。如果将输入模式由三维推广至 n 维,则学习过程可看作是由式 $\sum_{i=1}^n w_i x_i - \theta = 0$ 决定的 n 维超平面不断向正确划分输入模式的位置移动的过程。

感知机模型网络并不限定只有一个输出单元,它也可以有如

图 3.4 所示的多输出单元形式。在这种情况下感知机网络学习规则修改如下：设网络有 n 个输入单元， p 个连接层单元， q 个输出单元，学习模式对为： $S_k = (s_1^k, s_2^k, \dots, s_n^k)$ 、 $Y_k = (y_1^k, y_2^k, \dots, y_q^k)$ ， $k = 1, 2, \dots, m$ ；并设连接层的输出向量为： $A_k = (a_1^k, a_2^k, \dots, a_p^k)$

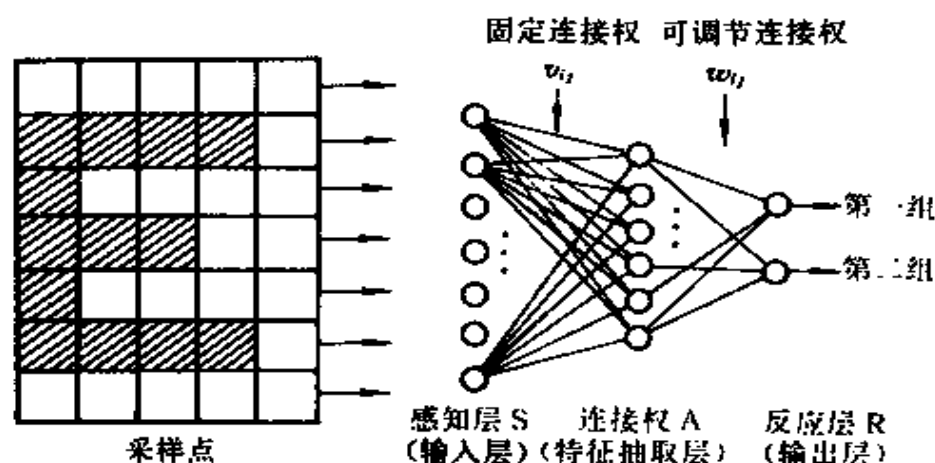


图 3.4 多输出单元感知机模型网络

① 初始化 将 S 层单元至连接层单元的连接权矩阵

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1p} \\ \vdots & \vdots & & \vdots \\ v_{N1} & v_{N2} & \dots & v_{Np} \end{bmatrix}$$

的各个元素及 A 层各单元的输出阈值赋予 $[-1, +1]$ 之间的随机值，一般情况下 $v_{ij} = 1$ ， $i = 1, 2, \dots, N$ ， $j = 1, 2, \dots, p$ ， $\theta_j = 0$ ，

N 为输入单元的个数， p 为连接层单元的个数。注意 V 与 θ 在整个学习过程中固定不变。

将 A 层单元至输出层单元的连接权矩阵

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1q} \\ \vdots & \vdots & & \vdots \\ w_{p1} & w_{p2} & \dots & w_{pq} \end{bmatrix}$$

的各个元素及连接层单元和输出单元的各个阈值向量

$$r_j = [r_1, r_2, \dots, r_p]$$

$$\theta_j = [\theta_1, \theta_2, \dots, \theta_q]$$

的各元素赋予 $[-1, +1]$ 之间的随机值。 p 为连接层单元的个数, q 为输出层单元个数。

② 连接权修正 对每一个输入模式对 (A_k, Y_k) $k=1, 2, \dots, m$ 完成如下计算:

(a) 计算连接层各单元输出

$$a_j^k = f\left[\sum_{i=1}^n v_{ij} s_i^k - v_j\right] \quad (3.9)$$

$$j = 1, 2, \dots, p$$

以 a_i 作为输出层各单元的输入计算输出层各单元的输出值

$$y_j = f\left[\sum_{i=1}^p w_{ij} \cdot a_i - \theta_j\right] \quad (3.10)$$

$$j = 1, 2, \dots, q$$

以上两式中 $f(x)$ 为双极值函数。

(b) 计算各输出单元的期望 y_j^k 与实际输出 y_j 之间的误差

$$d_j^k = y_j^k - y_j \quad j = 1, 2, \dots, q \quad (3.11)$$

(c) 修正A层与R层之间的连接权与R层输出阈值

$$w_{ij}(N+1) = w_{ij}(N) + \Delta w_{ij}(N) \quad (3.12)$$

$$\Delta w_{ij}(N) = \alpha \cdot a_i^k \cdot d_j^k \quad (3.13)$$

$$i = 1, 2, \dots, p \quad j = 1, 2, \dots, q$$

$$\theta_j(N+1) = \theta_j(N) + \Delta \theta_j(N) \quad (3.14)$$

$$\Delta \theta_j(N) = \beta \cdot d_j^k \quad (3.15)$$

$$i = 1, 2, \dots, p \quad j = 1, 2, \dots, q$$

式3.12~3.15中, $0 < \alpha < 1, 0 < \beta < 1, N$ 为学习次数。

③ 对 m 个输入学习模式重复步骤②,直至 d_j^k $j=1, 2, \dots, q$
 $k=1, 2, \dots, m$ 趋于零或小于预先给定的误差限 ϵ 。

我们把提供给网络学习的所有模式的集合称为网络的学习样本集合,每个学习模式由输入向量和希望输出向量对组成。图3.5所示的是一些二值模式对。

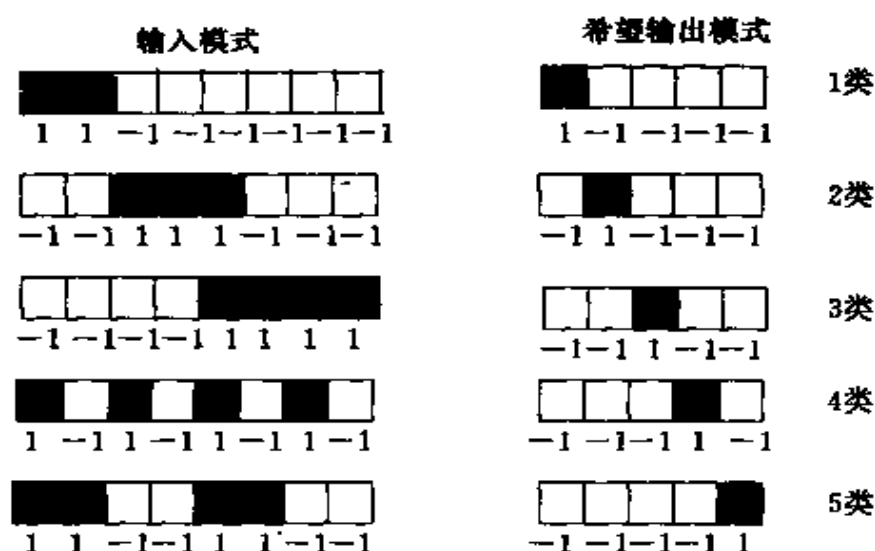


图3.5 学习样本集合举例

这里所谓希望输出模式就是对输入模式的一种人为分类。由式 3.11 可知：

$$d_j = y_j^t - y_j = \begin{cases} 2 & \text{当 } y_j^t = 1, y_j = -1 \text{ 时} \\ 0 & \text{当 } y_j^t = y_j \text{ 时} \\ -2 & \text{当 } y_j^t = -1, y_j = 1 \text{ 时} \end{cases}$$

当实际输出与希望输出相同时误差为零，不需对连接权和输出阈值进行调整；当实际输出与希望输出不相同，如 $y_j^t = 1, y_j = -1$ ，而 $a > 0$ 时，说明权值 w_{ij} 太小，使超平面偏离了正确划分输入模式的位置。因此应以两倍 $a_i (a_i > 0)$ 的调整量增大 w_{ij} 的值，使超平面向正确划分输入模式的坐标位置移动。在学习过程中，当所有 m 个样本模式对提供给网络学习一遍之后，还需从头再继续提供给网络学习。如此反复进行下去，直至超平面十分接近其正确位置为止。这个接近过程称为**网络的收敛过程**。

在实际应用中，作为学习过程的结束标志一般给出两个限制：一是为了防止因学习过程发散（下面将详述这个问题），使学习过程进入无限振荡而设置的最大学习回数；另一个是作为衡量整个

学习过程收敛程度的综合误差(或称为全局误差)限值。当学习回数超过最大学习回数或综合误差小于预先设定的误差限值时,学习过程都将停止。关于综合误差有各种不同的定义,但本质上都是一致的。这里主要介绍两种:

;(1) 平均平方(RMS)误差

$$E = \sqrt{\frac{\sum_{k=1}^M \sum_{j=1}^Q (y_j^t - y_j^t)^2}{M \cdot Q}} \quad (3.16)$$

式中, M 为样本模式对的个数, Q 为输出单元的个数。

图 3.6 为图 3.5 所示的样本模式按 RMS 误差收敛曲线。由图可知,当所有模式对学习 5 次后,误差 RMS 收敛至极小值。

对于图 3.5 所示样本模式取 $\alpha = \beta = 0.1$, 初始权值 w_{ij} 为 $(-0.05, +0.05)$ 间的随机值, $v_{ij} = 1$ 。

(2) 误差平方和

把所有实际输出与希望输出的误差平方和作为检验网络收敛的误差标准。

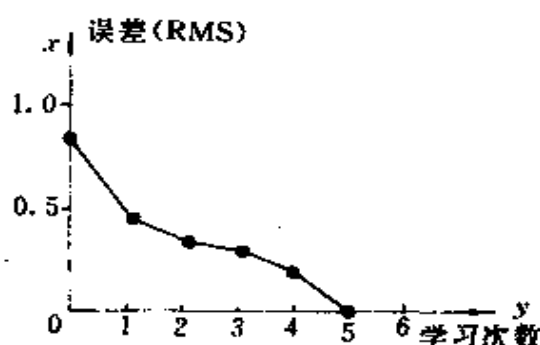


图 3.6 学习收敛曲线

$$E_k = \sum_{j=1}^q (y_j^t - y_j^t)^2 / 2 \quad (3.17)$$

$$E = \sum_{k=1}^M E_k \quad (3.18)$$

式中, E_k 为一对样本模式的所有输出单元的误差平方和, E 为所有样本模式对的误差平方和,也就是检验网络收敛的误差标准; q 为输出单元个数; M 为样本模式对数。

还需要指出的是学习率 α, β 的取值问题, α, β 反映了学习过程的进行速度。 α 小时,学习收敛过程缓慢,但却在学习模式符合规定的情况下能保证网络收敛到极小(或最小)点; α 大时学习速

度加快,但却容易引起学习过程振荡,最终可能使网络收敛不到极小(或最小)点。关于学习率,许多学者对其进行了各种研究,提出了许多静态或动态设置学习率的准则或方案。

3.2 感知机的局限性

感知机模型网络具有对输入模型进行自动分类的功能。那么,它是否可以对任意输入模型进行分类呢?回答是否定的。

一般来说,对于 n 维空间,凡可以用 $n-1$ 维超平面进行适当分割的点集合称为线性分割集合(如图 3.3 所示)。只要输入模式属于线性可分割集合就可以用感知机网络对其进行正确分类;反之,网络的学习过程将无法收敛,即不能做出正确的分类。对此有以下感知机学习收敛定理。

定理

当输入模式满足式 3.19 的线性可分离条件时,学习过程必能在有限次内收敛。

$$\begin{aligned} \sum w_{ij}a_i - \theta_j > 0, \text{ 当 } A_k = (a_1^k, a_2^k, \dots, a_n^k) \text{ 由 } \\ P_+ \text{ 类模式数据组成;} \\ \sum w_{ij}a_i - \theta_j < 0, \text{ 当 } A_k = (a_1^k, a_2^k, \dots, a_n^k) \text{ 由 } \\ P_- \text{ 类模式数据组成;} \end{aligned} \quad (3.19)$$

感知机模型网络收敛条件式 3.19 表明:由 P_+ 和 P_- 模式组成的输入模式,当模式维数为 2 时,可以用一条直线对其正确划分;当模式维数为 3 时,可以用一个空间平面划分;当模式维数大于 3 时,可以用超平面边界对其正确划分。

感知机的发明者 Rosenblatt 对这一定理进行了详细证明,读者可参考文献^[12]。

对于如图 3.7 所示的输入模式,不可能用一空间平面将由黑点和白点分别组成的顶点集合分开。这类输入模式称为线性不可分割模式。单层感知机对这类输入模式无法进行正确的分类。针对感知机无法解决非线性分类问题的缺陷,当时作为人工智能创始人之一的美国麻省理工学院 Minsky 和 Papert 在他们所著的《感知机》一书中否定了当时许多学者做的各种尝试。他们指出,对于如“异或”这类非线性输入模式,无论如何调节连接权,都不能获得所要求的非线性

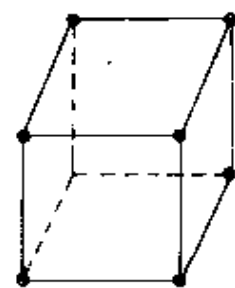


图 3.7 线性不可分割集合例子

映射。如图 3.8 所示,划分“异或”问题必须同时使用两条直线。Minsky 和 Papert 指出,造成感知机对“异或”问题分类失败的主要原因在于网络无中间隐含单元作为输入模式的“内部表示”。

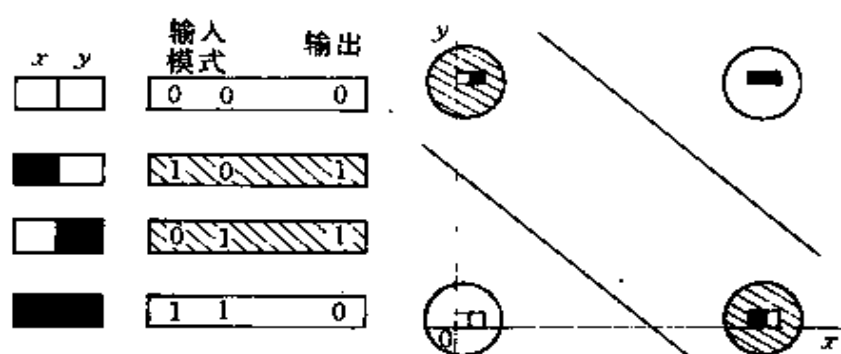


图 3.8 异或分类问题

因此解决这一问题的有效办法是给网络增加一些隐含单元,如图 3.4 所示例子。这里我们分析如何用三层感知机网络划分“异或”问题。如图 3.9 所示的三层网络,设 S 层至 A 层的连接权矩阵为:

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

A 层至 R 层的连接权向量为 $W=[1,1]$

按式 3.1, 设 $f(x)=x$ 为线性函数, 有:

$$\begin{cases} x_{11} = v_{11} \cdot x_{01} + v_{12} \cdot x_{02} - \gamma_1 & \text{①} \\ x_{12} = v_{21} \cdot x_{01} + v_{22} \cdot x_{02} - \gamma_2 & \text{②} \\ y = w_1 \cdot x_{11} + w_2 \cdot x_{12} - \theta & \text{③} \end{cases} \quad (3.20)$$

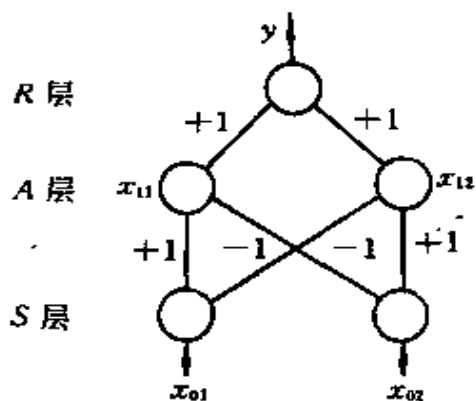
将连接权值代入得:

$$\begin{cases} x_{11} = 1 \cdot x_{01} - 1 \cdot x_{02} - \gamma_1 & \text{①} \\ x_{12} = 1 \cdot x_{01} - 1 \cdot x_{02} - \gamma_2 & \text{②} \\ y = 1 \cdot x_{11} + 1 \cdot x_{12} - \theta & \text{③} \end{cases} \quad (3.21)$$

取 $\theta=2$, 由式 3.21 可以看出, ③式是对①式和②式的逻辑“与”运算结果。当令①、②式分别为零时有:

$$\begin{cases} 1 \cdot x_{01} - 1 \cdot x_{02} = \gamma_1 \\ 1 \cdot x_{01} - 1 \cdot x_{02} = \gamma_2 \end{cases} \quad (3.22)$$

式 3.22 表示两条直线方程, 而这两条直线的逻辑“与”区域正好代表了“异或”问题“1”输出区域; 这个区域的“非”区域代表了“异或”问题的“0”输出, 正如图 3.8 所示那样。由这个例子可以看出, 正是因为有了式 3.21 中①、②式的“内部表示”加上③式的逻辑运算, 才使感知机模型网络对非线性的“异或”输入模式进行正确分类。



但是, 问题是否就此全部解决 图 3.9 三层网络解决“异或”问题了呢? 我们在介绍三层感知机模型网络时曾明确指出, 只有中间层至输入层的连接权能够随学习过程按一定规则进行修正, 而输入层至中间层的连接权在整个学习过程中是固定不变的。在上例中, 我们在初始时就将各个连接权的正确方案设定下来, 所以能够得出正确的分类结果, 但在实际问题中往往不可能不经过学习而

事先完全确定正确的连接权值。一旦输出存在误差时如何判断是哪一个连接权的过错,并按什么规则对其进行调整,这些都还是没有解决的问题。当时 Minsky 和 papert 指出:“将感知机扩展到多层装置是没有意义的。这种看法的正确与否是一个很大的研究课题。或许在将来,一些更强有力的收敛定理会被发现,或者能明白:无法在多层装置中获得有意义的学习定理的道理。”

幸运的是,他们所说的第一种可能终于在后来成为现实。这就是在下一章将要介绍的误差逆传播神经网络学习定理。应该指出,虽然感知机只能对线性输入模式进行正确分类,但它作为人工神经网络的初期模型,特别是其反映出的自学习、自组织思想,对整个神经网络理论的研究和发展确实产生了深远的影响。

3.3 自适应线性神经网络

感知机模型网络是以二值数字量作为输入模式的离散时间型神经网络。作为神经网络的初期模型,与感知机模型相对应,自适应线性(Adaline-Mada Line)神经网络是以连续线性模拟量为输入模式,在拓扑结构上与感知机网络十分相似的一种连续时间型线性神经网络。这种网络模型是美国学者 Widrow 和 Hoff 于 1960 年提出的,简称 Adaline 网络。这种网络实际上是一种自适应阈值逻辑元件,主要应用于自适应系统等连续可调过程。在这一网络的基础上, Widrow 又提出了一种多层自适应线性神经网络(Multilayer Adaline),简称 Madaline 网络。它是由两个以上 Adaline 网络组成的一种具有三层结构的神经网络。

首先介绍 Adaline 网络,其网络结构如图 3.10 所示。

网络的模拟输入向量为 $A_k = (a_0^k, a_1^k, a_2^k, \dots, a_n^k)$; 希望模拟输出为 $y^k, k=1, 2, \dots, m$; 连接权向量为 $W = (w_0, w_1, w_2, \dots, w_n)$, 其中

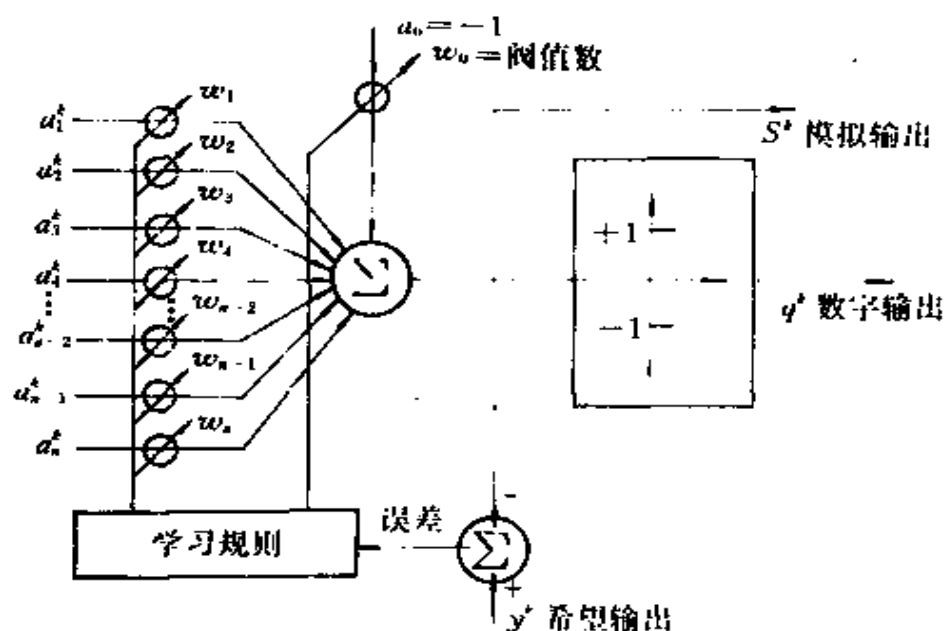


图 3.10 自适应线性神经网络(Adaline)

a_0^k 恒为 -1 , w_0 是控制阈值。网络的模拟输出按式 3.23 求得; 数字输出按式 3.24 求得。

$$S^k = \sum_{i=0}^n w_i a_i^k \quad (3.23)$$

$$q^k = f(s) \quad (3.24)$$

式中 $f(s)$ 为双极值函数。

这一网络的学习过程是按使误差平方和最小化原则, 即 LMS (Least Mean Square) 算法, 反复对各连接权进行修正的过程。这里误差定义为希望输出与实际输出之差, 即

$$\begin{aligned} \Delta w_i &= \alpha \cdot a_i (y^k - S^k) \\ i &= 1, 2, \dots, n \end{aligned} \quad (3.25)$$

式中, α 为正的常数, 即学习率 ($0 < \alpha < 1$)。式 3.25 称为 **Widrow-Hoff 学习规则**。

下面在介绍 Adaline 网络的基础上, 介绍 Madaline 网络。网络的结构如图 3.11 所示。由图可知它是由多个 Adaline 网络和

AND 逻辑单元构成的。

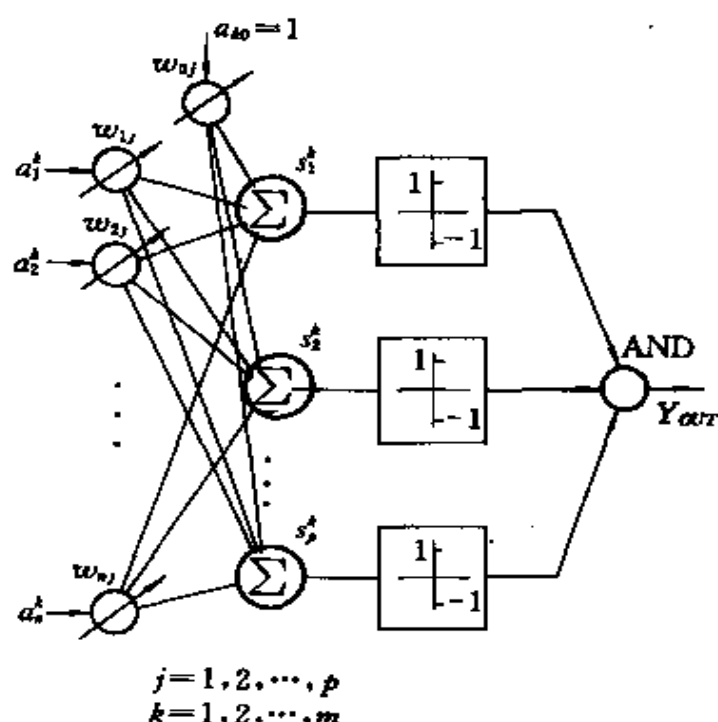


图 3.11 多层自适应线性神经网络结构

设网络的模拟输入向量为: $A_k = (a_0^k, a_1^k, a_2^k, \dots, a_n^k)$, 希望输出向量为 $Y_k = (y_1^k, y_2^k, \dots, y_p^k)$, $k=1, 2, \dots, m$; 连接权向量为 $W_{ij} = (w_{i1}, w_{i2}, \dots, w_{ip})$, $i=1, 2, \dots, n$; a_0^k 恒为 +1。 m 为输入模式对数, n 为输入单元数, p 为中间层单元数。网络的实际数字输出服从于多数表决原则, 即当 AND 单元的 1 输入占多数时, 输出为 1; 反之为 -1。网络的学习规则同样是按最小均方误差原则进行。

设网络对所有输入模式的误差平方和为

$$E = \sum_{k=1}^m \left\{ \frac{1}{2} \left[\sum_{j=1}^p (y_j^k - s_j^k)^2 \right] \right\} \quad (3.26)$$

式中

$$s_j^k = \sum_{i=1}^n w_{ij} a_i^k \quad (3.27)$$

则网络对第 k 个输入模式的误差为

$$E_k = \frac{1}{2} \left[\sum_{j=1}^p (y_j^k - s_j^k)^2 \right] \quad (3.28)$$

$$k = 1, 2, \dots, m$$

求 E_k 对连接权 w_{ij} 的偏导为:

$$\begin{aligned} \frac{\partial E_k}{\partial w_{ij}} &= -(y_j^k - s_j^k) \cdot \frac{\partial s_j^k}{\partial w_{ij}} \\ &= -(y_j^k - s_j^k) \cdot a_i^k \end{aligned}$$

根据最小均方误差的原则, 应使连接权 w_{ij} 按正比于负的偏导 $\left(-\frac{\partial E_k}{\partial w_{ij}} \right)$ 的方向变化, 即:

$$\begin{aligned} \Delta w_{ij} &= -\alpha \cdot \frac{\partial E_k}{\partial w_{ij}} \\ &= \alpha \cdot (y_j^k - s_j^k) \cdot a_i^k \\ &= \alpha \cdot d_j^k \cdot a_i^k \end{aligned} \quad (3.29)$$

其中, α 为学习率 ($0 < \alpha < 1$); $d_j^k = (y_j^k - s_j^k)$ 为误差。

从数学观点来看, 按式 (3.29) 调整连接权, 实际上就是使误差函数 E_k 按梯度下降 (gradient descent), 从而使 E_k 达到最小。

但是, 网络的学习目的并不只是使一个输入模式的误差函数达到最小, 而是要使全部用于学习的输入模式的误差都达到最小, 即使式 3.26 所定义的误差达到最小。为此应按下式调整连接权 w_{ij} :

$$\begin{aligned} \Delta w_{ij} &= -\alpha \cdot \frac{\partial E}{\partial w_{ij}} = \sum_{k=1}^m \left(-\alpha \cdot \frac{\partial E_k}{\partial w_{ij}} \right) \\ &= \alpha \cdot \sum_{k=1}^m d_j^k \cdot a_i^k \end{aligned} \quad (3.30)$$

由式 3.30 可知, 连接权的修正正比于整个学习模式集合内的各个模式所对应的负梯度之和, 因此这是一种累积误差修正方法。在实际应用中, Madaline 的 LMS 学习往往是每提供一种学习模式都要进行修正, 这实际上偏离了全局误差式 3.26 上的真正梯度下降方向。但是当学习率 α 充分小时, 这种偏离是可以忽略的, 如

图 3.12 所示。

下面将 Madaline 的 LMS 学习规则总结如下：

① 初始化 给各连接权 w_{ij} 赋予 $[-1, +1]$ 区间内的随机值。

② 任选一学习模式对提供给网络。

③ 计算网络输出值

$$s_j^k = \sum_{i=0}^n w_{ij} \cdot a_i \quad (3.31)$$

$$j = 1, 2, \dots, p$$

④ 计算网络各输出单元的希望输出 y_j^k 与实际输出 s_j^k 之间的误差

$$d_j^k = y_j^k - s_j^k \quad (3.32)$$

$$j = 1, 2, \dots, p$$

⑤ 进行连接权修正

$$w_{ij}(N+1) = w_{ij}(N) + \Delta w_{ij}(N) \quad (3.33)$$

$$\Delta w_{ij}(N) = a \cdot a_i^k \cdot d_j^k \quad (3.34)$$

$$i = 0, 1, 2, \dots, n \quad j = 1, 2, \dots, p \quad (0 < a < 1)$$

⑥ 取下一个学习模式对提供给网络,重复步骤③~⑤,直到误差 $d_j^k (j=1, 2, \dots, p; k=1, 2, \dots, m)$ 变得足够小为止。

⑦ 计算最终输出

$$Y_{out} = f(s_1) \text{AND} f(s_2) \dots \text{AND} f(s_p) \quad (3.35)$$

其中 $f(s)$ 为双极值函数。

网络学习结束后,需要进行模式分类或识别时,可按式 3.31 进行回想,根据各个输出值 $s_j \quad j=1, 2, \dots, p$ 对输入模式进行分类,也可通过双极值函数的输出进行近似分类。这一学习规则有时

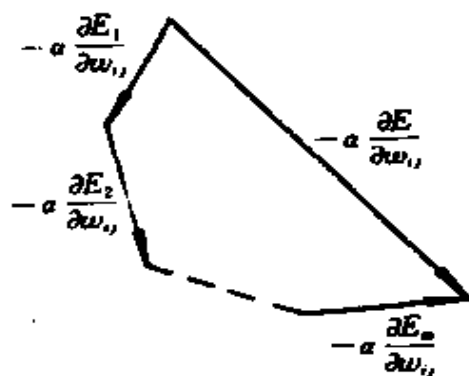


图 3.12 近似梯度下降

也称为 δ 学习规则。

需要指出的是,对于 Madaline 网络的最终输出 Y_{out} ,由于它经过了非线性函数——AND 逻辑单元的作用,因此使得网络具有对非线性输入模式进行分类的能力。如图 3.13 所示是由两个 Adaline 网络组成的 Madaline 网络。

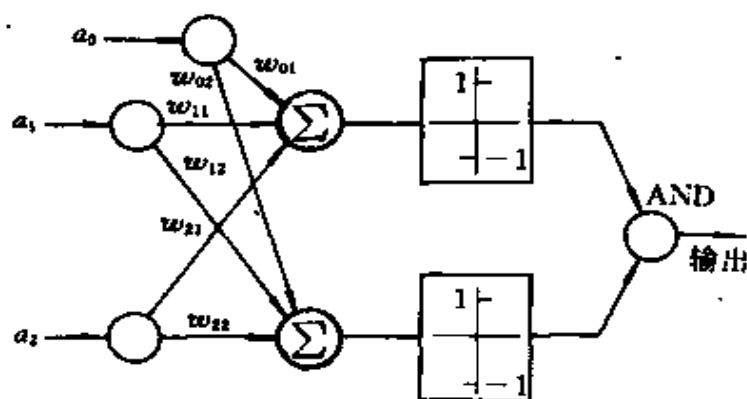


图 3.13 由两个 Adaline 组成的 Madaline 网络

当设 $w_{11}=w_{22}=1, w_{21}=w_{12}=-1, w_{01}=w_{02}=1, a_0=1$ 时,这一网络可对 a_1, a_2 输入“异或”值进行正确分类。但是同感知机的情况一样,这种对非线性输入模式进行正确分类的能力,是在预先能正确地确定各连接权值的条件下形成的,而没有一套完整的对非线性输入模式进行分类的学习规则。因此,从总体上说, Madaline 网络同感知机网络一样只具有对线性输入模式进行分类能力,即只有线性映射的能力。

最后我们来观察一下感知机学习规则与 Madaline 的 LMS 学习规则的主要区别。由感知机的输出方程式 3.10 可知,连接权的调整实际上是对超平面方程中各个系数进行调整,整个学习过程表现为超平面的位置调整。而 Madaline 的 LMS 学习规则则是建立在均方误差最小化的基础上的,由式 3.26 可知,误差函数 E 是连接权 w_{ij} 的平方函数。因此,它的学习过程表现为误差曲面上的梯度下降。同时,由于由 E 构成的连接权 w_{ij} 空间的抛物面只有一

个极小点即最小点,因此 Madaline 的 LMS 学习规则可保证最终误差函数最小。但这需要无限次学习,所以实际应用中有有限次的学习结果只能得到近似解。现已证明,当输入模式 $A_i = (a_0^i, a_1^i, \dots, a_n^i)$ 线性无关时,用 LMS 学习规则能使误差函数 E 最终为零,即实现对学习模式的完整联想^[13]。

3.4 自适应线性神经网络的应用举例

作为 Adaline 和 Madaline 网络的发明人,Widrow 在对网络的理论进行深入研究的同时,对这一网络的应用也做了大量工作,先后将网络应用于天气预报、语音识别、心电图诊断、图象分析等研究领域。特别值得一提的是,他将这一网络成功地应用于倒立振子自适应平衡系统。另外,作为与这一网络应用相关的一个重要实例,他与其他合作者(1976 年)提出了用于系统辨识的自适应滤波器的构想,并加以成功地应用。图 3.14 是自适应滤波器的基本结

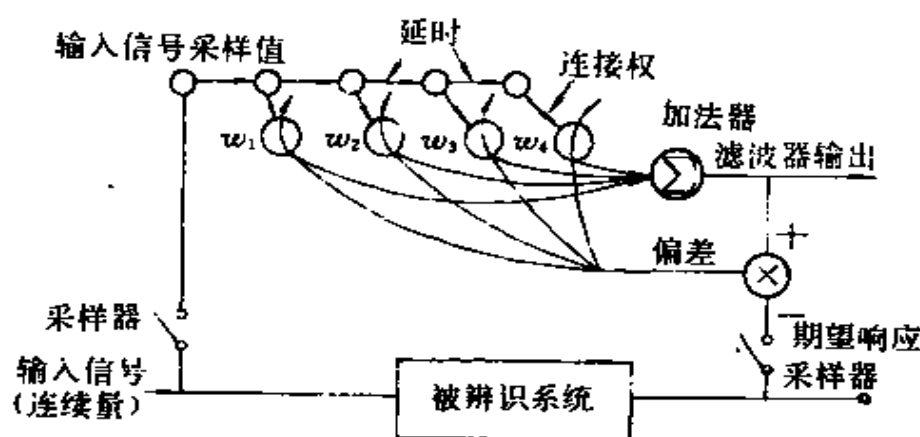


图 3.14 自适应滤波器辨识系统结构

构图。如图所示,系统辨识的主要部分是由不带双极值输出函数的

Adaline 网络组成。连续输入信号的采样值经不同延时后,作为网络的输入模式,被辨识系统输出量的采样值作为网络的希望输出模式。Adaline 网络的各连接权 w_1, w_2, w_3, w_4 实际上就是被测系统微分方程中各阶导数的系数。随着网络学习过程的进行,这些系数逐渐逼近系统的真实值,直至得到系统的数学模型。

第四章 多阶层神经网络与 误差逆传播算法

4.1 多阶层网络与误差逆传播算法的提出

感知机的发明,曾使神经网络的研究迈出了历史性的一步。但是正如第三章所述,尽管感知机具有很出色的学习和记忆功能,可由于它只适用于线性模式的识别,对非线性模式的识别显得无能为力,甚至不能解决“异或”这样简单的非线性运算问题。虽然人们当时已发现,造成感知机这种缺陷的主要原因是由于网络无隐含层作为输入模式的“内部表示”,并做了在输入层和输出层之间增加一层或多层隐单元的尝试,但是当时还找不到一个适用于多层网络的行之有效的学习规则,甚至对是否存在这样一条规则抱有怀疑。因此,一时使人们对神经网络的发展前途产生了动摇。尽管如此,神经网络本身那种“无穷奥秘”的魅力,仍然吸引着一些致力于这一领域研究的学者。“有志者事竟成”,有的研究者另辟蹊径,撇开“阶层”概念,创造出无阶层的全连接型神经元网络,并提出可识别非线性模式的学习规则,其中最著名的就是将在第五章介绍的 Hopfield 神经网络。而另一些研究者,通过艰苦的探索和努力,终于在阶层型神经网络的研究中,打开了一条希望的通路,这就是目前应用最广,其基本思想最直观、最容易理解的多阶层神经网络及误差逆传播学习算法(Error Back-Propagation)。有时也将按这

一学习算法进行训练的多阶层神经网络直接称为误差逆传播神经网络。

误差逆传播神经网络是一种具有三层或三层以上的阶层型神经网络。上、下层之间各神经元实现全连接,即下层的每一个单元与上层的每个单元都实现权连接,而每层各神经元之间无连接。网络按有教师示教的方式进行学习,当一对学习模式提供给网络后,神经元的激活值,从输入层经各中间层向输出层传播,在输出层的各神经元获得网络的输入响应。在这之后,按减小希望输出与实际输出误差的方向,从输出层经各中间层逐层修正各连接权,最后回到输入层,故得名“误差逆传播算法”。随着这种误差逆传播修正的不断进行,网络对输入模式响应的正确率也不断上升。

完整提出并被广泛接受误差逆传播学习算法的是以 Rumelhart 和 McClelland 为首的科学家小组。他们在 1986 年出版的名为《Parallel Distributed Processing》^[13] (《并行分布信息处理》)一书中,对误差逆传播学习算法进行了详尽分析与介绍,并对这一算法的潜在能力进行了深入探讨。实现了当年 Minsky 预想的第一种可能。其实早在这之前,Paul Werbas 博士于 1974 年在他的博士论文中就已独立的提出过误差逆传播学习算法^[14]。

由于误差逆传播网络及其算法增加了中间隐层并有相应学习规则可循,使其具有对非线性模式的识别能力。特别是其数学意义明确、步骤分明的学习算法,更使其具有广泛的应用前景。

4.2 误差逆传播神经网络结构与学习规则

为方便起见,把误差逆传播网络简称为 BP (Back Propagation)网络。

典型的 BP 网络是三层、前馈阶层网络,即:输入层、隐含层

(也称中间层)和输出层。各层之间实行全连接。如图 4.1 所示。

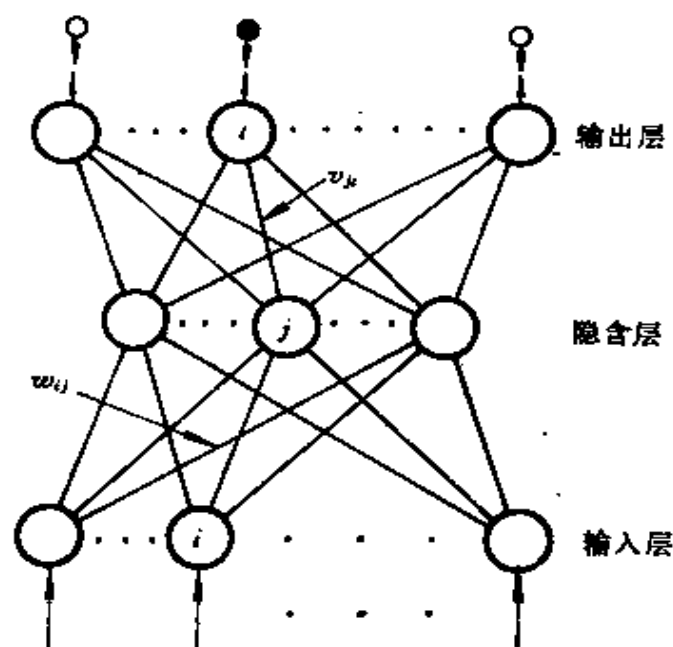


图 4.1 三层 BP 网络结构

BP 网络的学习,由四个过程组成:输入模式由输入层经中间层向输出层的“模式顺传播”过程,网络的希望输出与网络实际输出之差的误差信号由输出层经中间层向输入层逐层修正连接权的“误差逆传播”过程,由“模式顺传播”与“误差逆传播”的反复交替进行的网络“记忆训练”过程,网络趋向收敛即网络的全局误差趋向极小值的“学习收敛”过程。归结起来为,“模式顺传播”→“误差逆传播”→“记忆训练”→“学习收敛”过程。BP 网络学习规则有时也称广义 δ 规则。

下面分别介绍和分析这四个过程。

(1) 模式的顺传播

模式顺传播过程是由输入模式提供给网络的输入层开始的。输入层各个单元对应于输入模式向量的各个元素。设输入模式向量为

$$A_k = (a_1, a_2, \dots, a_n) \quad k = 1, 2, \dots, m; m \text{——学习模式对数,}$$

n ——输入层单元个数；

对应输入模式的希望输出向量为

$Y_k = (y_1, y_2, \dots, y_q)$ q ——输出层单元数；

根据 M-P 神经元模式原理, 首先按式 4.1 计算中间层各单元的输入。

$$s_j = \sum_{i=1}^n w_{ij} a_i - \theta_j$$

$$j = 1, 2, \dots, p \quad (4.1)$$

式中, w_{ij} ——输入层至中间层的连接权；

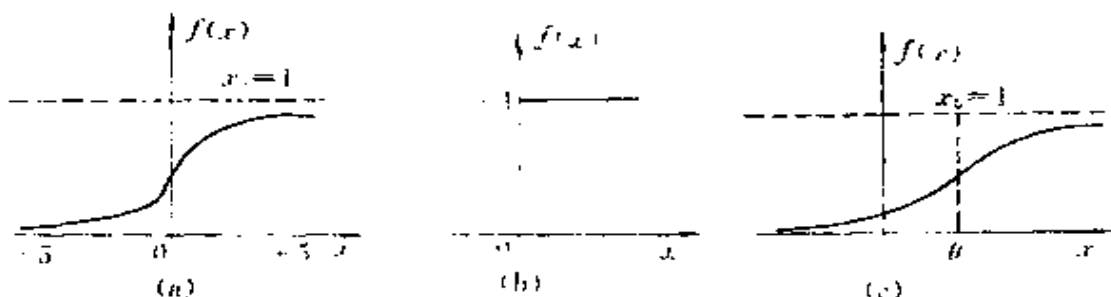
θ_j ——中间层单元的阈值；

p ——中间层单元个数。

为模拟生物神经元的非线性特性, 以 s_j 作为 S 函数 (Sigmoid 函数) 的自变量, 计算中间层各单元的输出。 S 函数的数学表达式如式 4.2 所示:

$$f(x) = \frac{1}{1 + e^{-\frac{x}{x_0}}} \quad (4.2)$$

S 函数的输出曲线如图 4.2(a) 所示:



(a) S 函数曲线 (b) 阶跃函数曲线 (c) 向右平移 θ 个单位的 S 函数曲线

图 4.2 非线性输出曲线

由图可知, S 函数输出曲线两端平坦, 中间部分变化剧烈。当 $x < -5$ 时, $f(x)$ 接近于 0; 当 $x > +5$ 时, $f(x)$ 接近于 1; 而当 x 在 0 附近时 ($-5 < x < +5$), $f(x)$ 才真正起转移作用。 S 函数与阶跃函数 (如图 4.2(b) 所示) 相比, 从形式上看具有“柔软性”, 从生理

学角度看,一个人对远远低于或高于他智力和知识水平的问题,往往很难产生强烈的思维反映;从数学角度看, S 函数具有可微分性。正是因为 S 函数更接近于生物神经元的信号输出形式,所以选用 S 函数作为 BP 网络的输出函数。同时 BP 学习规则本身也要求网络的输入,输出函数是可微分的(后面将介绍其原因), S 函数不但具有可微分性,而且具有饱和非线性特性,这又增强了网络的非线性映射能力。把式 4.1 代入式 4.2 得:

$$b_j = f(s_j) = \frac{1}{1 + e^{-s_j}} = \frac{1}{1 + e^{-\sum_{i=1}^n w_{ij}x_i + \theta_j}} \quad (4.3)$$

$$j = 1, 2, \dots, p$$

式中, b_j 为中间层 j 单元的激活值。单元输出阈值 θ_j 是为模拟生物神经元的阈值电位而设置的。实际上在网络的学习过程中,它和各连接权一样也不断地被修正。阈值 θ_j 的作用反映在 S 函数的输出曲线上,是使曲线向右平行移动了 θ_j 个单位,如图 4.2(c) 所示。它起到了调节神经元兴奋水平的作用。

按模式顺传播的思路,以式 4.4、式 4.5 计算输出层各单元的输入、输出。

$$L_t = \sum_{j=1}^n v_{jt} \cdot b_j - \gamma_t \quad (4.4)$$

$$C_t = f(L_t) \quad (4.5)$$

$$t = 1, 2, \dots, q$$

式中, v_{jt} ——中间层至输出层连接权;

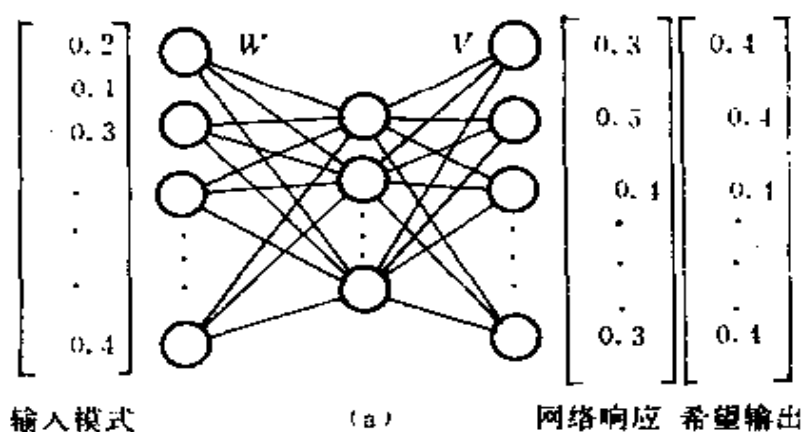
γ_t ——输出层单元阈值;

f —— S 函数;

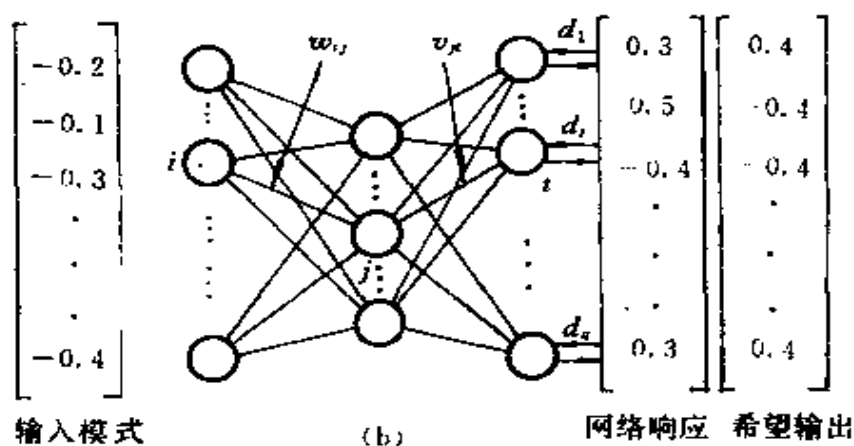
至此,一个输入模式完成了一遍顺传播过程。

(2) 误差的逆传播

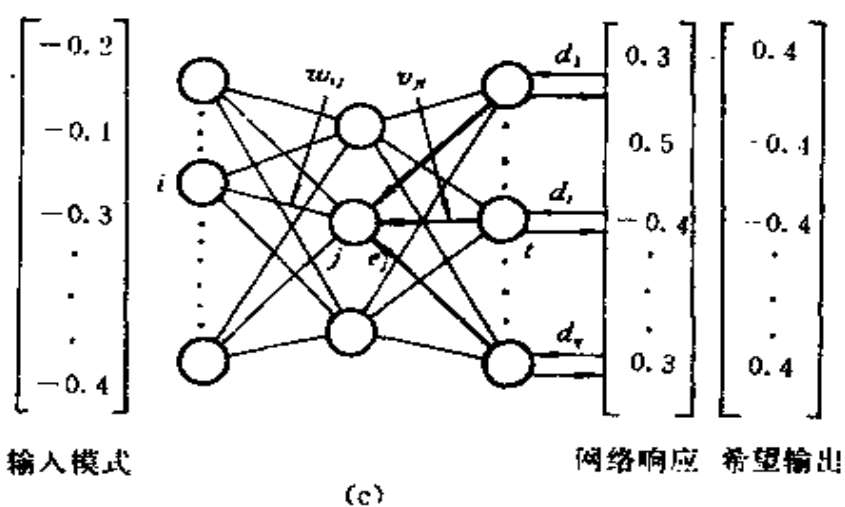
作为误差逆传播的第一步,是进行误差计算。如图 4.3 所示,误差逆传播过程是由输出层的误差 d_j 向中间层的误差 e_i 传递的过程:



(a) 模式顺传播后网络响应与希望输出相比较



(b) 计算输出层校正误差 d_i , 接箭头方向修正连接权 v_x



(c) 计算中间层校正误差 e_j , 粗线表示输出层误差向中间层的传递

图 4.3 “顺传播”与“逆传播”示意图

在这里输出层的校正误差按式 4.6 计算

$$d_i^k = (y_i^k - C_i^k) f'(L_i) \quad (4.6)$$

$$i = 1, 2, \dots, q \quad k = 1, 2, \dots, m$$

式 4.6 的物理意义是: $(y_i^k - C_i^k)$ 项表示网络希望输出与实际输出的绝对误差; $f'(L_i)$ 项根据各单元的实际响应调整偏差量。如 4.4 所示, 当输出层某单元的输入 L_i 在 0 附近时, 其输出变化幅度较大, 而此时 $f(L_i)$ 的导数 $f'(L_i)$ 正好处于峰值附近, 当与 $(y_i^k - C_i^k)$ 项相乘后, 增强了偏差的校正作用。反之, 当 L_i 的绝对值较大时, 其输出变化幅度很小, 也就是说此单元所受激励对此单元这一次学习过程所起的校正作用不大, 而此时 $f'(L_i)$ 正好处于较小值的部分, 当与 $(y_i^k - C_i^k)$ 项相乘后, 减弱了偏差的校正作用。

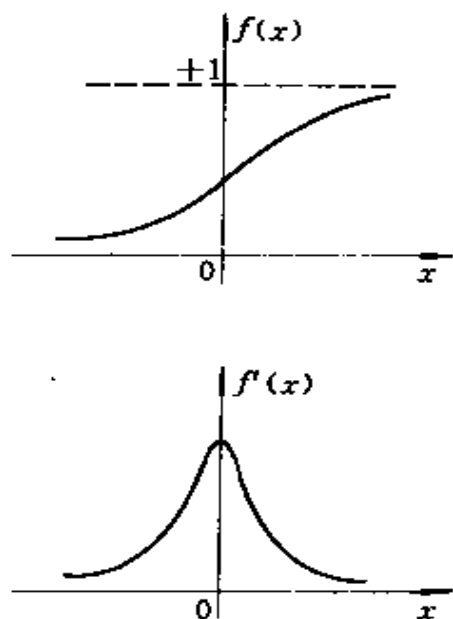


图 4.4 S 函数及其导数

为完成误差向中间层的传递, 需要按式 4.7 计算中间层各单元的校正误差:

$$e_j^k = \left[\sum_{i=1}^q v_{ji} \cdot d_i^k \right] f'(s_j) \quad (4.7)$$

$$j = 1, 2, \dots, p \quad k = 1, 2, \dots, m$$

式 4.7 的物理意义与式 4.6 相似, 只不过每一个中间单元的校正误差都是由 q 个输出层单元校正误差传递而产生的。

得到校正误差 d_i^k 与 e_j^k 之后, 沿逆方向调整输出层至中间层、中间层至输入层之间的连接权, 以及各单元的输出阈值。其调整量按式 4.8~4.11 计算

$$\Delta v_{ji} = \eta \cdot d_i^k \cdot b_j^k \quad (4.8)$$

$$\Delta \gamma_i = \eta \cdot d_i^k \quad (4.9)$$

$$\begin{aligned} j &= 1, 2, \dots, p & t &= 1, 2, \dots, q \\ k &= 1, 2, \dots, m & 0 < \alpha < 1 & (\text{学习系数}) \end{aligned}$$

$$\Delta w_{ij} = \beta \cdot e_j^k \cdot a_i^k \quad (4.10)$$

$$\Delta \theta_j = \beta \cdot e_j^k \quad (4.11)$$

$$0 < \beta < 1 (\text{学习系数}) \quad i = 1, 2, \dots, n$$

$$j = 1, 2, \dots, p \quad k = 1, 2, \dots, m$$

下面以式 4.8 为例,说明连接权的调整机理。由式 4.8 可知,连接权的调整量依赖于三个因素: α 、 d_i^k 和 b_j^k 。首先,调整量与误差 d_i^k 成比例,即误差越大,调整的幅度也就越大,这一物理意义是显见的。其次,调整量与连接权所对应的中间层单元的输出值 b_j 成比例。其物理意义是,某中间层单元的激活值越高,它在这次学习过程中就显得越活跃,则与其相关的连接权的调整幅度也就应该越大。调整量与学习率 α 成比例,通常 α 的值在 0.25 至 0.75 之间。如第三章所述, α 的大小影响学习的速度,为了使整个学习过程适当加快,又不至于引起振荡,可采用变学习率的办法,即在学习初期取较大 α 值,随着学习过程的进行逐渐减小 α 值。

(3) 训练过程

所谓训练过程,是指反复学习的过程,也就是根据教师示教的希望输出与网络实际输出的误差调整连接权的过程。而希望输出实际上是对输入模式分类的一种表示,是人为设定的,所以也因人而异。随着“模式顺传播”与“误差逆传播”过程的反复进行,网络的实际输出逐渐向各自所对应的希望输出逼近。图 4.5 所示的是一组由三个元素构成的三种学习模式对。

对于典型的 BP 网络,一组训练模式,一般要经过数百次乃至几千次的学习过程,才能使网络收敛。

(4) 收敛过程

前面已多次指出,学习或者说训练的收敛过程就是网络全局误差趋向于极小值的过程。一般全局误差有式 3.17 和式 3.19 两种定义。

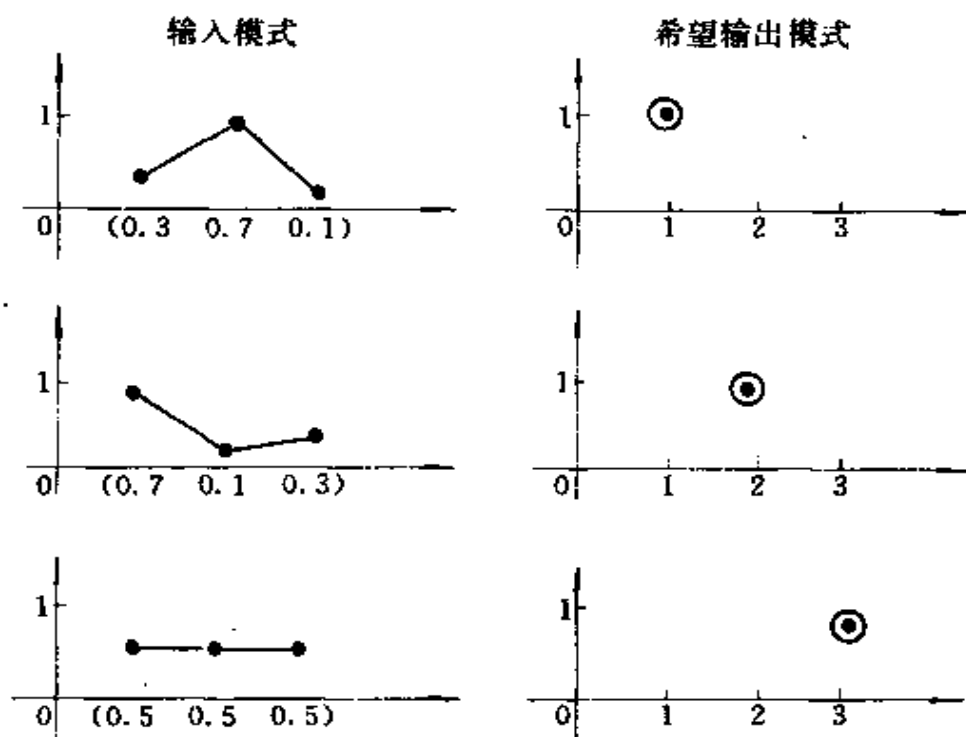


图 4.5 训练集合举例

值得注意的是,虽然网络的收敛标志着网络的响应非常逼近了希望输出模式。但是,这种逼近不能简单地用“是”或“否”来衡量。网络的响应往往是以实数的形式出现,而不是逻辑值。在这里“逼近”是一个模糊的概念,有点象模糊数学中的隶属度。例如,当某 BP 网络的希望输出模式为 $Y=(1,0,0)$ 时,学习收敛后的网络实际响应为 $C=(0.988,0.012,0.036)$,这时可以说网络的实际响应已十分逼近希望输出模式,但不能认为这就是最好的逼近。在网络进行回想时,网络的响应也往往是一个近似的模式分类。例如,当三种标准分类 A、B、C 分别用希望输出模式 $Y_A=(1,0,0)$, $Y_B=(0,1,0)$, $Y_C=(0,0,1)$ 表示时,当某一被识别的输入模式提供给网络回想后,其输出为 $(0.03, 2.76, 0.99)$ 。这时,我们应该认为,输入模式的第一可能类别是 C 类,第二可能类别是 B 类,而不可能是 A 类。

对于 BP 网络,其收敛过程存在着两个很大的缺陷:收敛速度

慢;存在所谓“局部最小值”问题。在学习过程中有时会发现,当学习反复进行到一定次数以后,虽然网络的实际输出与希望输出还存在很大的误差,但无论再如何学习下去,网络全局误差的减小速度都变得十分缓慢,或者根本不再变化。这种现象就是因网络收敛于局部极小点所致。从后面的数学推导我们将知道,BP 网络全局误差函数 E (也称代价函数或能量函数),是一个以 S 函数为自变量的非线性函数:这就意味着由 E 构成的连接权空间不是只有一个极小点的抛物面,而是存在多个局部极小点的超曲面。因此,BP 网络的收敛过程,很可能在遇到局部极小值便被“冻结”住,而无法最终收敛于全局最小点,也就无法对学习模式准确记忆。导致 BP 网络这一缺陷的原因,是由于 BP 学习规则同 Madaline 算法类似,采用了按误差函数梯度下降的方向进行收敛,如图 4.6 所示。



图 4.6 BP 网络的非线性空间

梯度下降学习法,有些象高山滑雪运动员,总是在寻找坡度最大的地段向下滑行,当他处于 A 点位置时,沿最大坡度路线下降,到达局部极小点,则停止滑行。如果他是从 B 点开始向下滑行,则最终他将到达全局最小点。由此可知,BP 网络的收敛依赖于学习模式的初始位置。适当改变 BP 网络中间层的单元个数、或给每个连接权加上一个很小的随机数,都有可能使收敛过程避开局部极小点。但是保证网络收敛于全局最小值的有效办法是随机学习算法,将在第六章中介绍。

4.3 误差逆传播学习规则的数学推导

如图 4.1 所示的三层 BP 网络, 设输入模式向量 $A_k = (a_1, a_2, \dots, a_n)$, 希望输出向量 $Y_k = (y_1, y_2, \dots, y_q)$; 中间层单元输入向量 $S_k = (s_1, s_2, \dots, s_p)$, 输出向量 $B_k = (b_1, b_2, \dots, b_p)$; 输出层单元输入向量 $L_k = (l_1, l_2, \dots, l_q)$, 输出向量 $C_k = (c_1, c_2, \dots, c_q)$; 输入层至中间层连接权 $\{w_{ij}\} \quad i=1, 2, \dots, n \quad j=1, 2, \dots, p$; 中间层至输出层连接权 $\{v_{jt}\} \quad j=1, 2, \dots, p \quad t=1, 2, \dots, q$; 中间层各单元输出阈值为 $\{\theta_j\} \quad j=1, 2, \dots, p$; 输出层各单元输出阈值为 $\{\gamma_t\} \quad t=1, 2, \dots, q$ 。以上 $k=1, 2, \dots, m$ 。

首先介绍网络响应函数——S 函数的一个重要特征: S 函数的导数可用 S 函数自身表示。如式 4.12、4.13 所示

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.12)$$

$$f'(x) = f(x)[1 - f(x)] \quad (4.13)$$

设第 k 个学习模式网络希望输出与实际输出的偏差为:

$$\delta_j^k = (y_j^k - C_j^k) \quad j = 1, 2, \dots, q$$

δ_j^k 的均方值为

$$\begin{aligned} E_k &= \sum_{j=1}^q (y_j^k - C_j^k)^2 / 2 \\ &= \sum_{j=1}^q (\delta_j^k)^2 / 2 \end{aligned} \quad (4.14)$$

为使 E_k 随连接权的修正按梯度下降, 则需求 E_k 对网络实际输出 $\{C_j\}$ 的偏导

$$\frac{\partial E_k}{\partial C_j} = - (y_j^k - C_j^k) = - \delta_j^k \quad (4.15)$$

由于

$$L_t = \sum_{j=1}^p v_{jt} \cdot b_j - \gamma_t \quad t = 1, 2, \dots, q \quad (4.16)$$

$$C_t = f(L_t) \quad t = 1, 2, \dots, q \quad (4.17)$$

连接权 v_{jt} 的微小变化, 对输出层响应的影响可由式 4.13、4.16 推得式 4.18

$$\frac{\partial C_t}{\partial v_{jt}} = \frac{\partial C_t}{\partial L_t} \cdot \frac{\partial L_t}{\partial v_{jt}} = f'(L_t) \cdot b_j = C_t(1 - C_t) \cdot b_j \quad (4.18)$$

$$t = 1, 2, \dots, q \quad j = 1, 2, \dots, p$$

则连接权 v_{jt} 的微小变化对第 k 个模式的均方差 E_k 的影响, 可由式 4.15、4.18 推得

$$\frac{\partial E_k}{\partial v_{jt}} = \frac{\partial E_k}{\partial C_t} \cdot \frac{\partial C_t}{\partial v_{jt}} = -\delta_t^k C_t(1 - C_t) \cdot b_j \quad (4.19)$$

$$t = 1, 2, \dots, q \quad j = 1, 2, \dots, p$$

按梯度下降原则, 使连接权 v_{jt} 的调整量 Δv_{jt} 与 $\frac{\partial E_k}{\partial v_{jt}}$ 的负值成比例变化, 则由式 4.19 可得

$$\Delta v_{jt} = -\alpha \left[\frac{\partial E_k}{\partial v_{jt}} \right] = \alpha \delta_t^k C_t(1 - C_t) \cdot b_j \quad (4.20)$$

$$0 < \alpha < 1 \quad t = 1, 2, \dots, q \quad j = 1, 2, \dots, p$$

设输出层各单元的一般化误差为 d_t^k $t=1, 2, \dots, q; k=1, 2, \dots, m$ 。 d_t^k 定义为 E_k 对输出层输入 L_t 的负偏导, 由式 4.15、4.13 可得

$$d_t^k = -\frac{\partial E_k}{\partial L_t} = -\frac{\partial E_k}{\partial C_t} \cdot \frac{\partial C_t}{\partial L_t} = \delta_t^k C_t(1 - C_t) \quad (4.21)$$

$$t = 1, 2, \dots, q \quad k = 1, 2, \dots, m$$

则连接权 v_{jt} 的调整量 Δv_{jt} 可表示为

$$\Delta v_{jt} = \alpha \cdot d_t^k \cdot b_j \quad (4.22)$$

$$t = 1, 2, \dots, q; j = 1, 2, \dots, p \quad k = 1, 2, \dots, m$$

同理, 由输入层至中间层连接权的调整, 仍然按梯度下降法的

原则进行。中间层各单元的输入 $\{s_j\}$ 为

$$s_j = \sum_{i=1}^n w_{ij} \cdot a_i + \theta_j \quad j = 1, 2, \dots, p \quad (4.23)$$

其输出 $\{b_j\}$ 为

$$b_j = f(s_j) \quad j = 1, 2, \dots, p \quad (4.24)$$

连接权 w_{ij} 的微小变化, 对第 k 个学习模式的均方误差的影响, 可由式 4.21、4.16、4.24、4.13、4.23 推得式 4.25

$$\begin{aligned} \frac{\partial E_k}{\partial w_{ij}} &= \left[\sum_{l=1}^q \frac{\partial E_k}{\partial L_l} \cdot \frac{\partial L_l}{\partial b_j} \right] \cdot \frac{\partial b_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}} \\ &= \left[\sum_{l=1}^q (-d_l^k) v_{jl} \right] \cdot f'(s_j) \cdot a_i \\ &= - \left[\sum_{l=1}^q d_l^k v_{jl} \right] b_j (1 - b_j) \cdot a_i \\ &\quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, p \end{aligned} \quad (4.25)$$

设中间层各单元的一般化误差为 $\{e_j^k\}$ $j=1, 2, \dots, p; k=1, 2, \dots, m$ 。 e_j^k 定义为 E_k 对中间层输入 s_j 的负偏导。由式 4.21、4.16、4.24、4.13 可得

$$\begin{aligned} e_j^k &= \frac{\partial E_k}{\partial s_j} = - \left[\sum_{l=1}^q \frac{\partial E_k}{\partial L_l} \cdot \frac{\partial L_l}{\partial b_j} \right] \cdot \frac{\partial b_j}{\partial s_j} \\ &= \left[\sum_{l=1}^q d_l^k \cdot v_{jl} \right] \cdot b_j (1 - b_j) \\ &\quad j = 1, 2, \dots, p, k = 1, 2, \dots, m \end{aligned} \quad (4.26)$$

则式 4.25 可表示为

$$\begin{aligned} \frac{\partial E_k}{\partial w_{ij}} &= - e_j^k \cdot a_i \\ &\quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, p \end{aligned} \quad (4.27)$$

与式 4.20 类似, 连接权 w_{ij} 的调整量应为

$$\Delta w_{ij} = - \beta \frac{\partial E_k}{\partial w_{ij}} = - \beta \cdot e_j^k \cdot a_i \quad (4.28)$$

$$i = 1, 2, \dots, n \quad j = 1, 2, \dots, p$$

同理阈值 $\{\gamma_i\}$ 的调整量为

$$\Delta \gamma_i = \alpha \cdot d_i^k \quad t = 1, 2, \dots, q \quad (4.29)$$

$$\Delta \theta_j = \beta \cdot e_j^k \quad j = 1, 2, \dots, p \quad (4.30)$$

以上的推导仅是针对一组学习模式进行的。设网络的全局误差为 E , 则

$$E = \sum_{k=1}^m E_k = \sum_{k=1}^m \sum_{i=1}^q (y_i^k - C_i)^2 / 2 \quad (4.31)$$

从以上的推导可以看出, 各个连接权的调整量是分别与各个学习模式对的误差函数 E_k 成比例变化的, 这种方法称为**标准误差逆传播算法**。而相对于全局误差函数 E 的连接权的调整, 应该在所有 m 个学习模式全部提供给网络之后统一进行。这种算法称为**累积误差逆传播算法**。当学习模式集合不太大时, 即学习模式对较少时, 累积逆传播算法比标准逆传播算法收敛速度要快一些。另外应特别注意, BP 学习规则实现的是一组学习模式集合上平方误差 E_k (或 E) 的梯度下降, 而不是特定某个模式的绝对误差 δ_j^k 的梯度下降。

下面给出整个学习过程的具体步骤和流程图。

① 初始化 给各连接权 $\{w_{ij}\}$ 、 $\{v_{jk}\}$ 及阈值 $\{\theta_j\}$ 、 $\{\gamma_i\}$ 赋予 $(-1, +1)$ 间的随机值。

② 随机选取一模式对 $A_k = (a_1^k, a_2^k, \dots, a_n^k)$, $Y_k = (y_1^k, y_2^k, \dots, y_r^k)$ 提供给网络。

③ 用输入模式 $A_k = (a_1^k, a_2^k, \dots, a_n^k)$ 、连接权 $\{w_{ij}\}$ 和阈值 $\{\theta_j\}$ 计算中间层各单元的输入 s_j ; 然后用 $\{s_j\}$ 通过 S 函数计算中间层各单元的输出生 $\{b_j\}$

$$s_j = \sum_{i=1}^n w_{ij} \cdot a_i - \theta_j \quad j = 1, 2, \dots, p \quad (4.32)$$

$$b_j = f(s_j) \quad j = 1, 2, \dots, p \quad (4.33)$$

④ 用中间层的输出 $\{b_j\}$ 、连接权 $\{v_{jk}\}$ 和阈值 $\{\gamma_i\}$ 计算输出层

各单元的输入 $\{L_t\}$, 然后用 $\{L_t\}$ 通过 S 函数计算输出层各单元的响应 $\{C_t\}$

$$L_t = \sum_{j=1}^p v_{jt} \cdot b_j - \gamma_t \quad t = 1, 2, \dots, q \quad (4.34)$$

$$C_t = f(L_t) \quad t = 1, 2, \dots, q \quad (4.35)$$

⑤ 用希望输出模式 $Y_t = (y_1^t, y_2^t, \dots, y_q^t)$ 、网络实际输出 $\{C_t\}$, 计算输出层的各单元的一般化误差 $\{d_t^t\}$

$$d_t^t = (y_t^t - C_t) \cdot C_t(1 - C_t) \quad t = 1, 2, \dots, q \quad (4.36)$$

⑥ 用连接权 $\{v_{jt}\}$ 、输出层的一般化误差 $\{d_t^t\}$ 、中间层的输出 $\{b_j\}$ 计算中间层各单元的一般化误差 $\{e_j^t\}$

$$e_j^t = \left[\sum_{t=1}^q d_t^t \cdot v_{jt} \right] b_j(1 - b_j) \quad j = 1, 2, \dots, p \quad (4.37)$$

⑦ 用输出层各单元的一般化误差 $\{d_t^t\}$ 、中间层各单元的输出 $\{b_j\}$ 修正连接权 $\{v_{jt}\}$ 和阈值 $\{\gamma_t\}$

$$v_{jt}(N+1) = v_{jt}(N) + \alpha \cdot d_t^t \cdot b_j \quad (4.38)$$

$$j = 1, 2, \dots, p \quad t = 1, 2, \dots, q \quad (0 < \alpha < 1)$$

$$\gamma_t(N+1) = \gamma_t(N) + \alpha \cdot d_t^t \quad (4.39)$$

$$t = 1, 2, \dots, q$$

⑧ 用中间层各单元的一般化误差 $\{e_j^t\}$ 、输入层各单元的输入 $A_t = (a_1^t, a_2^t, \dots, a_n^t)$ 修正连接权 $\{w_{ij}\}$ 和阈值 $\{\theta_j\}$

$$w_{ij}(N+1) = w_{ij}(N) + \beta \cdot e_j^t \cdot a_i^t \quad (4.40)$$

$$i = 1, 2, \dots, n \quad j = 1, 2, \dots, p$$

$$\theta_j(N+1) = \theta_j(N) + \beta \cdot e_j^t \quad (4.41)$$

$$j = 1, 2, \dots, p$$

⑨ 随机选取下一个学习模式对提供给网络, 返回到步骤③, 直至全部 m 个模式对训练完毕。

⑩ 重新从 m 个学习模式对中随机选取一个模式对, 返回步骤③, 直至网络全局误差函数 E 小于预先设定的一个极小值, 即网络收敛或学习回数大于预先设定的值, 即网络无法收敛。

⑪ 结束学习。

在以上的学习步骤中,③~⑥为输入学习模式的“顺传播过程”,⑦~⑧为网络误差的“逆传播过程”,步骤⑨、⑩则完成训练和收敛过程。

学习过程的流程图如图 4.7 所示。

通常,经过训练的网络还应该进行性能测试。测试的方法就是选取测试模式集合,将其提供给网络,检验网络对其分类的准确度。各个测试模式应包含今后网络应用中将要遇到的主要典型模式。这些模式数据可以是实例中直接测取的,也可以是通过仿真得到的,在模式数据较少或较难得到的情况下,也可以通过给学习模式加上适当噪声量或按一定规则插值得到。总之,一个好的测试集合应该不含有和学习模式集合完全相同的模式。

训练与测试后的网络按上述步骤③~④进行网络回想,即进行网络工作。

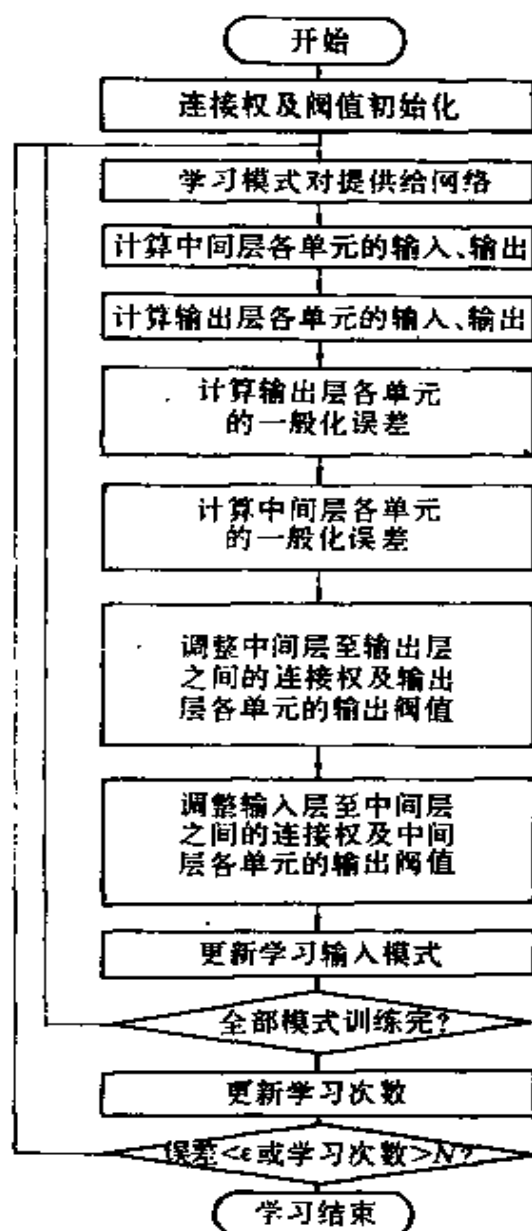


图 4.7 BP 学习规则流程图

4.4 隐含层——特征抽取器的作用

使 BP 网络具有识别非线性模式的主要原因,是由于在网络

的输入与输出层之间增加了一个或一个以上的中间层,即隐含层,作为输入模式的“内部表示”。所谓“内部表示”,实际上就是对一类输入模式中所含的区别于其它类别的输入模式的特征进行抽取,并将抽取出的特征传递给输出层。由输出层对输入模式的类别作最后的判别。因此,有时也将中间层称为**特征抽取层**。隐含层对输入模式进行特征抽取的过程,实际上也就是输入层与隐含层之间连接权进行“自组织化”的过程。为了清楚地说明这个问题,下面分析一个具体的例子。图 4.8 所示的是一个具有 7 个输入单元、2 个中间层单元和 3 个输出层单元的三层 BP 网络。

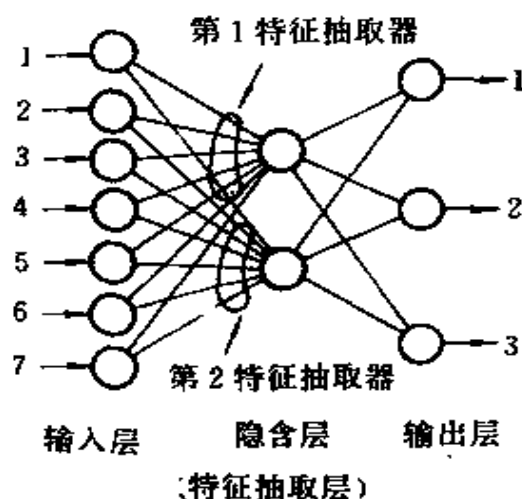


图 4.8 三层 BP 网络

现选取如图 4.9 所示的三个模式作为学习模式,对网络进行训练。图中右侧表示的是进行了 460 回学习训练,学习率为 0.6,网络全局误差 RMS 值收敛到 0.06 时的网络输出层三个单元的输出值。

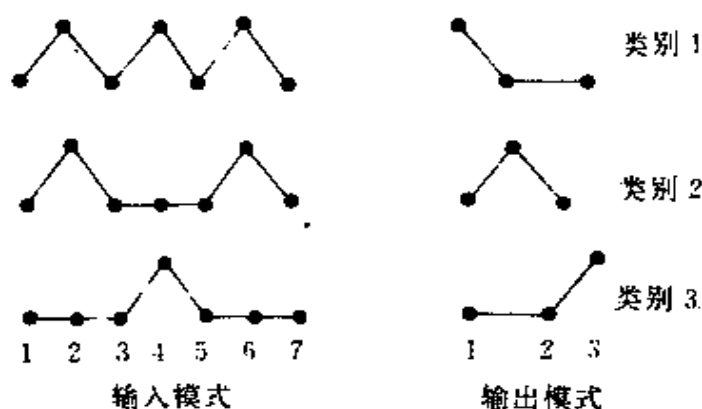


图 4.9 图 4.8 所示网络的输入、输出模式

由图可以看出,三个输入模式并不是完全相互独立的。其中,第2、第3个模式的线性组合构成了第1个模式。也就是说,三个输入模式中实际上只有两个独立特征。因此,隐含层只需设置2个单元,即2个特征抽取器。当两个特征抽取器同时被激活时,说明此时输入的是第一类别模式。图4.10所示,分别是与第1特征抽取器和第2特征抽取器相连接的各连接权在网络收敛后的最终值示意图。

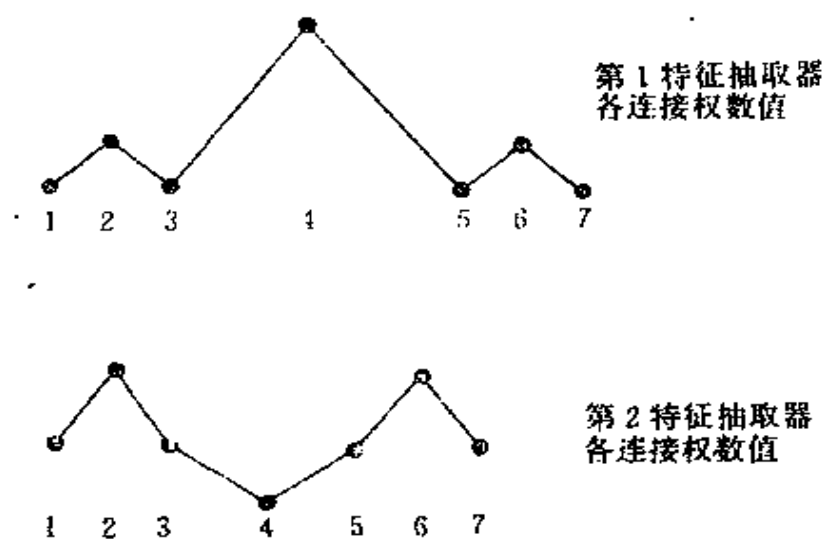


图4.10 特征抽取器连接权值示意图

如图所示,这两组连接权值示意图与第3、第2类输入模式的示意图在形状上大致相似,说明这些连接权值实际上就是对输入模式的一种“内部表示”。

隐含层两个特征抽取器的输出,通过它们与三个输出单元的连接权,将输入模式的特征传递给输出层,两层之间的连接权起着对传递特征选择、增强的作用,即第1特征抽取器的输出,经与第三个输出单元的连接权的作用而被增强。当输入为第3类模式时,激活第三个输出单元,另外两个单元则被相对抑制,表明此时的输入模式为第3类;第2特征抽取器的输出,经与第二个输出单元连

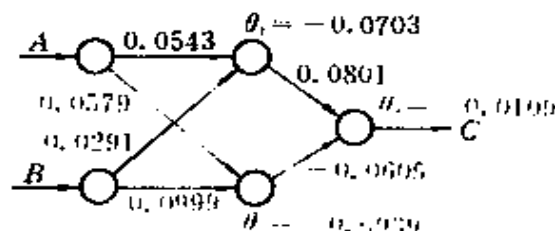
接权的作用而被增强。当输入为第2类模式时,激活第二个输出单元,相对抑制一、三单元的输出,表明此时的输入模式为第2类。而当输入为第7类模式时,两个特征抽取器同时“兴奋”,并激活第一个输出单元,相对抑制二、三单元的输出,表明此时的输入模式为第1类。

从以上的分析可以看出,在网络的训练过程中,各层之间的连接权起着“传递特征”的作用。各连接权从初始的随机值逐渐演变,最终达到能够表征输入模式特征的过程,就是所谓网络的“自组织化过程”。但是应该强调一点,网络的“自组织化”能力并不是万能的,它比起人类的思维要差得很多。这主要表现在它的特征表示清晰度远低于人类。例如,如果将上例中某一个模式换为7个输入单元具有相同值的“平坦型”输入模式,则无论哪一个特征抽取器都不呈现与之大致相似的模式特征。

4.5 BP 网络应用举例

本节仅举一个对“异或”问题分类的简单例子,较复杂的实际应用例子将在第十章中详细介绍。

图 4.11 为“异或”(XOR)问题分类的网络及学习模式示意图。该网络由输入、中间、输出三层构成。所对应处理单元分别为2—2—1。四个模式对轮流提供给网络,按“模式顺传播”、“误差逆传播”进行反复的网络训练,直至网络输出的全局误差满足要求为止。这里网络学习率取 $\alpha = \beta = 0.6$ 。



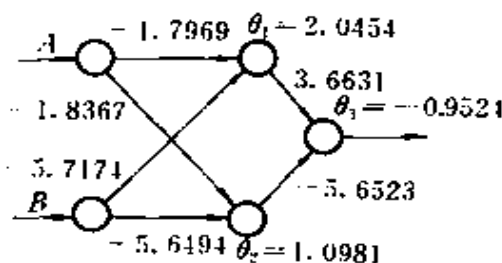
(a) 学习初始状态

输入		希望 输出	实际 输出	全局 误差
A	B			
0	0	0	0.5	0.5
0	1	1	0.5	
1	0	1	0.5	
1	1	0	0.5	

(b)学习模式

图 4.11 XOR 问题 BP 网络

首先给网络的各连接权及阈值赋予 $[-0.1, 0.1]$ 之间的随机值,此时网络对应四个输入模式所产生的实际输出均为 0.5 左右,全局误差约为 0.5。当进行 8000 次学习之后,全局误差为 0.1659。若以输出小于 0.5 判为“0”,输出大于 0.5 判为“1”,则可以认为此时网络已完成了模式记忆,具有正确分类的能力,如图 4.12 所示。



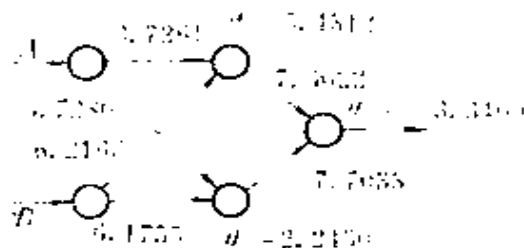
(a)

输入		希望 输出	实际 输出	全局 误差
A	B			
0	0	0	0.119	0.166
0	1	1	0.727	
1	0	1	0.734	
1	1	0	0.415	

(b)

图 4.12 图 4.11 所示网络学习 8000 次之后的状态

当网络进一步学习达 11050 次后,全局误差减少为 0.0078。若以输出小于 0.1 判为“0”,大于“0.9 判为“1”,则可认为该网络已完全学会了解决“异或”问题,如图 4.13 所示。



(a)

输入		希望 输出	实际 输出	全局 误差
A	B			
0	0	0	0.05	0.008
0	1	2	0.941	
1	0	1	0.941	
1	1	0	0.078	

(b)

图 4.13 图 4.11 所示网络学习 11050 次后的状态

上例中,从所设置的初始状态出发,网络最终收敛于全局最小点。但正如前几节所分析的那样,BP 算法并不能保证网络一定能收敛到全局最小点,即不能保证学习一定能够成功。如图 4.14(a)所示的 XOR 问题的 BP 网络在经过若干次学习之后,就陷入了图 4.14(b)所示的局部极小的状态中。

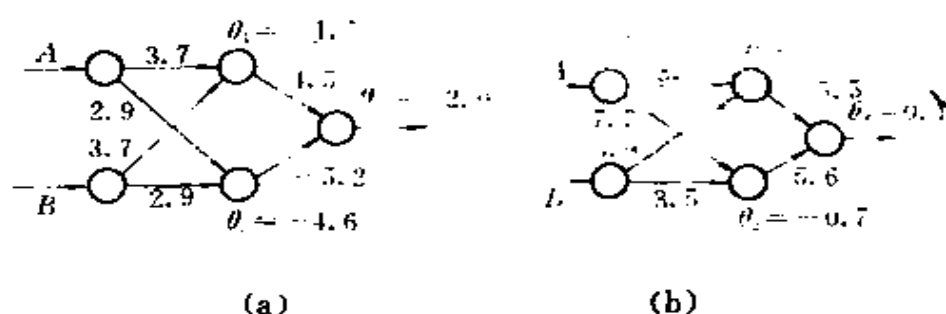


图 4.14 陷入局部极小点的 BP 网络

4.6 BP 网络小结

比起初期的神经网络,误差逆传播神经网络无论在网络理论还是网络性能方面都更加成熟。其突出的优点就是具有很强的非线性映射能力和柔性的网络结构。网络的中间层数、各层的处理单元数及网络学习系数可根据具体情况任意设定,并且随着结构的差异其性能也有所不同。

但是,误差逆传播神经网络并不是一个十分完善的网络,它存在以下一些主要缺陷。

① 学习收敛速度太慢,即使一个比较简单的问题,也需要几百次甚至上千次的学习才能收敛。

② 不能保证收敛到全局最小点。

③ 网络隐含层的层数及隐含层的单元数的选取尚无理论上的指导,而是根据经验确定。因此,网络往往有很大的冗余性,无形中也增加了网络学习的时间。

④ 网络的学习、记忆具有不稳定性。一个训练结束的 BP 网络,当给它提供新的记忆模式时,将使已有的连接权打乱,导致已

记忆的学习模式的信息消失。要避免这种现象,必须将原来的学习模式连同新加入的新学习模式一起重新进行训练。而对于人类的大脑来说,新信息的记忆不会影响已记忆的信息,这就是人类大脑记忆的稳定性。

对于 BP 网络存在的以上几种缺陷,在后面几章介绍的其它形式神经网络和学习算法中,都得到了不同程度的改善或补充。然而就 BP 网络本身来说,人们也对其性能的改善作了大量的工作,提出了许多改进方案。其中研究最多的是如何加速 BP 网络的收敛速度和尽量避免陷入局部最小点的问题。

4.7 几种改进方案

本节主要介绍几种旨在加速 BP 网络收敛的改进算法方案。

(1) 累积误差校正算法

4.3 节中曾经提到,一般的 BP 算法称为标准误差逆传播算法,这种算法偏离了真正全局误差意义上的梯度下降。真正的全局误差意义上的梯度算法称为累积误差校正算法。其具体算法是,按式 4.36、4.37 分别计算出 m 个学习模式的一般化误差,并将这 m 个误差进行累加,用累加后的误差校正输出层与中间层、中间层与输入层之间的连接权以及各个输出阈值。其算法流程图在图 4.7 的基础上的改进部分如图 4.15 所示。

这种算法与标准误差逆传播算法相比,每个连接权及阈值的校正次数明显减少(每一次学习减少 $m-1$ 次校正),因此,学习时间也随之缩短。但是,这种算法将各个学习模式的误差平均化,在某些情况容易引起网络的振荡。

(2) S 函数输出限幅算法

由 4.3 节可知,连接权校正量 $\{v_{jk}\}$ 与 $\{w_{ij}\}$ 的校正量,如式

4.38、4.40 所示,它们都与中间层的输出 $\{b_j\}$ 有关。

$$\Delta v_{jk} = \alpha \cdot d_i^k \cdot b_j \quad (4.42)$$

$$\Delta w_{ij} = \beta \cdot e_j^k \cdot a_i \quad (4.43)$$

$$e_j^k = \left[\sum_{i=1}^q d_i^k \cdot v_{jk} \right] \cdot b_j (1 - b_j) \quad (4.44)$$

因此,当中间层的输出为“0”或“1”时,连接权校正量 Δv_{jk} 或 Δw_{ij} 为“0”,不起校正作用。中间层的输出是由 S 函数的输出所决定的。由 S 函数的饱和非线性输出特性可知,当其输入小于或大于某数值后,其输出接近于“0”或“1”。因而在相当次数的学习过程中,真正的校正量很小,校正进程十分缓慢。为此,限制 S 函数的输出,是加快网络校正的一个有效方法。具体做法是,在网络的学习计算过程中,当 S 函数的实际输出小于 0.01 或大于 0.99 时,将其输出值直接取为 0.01 或 0.99。这样做保证了每次学习都能进行有效的校正,从而加快收敛过程。

(3) 惯性校正法

所谓惯性校正法,就是在每一次对连接权或输出阈值进行校正时,按一定比例加上前一次学习时的校正量,即惯性项,从此加速网络学习的收敛。具体作法如式 4.45 所示

$$\Delta W(N) = d + \eta \Delta W(N-1) \quad (4.45)$$

式中, $\Delta W(N)$ 为本次应得校正量, $\Delta W(N-1)$ 为前次校正量,

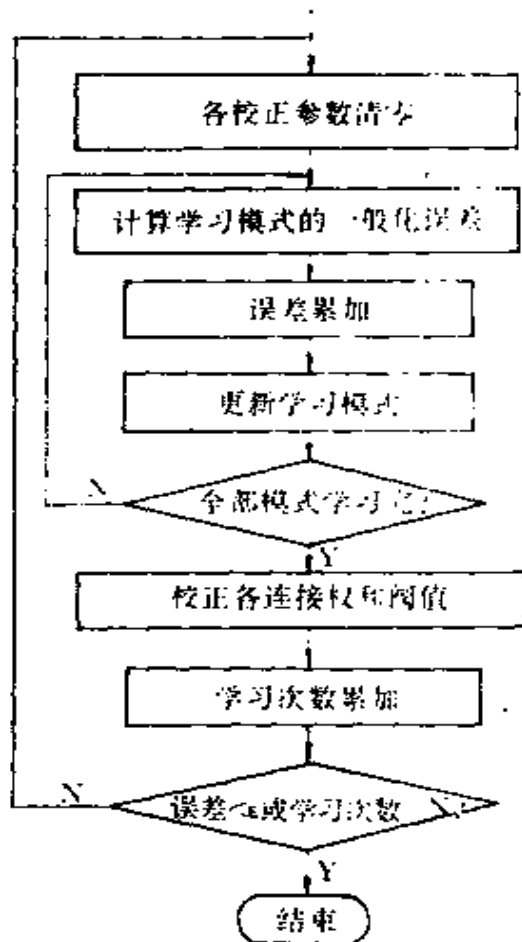


图 4.15 累积误差算法流程图

d 为由本次误差计算得到的校正量, η 为惯性系数 ($0 < \eta < 1$)。由式 4.45 可知, 当前一次的校正量过调时, 惯性项与本次误差校正项 d 符号相反, 使得本次实际校正量 $\Delta W(N)$ 减小, 起到减小振荡的作用; 而当前次校正量欠调时, 惯性项与本次误差计算校正项符号相同, 本次实际校正量增加, 起到加速校正的作用。

(4) 改进的惯性校正法

以上介绍的惯性校正法中, 惯性项在每次学习校正时所起作用的比重是相等的, 即惯性系数在整个校正过程中为常数。在改进的惯性校正法中, 惯性系数是一个变量, 随着校正的不断进行, 惯性系数逐渐增大, 惯性项在本次校正量中所占的比重逐渐增大, 如式 4.46、4.47 所示

$$\Delta W(N) = d + \eta(N) \cdot \Delta W(N-1) \quad (4.46)$$

$$\eta(N) = \eta(N-1) + \Delta\eta \quad (4.47)$$

这一改进的目的是使被校正量随着学习进程的发展, 逐渐沿前一次校正方向变化, 以此达到加速收敛的目的。但是应该注意, 如果惯性系数太大, 惯性项所占的比例过重, 则误差修正项的作用削减太多以至完全不起作用, 反而会延长收敛时间, 甚至引起振荡。所以, 一般应设置惯性系数的上限值, 以确保误差修正项的校正作用。通常上限值取 0.90 左右。另外也有采用惯性系数按指数形式增长的办法进行惯性校正的方案。

除以上介绍的几种加速 BP 网络收敛的改进方案之外, 还有许多其它改进方案。如按最小均方误差选择学习系数, 以变系数代替常系数加速收敛的方案; 用双曲正切函数代替 S 函数改善学习速度的方案; 用限制连接权的取值范围避免学习过程中的振荡, 提高收敛速度的方案等等。

第五章 Hopfield 神经网络

5.1 Hopfield 网络的基本思想

Hopfield 网络作为一种全连接型神经网络,曾经在人工神经网络研究发展历程中起过唤起希望、开辟研究新途径的作用。它与阶层型神经网络不同的结构特征和学习方法,模拟生物神经网络的记忆机理,获得了令人满意的结果。这一网络及学习算法最初是由美国物理学家 J. J Hopfield 于 1982 年首先提出的^[15],故称为 Hopfield 网络。1985 年 Hopfield 和 D. W. Tank 用这种网络模型成功地求解了优化组合问题中的具有典型意义的旅行商(TSP)问题,在所有随机选择的路径中找到了其中十万分之一的最优路径,这在当时是神经网络研究工作中所取得的突破性进展。

Hopfield 是从物理学磁场理论中受到启发,结合生物神经网络的思维机理而提出这一网络模型的。我们知道,磁场也是一种具有记忆功能的物质,人们很早就利用磁场的记忆功能创造出许多很有价值的产品,如目前广泛使用的计算机磁盘。由物理学知识可知,在磁性材料中游动着大量的磁旋,正是由于这些带有方向的磁旋的相互作用,才产生了磁场本身所具有的各种性质。在永久磁铁中,由于所有的磁旋都朝向一个方向,构成了磁铁的 N 极和 S 极的两极特性。Hopfield 网络的基本思想,就是用人工神经元模拟的磁旋,用神经元之间的连接权模拟磁场中磁旋的相互作用;用各神经元的“激活”和“抑制”两种状态,模拟磁场中磁旋的上、下

两个方向,构成一个具有记忆功能的神经网络系统。并且引用物理学中有关能量的概念,用“计算能量函数”(Computational Energy Function)来评价和指导整个网络的记忆功能。磁场与神经网络的对照示意图如图 5.1 所示。

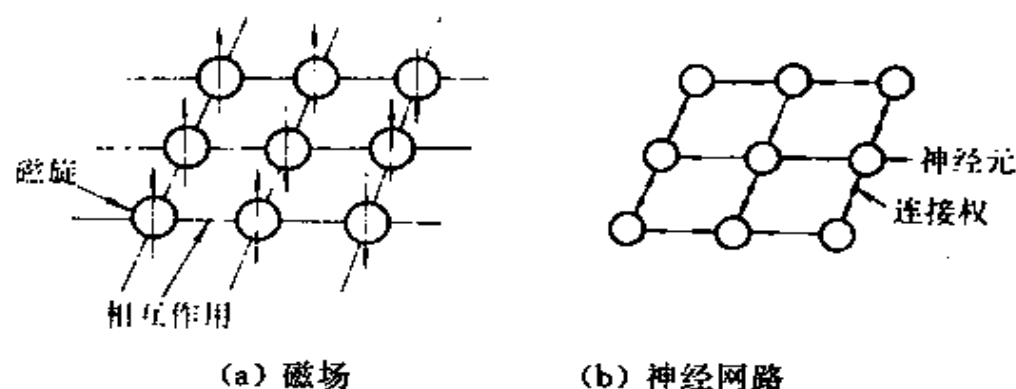


图 5.1 磁场与神经网络对照

5.2 Hopfield 网络的结构与算法

离散型 Hopfield 网络结构如图 5.2 所示。这是一个只有四个神经元的离散型 Hopfield 网络。其中每个神经元只能取“1”或“0”两个状态。设网络有 n 个神经元,则各个神经元的状态可用向量 U 表示:

$$U = (u_1, u_2, \dots, u_n) \quad (5.1)$$

其中, $u_i = 1$ 或 0 ($i = 1, 2, \dots, n$)

Hopfield 网络的各个神经元都是相互连接的,即每一个神经元都将自己的输出通过连接权传送给所有其它神经元,同时每个神经元又都接收所有其它

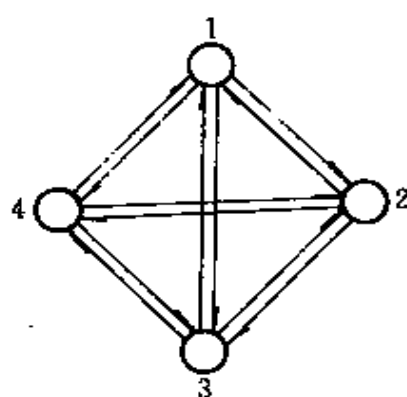


图 5.2 离散型 Hopfield 网络结构

神经元传递过来的信息。特别值得注意的是,由于 Hopfield 网络的这种结构特征,对于每一个神经元来说,自己输出的信号经过其它神经元又反馈回自己,所以也可以认为 Hopfield 网络是一种反馈型神经网络。图 5.3 所示的是其它几种形式的 Hopfield 网络结构。

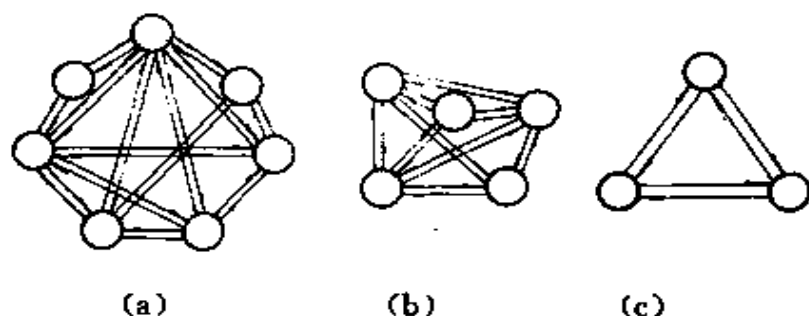


图 5.3 离散型 Hopfield 网络

对于 Hopfield 网络已有定理证明,当网络满足以下两个条件时, Hopfield 学习算法总是收敛的。

① 网络的连接权矩阵无自连接且具有对称性,即

$$w_{ii} = 0 \quad i = 1, 2, \dots, n \quad (5.2)$$

$$w_{ij} = w_{ji} \quad i, j = 1, 2, \dots, n \quad (5.3)$$

这一假设条件虽然不符合生物神经网络的实际情况(生物神经元之间连接强度通常是不对称的),但是却与磁场中各磁旋的相互作用情况相一致。

② 网络中各神经元以非同步或串行方式,依据运行规则改变其状态,即各神经元按随机选取方式,依据运行规则改变状态;且当某个神经元改变状态时,其它所有神经元保持原状态不变。这一点符合生物神经网络的情况。

5.3 Hopfield 网络运行规则

神经网络主要有两种运行方式,一种是前面介绍过的学习运行方式,即通过对训练模式的学习,调整连接权达到模式记忆的目的;另一种就是下面将要介绍的工作运行方式。在这种运行方式中,各连接权值是固定的,只是通过按一定规则的计算,更新网络的状态,以求达到网络的稳定状态。

图 5.4 是 Hopfield 网络中某个神经元的结构图。设网络由 n 个这样的神经元构成。时刻 t 第 i 个神经元的输出为

$$u_i(t) = \text{Sgn}(H_i) \quad (5.4)$$

$$H_i = \sum_{j=1}^n w_{ij} u_j - \theta_i \quad (5.5)$$

式中, $\text{Sgn}(x)$ 为符号函数:

$$\text{Sng}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (5.6)$$

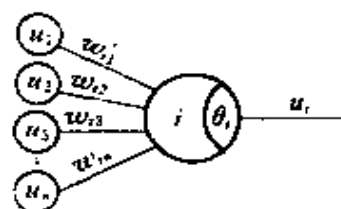


图 5.4 Hopfield 网络
神经元结构
图

式 5.4 表明,当所有其它神经元输出的加权总和超过第 i 个神经元的输出阈值时,此神经元被“激活”,否则将受到“抑制”。这里特别应该注意的是,按式 5.4 改变状态的神经元 u_i ,并不是按顺序进行的,而是按随机的方式选取的。下面将 Hopfield 工作运行规则总结如下:

- ① 从网络中随机选取一个神经元 u_i 。
- ② 按式 5.5 求所选中的神经元 u_i 的所有输入的加权总和:

$$H_i = \sum_{j=1}^n w_{ij} u_j - \theta_i$$

- ③ 按式 5.4 计算 u_i 的第 $t+1$ 时刻的输出值,即:

IF $[H_i(t) \geq 0]$ THEN

$$u_i(t+1) = 1$$

ELSE

$$u_i(t+1) = 0$$

④ u_i 以外的所有神经元输出保持不变

$$u_j(t+1) = u_j(t)$$

$$j = 1, 2, \dots, n$$

$$j \neq i$$

⑤ 返回到第一步,直至网络进入稳定状态。

前节曾经指出, Hopfield 网络是一种具有反馈性质的网络,而反馈网络的一个重要特点就是它具有稳定状态,也称为吸引子。那么 Hopfield 网络的稳定状态是怎样的呢? 当网络结构满足前节所指出的两个条件时,按上述工作运行规则反复更新状态,当更新进行到一定程度之后,我们会发现无论再怎样更新下去,网络各神经元的输出状态不再改变,这就是 Hopfield 网络的稳定状态,用数学表示为

$$u_i(t+1) = u_i(t) = \text{Sgn}(H_i) \quad (5.7)$$

$$i = 1, 2, \dots, n$$

一般情况下,一个 Hopfield 网络必须经过多次反复更新才能达到稳定状态。

5.4 网络计算能量函数与网络收敛

从 Hopfield 网络工作运行规则可以看出,网络中某个神经元 t 时刻的输出状态,通过其它神经元间接地与自己的 $t-1$ 时刻的输出状态发生联系。这一特性从数学的观点看,网络的状态变化可用差分方程表征;从系统动力学的观点看,此时的网络已不象误差

逆传播那样只是非线性映射的网络,而是一个反馈动力学系统。准确地说,是一个多输入、多输出、带阈值的二态非线性动力学系统。我们知道,一个抽象的动力学系统,与一个具有实际物理意义的动力学系统比较,抽象系统的动态过程必定是使某个与实际系统形式上一致的“能量函数”减小的过程。Hopfield 网络也同样如此。在满足一定的参数条件下,某种“能量函数”的能量在网络运行过程中不断地降低,最后趋于稳定的平衡状态。

设 t 时刻网络的状态用 n 个神经元的输出向量 $U(t)$ 表示

$$U(t) = [u_1(t), u_2(t), u_3(t), \dots, u_n(t)] \quad (5.8)$$

而每个神经元只有“1”或“0”两种状态,所以 n 个神经元共有 2^n 个组合状态,即网络具有 2^n 种状态,从几何学的角度看,这 2^n 种状态正好对应一个 n 维超立方体的各个顶点。以 $n=3$ 为例,一个立方体的八个顶点正好对应网络的八种状态。如图 5.5 所示。网络的能量函数可定义为网络状态的二次函数

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} u_i u_j + \sum_{i=1}^n \theta_i \cdot u_i \quad (5.9)$$

注意,式 5.9 的能量函数已不是物理学意义上的能量函数,而是在表达形式上与物理意义上的能量概念一致;表征网络状态的变化趋势,并可依据 Hopfield 工作运行规则不断进行状态变化,最终能够达到某个极小值的目标函数。所谓网络的收敛,就是指能量函数达到极小值。

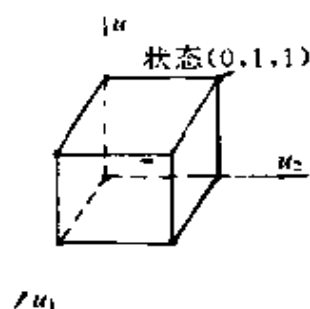


图 5.5 网络状态($n=3$)

下面证明,按照 Hopfield 工作运行规则,改变网络状态,能量函数式 5.9 将单调减小。

由式 5.9 可知,对应第 i 个神经元的能量函数为

$$E_i = -\frac{1}{2} \sum_{j=1}^n w_{ij} u_i u_j + \theta_i \cdot u_i \quad (5.10)$$

则由时刻 t 至时刻 $t+1$ 的能量 E_i 的变化量 ΔE_i 为

$$\begin{aligned}
\Delta E_i &= E_i(t+1) - E(t) \\
&= -\frac{1}{2} [u_i(t+1) - u_i(t)] \left[\sum_{j=1}^n w_{ij} u_j(t) + \theta_i \right] \\
&= -\frac{1}{2} [u_i(t+1) - u_i(t)] H_i(t)
\end{aligned} \tag{5.11}$$

由 Hopfield 网络工作运行规则可知, 当 $H_i(t) \geq 0$ 时, 方括号中的值大于或等于零, 故 $\Delta E_i \leq 0$; 当 $H_i(t) < 0$ 时, 方括号中的值小于或等于零, 故 $\Delta E_i \leq 0$ 。总之, $\Delta E_i \leq 0$ 。因为所有神经元都是按同一个工作运行规则进行状态更新的, 所以有 $\Delta E \leq 0$, 即

$$E(t+1) \leq E(t) \tag{5.12}$$

式 5.12 说明, 随着网络状态的更新, 网络能量函数是单调递减的。

图 5.6 是 Hopfield 网络能量函数的示意图, 为简单起见, 假设网络的状态是一维的。横轴为网络状态, 纵轴为网络能量函数, 当网络的状态随时间发生变化时, 网络能量沿其减小的方向变化, 最后落入能量的极小点。一旦能量落入某个极小点之后, 按 Hopfield 工作运行规则, 网络能量函数将会“冻结”在那里。也就是说网络不见得一定收敛到全局的最小点, 这是 Hopfield 网络的一个很大缺陷。但尽管如此, 由以上分析可知, Hopfield 网络已具有了寻找能量函数极小点的功能, 这就为网络的模式记忆打下了基础。

为更深理解 Hopfield 网络的收敛过程, 下面举一个具有四个神经元的 Hopfield 网络的实例, 如图 5.7 所示。因每个神经元只有“0”、“1”两种状态, 故四个神经元共有 $2^4 = 16$ 种状态组合, 也就是说, 函数共有 16 种状态, 分别用 16 进制数 1~F 表示。

分别让网络从 0, 1, 2, ..., F 状态开始变化, 每次共进行 30 回学习, 观察网络状态变化的次序及网络的收敛情况。

如图 5.8 所示, 左边 [:] 内分别表示网络的初始状态和对应的网络能量, 中间部分表示网络状态在 30 次学习过程中的变化次序, 最右侧“*”号表示网络收敛到全局最小点, “·”号表示网络落



图 5.6 网络能量函数示意图 图 5.7 四神经元的 Hopfield 网络
入局部极小点。网络的连接权初始值为 $[-1, +1]$ 内的随机值。

初始 状态	对应 能量	状态随时间变化次序	收敛情况
[0 : 0]		0 04444444444444444444444444444444	*
[1 : 17]		5 54444444444444444444444444444444	*
[2 : 122]		0 00004444444444444444444444444444	*
[3 : 17]		33333333333333333333333333333333	*
[4 : -115]		4 44444444444444444444444444444444	*
[5 : 10]		5 44444444444444444444444444444444	*
[6 : 77]		6 66444444444444444444444444444444	*
[7 : 80]		55544444444444444444444444444444	*
[8 : 122]		80004444444444444444444444444444	*
[9 : 187]		99dd5555544444444444444444444444	*
[A : 351]		B333E33333333333333333333333333	*
[B : 294]		FFFFDD54444444444444444444444444	*
[C : -66]		CC444444444444444444444444444444	*
[D : 107]		CC444444444444444444444444444444	*
[E : 233]		ECCCCC44444444444444444444444444	*
[F : 284]		FFDD5444444444444444444444444444	*

图 5.8 四神经元 Hopfield 网络状态变化图

由图可知,除去从 3 和 A 两个初始状态开始变化外,其它 14 个初始状态最后都能收敛至全局最小点状态 4(能量为-115)。例如初始状态 9,其能量为 187,随着学习的进行,经 $9 \rightarrow d \rightarrow 5 \rightarrow 4$ 最后收敛于状态 4;而当从状态 A 开始变化时,由 $A \rightarrow 3 \rightarrow 3 \cdots \rightarrow 3$,最

后收敛于局部极小点状态了,其能量为 17。这说明网络的收敛情况依赖于网络的初始状态。另外需要注意的是,这里网络能量的具体数值并不具有一定的物理意义,它只表明网络某一状态在整个网络的所有状态中所处的地位。当网络连接权的初始值改变时,各个状态所对应的能量具体数值也将随之改变。

下面我们再介绍一个值得注意的有趣现象,在上例中,只改变连接权初始随机值的取值方法(程序语言中,设置了产生不同种类随机数的方法),其它与上例完全一样,则其状态的变化规律如图 5.9 所示。由图中可知,此时无论网络从哪一个状态出发,最后都将收敛于网络的全局最小点状态 e,其能量为 -248。这说明整个网络只有一个极小点,即全局最小点。由此可知,对于同样结构的网络,当网络参数有所变化时,如连接权初始值随机数的取值形式或随机数的振幅有所不同时,网络的能量函数也将发生变化,其曲线或曲面(超曲面)对应的极小点的个数和大小也将变化。

初始 状态	对应 能量		
[0 : 0]	8cccccccccccccccccccccccccccccccc	*	
[2 : -72]	→ 9ddddddcccccccccccccccccccccccc	*	
[2 : 93]	6445ddddcccccccccccccccccccccccc	*	
[3 : 114]	155ddcccccccccccccccccccccccccccc	*	
[4 : -115]	4cccccccccccccccccccccccccccccccc	*	
[5 : -137]	5555555555ddcccccccccccccccccccc	*	
[6 : -39]	45cccccccccccccccccccccccccccccccc	*	
[7 : 32]	7Fcccccccccccccccccccccccccccccccc	*	
[8 : -89]	accccccccccccccccccccccccccccccccc	*	
[A : -87]	accccccccccccccccccccccccccccccccc	*	
[B : -21]	999999ddcccccccccccccccccccccccc	*	
[C : -233]	cccccccccccccccccccccccccccccccc	*	
[D : -210]	ddcccccccccccccccccccccccccccccccc	*	
[E : -248]	cccccccccccccccccccccccccccccccc	*	
[F : -132]	ffcccccccccccccccccccccccccccccccc	*	

图 5.9 四神经元 Hopfield 网络状态变化图

5.5 联想记忆

记忆是生物神经网络一个最基本也是最重要的功能。对于人工神经网络,记忆功能的强弱同样是衡量其综合性能的一个重要指标。而联想记忆又是人工神经网络模拟生物神经网络记忆特征的一个主要方法。在实际生活中,联想记忆的例子数不胜数。如分析下面甲、乙两人的对话。

甲:你还记得星期六舞会上遇见的那个小伙子吗?

乙:哪个?

甲:就是那个高个子的。

乙:噢,你说的是那个带眼镜的吗?

甲:不是。是穿蓝、白格子衬衫的那个。

乙:知道了,你说的那个人叫李华,他舞跳得不错。

在这段简短的对话中,甲以一些表面特征的提示使乙回想出甲所指的人叫李华。这些提示包括:① 时间:星期六;② 地点:舞会场;③ 人物特征:小伙子,高个子,穿蓝、白格衬衫。乙的回想包括:① 姓名:李华;② 人物行为:舞跳得好。当然乙很可能还会联想起有关李华的一些其他信息,如性格、爱好等等。这段对话体现了联想记忆的基本特征;这就是由某个模式的部分信息联想起这个模式的全部信息。联想记忆又分为自联想记忆和互联想记忆。自联想记忆就是通过对某个事物的部分信息或含有噪声的信息,再现出事物的本来信息;而互联想则是由某一事物的信息联想出另一事物的信息。事实上,可以将互联想记忆看作是自联想记忆的一个特例。当把与互联想有关的两个事物的信息当作一个统一的事

物的信息时,互联想过程就自然过渡到自联想过程。下面再举一个例子。对于中学时代某个非常熟悉的老同学,尽管相隔多年未见,他的外貌发生了很大的变化(相对于原记忆已混杂了大量噪声),但当再次相遇时,仍能很快辨认出他来。这实际上就是大脑自联想记忆功能的作用。图 5.10 为自联想和互联想的示意图。

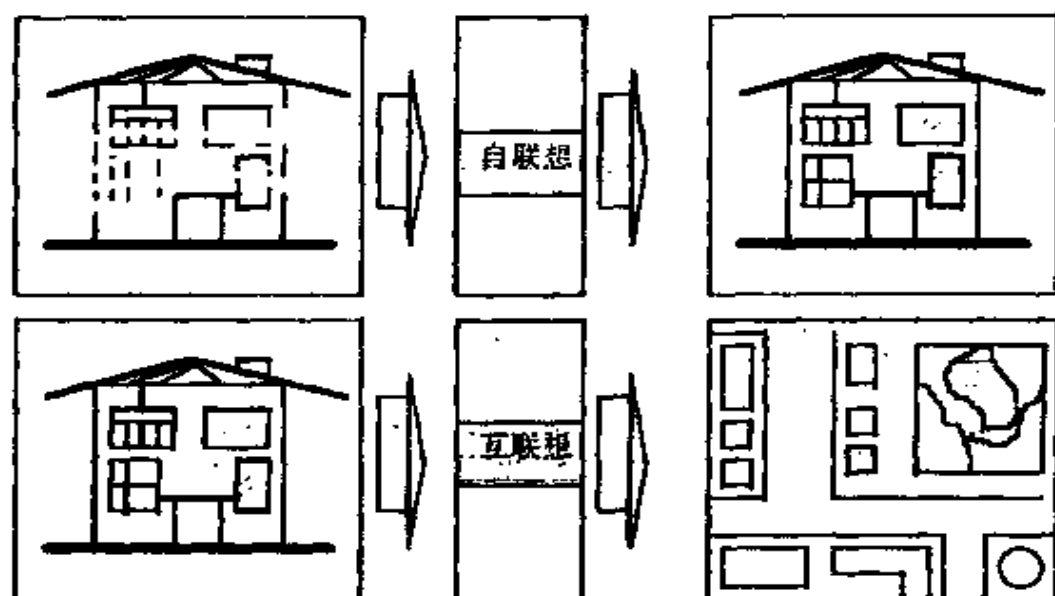


图 5.10 自联想和互联想示意图

Hopfield 网络是如何实现联想记忆的呢? 前面曾经指出, Hopfield 网络是一个非线性动力学系统,网络的“能量函数”存在着一个或多个极小点或称为平衡点、平衡状态。当某一时刻各神经元的状态确定之后,即网络的初始状态确定之后,网络的状态将按动力学方程——即 Hopfield 工作运行规则向能量递减的方向变化,最后接近或达到网络的平衡状态。这种实际上可观测的平衡状态又称为吸引子(Attractor)。实际上,吸引子可以是稳定的,也可以是不稳定的;可以是平衡点,也可以是极限环或混沌吸引子(奇异吸引子)。不随时间变化的吸引子称为稳定平衡点(Stable Equilibrium Point)。Hopfield 网络的这一特性提示我们,如果设法把所需记忆的模式设计成某个确定网络状态的一个稳定平衡点

(极小值),则当网络从与记忆模式较靠近的某个初始状态(即发生某些变形或含有某些噪声的记忆模式)出发后,网络按 Hopfield 工作运行规则进行状态更新,最后网络的状态将稳定在能量函数的极小点,即记忆模式所对应的状态。这样就完成了由部分信息(含有噪声的记忆模式)到全部信息(记忆模式)的联想过程。图 5.11 所表示的是一个稳定吸引子的示意图。网络中存放的一个记忆相当于一稳定吸引子,如图中黑点所示位置。黑点以外为不稳定状态,不稳定状态可看成是某记忆事物的部分已知信息。网络的能量函数总是由不稳定状态朝某局部稳定吸引子流动,这就相当于由事物的一部分信息自动联想出事物的全部信息。

这里应特别注意的是神经网络的记忆方式。显然,它的记忆方式与 Von Neumann 数字计算机的记忆方式完全不同。在数字计算机中,一组信息对应一组存贮单元,当给定了这组信息的正确地址后就可恢复这组信息,这就是所谓的地址寻址存贮方式(Addressable Memory),简称 AAM 方式。而神经网络的联想记忆,则是由一组信息的部分信息通过网络的演化过程回忆出这组信息的全部内容。它模拟了生物神经网络中的信息存贮机理,也就是说,用一个具有内部连接的非线性网络的动态行为模拟了联想记忆的过程。这种记忆方式称为内容可寻址存贮方式(Content Addressable Memory),简称为 CAM 方式。这种记忆方式的特点是信息存贮量大,信息检索时间与信息存贮量大小无关,可高速读取。当用记忆模式的部分信息激励网络时,可恢复出每个记忆模式的全部信息。

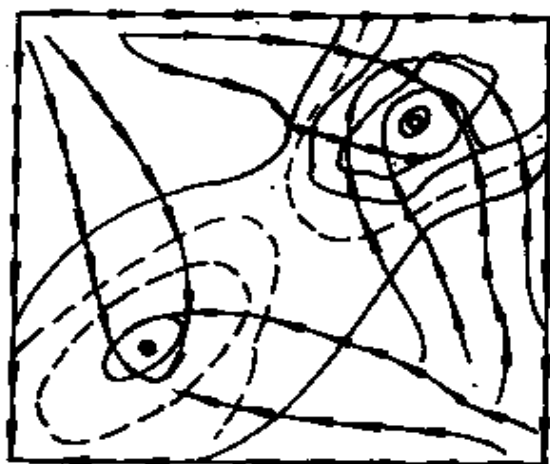


图 5.11 稳定吸引子示意图

图 5.12 是 Hopfield 神经网络联想记忆的一个实例。这个具

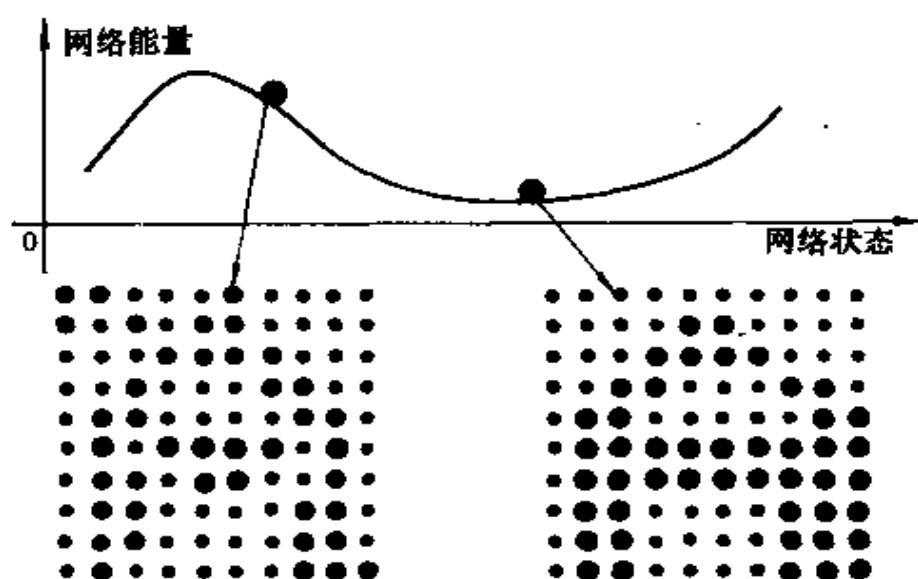


图 5.12 Hopfield 网络联想记忆实例

有 $10 \times 10 = 100$ 个神经元的 Hopfield 网络, 由“0”或“1”组成的网络各个状态之中, 能量极小点状态对应于图象 A 的完整形状; 而含有某些噪声, 即发生一些扭曲的图象 A 对应网络的某个初始状态。随着网络沿其能量减小的方向逐渐变化, 最终将由初始状态过渡到能量的极小状态, 即将图象 A 的完整形状恢复出来。

5.6 Hopfield 网络联想记忆的设计方法

实现 Hopfield 网络联想记忆的关键, 是使被记忆的模式样本对应网络能量函数的极小值。具体说就是, 设有 M 个 N 维记忆模式, 通过对网络 N 个神经元之间连接权 w_{ij} 和 N 个神经元的输出阈值 θ_i 的设计, 让这 M 个记忆模式所对应的网络状态正好是网络能量函数的 M 个极小值, 这是一个比较复杂、困难的问题。目前还没有一个适应任意形式的记忆模式、且有效的通用设计方法。针

对各种不同的情况,主要的设计方法有外积法(Outer Product Method);相关存贮法、投影学习规则法、特征结构法(Eigenstructure Method)、非对称连接矩阵网络综合法等。这些方法在使用上各有利弊。对这些方法的介绍和证明需要较深数学基础和对非线性网络平衡点的稳定性、收敛轨道的有界性、吸引子的动态范围的深入了解。因此,本书对这些方法不再作详细介绍。这里仅介绍一种比较简单、在一定约束条件下行之有效的网络联想记忆设计方法。

定理 5.1:

设 M 个 N 维记忆模式为

$$V_k = (v_k(0), v_k(1), \dots, v_k(N-1))$$

$$k = 1, 2, \dots, M$$

当连接权 w_{ij} 和输出阈值 θ_i 按式 5.13 设计,且满足式 5.14 条件时,这 M 个记忆模式将对应网络的 M 个极小值,即网络的 M 个吸引子。

$$w_{ij} = \sum_{k=1}^M [2v_k(i) - 1][2v_k(j) - 1]$$

$$w_{ii} = 0 \quad \theta_i = 0 \quad (5.13)$$

$$2 \sum_{j=0}^{N-1} v_k(j) \cdot v_p(j) - \sum_{j=1}^{N-1} v_p(j) = C_p \cdot \delta_{kp} \quad (5.14)$$

C_p ——正的常数; $p \geq 1, K \leq M$

式中,
$$\delta_{kp} = \begin{cases} 1 & k=p \\ 0 & k \neq p \end{cases} \quad (5.15)$$

我们先对以上两式作一般性解释,然后再给出严格的证明。

首先从式 5.13 可知,网络是对称的(即 $w_{ij} = w_{ji}$)、无自连接($w_{ii} = 0$)、无输出阈值($\theta_i = 0$)的。又因为

$$2v_k(i) - 1 = \begin{cases} 1 & v_k(i) = 1 \\ -1 & v_k(i) = 0 \end{cases} \quad (5.16)$$

$$k = 1, 2, \dots, M$$

$$i = 0, 1, 2, \dots, N - 1$$

所以,当 $v_k(i)$ 与 $v_k(j)$ 相等(同时为“1”或同时为“0”)时, w_{ij} 加 1; 当 $v_k(i)$ 与 $v_k(j)$ 不相等时, (w_{ij}) 减 1。所以 w_{ij} 反映了 M 个记忆模式中 $v_k(i)$ 与 $v_k(j)$ 相同或不相同的数量关系。条件式 5.14 表示的是这样一种关系: 在 M 个 N 维记忆模式中, 任取两个模式的 N 维向量, 一一对应同时为“1”的元素个数是两个向量各自为“1”的元素个数的一半。从几何角度看, 指这两个 N 维向量重叠后同时为“1”的元素个数正好是各自向量中元素值为“1”的个数的一半。后面我们将举出满足这一条件的模式的例子。

下面证明定理 5.1。

证明: 设 v_p 为 M 个记忆模式中的第 p 个模式。由式 5.5, 第 i 个神经元的输入 H_i 为

$$H_i = \sum_{\substack{j=0 \\ j \neq i}}^{N-1} w_{ij} v_p(j) - \theta_i \quad (5.17)$$

将式 5.13 代入式 5.17 得

$$H_i = \sum_{k=1}^M [2v_k(i) - 1] \sum_{\substack{j=0 \\ j \neq i}}^{N-1} [2v_k(j) - 1] \cdot v_p(j) \quad (5.18)$$

将式 5.14 代入式 5.18 得

$$\begin{aligned} H_i &= \sum_{k=1}^M [2v_k(i) - 1] C_p \cdot \delta_{kp} \\ &= C_p [2v_p(i) - 1] \end{aligned} \quad (5.19)$$

由式 5.19 可知:

$$H_i \begin{cases} > 0 & v_p(i) = 1 \\ < 0 & v_p(i) = 0 \end{cases} \quad (5.20)$$

由式 5.20 和式 5.19 可知 $v_p(i)$ 下一个时刻的值将保持不变。同理, 网络的其它神经元也将保持前一时刻的值不变。这说明此时网络已落入能量函数的极小值点。

值得注意的是,如果式 5.13 稍加变形,得式 5.21 后可以看出,网络在记忆模式 $v_k(i)$ 时,同时也记忆了模式 $v_k(i)$ 的反转模式,即将 $v_k(i)$ 向量中 N 个元素值取反后($0 \rightarrow 1, 1 \rightarrow 0$)的模式。

$$\begin{aligned} w_{ij} &= \sum_{k=1}^M [2v_k(i) - 1][2v_k(j) - 1] \\ &= \sum_{k=1}^M \{2[1 - v_k(i)] - 1\} \{2[1 - v_k(j)] - 1\} \quad (5.21) \end{aligned}$$

式 5.21 中,模式 $[1 - v_k(i)]$ 正好是模式 $v_k(i)$ 的反转模式。这说明,一个 Hopfield 网络记忆了 $2M$ 个模式。

为加深对式 5.13、5.14、5.21 的理解,下面举一简单的例子。

图 5.13 所示为两幅由 $10 \times 10 = 100$ 个元素组成的图象,各个元素根据图象的黑、白点分别取 0 或 1,将这两幅图象的 100 个元素对应两个 100 维的向量作为记忆模式。仔细观察这两幅图象,黑点重叠的个数(25 个),正好是各个图象黑点个数(50 个)的一半,即这两个记忆模式满足条件式 5.14。

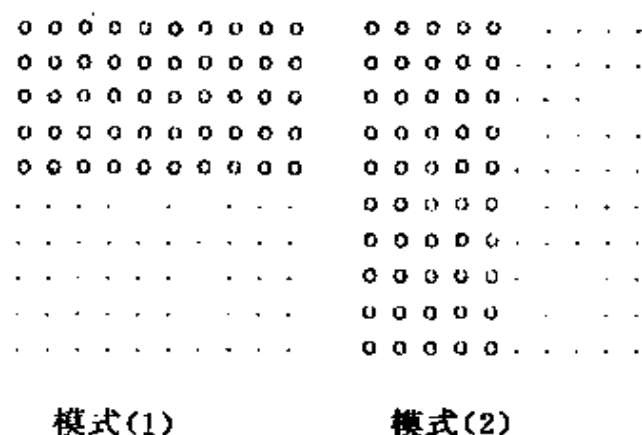


图 5.13 记忆模式

首先按式 5.13 设置 100 个神经元之间的连接权 w_{ij} $i=1,$

2, ..., 100; $j=1, 2, \dots, 100$ 。将所有输出阈值置“0”。此时,这两个记忆模式已对应于网络的两个吸引子。为检验网络联想记忆的能力,设计一个如图 5.14 所示的输入模式作为网络的初始状态,按式 5.4、5.5 的学习规则改变网络的状态,观察网络的变化趋势。图的下端 Pat 表示模式的序号, D 表示记忆模式与网络初始状态模式之间的汉明距离(即两个模式向量对应元素值之差的绝对值之和), E 表示记忆模式的网络能量, Rev 表示与两个记忆模式对应的反转模式。图的右侧 $10 \times 10 = 100$ 个单元的数字,表示该单元对应的神经元按学习规则进行状态更新的次数(初始时全为“0”)。

初始状态																			
E = 4																			
0	.	0	.	0	0	0	0	0	0	0	0	0	0	0
0	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	.	.	0	0	.	0	0	0	.	0	0	0	0	0	0	0	0	0	0
0	0	0	.	0	0	0	.	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	.	.	0	0	0	.	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	.	0	0	0	0	0	0	0	0	0	0
.	0	.	0	.	.	.	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	.	.	0	.	.	.	0	.	0	0	0	0	0	0	0	0	0	0
0	0	.	.	0	.	.	.	0	.	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	.	0	0	0	0	0	0	0	0	0	0

Pat	D	E	Rev	D	E
[0]	40	-1200	([0]	60	-1200)
[1]	50	-1200	([1]	50	-1200)

图 5.14 测试输入模式

图 5.15(a)、(b)分别表示网络经 100 回和 200 回状态变化后网络的状态。由图可知,当经过 100 次状态变化之后,原输入模式已十分接近记忆模式 1,它与记忆模式 1 的汉明距离由原来的 40 缩短到 13,网络能量由 4 下降到 -628;当经过 200 回状态变化之后,原输入模式已完全恢复为记忆模式 1。从这例子我们还可以看出,网络的初始状态总是朝着与其最邻近的记忆模式变化,也就是说朝着与其汉明距离最短的那个记忆模式变化。

100 回后

$E = -628$

0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0	1 2 1 1 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0	0 1 3 0 1 3 1 0 2 1
0 0 0 0 0 0 0 0 0 0 0 0	2 1 2 0 1 0 1 2 1 0
0 0 0 0 0 0 0 0 0 0 0 0	1 1 2 1 1 1 2 2 1 1
0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 3 0 1
0 0 0 0 0 0 0 0 0 0 0 0	0 0 1 4 2 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0	2 1 2 0 2 0 0 1 0 2
0 0 0 0 0 0 0 0 0 0 0 0	0 4 0 0 2 2 1 0 3 0
0 0 0 0 0 0 0 0 0 0 0 0	1 1 2 0 1 2 1 1 0 4

Pat	D	E	Rev	D	E
[0]	12	1200	[0]	67	1250
[1]	11	1220	[1]	48	1200

(a)

200 回后

$E = -1200$

0 0 0 0 0 0 0 0 0 0 0 0	4 3 5 1 2 5 3 2 5 3
0 0 0 0 0 0 0 0 0 0 0 0	2 5 4 4 1 1 2 4 4 1
0 0 0 0 0 0 0 0 0 0 0 0	1 2 7 1 5 4 5 2 6 2
0 0 0 0 0 0 0 0 0 0 0 0	2 1 6 1 4 2 4 3 5 1
0 0 0 0 0 0 0 0 0 0 0 0	1 3 3 4 2 3 2 3 3 3
0 0 0 0 0 0 0 0 0 0 0 0	3 1 1 1 6 1 2 4 3 1
0 0 0 0 0 0 0 0 0 0 0 0	2 5 4 6 3 2 2 3 3 5
0 0 0 0 0 0 0 0 0 0 0 0	3 2 4 2 3 0 2 2 3 2
0 0 0 0 0 0 0 0 0 0 0 0	1 5 2 2 2 5 7 2 5 2
0 0 0 0 0 0 0 0 0 0 0 0	1 5 6 1 3 5 1 2 2 5

Pat	D	E	Rev	D	E
[0]	9	960	[0]	112	-1200
[1]	10	1200	[1]	10	1200

(b)

图 5.15 网络状态变化过程

为说明网络对记忆模式的反转模式的记忆能力,我们再观察一下图 5.16 所示输入模式的变化过程。

初始状态

$E = -2$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Pat	D	E	Rev	D	E
[0]	80	-1200	([0]	40	-1200)
[1]	54	-1200	([1]	46	-1200)

(a)

100 回后

$E = -502$

1	1	0	1	1	0	3	0	1	0
1	1	0	2	3	0	1	1	1	1
0	0	1	2	1	2	1	0	1	1
2	0	1	1	0	1	0	3	0	2
0	1	0	1	2	1	2	0	2	2
0	0	0	0	0	1	0	0	0	2
2	1	0	1	1	1	0	2	1	0
4	1	3	0	1	1	0	1	0	2
3	7	1	3	0	0	4	0	0	4
1	0	1	0	2	0	0	0	0	1

Pat	D	E	Rev	D	E
[0]	83	-1200	([0]	17	-1200)
[1]	47	-1200	([1]	53	-1200)

(b)

由图可知,网络的初始状态与第一记忆模式的反转模式的汉明距离最短,所以经过 500 次状态更新,网络最终收敛于第一记忆模式的反转模式的状态上。

300 回后

E=-1060

.....	4	1	1	3	3	1	4	5	4	1
.....	5	3	2	4	3	0	2	2	5	2
0.....	0	4	4	3	3	2	4	2	2	2
.....	2	5	2	3	2	2	1	3	0	5
.....	4	4	2	1	3	2	6	1	3	6
000000000000	3	1	2	2	1	4	2	5	3	5
000000000000	5	2	3	2	4	2	0	5	3	1
000000000000	7	3	3	3	1	4	1	3	2	5
000000000000	4	10	2	6	1	4	6	1	2	6
000000000000	3	2	3	2	3	4	3	4	0	4

Pat	D	E	Rev	D	E
[0] :	97	-1200	([0] :	3	-1200)
[1] :	47	-1200	([1] :	53	-1200)

(c)

500 回后

E=-1200

.....	5	6	4	8	6	5	8	5	4	2
.....	8	6	3	6	5	0	3	3	6	7
.....	3	6	6	3	6	4	6	4	4	3
.....	5	9	5	7	6	5	5	6	2	6
.....	7	5	4	1	4	4	6	5	3	9
000000000000	6	1	2	8	2	8	3	8	3	6
000000000000	5	4	7	2	5	4	2	7	5	1
000000000000	8	5	4	5	2	7	2	4	4	5
000000000000	8	13	4	9	2	7	7	4	3	9
000000000000	4	3	11	4	7	5	5	8	1	5

(d)

图 5.16 向反转记忆模式接近的过程

5.7 Hopfield 网络联想记忆的缺陷

Hopfield 网络虽然具有很强的联想记忆功能,但是形成联想记忆的网络综合却是一个十分复杂、且往往需要满足一定约束条件的棘手问题。在许多情况下,这些约束条件很难得到满足,如网络的对称性、式 5.14 所示的条件等。这样就给 Hopfield 网络的联想记忆带来了一些缺陷和使用上的限制。总结起来主要有以下三点:

① 由网络联想出或恢复出的模式未必是与输入模式最接近的记忆模式。也就是说,在约束条件不满足的情况下,由网络的输入模式回想出的记忆模式,有时并不是与其汉明距离最短的那个记忆模式。

② 所有的记忆模式并不是以同样的记忆强度回想出来。我们可以从网络的能量函数曲线直观地解释这个问题。网络的记忆模式对应着网络能量函数的各个极小值,即能量函数曲线的各个“谷底”,如图 5.17 所示。而这些“谷底”有深有浅,“谷面”大小有宽有狭,“山谷”的“坡度”有大有小。因此,“谷底”深、“谷面”宽、坡度大的比“谷底”浅、“谷面”狭、“坡度”小的那些“谷底”有更多的被回想出来的机会,即前者比后者对应更多的初始状态,或者说前者比后者具有更大的吸引子控制域。

③ 在某些情况下,网络回想出的模式不是记忆模式中任何一个模式,而落入“伪状态”。这里再举一例说明 Hopfield 网络存在的缺陷。

如图 5.18 所示有三个记忆模式,这三个模式不符合条件式 5.14。图 5.19(a)为一个与模式 1 最接近(汉明距离为 14)的输入模式,但是当网络依据 Hopfield 工作运行规则进行网络状态改变

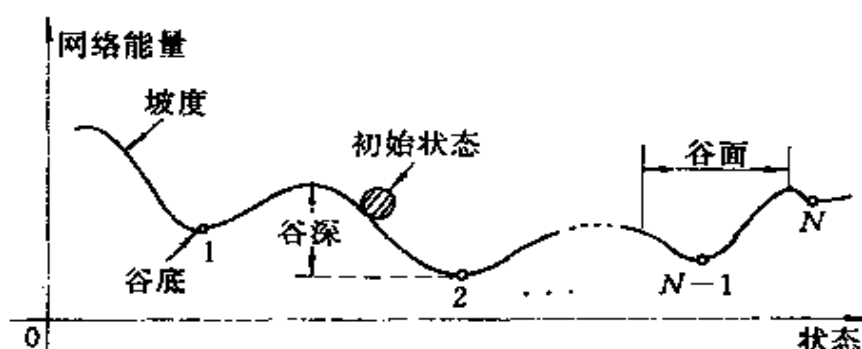


图 5.17 记忆模式对应的能量函数各个极小值

500 次后,回想出来的却是一个与模式 3 接近的模式(汉明距离为 11)。这是因为虽然输入模式与记忆模式 1 接近,但是记忆模式 3 却比记忆模式 1 具有更小的网络能量,并且记忆模式 1 与 3 之间又比较接近(汉明距离为 38),这表明这两个记忆模式的极小值是邻近的,加之模式 3 比模式 1 有更宽的吸引子控制范围。

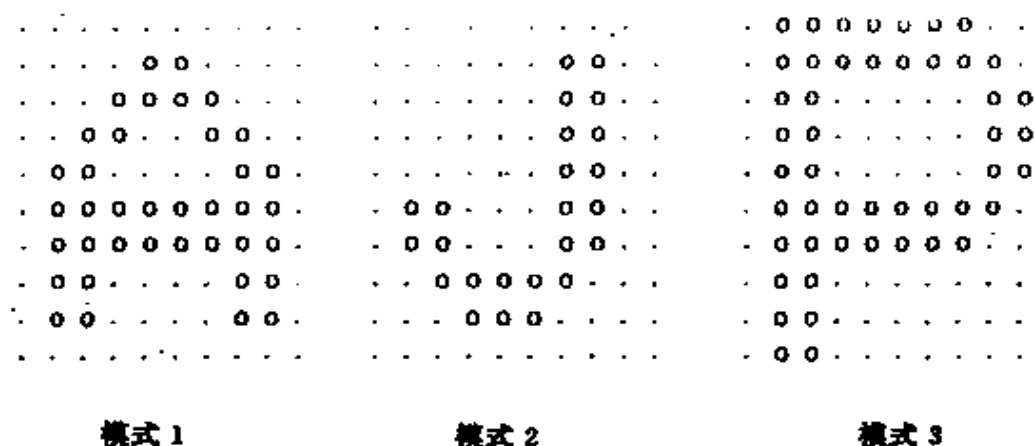


图 5.18 三个不符合约束条件式 5.14 的记忆模式

针对 Hopfield 网络存在的这些问题,许多研究者提出了各种改进方案。其中 Hopfield 本人提出过一种称为“反学习”(Unlearning)的方法^[16]。这一方法的主要思想是:当网络连接权按式 5.13 设定好之后,针对需要联想的输入模式,每次按工作运行规则改变一次网络状态时,网络的连接权也按式 5.22 做一次“微调”

初始状态									
E = -425									
. 0 0	0	0	0	0	0	0	0	0	0
0 0 0 . 0 0	0	0	0	0	0	0	0	0	0
. . . . 0 0 0 0	0	0	0	0	0	0	0	0	0
. . 0 0 0 . .	0	0	0	0	0	0	0	0	0
. 0 0 0 . .	0	0	0	0	0	0	0	0	0
. 0 . 0 0 0 0 . 0 .	0	0	0	0	0	0	0	0	0
. 0 0 . 0 0 . 0 0 .	0	0	0	0	0	0	0	0	0
. 0 0 0 0 .	0	0	0	0	0	0	0	0	0
. . 0 0 0 0	0	0	0	0	0	0	0	0	0
. 0	0	0	0	0	0	0	0	0	0
Pat	D	E	Rev	D	E				
[0] :	14	-809	([0] :	86	-2757)				
[1] :	42	-256	([1] :	58	-3114)				
[2] :	40	-1420	([2] :	60	-1950)				
500 回后									
(a)									
E = -1371									
. 0 0 0 0 0 0 0 0 . .	4	4	7	3	5	6	4	2	5
. 0 0 0 0 0 0 . . 0 .	4	5	5	4	1	1	3	5	8
. 0 0 0 0 0 0 . . 0 0	3	4	9	2	5	5	6	3	4
. 0 0 0 0 0	2	1	4	3	4	4	4	7	2
. 0 0 0 0	7	6	3	7	3	3	2	5	3
. 0 0 0 0 0 0 0 0 0 .	5	3	1	2	7	1	4	7	5
. 0 0 0 0 0 0 0 0 0	4	6	4	6	3	3	3	4	4
. 0 0 0 0	4	3	4	4	4	0	2	3	4
. 0 0 0 0	2	6	2	3	4	6	10	3	5
. 0 0	3	4	5	1	3	8	5	4	3
Pat	D	E	Rev	D	E				
[0] :	25	-309	([0] :	75	-2757)				
[1] :	61	-256	([1] :	39	-3114)				
[2] :	11	-1420	([2] :	89	-1950)				
(b)									

图 5.19 与图 5.18 中模式 1 接近输入模式及输出模式

$$\Delta w_{ij} = -\epsilon(2v_i - 1)(2v_j - 1)$$

$$(0 < \epsilon \leq 1) \quad (5.22)$$

做这种“微调”的目的,是使网络能量函数的吸引子的控制范围趋于均等,从而抑制“伪状态”现象的出现。

对于 Hopfield 网络,另一个十分有意义的问题是,一个具有 N 个神经元的 Hopfield 网络,最多能存贮多少个记忆模式,即 Hopfield 网络的信息存贮容量是多少? 如果单从网络的两个基本

参数 w_{ij} 和 θ_i 来看, 由于它们可以是任意实数, 可能有无穷多种参数组合, 因而存贮信息的能力似乎是无穷大的。但实际上 Hopfield 网络的存贮能力决定于由 w_{ij} 和 θ_i 的适当组合而能够形成网络能量函数稳定平衡点的个数。各种理论和实验表明, Hopfield 网络的信息存贮容量的上限值为 $0.15N$, 当超过这个上限后, 记忆错误将明显增多。目前还没有一个精确地解释和计算神经网络信息存贮容量的方法, 这是一个很有意义的研究课题, 特别是对设计神经网络计算机来说, 具有重要的实用价值。

5.8 连续时间型 Hopfield 神经网络

连续时间型 Hopfield 神经网络与离散型 Hopfield 网络的基本原理是一致的。但由于连续时间型 Hopfield 神经网络是以模拟量作为网络的输入、输出量, 各神经元采用同步工作方式, 因而它比离散型网络在信息处理的并行性、联想性、存贮分布性、实时性、协同性等方面更接近于生物神经网络。

连续时间型 Hopfield 网络是 Hopfield 于 1984 年在离散型神经网络的基础上提出来的, 其基本结构如图 5.20 所示。

图中电阻 R_i 和电容 C_i 并联, 模拟生物神经元的延时特性; 运算放大器是一个输入、输出按 S 函数(非线性饱和特性)关系变化的非线性元件, 它模拟生物神经元的非线性特性即: $v_i = f(u_i)$; 各放大器输出的反馈权值 w_{ij} , 反映神经元之间的突触特性, 但其中不反馈回自身, 即自反馈耦合 $w_{ii} = 0$ 。

电路中第 i 个节点的节点方程为

$$I_{out} = I_{in} + w_{ij}v_j \quad (5.23)$$

式中, $w_{ij} = 1/R_{ij}$ 是 i 放大器与 j 放大器之间的反馈耦合系数, R_{ij} 是反馈电阻。根据理想放大器的特性, 其输入端口无电流流入, 则

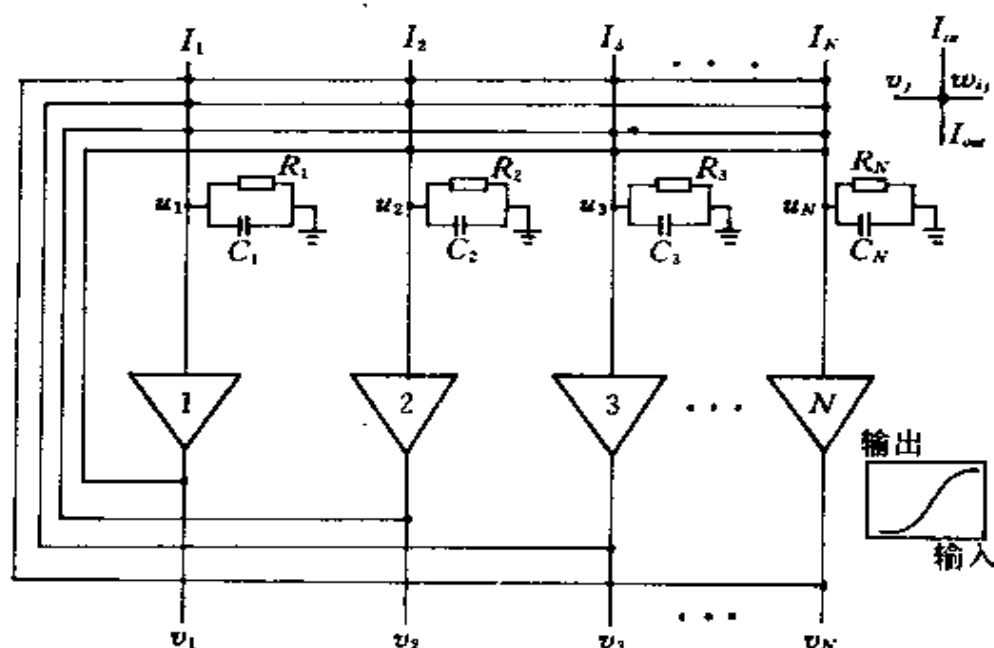


图 5.20 连续时间型 Hopfield 网络模型

第 i 个放大器的输入方程应为

$$C_i \frac{du_i}{dt} = -\frac{u_i}{R_i} + \sum_{j=1}^N w_{ij}v_j + I_i \quad (5.24)$$

由放大器非饱和特性决定的 v_i 与 u_i 的关系为

$$v_i = \frac{1}{1 + e^{-u_i}} \quad (5.25)$$

与离散型网络相同,连续型网络也具有对称性(即 $w_{ij}=w_{ji}$)。同时,也给网络定义了一个能量函数。

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij}v_i v_j - \sum_{i=1}^N v_i I_i + \sum_{i=1}^N \frac{1}{R_i} \int_0^{v_i} g^{-1}(t) dt \quad (5.26)$$

式中 $g(t)$ 为 S 函数: $g(t) = \frac{1}{1 + e^{-t}}$, $g(t)^{-1}$ 是 $g(t)$ 的反函数。

定理 5.2:

对于式 5.24 所示网络,若 $g^{-1}(t)$ 为单调递增且连续的函数,

并有 $C_i > 0, w_{ij} = w_{ji}$, 则随网络的状态变化有:

$$\frac{dE}{dt} \leq 0, \quad \text{当且仅当} \quad \frac{dv_i}{dt} = 0 \text{ 时, } \frac{dE}{dt} = 0$$

$$i = 1, 2, \dots, N$$

定理 5.2 所说的意思是, 网络状态改变总是朝着其能量函数减小的方向运动, 并且最终收敛于网络的稳定平衡点, 即 E 的极小值点。注意: 此时网络状态变化所遵循的规则与离散型网络时基本相同, 只要把式 5.4 的符号函数改成式 5.25 的 S 函数即可。

定理 5.2 的证明:

$$\because \quad \frac{dE}{dt} = \sum_{i=1}^N \frac{dE}{dv_i} \cdot \frac{dv_i}{dt}$$

$$\begin{aligned} \text{而} \quad \frac{dE}{dt} &= - \left(\sum_{\substack{j=1 \\ j \neq i}}^N w_{ij} v_j + I_i - \frac{u_i}{R_i} \right) \\ &= - C_i \frac{du_i}{dt} \quad \leftarrow \text{由式(5.24)} \end{aligned}$$

$$\begin{aligned} \text{则} \quad \frac{dE}{dt} &= - \sum_{i=1}^N \frac{dv_i}{dt} \cdot C_i \frac{du_i}{dt} = - \sum_{i=1}^N C_i \frac{du_i}{dv_i} \left(\frac{dv_i}{dt} \right)^2 \\ &= - \sum_{i=1}^N C_i \frac{d[g^{-1}(v_i)]}{dv_i} \left(\frac{dv_i}{dt} \right)^2 \end{aligned} \quad (5.27)$$

因为 $g^{-1}(t)$ 单调递增且连续, 所以 $\frac{d[g^{-1}(v_i)]}{dv_i} > 0, C_i > 0$, 故有 $\frac{dE}{dt} \leq 0$ 。显然可证: 当且仅当 $\frac{dv_i}{dt} = 0$ 时, $\frac{dE}{dt} = 0$ 。

当网络模型中的运放为理想放大器时, (即 $u_i = 0$ 时, 仍可有输出 v_i), 能量函数式 5.26 可简化为

$$E = - \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N w_{ij} v_i v_j - \sum_{i=1}^N v_i I_i \quad (5.28)$$

最后将式 5.24、5.25 整理后, 得到连续时间型 Hopfield 网络的运行方程为

$$\frac{du_i}{dt} = - \frac{u_i}{\tau} + \sum_{j=1}^N w_{ij} v_j + I_i \quad (5.29)$$

$$v_j = g(u_j) \quad (5.30)$$

式中 $g(t)$ 为 S 函数, $\tau = R_i$ 。

5.9 Hopfield 网络在优化组合问题中的应用

用神经网络解决优化组合问题,即寻找问题的最优解,是神经网络应用的一个重要方面。所谓最优解问题,就是指在给定的约束条件内,求出使某目标函数最小(或最大)化的变量组合问题。我们通过对神经网络能量函数的分析,可以得到这样一种启发:既然网络的能量函数在网络的状态按一定规则变化时,可以自动地朝着其稳定的平衡点即极小值点运动,并将最终收敛于极值点。如果把一个需要求解的问题的目标函数转换为网络的能量函数,把问题的变量对应于网络的状态。这样当网络的能量函数收敛于极小值时,问题的最优解也就随之求出。为了说明这个问题,这里举一个最有代表性的优化组合实例——旅行商问题。简称 TSP (the Traveling Salesman Problem)。这个问题同时也是用神经网络解决优化组合问题的最典型、最具有范例意义的问题。当年正是因为 Hopfield 用其连续时间型神经网络成功地求解了这个具有相当难度的组合优化问题^[17],才使人工神经网络的研究工作走出“低谷”,并重新兴盛起来。

TSP 的提法是:

设有 n 个城市 c_1, c_2, \dots, c_n , 记为: $C = \{c_1, c_2, \dots, c_n\}$; 用 d_{ij} 表示 c_i 与 c_j 之间的距离, $d_{ij} > 0$, $i, j = 1, 2, \dots, n$ 。有一旅行商从某一城市出发,访问各个城市一次且仅一次后,再回到原出发城市。要求找出一条最短的巡回路线。

目前对一问题已有许多种解法,如穷举搜索法(Exhaustive Search Method)、贪心法(Greedy Method)、动态规划法(Dynamic

Programming Method)、分枝定界法(Branch-And-Bound)等等。这些方法都存在着一个共同问题,就是当城市数 n 较大时,会产生所谓“组合爆炸”问题。如:当 $n=50$ 时,用每秒运算一亿次的巨型计算机按穷举搜索法计算 TSP 所需的时间为 5×10^{46} 年。即使城市数减少到 20 个,用这一方法求解仍需 350 年。显然这是无法实现的。下面我们看一下如何用 Hopfield 神经网络解决这一问题。

为了简单明了起见,我们仅举 $n=5$ 的例子。设这五个城市分别为 A、B、C、D、E。当任选一条路径如 $B \rightarrow D \rightarrow E \rightarrow A \rightarrow C$ 时,其总的路径长度 s 可表示为

$$s = d_{BD} + d_{DE} + d_{EA} + d_{AC} + d_{CB} \quad (5.31)$$

求解这五个城市 TSP 的第一个关键问题是找到一个简单、能够充分说明问题、又便于计算的表达形式。为此先观察图 5.21。

城市 \ 次序	1	2	3	4	5
A	0	0	0	1	0
B	1	0	0	0	0
C	0	0	0	0	1
D	0	1	0	0	0
E	0	0	1	0	0

图5.21 五城市巡回路线

图中的行表示城市,列表示城市巡回次序。按照式 5.31 提出的路线,可得到如图所示的表现形式。如果把图看作一个矩阵的话,这样形式的矩阵称为换位矩阵(Permutation Matrix)。进一步,如果把矩阵中每个元素对应于神经网络中的每个神经元的话,则可用一个 $n \times n = 5 \times 5 = 25$ 个神经元组成的 Hopfield 网络来解决这个问题。

有了明确的表达形式之后,第二个关键问题是如何把问题的目标函数表示为网络的能量函数,并将问题的变量对应为网络的

状态。解决这个具体问题有一定的复杂性,并需要一定的技巧。首先,对应换位矩阵,把问题的约束条件和最优要求分解出来:

- ① 一次只能访问一个城市 \Rightarrow 换位矩阵每列只能有一个“1”。
- ② 一个城市只能被访问一次 \Rightarrow 换位矩阵每行只能有一个“1”。
- ③ 一共有 N 个城市 \Rightarrow 换位矩阵中元素“1”之和应为 N 。
- ④ 要求巡回路径最短 \Rightarrow 网络能量函数最小值对应于 TSP 的最短路径。

如果用 v_{ij} 表示为换位矩阵第 i 行、第 j 列的元素,显然 v_{ij} 只能取“1”或“0”。当然 v_{ij} 同时也是网络的一个神经元的状态。则以上第 4 点也可表达为:构成最短路径的换位矩阵一定是形成网络能量函数极小点的网络状态。

对于上述第一个约束条件:

(a) 第 x 行的所有元素 v_{xi} 按顺序两两相乘之和 $\sum_{i=1}^{N-1} \sum_{j=i+1}^N v_{xi}$ 应为 0。

(b) N 行的所有元素按顺序两两相乘之和 $\sum_{x=1}^N \sum_{i=1}^{N-1} \sum_{j=i+1}^N v_{xi} v_{xj}$ 也应为 0。

(c) 将(b)项式前面乘上一系数 $A/2$ 则可作为网络能量函数的第一项:

$$A/2 \sum_{x=1}^N \sum_{i=1}^{N-1} \sum_{j=i+1}^N v_{xi} v_{xj}$$

同理对应第二个约束条件,可得网络能量函数的第二项:

$$B/2 \sum_{i=1}^N \sum_{x=1}^{N-1} \sum_{j=x+1}^N v_{xi} v_{xj}, \text{ 其中 } B/2 \text{ 为系数。}$$

对应第三个约束条件,换位矩阵中所有为“1”的元素之和应等于 N :

$$\left(\sum_{x=1}^N \sum_{i=1}^N v_{xi} \right) - N = 0$$

由此可得网络能量函数的第三项:

$$C/2 \left[\left(\sum_{x=1}^N \sum_{i=1}^N v_{xi} \right) - N \right]^2$$

式中,取平方值是为了使这项符合能量的表达形式,同时也表示了当总元素之和大于 N 时的一种惩罚, $C/2$ 为系数。

对应第四个约束条件:

(a) 设任意两个城市 x, y 之间的距离为 d_{xy} ;

(b) 访问这两个城市有两种途径:

$x \rightarrow y$ 和 $y \rightarrow x$, 其表达式为:

$$d_{xy} \cdot v_{xi} \cdot v_{yi+1} \text{ 和 } d_{xy} \cdot v_{xi} \cdot v_{yi-1}$$

由前三个约束条件可知,这两项至少有一项为 0;

$$\begin{aligned} (c) \quad \text{式} \quad & \sum_{i=1}^N [d_{xy} \cdot v_{xi} \cdot v_{yi+1} + d_{xy} \cdot v_{xi} \cdot v_{yi-1}] \\ &= \sum_{i=1}^N d_{xy} \cdot v_{xi} \cdot [v_{yi+1} + v_{yi-1}] \end{aligned}$$

表示顺序访问 x, y 两城市所有可能的途径(长度),按前三个约束条件,所有 N 项之和中最多只能有一项不为 0,而这一项或为 $d_{xy} \cdot v_{xi} \cdot v_{yi+1}$, 或为 $d_{xy} \cdot v_{xi} \cdot v_{yi-1}$ 。如果所有 N 项均为 0,则说明不是按相邻顺序访问这两个城市的。

(d) 式 $\sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^N d_{xy} v_{xi} \cdot (v_{yi+1} + v_{yi-1})$ 表示 N 个城市两两之间所有可能的访问路径的长度。当这一项最小时,则表示访问 N 个城市的最短距离。由此得网络能量函数的第四项:

$$D/2 \sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^N d_{xy} \cdot v_{xi} (v_{yi+1} + v_{yi-1})$$

式中 $D/2$ 为系数。

以上前三项只有当满足问题的约束条件时才能为 0,因此这三项保证了所得路径的有效性。从一般意义上讲,这三项是针对优化组合问题约束条件而设置的,称为惩罚项(意思是:不满足约束

条件,这些项就不为 0,网络能量函数就不可能达到极小值)。第四项对应问题的目标,即优化要求,其最小值就是最短路径长度。综合这四项可得到网络能量函数的最后表达形式

$$\begin{aligned}
 E = & A/2 \sum_{x=1}^N \sum_{i=1}^N \sum_{j=i+1}^N v_{xi} v_{xj} + B/2 \sum_{i=1}^N \sum_{x=1}^{N-1} \sum_{y=x+1}^N v_{xi} v_{yj} \\
 & + C/2 \left[\left(\sum_{x=1}^N \sum_{i=1}^N v_{xi} \right) - N \right]^2 \\
 & + D/2 \sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^N d_{xy} v_{xi} \cdot (v_{yi+1} + v_{yi-1})
 \end{aligned} \quad (5.32)$$

式 5.32 符合网络能量函数的定义,且只能当达到问题的最优解时, E 取得极小值,由此时的网络状态 v_{xi} 构成的换位矩阵表达了最佳旅行路线。为使网络能收敛到全局极小值,可按以下式 5.33 设置网络各连接权的初值。

设网络 (x, i) 神经元与 (y, j) 神经元之间的连接权为 $w_{xi, yj}$, 神经元 (x, i) 的输出阈值为 I_{xi} , 则有

$$\begin{aligned}
 w_{xi, yj} = & -A\delta_{xy}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{xy}) \\
 & \text{(行约束)} \qquad \qquad \qquad \text{(列约束)}
 \end{aligned} \quad (5.33)$$

$$\begin{aligned}
 & -C - Dd_{xy}(\delta_{ji+1} + \delta_{ji-1}) \\
 & \text{(全局约束)} \quad \text{(路径长度约束)}
 \end{aligned}$$

$$I_{xi} = C \cdot N \quad (5.34)$$

$$\text{式中, } \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \quad (5.35)$$

实际上,将上式代入 Hopfield 网络能量函数式 5.24 则得到 TSP 问题能量函数式 5.32(只相差一常数 N^2)。也可以说,比较式 5.24 和式 5.32,则可得到连接权表达式 5.33。将式 5.33、5.34 代入 Hopfield 网络运行方程 5.29、5.30,则得求解 TSP 的网络迭代方程

$$\frac{du_{xi}}{dt} = -\frac{u_{xi}}{\tau} - A \sum_{\substack{j=1 \\ j \neq i}}^N v_{xj} - B \sum_{\substack{y=1 \\ y \neq x}}^N v_{yj} - C \left(\sum_{x=1}^N \sum_{y=1}^N v_{xy} - N \right)$$

$$-D \sum_{y=1}^N d_{xy}(v_{yi+1} + v_{yi-1}) \quad (5.36)$$

$$v_{xi} = g\left(\frac{u_i}{u_0}\right) = \frac{1}{2} \left[1 + \tanh\left(\frac{u_i}{u_0}\right) \right] \quad (5.37)$$

这里取神经元的 I/O 函数为双曲正切函数。式中 u_{xi} 表示各神经元的内部状态, v_{xi} 表示神经元的输出。

具体计算迭代步骤为:

① 初始化 取 $u_0 = 0.02$, 为保证问题收敛于正确解, 按式 5.38 取网络的 25 个神经元内部初始状态:

$$u_{xi} = u_{00} + \delta_{u_{xi}} \quad (5.38)$$

式中, $u_{00} = \frac{1}{2} u_0 \ln(N-1)$, N ——神经元个数;

$\delta_{u_{xi}}$ 为 $(-1, +1)$ 区间内的随机值。这样做的目的是使网络因初始状态的不同而引起竞争, 从而使网络朝收敛方向发展。

② 按式 5.37 $v_{xi}(t_0) = \frac{1}{2} (\tanh \frac{u_{xi}(t_0)}{u_0})$ 求出各神经元的输出 $v_{xi}(t_0)$ 。

③ 将 $v_{xi}(t_0)$ 代入式 5.36 中求得 $\frac{du_{xi}}{dt}|_{t=t_0}$, 即:

$$\begin{aligned} \frac{du_{xi}}{dt}|_{t=t_0} = & -\frac{u_{xi}}{\tau} - A \sum_{\substack{j=1 \\ j \neq i}}^N v_{xj} - B \sum_{\substack{y=1 \\ y \neq x}}^N v_{yi} \\ & - C \left[\sum_{x=1}^N \sum_{j=1}^N v_{xj} - N \right] - D \sum_{y=1}^N d_{xy}(v_{yi+1} + v_{yi-1}) \end{aligned}$$

④ 按式

$$u_{xi}(t_0 + \Delta t) = u_{xi}(t_0) + \frac{du_{xi}}{dt}|_{t=t_0} \Delta t \quad (5.39)$$

求出下一时刻的 $u_{xi}(t + \Delta t)$ 值, 返回步骤②。

注意: 在每进行一遍巡回之后, 要检查运行结果即旅行路径的合法性。主要有三方面:

(a) 每个神经元的输出状态必须是“0”或“1”。

(b) 换位矩阵每行有且仅有一个为 1 的单元。

(c) 换位矩阵每列有且仅有一个为 1 的单元。这里只要神经元输出 $v_{xi} < 0.01$ 则可视作 0, $v_{xi} > 0.99$ 则可视作 1。当网络的运行迭代次数大于事先给定的回数时, 经检查运行结果仍属非法时, 说明从这一初始状态网络不能收敛到全局最小值。这时需要更换一组网络初始状态(即重新设置 $u_{xi}(t_0) = u_{00} + \delta_{u_{xi}}$)。从步骤②开始再进行网络迭代, 直到网络达到稳定状态。如图 5.22 所示。

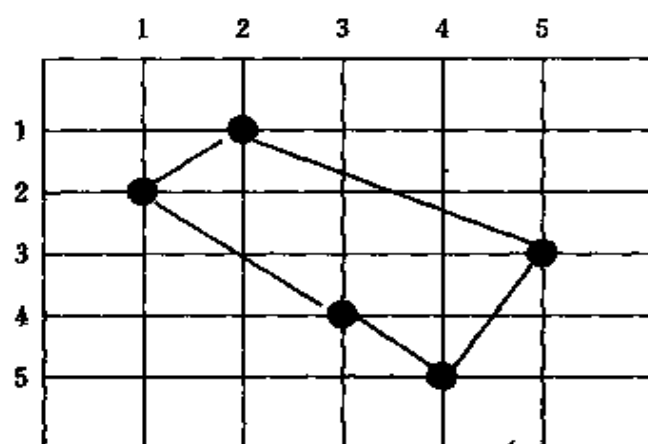


图5.22 收敛示意图

经验表明, 当选择合适的参数, 如 $A=500, B=500, C=200, D=500, S$ 函数时间常数 $\tau_0=100$ 时, 网络能得到较满意的收敛结果。

但是应该指出, 用 Hopfield 网络解决 TSP 问题并不是每次都能收敛到最小值, 而时常会“冻结”在无意义的旅行路线上。这说明 Hopfield 网络模型具有不稳健性(Non Robustness), 关于这方面的问题, 有关文献进行了深入的研究^[18], 并提出了一些改进设想^{[19][20]}, 这里不再详述。

总结以上用神经网络解决优化组合问题的方法, 其主要思想是根据问题的性质, 把目标函数与网络的能量函数联系在一起, 把问题的变量对应于网络单元状态, 通过网络运行时能量函数的最小化趋势得到问题的最优解。由于网络是以迅速收敛的迭代方式

运行的,所以那些有“组合爆炸”危险的复杂问题可以变为可计算的简单问题。而运用这一方法的关键是恰当地写出问题的网络能量函数,构造出网络能量函数的一般方法可用式 5.40 表示

$$E = \sum_i a_i E_i + a_0 E_0 \quad (5.40)$$

式中,第一项为惩罚项(即违反约束要付出代价), E_0 是需要优化的目标函数,常数 a_i 和 a_0 为平衡 E_i 和 E_0 在总能量函数中的作用,如最小化 E_0 则 $a_0 > 0$,而最大化 E_0 则 $a_0 < 0$ 。

5.10 网络应用与网络能量函数

我们已介绍了人工神经网络在三个主要方面的应用,即模式识别与分类、联想记忆和最优化。细心领会一下可以发现,这三方面的应用实际上都是在利用神经网络能量的收敛特性的基础上得以实现的,只不过利用的角度不同而已。

① 联想记忆:利用了网络能量函数的所有极小值,每一个极小值对应一个记忆模式。

② 只利用网络能量函数的全局极小值即最小值,最小值所对应的网络状态就是问题的最优解。

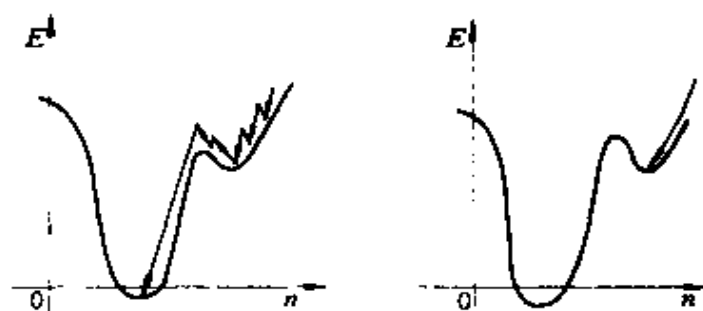
③ 利用网络能量函数的收敛特性进行网络训练,而没有直接利用其外在表现形式。

第六章 随机型神经网络

6.1 随机型神经网络的基本思想

误差逆传播算法的基本思想是通过对网络参数(连接权、输出阈值)的逐步调整实现网络学习的,它是在提供给网络的学习模式集合的全局误差按梯度下降的条件下达到网络记忆目的的。所以,当学习过程进行到全局误差开始有上升的趋势时就停止了,这往往导致网络误差陷入局部极小值,而达不到全局最小点。这种算法被形象地称为“贪心”算法,即急于找到最小解,结果却是欲速则不达。同样,对于 Hopfield 网络我们也曾指出过, Hopfield 网络的工作规则也是使网络的能量函数朝梯度下降的方向变化,即随着网络状态的不断更新,网络能量函数单调下降,其结果也往往是使网络陷入局部极小值,最终得不到网络的最优解。分析以上两种网络结构与算法的特点,我们会发现导致网络学习过程陷入局部极小值的原因主要有两点:① 网络结构上存在着输入与输出之间的非线性函数关系,从而使网络误差或能量函数所构成的空间是一个含有多极点的非线性空间;② 算法上,网络的误差或能量函数只能按单方向减小而不能有丝毫的上升趋势。对于第一点,是为保证网络具有非线性映射能力而不可少的。那么,解决网络收敛问题的途径就只能从第二点入手,即不但让网络的误差或能量函数向减小的方向变化,而且还可按某种方式向增大的方向变化,目的是使网络有可能跳出局部极小值而向全局最小点收敛。这就是随机型

神经网络算法的基本思想。图 6.1 是随机型算法与“贪心”算法的形象表示。



(a) 随机型算法 (b) “贪心”算法

图 6.1 两种类型算法的比较

6.2 模拟退火算法

模拟退火算法(Simulated Annealing Algorithm),顾名思义就是模拟金属构件退火过程的一种算法。金属或某类固体物质退火处理过程是:先用高温将其加热熔化,使其中的粒子可以自由运动,然后逐渐降低温度,粒子的自由运动趋势也逐渐减弱,并逐渐形成低能态晶格。若在凝结点附近温度下降的速度足够慢,则金属或固体物质一定会形成最低能量的基态,即最稳定结构状态。实际上,在整个降温的过程中,各个粒子都可能经历了由高能态向低能态、有时又暂时由低能态向高能态最终趋向低能态的变化过程。由此可以得到这样一种启发:如果把神经网络的状态看作金属内部的“粒子”,把网络在各个状态下的能量函数 E_i 看作是粒子所处的能态。在算法中设置一种控制参数 T ,当 T 较大时,网络能量由低向高变化的可能性也较大;随着 T 的减小,这种可能性也减小。如

果把这个参数看作温度,让其由高慢慢地下降,则整个网络状态变化过程就完全模拟了金属的退火过程,当参数 T 下降到一定程度时,网络将收敛于能量的最小值。在模拟退火算法中,有两点是算法的关键:① 控制参数 T ;② 能量由低向高变化的可能性。这两点必须结合起来考虑,当 T 大时,可能性也大; T 小时,可能性也小。把“可能性”当作参数 T 的函数,“可能性”用数学模型来表示就是概率。由此可以得到模拟退火算法如下:

设网络状态为: $u_k = \{u_i\}, i=1, 2, \dots, N, k=1, 2, \dots, 2^N$,

N ——网络神经元的个数;

各连接权为 $\{w_{ij}\} i, j=1, 2, \dots, N$;

神经元 i 的综合输入值,即内部状态为 $\{H_i\} i=1, 2, \dots, N$;

输出阈值为 $\{\theta_i\} i=1, 2, \dots, N$;

神经元 i 输出 u_i 为 1 和 0 的概率分别为 $P_{\omega}(0)$ 和 $P_{\omega}(1)$, 则

$$H_i = \sum_{j=1}^N w_{ij} \cdot u_j - \theta_i \quad (6.1)$$

$$P_{\omega}(1) = \frac{1}{1 + e^{-\frac{H_i}{T}}} \quad (6.2)$$

$$P_{\omega}(0) = 1 - P_{\omega}(1) = \frac{e^{-\frac{H_i}{T}}}{1 + e^{-\frac{H_i}{T}}} \quad (6.3)$$

式中, $i=1, 2, \dots, N$; T 称为网络温度(Network Temperature)。

式 6.2、6.3 表明,在模拟退火算法中,某神经元的输出不象 Hopfield 算法中那样,是由以内部状态 H_i 为输入的非线性函数的输出(阶跃函数)所决定的,而是由以 H_i 为变量的概率 $P_{\omega}(1)$ 或 $P_{\omega}(0)$ 所决定的。不同的 H_i 对应不同的概率 $P_{\omega}(1)$ 或 $P_{\omega}(0)$ 。进一步分析可知,以上各式体现了模拟退火算法的两个关键点。如图 6.2 所示是 $P_{\omega}(1)$ 的函数曲线。首先将 Hopfield 网络能量函数的变化式重写如下:

$$\Delta E_i = -[u_i(t+1) - u_i(t)]H_i(t) \quad (6.4)$$

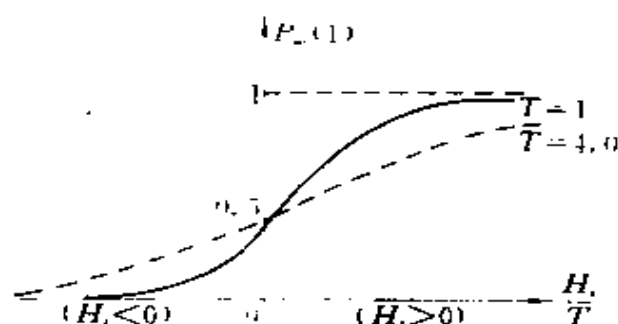


图 6.2 $P_m(1)$ 函数曲线

当神经元 i 按式 6.2 所给的概率, 在下一时刻的输出值取 1 时, 则其能量变化为

$$\Delta E_i = -[1 - u_i(t)]H_i(t) \quad (6.5)$$

由式 6.5 可以看出, 当 $H_i(t) \geq 0$ 时, $\Delta E_i \leq 0$, 表示能量随状态的变化是单调减小的; 而当 $H_i(t) < 0$ 时, $\Delta E_i \geq 0$, 表示能量将增加或不变化, 这在 Hopfield 算法中是不允许的, 而这里却允许比较小的概率(图 6.2 负横轴所对应的概率)接受这种变化。从图 6.2 中还可以看出, 当温度 T 较高时, $P_m(1)$ 相对于 H_i 的变化反应迟钝, 曲线趋于平坦, 特别是当 $T \rightarrow \infty$ 时, 曲线变为一条恒为 0.5 的直线。此时 u_i 取 1 和 0 的概率相等, 这意味着在 T 较高的期间, 网络各神经元有更多的机会进行状态选择, 相当于金属内部的粒子作激烈的自由运动。当温度降低时, $P_m(1)$ 曲线变陡, $P_m(1)$ 相对于 H_i 的变化相当敏感。当 $T \rightarrow 0$ 时, 曲线退化为一阶跃函数, 则算法也从模拟退火算法过渡到 Hopfield 算法, 所以可以说 Hopfield 算法是模拟退火算法在 $T \rightarrow 0$ 时的特例。

按式 6.1~6.3 反复进行网络的状态更新, 且更新次数 N 足够大以后, 我们会发现网络某状态出现的概率将服从式 6.6 所示的分布。

$$Q(E_i) = (1/Z)e^{-\frac{E_i}{T}} \quad (6.6)$$

$$Z = \sum_{i=1}^N e^{-\frac{E_i}{T}} \quad (6.7)$$

式中, E_i 为状态 $\{u_i\}$ 所对应的网络能量。这一概率分布就是由统计力学家 Ludwig Boltzmann (1844-1906 年) 提出的 Boltzmann 分布。式 6.6、6.7 中的 Z 是为使分布规一化而设置的常数(网络所有状态的能量之和为常数)。由这一分布可以看出, 状态的能量越小, 这一状态出现的概率就越大。这是 Boltzmann 分布的一大特点, 即“最小能量状态以最大的概率出现”。

6.3 Boltzmann 机与 Boltzmann 机工作规则

按模拟退火算法实现网络状态的更新时, 网络各状态出现的概率将服从 Boltzmann 分布。实际上, 模拟退火算法的提出并不是针对神经网络的, 它最早是由 Metropolis 于 1953 年针对模拟统计物理中液体结晶问题而提出的一种算法思想。1985 年, Hinton 等人把模拟退火算法引入神经网络中^[21], 提出了 Boltzman 机模型, 简称 BM 网络(Boltzmann Machine)。Boltzmann 机网络结构基本上与离散型 Hopfield 网络结构相似, 由 N 个神经元构成, 每个神经元取 0、1 二值输出, 且神经元之间以对称连接权相互连接。与 Hopfield 网络所不同的是, Boltzmann 机网络一般把整个神经元分为可视层与隐含层两大部分, 可视层又可分为输入部分和输出部分。但它与一般的阶层网络结构不同之处是网络没有明显的层次界限, 且神经元之间不是单向连接而是双向连接的, 如图 6.3 所示。

Boltzmann 机网络的算法根据其两大用途分为工作规则和学习规则。工作规则也就是网络的状态更新规则, 主要用于优化组合问题。学习规则也就是网络连接权和输出阈值的修正规则, 主要用于以网络作为一种外界概率分布的模拟机。这也是 Boltzmann 机网络的一个独特的用途, 我们在后面将作详细介绍。这里首先介绍 Boltzmann 机网络的工作规则。

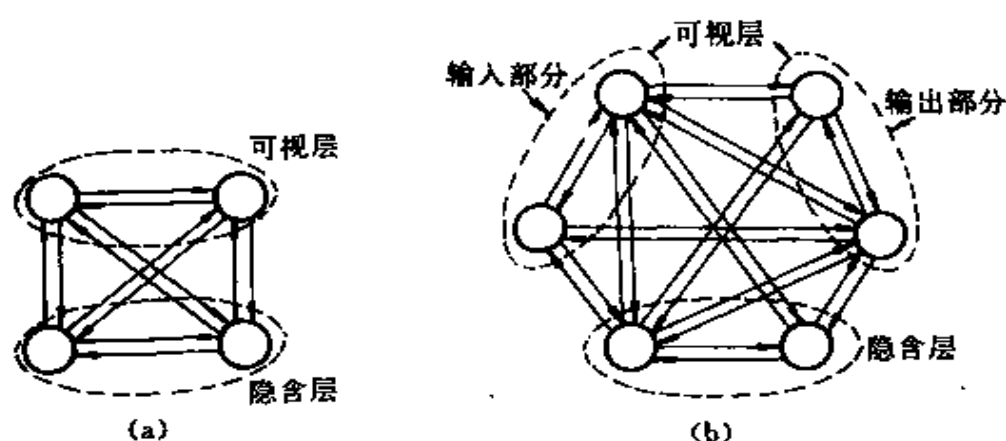


图 6.3 Boltzmann 机网络模型结构

Boltzmann 机网络工作规则与 Hopfield 网络工作规则十分相似,只是以概率方式取代阶跃函数方式来决定网络根据其神经元的内部状态而进行的状态更新,并且网络的温度参数随着网络状态更新的进行而逐渐减小。实际上,可以说 Boltzmann 机网络工作规则就是模拟退火算法的具体体现。现将其步骤归纳如下:

设网络有 N 个神经元,各神经元之间的连接权为 $\{w_{ij}\}$,各神经元的输出阈值为 $\{\theta_i\}$,输出为 $\{u_i\}$,神经元 i 的内部状态为 $\{H_i\}$ 。以上 $i=1,2,\dots,N; j=1,2,\dots,N$ 。

取 $T|_{t=0}=T_0$,给 $\{w_{ij}\}$ 和 $\{\theta_i\}$ 赋予 $[-1, +1]$ 区间内的随机值,并使 $w_{ij}=w_{ji}$, $i, j=1,2,\dots,N$ (实际应用中连接权是按联想记忆模式设计的)。

- ① 从 N 个神经元中随机选取一个神经元 i
- ② 按式 6.8 求出神经元 i 的输入总和,即内部状态 H_i

$$H_i(t) = \sum_{j=1}^N w_{ij} \cdot u_j(t) - \theta_i \quad (6.8)$$

- ③ 按式 6.9 所示概率将神经元 i 的状态更新为 1

$$P[u_i(t+1) = 1] = \frac{1}{1 + e^{-\frac{H_i(t)}{T}}} \quad (6.9)$$

④ i 以外的神经元的输出状态保持不变

$$u_j(t+1) = u_j(t) \quad (6.10)$$

$$j = 1, 2, \dots, N \quad j \neq i$$

⑤ 令 $t=t+1$, 按式 6.11 计算此时新的温度参数 $T(t+1)$

$$T(t+1) = \frac{T_0}{\log(t+1)} \quad (6.11)$$

式 6.11 称为降温策略。

⑥ 返回到步骤①, 直至温度参数 T 小于预先给定的一个截止温度 T_d 。

这里对工作规则作以下四点解释:

1) 步骤③中, 按式 6.9 更新网络状态使 $u(t+1)=1$, 一般有两种方法:

(a) 当 $H_i(t) > 0$ 时, 直接使 $u(t+1)=1$, 因为这种情况下 $\Delta E < 0$; 而当 $H_i(t) < 0$ 时, 在 $[0, 0.5]$ 区间内产生一随机数 $\epsilon(t)$, 当 $P[u_i(t+1)=1] > \epsilon(t)$ 时, 使 $u_i(t+1)=1$, 否则使 $u_i(t+1)=u_i(t)$ 。

(b) 当 $H_i(t) > 0$ 时, 使 $u(t+1)=1$; 当 $H_i(t) < 0$ 时, 当 $P[u_i(t+1)=1]$ 大于预先给定的概率值 ϵ_0 ($\epsilon_0 < 0.5$) 时, 使 $u(t+1)=1$, 否则使 $u_i(t+1)=u_i(t)$ 。

2) 关于初始温度 T_0 和结束温度 T_E , 目前还没有一个成熟的设定方法, 一般凭经验给出。

3) 关于连接权 $\{w_{ij}\}$ 和阈值 $\{\theta_i\}$ 的设定, 这里为叙述方便, 在初始化时赋予 $[-1, +1]$ 区间内的随机值。在实际应用中, 同 Hopfield 网络一样, 是需要按一定规则预先设定的。

4) 关于降温策略, 由于在 Boltzmann 机工作规则中引入了概率, 所以网络不是收敛于 1 个状态而是收敛于平衡状态。即在 N 次网络状态更新过程中, 网络各个状态出现的概率分布保持不变, 符合式 6.6 的 Boltzmann 分布。由这一分布可知, 温度参数 T 对收敛概率分布有很大影响。当温度参数 $T \rightarrow 0$ 时, 最小能量的状态实现概率为 1, 这也正是我们所期望的。但是如果开始就以 $T=0$

进行状态更新,则网络状态很可能陷入局部极小点。因此需采用逐渐降温法实现状态更新,而且只有降温速度合适才能保证网络状态收敛于全局最小点。现已证明^[22],按式 6.11 所示降温方案,能够保证网络收敛于全局最小值。但用此方法也存在着一个很大缺陷,即网络的收敛时间太长。为此,也可用式 6.12 所示方法实施降温

$$T(t) = \frac{T_0}{1+t} \quad (6.12)$$

这种降温方法称为**快速降温方案**。

同 Hopfield 网络在优化组合中的应用一样,当把问题的约束条件及目标函数转化为网络的目标函数之后,按 Boltzmann 工作规则进行网络的状态转移,最终能得到问题的最优解。

由于 Boltzmann 机网络的工作规则可使网络的状态转移,无论从任何初始状态出发,都可以收敛到网络能量函数的最小值,能量函数的各个局部极小值无法被利用来作为记忆模式的存贮点。所以 Boltzmann 机网络不能充当一般意义上的多记忆模式的联想记忆器使用。

为加深对 Boltzmann 机网络工作规则的感性认识,下面举一个由 4 个神经元组成的 Boltzmann 机网络的计算实例,观察该网络工作规则进行状态转移的过程和结果。4 个神经元对应了网络的 16 个状态,各个状态分别对应不同的能量值。首先把网络温度参数 T 固定在 100,按工作规则共进行 1000 次状态更新,把这 1000 次状态转移中网络中的各个状态出现的次数 S_i $i=1,2,\dots,16$ 记录下来,按式 6.13 计算各个状态出现的实际频率。

$$P_i = \frac{S_i}{\sum_{i=1}^N S_i} = \frac{S_i}{M} \quad (6.13)$$

式中, N ——网络神经元个数; M ——网络状态转移次数。

同时按式 6.6 计算网络各个状态出现的概率的理论值。注意,

式 6.6 中网络各个状态的能量 E_i 按式 6.14 计算

$$E_i = -\frac{1}{2} \sum_{j=1}^N w_{ij} \cdot u_i \cdot u_j + \theta_i \cdot u_i \quad (6.14)$$

实际计算结果如图 6.4 所示。

$$T=100, M=1000, N=4$$

网络状态	对应能量	状态出现 概率理论值	状态出现 概率实际值
0	0	0.0667867	0.0691000
1	244	0.0058211	0.0048000
2	236	0.0063060	0.0073000
3	588	0.0001866	0.0001000
4	-150	0.2993173	0.3117000
5	-17	0.0791626	0.0762000
6	315	0.0028619	0.0037000
7	556	0.0002570	0.0002000
8	-93	0.1692716	0.1534000
9	183	0.0107134	0.0118000
a	259	0.0050103	0.007500
b	643	0.0001076	0.000000
c	-149	0.2963391	0.2977000
d	16	0.0569119	0.0549000
e	432	0.0008882	0.0015000
f	705	0.0000579	0.0001000

图 6.4 Boltzmann 机工作规则实例

由图可以看出,网络状态的实际出现概率与理论计算概率非常接近,并且状态能量越小,出现的概率越大。作为对比,下面采用降温方案重新计算此题,采用式 6.12 的快速降温方案。温度参数

T 从 100 降至 0.01, 按网络工作规则进行网络状态更新。当 $T=0.01$ 时结束降温, 让 T 保持在 0.01 再进行 1000 次状态转移。按式 6.13 计算网络状态出现概率的实际值, 按式 6.6 计算概率的理论值, 其实际运算结果如图 6.5 所示。由图可以看出, 实施降温方案之后, 网络的状态更新完全模拟了退火过程。当网络温度 T 降到 0.01 之后, 具有最小能量的状态 4 (能量值为 -150) 出现的概率为 1, 这说明网络此时完全收敛于网络能量函数的最小值。这一实际运算结果与前面的理论分析完全一致。

$$T=100, M=1000, T_0=100, T_d=0.01$$

网络状态	对应能量值	网络状态出现 概率理论值	网络状态出现 概率的实际值
0	0	0.000000	0.000000
1	244	0.000000	0.000000
2	236	0.000000	0.000000
3	588	0.000000	0.000000
4	-150	1.000000	1.000000
5	-17	0.000000	0.000000
6	315	0.000000	0.000000
7	556	0.000000	0.000000
8	-93	0.000000	0.000000
9	183	0.000000	0.000000
a	259	0.000000	0.000000
b	643	0.000000	0.000000
c	-149	0.000000	0.000000
d	16	0.000000	0.000000
e	432	0.000000	0.000000
f	705	0.000000	0.000000

图 6.5 实施降温方案的 Boltzmann 机工作规则实例

6.4 Boltzmann 机学习规则

Boltzmann 机网络除了可以解决优化组合问题外,还可以通过网络训练模拟外界给出的概率分布,实现概率意义上的联想记忆。什么是概率意义上的联想记忆呢?简单地讲,就是网络所记忆的并不是记忆模式本身,而是记忆模式出现的概率。提供给网络训练的也不仅仅是学习模式,而且还有学习模式出现的概率。在 6.2 节中曾经指出,Boltzmann 机网络可划分为可视层与隐含层两大部分。可视层主要作为网络记忆的外部表现,即学习模式及用于回想的输入模式都是通过可视层提供给网络的,而隐含层主要用于网络记忆的内部运算。当把一组记忆模式 $V_k = \{v_1, v_2, \dots, v_N\}$ $k = 1, 2, \dots, m$ 及这组记忆模式中每一个模式应出现的概率(即这组记忆模式的概率分布函数)提供给网络的可视层之后,让网络按后面将介绍的学习规则进行学习,学习结束之后,当网络按上节介绍的工作规则进行不断地状态转移时,网络的各个状态将按记忆的学习模式的概率分布出现,即概率大的状态出现的频率高,概率小的状态出现的频率低。这时的网络相当于一个按既定概率分布输出的“概率发生器”。这种概率意义上的联想记忆称为自联想记忆。

进一步分析互联记忆方式,把网络的可视层分为输入部分和输出部分,且按下述方式提供给网络学习模式:把某个记忆模式加到网络的输入部分。同时,在网络的输出部分按一给定概率分布给出一组希望输出模式。此时所给出的概率分布函数实际上是输出模式相对于输入模式的条件概率分布。网络正是通过记忆这种条件概率分布函数来完成互联想记忆。例如,一个由 Boltzmann 机网络组成的柴油机故障诊断系统,当给网络提供一个表示排气筒有黑烟的故障输入模式后,在网络的输出部分(即故障诊断系统的诊断输出端)按产生这种故障现象的原因的概率的大小提供一系列

输出模式,如汽缸点火位置不准、油料中含有杂质等等。从此构成网络的学习模式对。

总之,无论是自联想记忆还是互联想记忆,其实质就是网络通过学习目标概率分布函数,将其记忆并在以后的回想过程中将这一概率分布再现出来。应该注意的是,Boltzmann 机网络与一般的阶层网络不同,它没有明显的层次界限。一般是根据问题的需要,在全互连接的各个神经元中选择一些作为可视层,另一些作为隐含层。可视层神经元的个数可以根据记忆模式的形式确定,而隐含层神经元的个数目前则需凭经验确定。

Boltzmann 机网络是怎样记忆目标分布函数的呢?前面曾经指出,按 Boltzmann 机工作规则进行网络的状态转移,当转移的次数足够大时,网络的状态出现将服从于 Boltzmann 分布。Boltzmann 分布函数是由网络状态的能量函数决定的,而状态的能量函数又是由网络的连接权 $\{w_{ij}\}$ 和输出阈值所决定的。因此,通过适当地调整网络的连接权和输出阈值,就可实现所希望的 Boltzmann 概率分布。连接权和阈值的调整过程也就是网络的学习过程。

下面首先介绍自联想记忆学习规则。

设网络共有 N 个神经元,其中可视层有 n 个,隐含层有 m 个 ($n+m=N$)。可视层有 $2^n=p$ 个状态,隐含层有 $2^m=q$ 个状态,整个网络共有 $2^N=M$ 个状态 ($p+q=M$)。各层状态可表示为:可视层状态: $U_a=(u_1, u_2, \dots, u_n)$ $a=1, 2, \dots, p$; 隐含层状态: $U_b=(u_1, u_2, \dots, u_m)$ $b=1, 2, \dots, q$ 。由 N 个神经元构成的整个网络状态的概率分布函数为 $Q(U_a, U_b)$, 网络的连接权及输出阈值为 $\{w_{ij}\}$ 和 $\{\theta_j\}$ $i, j=1, 2, \dots, N$, 网络在第 k 个状态时的能量为 $E_k(U_a, U_b)$ 。

根据式 6.6 给出的 Boltzmann 概率分布,网络的状态概率分布函数 $Q(U_a, U_b)$ 为

$$Q(U_a, U_b) = (1/Z) e^{-E_k(U_a, U_b)} \quad (6.15)$$

$$Z = \sum_{k=1}^M e^{-E_k(U_a, U_b)} \quad (6.16)$$

$$k = 1, 2, \dots, M$$

可视层实际输出的状态概率分布 $Q(U_a)$ 为

$$Q(U_a) = \sum_{b=1}^q Q(U_a, U_b) \quad (6.17)$$

$$a = 1, 2, \dots, p$$

为表示目标概率分布 $P(U_a)$ 与实际概率分布 $Q(U_a)$ 的偏差, 引用统计学上的 Kullback 偏差 G (也称交叉熵) 的定义

$$G(w_{ij}) = \sum_{a=1}^p P(U_a) \ln \frac{PU_a}{Q(U_a)} \quad (6.18)$$

$G(w_{ij}) \geq 0$, 且只有在 $P(U_a) = Q(U_a)$ 时, $G(w_{ij}) = 0$ 。显然, $G(w_{ij})$ 越小, 实际输出状态概率分布就越接近目标状态概率分布。因此网络的学习过程也就是求 $G(w_{ij})$ 极小值的过程。

相对 w_{ij} 的微小变化量 Δw_{ij} , $G(w_{ij})$ 的变化量为

$$G(w_{ij} + \Delta w_{ij}) = G(w_{ij}) + \Delta w_{ij} \frac{\partial G(w_{ij})}{\partial w_{ij}} \quad (6.19)$$

如果设 Δw_{ij} 为

$$\Delta w_{ij} = -\epsilon \cdot \frac{\partial G(w_{ij})}{\partial w_{ij}} \quad (6.20)$$

则必有

$$G(w_{ij} + \Delta w_{ij}) \leq G(w_{ij}) \quad (6.21)$$

式 6.21 说明, 如果按式 6.20 进行连接权调整, 则网络的交叉熵 $G(w_{ij})$ 必定呈单调减小趋势。随着连接权调整的反复进行, $G(w_{ij})$ 将收敛于极小值, 即可实现目标分布函数。

式 6.20 中 $G(w_{ij})$ 对 w_{ij} 的偏微分为

$$\frac{\partial G(w_{ij})}{\partial w_{ij}} = -\frac{1}{T} (P_{ij}^{(+)} - P_{ij}^{(-)}) \quad (6.22)$$

其中

$$P_{ij}^{(+)} = \sum_{a=1}^p P(U_a) \frac{\sum_{b=1}^q u_i u_j e^{-\frac{E_k(U_a, U_b)}{T}}}{\sum_{b=1}^q e^{-\frac{E_k(U_a, U_b)}{T}}} \quad (6.23)$$

$$P_{ij}^{(-)} = \frac{1}{Z} \sum_{k=1}^M u_i u_j e^{-\frac{E_k(u_i, u_j)}{T}} \quad (6.24)$$

式 6.23 的 $P_{ij}^{(+)}$ 表示网络的可视层各神经元的输出按所希望的概率分布固定在某一状态下, 仅让隐含层的各神经元按 Boltzmann 工作规则进行状态更新, 当更新次数足够大并认为网络已达到平衡状态之后神经元 i 和 j 同时输出为 1 的概率(也称 u_i 与 u_j 之间的对称概率)。

式 6.24 的 $P_{ij}^{(-)}$ 表示当整个神经元按 Boltzmann 工作规则进行状态更新、并达到平衡状态之后, 神经元 i 与 j 同时输出为 1 的概率。

将式 2.22 代入式 2.20 则得到网络连接权的修正值

$$\Delta w_{ij} = \frac{\epsilon}{T} (P_{ij}^{(+)} - P_{ij}^{(-)}) \quad (6.25)$$

式 6.25 表示了 Boltzmann 机网络的自联想记忆的学习方法。式中 $\frac{\epsilon}{T} P_{ij}^{(+)}$ 项表示连接权的调整量 Δw_{ij} 与 $P_{ij}^{(+)}$ 成比例增加。而 $P_{ij}^{(+)}$ 是神经元 i 与 j 之间的对称概率, 即 u_i 与 u_j 同时为 1 的数量越多, $P_{ij}^{(+)}$ 越大; 反之亦然。这类似于 Hebb 学习原理: 两个神经元同时兴奋, 则它们之间的连接权得以增强。 $P_{ij}^{(+)}$ 形成过程称为 Boltzmann 机网络的学习过程。式中 $-\frac{\epsilon}{T} \cdot P_{ij}^{(-)}$ 项表示连接权的调整量 Δw_{ij} 与 $P_{ij}^{(-)}$ 成比例减小, 这正好与 Hebb 学习原理相反, 因此这一项称为反学习项。 $P_{ij}^{(-)}$ 形成的过程称为 Boltzmann 机网络的反学习过程, 即 Boltzmann 机网络, 通过可视层与“外界环境”接触时(也形象地称为“清醒”时)进行 Hebb 学习; 而当与“外界环境”切断联系时(也形象地称为“睡眠”时)进行反学习。

现将 Boltzmann 机网络自联想记忆学习规则步骤总结如下:

① 初始化 把连接权 $\{w_{ij}\}$ 赋予 $[-1, +1]$ 区间内随机值并令 $\{Q_i\} = 0 \quad i, j = 1, 2, \dots, N$ 。

② 按所给定的外界概率(目标概率分布) $P(U_a)$ 将网络可视层

各神经元的输出固定在某一状态:

$$U_a = (u_1, u_2, \dots, u_n)$$

③ 从 T_0 温度开始,按网络工作规则(模拟退火算法)对网络隐含层各神经元的输出进行状态更新,直至达到 T_d 温度下的平衡状态:

$$U_b = (u_1, u_2, \dots, u_m)$$

其中, T_0 为初始温度, T_d 为结束温度。

④ 在隐含层的平衡状态下,保持温度 $T=T_d$ 不变,再进行 L 次全网络的状态更新,每次更新后,当神经元 i 和 j 同时为 1 时,计算式 6.26(学习过程)

$$n_{ij}^{(+)} = n_{ij}^{(+)} + 1 \quad i, j = 1, 2, \dots, N \quad (6.26)$$

⑤ 重新从温度 T_0 开始,按工作规则对全网络神经元 $\{u_i\} \ i=1, 2, \dots, N$ 进行状态更新,直至达到 T_d 温度下的平衡状态:

$$U = (u_1, u_2, \dots, u_n, u_{n+1}, u_{n+2}, \dots, u_{n+m}) \quad n+m \leq N.$$

⑥ 在网络的平衡状态下,保持温度 $T=T_d$ 不变,再进行 L 次全网络的状态更新。每次更新后,当神经元 i 和 j 同时为 1 时,计算式 6.27(反学习过程)

$$n_{ij}^{(-)} = n_{ij}^{(-)} + 1 \quad i, j = 1, 2, \dots, N \quad (6.27)$$

⑦ 返回步骤②,共进行 M 次循环,并要求 $M > P$, P 为可视层的状态个数。

⑧ 按式 6.28、6.29 计算对称概率 $P_{ij}^{(+)}$ 、 $P_{ij}^{(-)}$

$$P_{ij}^{(+)} = \frac{1}{M \times L} n_{ij}^{(+)} \quad (6.28)$$

$$P_{ij}^{(-)} = \frac{1}{M \times L} n_{ij}^{(-)} \quad i, j = 1, 2, \dots, N \quad (6.29)$$

⑨ 按式 6.30 修正网络的各个连接权 w_{ij}

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (6.30)$$

$$\Delta w_{ij} = \frac{\epsilon}{T_d} (P_{ij}^{(+)} - P_{ij}^{(-)}) \quad i, j = 1, 2, \dots, N \quad (6.31)$$

⑩ 返回步骤②,使循环次数大于或等于预先给定的值 Y 。

在以上步骤中,需预先确定的数值有: $P(U_a), \epsilon, T_0, T_d, L, M, Y$ 。

学习结束后的网络在进行回想时,可使网络从任何初始状态出发,按工作规则进行若干次状态转移,则网络可视层各个状态的出现概率分布将符合网络学习时所希望的概率分布。

下面介绍 Boltzmann 机网络的互联想记忆学习规则。这一规则与自联想记忆学习规则十分相似。

设网络共有 N 个神经元。其中,可视层中输入部分有 n_i 个,输出部分有 n_o 个,隐含层有 m 个 ($n_i + n_o + m = N$)。输入部分有 $2^{n_i} = p_i$ 个状态,输出部分有 $2^{n_o} = p_o$ 个状态,隐含层有 $2^m = q$ 个状态,全网络共有 $2^N = M$ 个状态,其中, $p_i + p_o + q = M$ 。各个部分状态可表示为:

可视层输入部分状态: $U_a = (u_1, u_2, \dots, u_{n_i}) \quad a = 1, 2, \dots, p_i$;

可视层输出部分状态: $L_c = (l_1, l_2, \dots, l_{n_o}) \quad c = 1, 2, \dots, p_o$;

隐含层状态: $U_b = (u_1, u_2, \dots, u_m) \quad b = 1, 2, \dots, q$;

希望联合概率分布为: $P_{ac}(U_a, U_c) = P(U_a)P(L_c|U_a)$

其中, $P(L_c|U_a)$ 为在输入模式 U_a 下的输出模式 $L_c \quad c = 1, 2, \dots, p_o$ 出现的希望条件概率分布。

学习步骤如下:

① 初始化 在连接权 $\{w_{ij}\} \quad i, j = 1, 2, \dots, N$ 赋予 $[-1, +1]$ 区间内的随机值,令 $\{\theta_i\} \quad i = 1, 2, \dots, N$ 为 0。

② 随机地选取输入模式 $U_a = (u_1, u_2, \dots, u_{n_i})$ 加到可视层的输入部分。

③ 按希望条件概率分布 $P(U_b|U_a)$,把输出模式 $U_b = (u_1, u_2, \dots, u_m)$ 固定在可视层输出部分上。

④ 按网络工作规则(模拟退火算法),从温度 T_0 开始对网络隐含层的各神经元输出进行状态更新,直至达到 T_d 温度下的平衡状态。

⑤ 在隐含层的平衡状态下,再对其进行 T_d 温度下的 L 次状态更新。每次更新后,当整个网络中的神经元 i 和 j 同时为 1 时,计算式 6.31(学习过程)

$$n_{ij}^{(+)} = n_{ij}^{(+)} + 1 \quad i, j = 1, 2, \dots, N \quad (6.32)$$

⑥ 重新从 T_0 温度开始,按网络工作规则对网络中除可视层输入部分以外的所有神经元进行状态更新,直至达到 T_d 温度下的平衡状态。

⑦ 在可视层输出部分和隐含层达到平衡状态下,再对其进行 L 次状态更新,当全网络中神经元 i 和 j 同时为 1 时,计算式 6.33(反学习过程)

$$n_{ij}^{(-)} = n_{ij}^{(-)} + 1 \quad i, j = 1, 2, \dots, N \quad (6.33)$$

⑧ 返回步骤③共进行 M_1 次循环,并要求 $M_1 > P_0$, P_0 为输出层的状态个数。

⑨ 按式 6.34、6.35 计算对称概率 $P_{ij}^{(+)}$ 、 $P_{ij}^{(-)}$

$$P_{ij}^{(+)} = \frac{1}{M \times L} n_{ij}^{(+)} \quad (6.34)$$

$$P_{ij}^{(-)} = \frac{1}{M \times L} n_{ij}^{(-)} \quad (6.35)$$

⑩ 按式 6.35 修正网络的各个连接权 w_{ij}

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (6.36)$$

$$\Delta w_{ij} = \frac{\epsilon}{T_d} (P_{ij}^{(+)} - P_{ij}^{(-)}) \quad (6.37)$$

$$i, j = 1, 2, \dots, N$$

⑪ 返回步骤②,选取下一组学习模式进行 M_2 次循环,并要求 $M_2 > p_i$, p_i 为输入部分的状态个数。

⑫ 从步骤②~⑩进行 Y 次循环,结束学习。

在以上步骤中,需预先确定的数值有: $P(L_c | U_c)$, ϵ , T_0 , T_d , L , M_1 , M_2 , Y 。

注意,在对每一组学习模式进行训练时,输入部分的状态总是

固定在某个输入模式的状态。

学习结束后的网络在进行回想时,当给网络的输入部分提供一输入模式 U_s 后,对网络除输入部分按网络工作规则进行状态更新,在网络的输出部分各状态出现的概率分布将符合学习过的希望概率分布 $P(L_s|U_s)$ 。

这里举一个 Boltzmann 机网络自联想学习规则的计算实例。

为简单起见,设网络只有两个神经元,一为可视层神经元 V_0 ,一个为隐含层神经元为 V_1 。两个神经元各有 0,1 两种状态。可视层输出为 1 的概率为 0.1。网络如图 6.6 所示。

首先赋予连接权初值 $w_{10} = -10$,
 $w_{01} = -10$ 。

设网络温度 $T_0 = 100, T_d = 10$,以快速降温方式,按网络工作规则对网络进行状态更新,以此得到在没有进行学习、训练的情况下的状态概率分布,即网络状态的初始分布情况。其结果如图 6.7 所示:

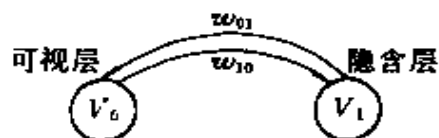


图 6.6 含有两个神经元的 BM 网络

网络状态		对应能量	状态出现的理论概率	状态出现的实际概率
可视层	隐含层			
0	0	0.0000	0.0000	0.1310
0	1	-8.0000	0.0000	0.2530
1	0	-10.0000	0.0000	0.3580
1	1	-8.0000	0.0000	0.2580

图 6.7 网络状态的初始分布情况

图中状态能量按式 6.14 计算,状态出现的理论概率按式 6.6 计算。由图可以得到网络可视层输出为 1 的概率: $0.3580 + 0.2580$

$=0.6160$ 。按式 6.18 计算网络的 Kullack 偏差为 $G=0.5848$ 。

下面进行网络学习。已知 $P(U_a)=P(U_0=1)=1$ 。设 $T_0=100$, $T_d=10$, $L=1000$, $M=128$, $Y=6$, $\varepsilon=0.1$ 。图 6.8 所示的是每次大循环后,网络可视层输出为 1 的概率分布及 Kullack 偏差。

次数 项目	1	2	3	4	5	6
$P(U_0=1)$	0.3910	0.3190	0.2260	0.2260	0.1590	0.1390
G	0.2152	0.1349	0.0542	0.0542	0.0146	0.0069

图 6.8 网络学习的中间结果

网络学习结束后,,以网络温度 $T_0=100$, $T_d=10$ 的快速降温方式,按网络的工作规则让网络回想。网络连接权的初值为 $w_{01}=-16.7566$, $w_{10}=-16.7656$,网络回想结果如图 6.9 所示。

网络状态		对 应 能 量	网络状态出现 概率的理论值	网络状态出现 概率的实际值
可视层	隐含层			
0	0	0.0000	0.4922	0.1460
0	1	-16.9858	0.4219	0.7560
1	0	7.8008	0.0156	0.0430
1	1	7.5806	0.0703	0.0550

图 6.9 网络回想结果

由图可以看出,网络可视层输出为 1 的概率: $0.0430+0.0550=0.0980$ 。按式 6.18 计算此时网络的 kullack 偏差为 $G=0.000022$ 。可见,网络的实际概率分布与希望概率分布已十分接近。作为网络误差测度的 Kullack 偏差也已经很小。

6.5 网络小结

我们介绍模拟退火算法及 Boltzmann 机网络的工作和学习规则时曾多次指出,这一算法可使网络的能量函数收敛于全局最小值,从而可以得到应用问题的最优解。但是实际情况有时并不是这样,所得到的解是近似的最优解。这是什么原因呢?分析一下这一算法的收敛过程就可以得到答案。网络的各个状态随着更新过程的不断进行,形成一状态序列: $V(0), V(1), \dots, V(i), \dots, V(k)$,而前后状态所对应的网络能量不外乎以下三种情况:

$$\begin{aligned} E[v(k+1)] &> E[v(k)] \\ E[v(k+1)] &= E[v(k)] \\ E[v(k+1)] &< E[v(k)] \end{aligned} \quad (6.38)$$

只不过前两种出现的概率较小而已。由于算法的这一特点,使网络在陷入局部最小时有机会跳出来;但也正是由于这一点,使得当前状态 $v(k)$ 所对应的网络能量 $E[v(k)]$ 有可能比前一个状态 $v(k-1)$ 所对应的网络能量 $E[v(k-1)]$ 大。尤其是当网络的初始温度 T_0 不够大、降温过程太快且结束温度 T_d 不够小的情况下,这种可能性会更大。甚至会产生当前解比中间所经历的最好解差得多的现象。这就是为什么有时模拟退火算法的结果反不如其它算法结果好的原因。针对模拟退火算法存在的这种缺点,人们又提出了一种改进的算法(Improved Annealing Procedure,简称 LAP 算法)。本书不再详述,读者可以参考有关文献^[23]。

尽管模拟退火算法仍然存在一些不足,但它比快速下降的“贪心”算法得到最优解的概率高得多;且这一算法具有很强的通用性,除了适用于神经网络之外,还可以应用于图象识别、大规模集成电路设计等众多的组合优化问题。特别是当待解决的问题复

杂性较高、规模较大,对问题所处的领域的有关知识了解甚少的情況下,这一算法具有明显的优越性。因为它不象其它算法那样,需要过多的依赖于问题所处领域的知识来提高算法的性能。但是也应当看到,当已知了大量有关待解问题的背景知识之后,这一算法却无法充分利用它们,算法的优点变成了缺点。另外,在 Boltzmann 学习规则中,包含着其工作规则,学习与反学习交替进行,因此,网络计算量猛增。特别是当温度 T 下降速度较慢时,网络收敛过程十分缓慢,这也是制约这种网络应用的主要障碍。

第七章 竞争型神经网络

7.1 竞争型神经网络的基本思想

竞争型神经网络是一种以无教师示教方式进行网络训练的网络。网络通过自身训练,自动对输入模式进行分类,这一点与Hopfield 网络的模式分类功能十分相似。但是,竞争型神经网络及其学习规则与其它类型的神经网络和学习规则相比,有着自己鲜明的特点。在网络结构上,它既不象阶层型网络那样各层神经元之间只有单向连接,也不象全连接型网络那样网络结构上没有明显的层次界限。它一般是由输入层和竞争层构成的两层网络。两层之间各神经元实现双向全连接,而且网络中没有隐含层。有时竞争层各神经元之间还存在横向连接。在学习算法上,它模拟生物神经系统依靠神经元之间的兴奋、协调与抑制、竞争的作用来进行信息处理的基本动力学原理,指导网络的学习与工作,而不是象其它大多数网络学习规则那样,是以网络的误差或能量函数的单调递减作为算法准则的。竞争型神经网络的基本思想是网络竞争层各神经元竞争对输入模式的响应机会,最后仅一个神经元成为竞争的胜者,并将与获胜神经元有关的各连接权朝着更有利于它竞争的方向调整。这一获胜的神经元则表示对输入模式的分类。除了竞争的方法外,还有通过抑制手段获取胜利的方法,即网络竞争层各神经元抑制所有其它神经元对输入模式的响应机会,从而使自己“脱颖而出”,成为取胜神经元。除此之外还有一种称为侧抑制的方法,即每个神经元只抑制与自己邻近的神经元,而对远离自己的神

神经元不抑制。这种方法常常用于图象处理,解决图象边缘的缺陷问题。竞争型神经网络更一般的学习规则允许出现一个以上的胜者,连接权的调整发生在获胜神经元的集合中。这种学习规则也称为协同学习或泄漏竞争学习。

竞争型神经网络经常作为基本的网络形式,构成其它一些具有自组织能力的网络。如本章将要介绍的自适应共振理论(Adaptive Resonance Theory)网络、第八章将要介绍的自组织映射(Self-Organizing Feature Map)网络、第九章将要介绍的对向传播(Counter Propagation)网络等。

7.2 基本竞争型神经网络及学习规则

竞争型神经网络有许多具体形式和学习算法,本节介绍其中一种比较简单的网络结构和学习算法^[24]。网络的基本结构如图 7.1 所示。

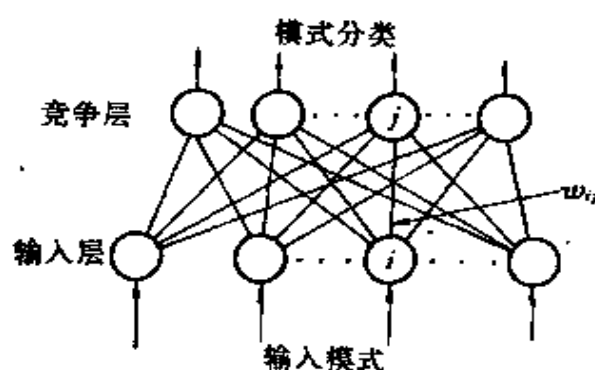


图 7.1 基本竞争型网络结构

网络分为输入层和竞争层。其中输入层有 N 个神经元,竞争型有 M 个神经元。网络的连接权为 $\{w_{ij} \mid i=1, 2, \dots, N; j=1, 2, \dots, M\}$, 且约束条件为:

$$\sum_{i=1}^N w_{ij} = 1 \quad (7.1)$$

网络的 P 个二值输入学习模式为: $A_k = (a_1^k, a_2^k, \dots, a_N^k)$, 与其对应的竞争层输出模式为 $B_k = (b_1^k, b_2^k, \dots, b_M^k)$ $k=1, 2, \dots, p$ 。

则网络的学习规则如下:

① 初始化 按式 7.1 的约束条件赋予 $\{w_{ij}\}$ $i=1, 2, \dots, N; j=1, 2, \dots, M$ $[0, 1]$ 区间内的随机值。

② 任选 P 个学习模式中的一个模式 A_k 提供给网络的输入层。

③ 按式 7.2 计算竞争层各神经元的输入值 s_j :

$$s_j = \sum_{i=1}^N w_{ij} a_i^k \quad i = 1, 2, \dots, M \quad (7.2)$$

④ 按“胜者为王”的原则, 以 s_j ($j=1, 2, \dots, M$) 中最大值所对应的神经元作为胜者, 将其输出状态置为 1, 而其它所有神经元的输出状态置为 0, 即:

$$\begin{aligned} b_j &= 1 & s_j > s_i (i = 1, 2, \dots, M; i \neq j) \\ b_i &= 0 & i \neq j \end{aligned} \quad (7.3)$$

如果出现 $s_j = s_i$ 的现象, 则按统一约定取左边的神经元为获胜神经元。

⑤ 与获胜神经元相连的各连接权按式 7.4 进行修正。而其它所有连接权保持不变。

$$\begin{aligned} w_{ij} &= w_{ij} + \Delta w_{ij} \\ \Delta w_{ij} &= \eta \cdot \left(\frac{a_i^k}{m} - w_{ij} \right) \end{aligned} \quad (7.4)$$

$$i = 1, 2, \dots, N \quad (0 < \eta < 1)$$

式中, η 为学习系数, m 为第 k 个学习模式向量 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 中元素为 1 的个数。

⑥ 选取另一个学习模式, 返回步骤③, 直至 P 个学习模式全部提供给网络。

⑦返回步骤②,直至各连接权的调整量变得很小为止。

对以上学习规则做如下分析:

① 式 7.4 中的学习系数 η 反映了学习过程中连接权调整量的大小。 η 的典型值一般为 0.01~0.03。

② 由式 7.4 可知,竞争层获胜神经元 j 与输入层神经 i 之间的连接权,当 a_i 为 1 时,由于 w_{ij} 在满足式 7.1 的约束条件下有 $w_{ij} \ll 1$,所以其调整量为正,即连接权向增大的方向变化;当 a_i 为 0 时,调整量为负,即连接权向减小的方向变化。所有连接权始终将在 (0,1) 之间变化。如图 7.2 所示的一个例子,由于输入模式第 1、5、6 元素为 1,所以其到竞争层第 j 个神经元的连接权向增大方向发展,而元素值为 0 的神经元的连接权则向减小的方向变化。

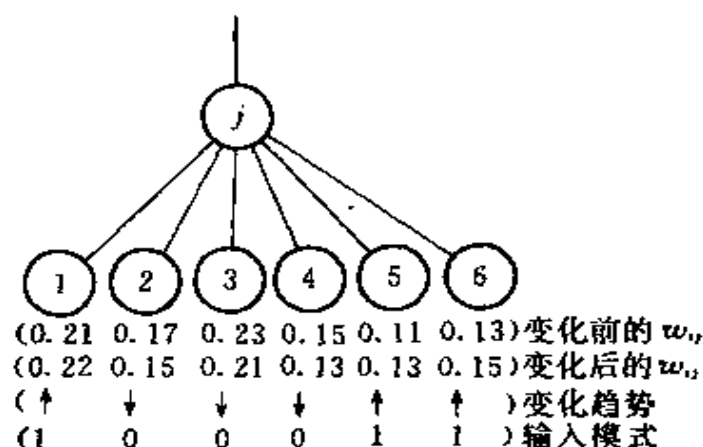


图 7.2 竞争学习的连接权调整

③ 当同一个学习模式反复提供给网络学习后,则这一模式前次所对应的竞争层获胜神经元的输入值 s_j 会逐渐增大,继续保持其胜者的地位。进而,当与这一学习模式非常近似的模式提供给网络时,也将促使同一神经元在竞争中获胜。这样,在网络回想时,就可以根据所记忆的学习模式按式 7.3 对输入模式作出最邻近分类,即以竞争层获胜神经元表示分类结果。

④ 分析式 7.4 可知,无论网络连接权如何调整,终将满足约束条件式 7.1。

$$\begin{aligned}\sum_{i=1}^N \Delta w_{ij} &= \eta \cdot \left[\underbrace{\frac{1}{m} \sum_{i=1}^N a_i^b}_m - \underbrace{\sum_{i=1}^N N_{ij}}_1 \right] \\ &= \eta \cdot (1 - 1) \\ &= 0\end{aligned}\quad (7.5)$$

下面举两个简单的例子,进一步说明基本竞争型神经网络的学习及分类功能。

(1) 三维模式组的分类

设由四个模式构成的一组三维学习模式为: $P_1 = (1, 0, 1)$, $P_2 = (1, 0, 0)$, $P_3 = (0, 1, 0)$, $P_4 = (0, 1, 1)$, 要求把这四个学习模式分为两类。图 7.3 为网络结构图。在网络分类之前,对学习模式之间的汉明距离作一分析,得到如图 7.4 所示的结果。

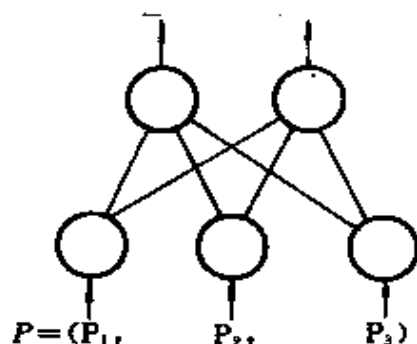


图 7.3 网络结构图

	P_1	P_2	P_3	P_4
P_1	0	1	3	2
P_2	1	0	2	3
P_3	3	2	0	1
P_4	2	3	1	0

图 7.4 模式间的汉明距离

按网络学习规则对这四个学习模式进行记忆训练,最后得到的网络稳定分类结果为:

$$\begin{aligned}P_1 = (1, 0, 1) \\ P_2 = (1, 0, 0)\end{aligned} \left. \vphantom{\begin{aligned}P_1 = (1, 0, 1) \\ P_2 = (1, 0, 0)\end{aligned}} \right\} \text{第一类}$$

$$\begin{aligned}P_3 = (0, 1, 0) \\ P_4 = (0, 1, 1)\end{aligned} \left. \vphantom{\begin{aligned}P_3 = (0, 1, 0) \\ P_4 = (0, 1, 1)\end{aligned}} \right\} \text{第二类}$$

从这一分类结果可以看出,网络的分类结果完全符合最邻近

分类原则。如图 7.4 所示, P_1 、 P_2 之间和 P_3 、 P_4 之间具有相对最短的汉明距离。

(2) 平面域模式组的分类

设有一个 4×4 点阵组成的平面域, 如图 7.5 所示。每点可取 0 或 1, 要求将平面域中的 16 点分成两类。为此, 采用如图 7.6 所示的结构网络, 并将平面域中所有邻近的点 (共 24 对) 组成如图 7.7 所示的学习模式。将这 24 个学习模式分别提供给网络, 按学习规则进行 1000 次学习, η 取 0.05。最后的分类结果如图 7.8 所示。平面域右上角的点归为一类, 左下角的点归为另一类。

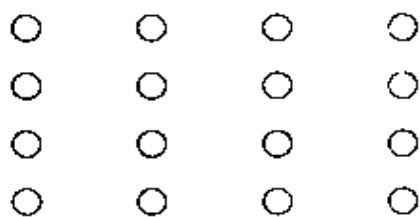


图 7.5 4×4 点组成的平面域

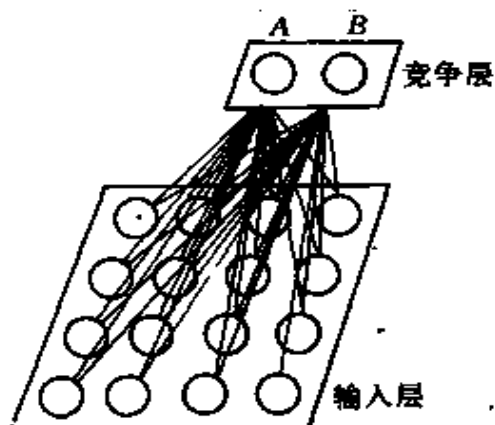


图 7.6 平面域分类网络结构

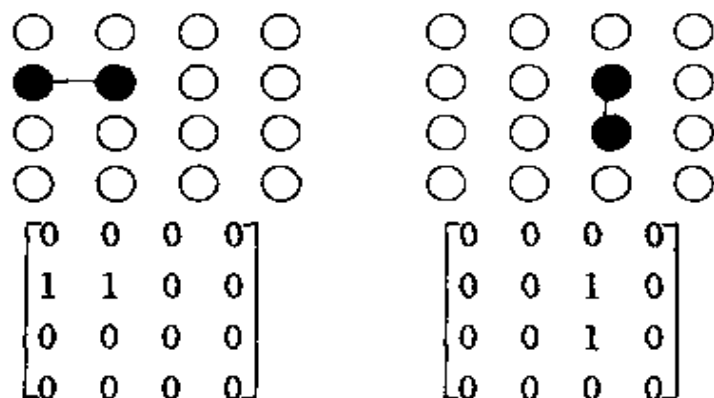


图 7.7 学习模式示例

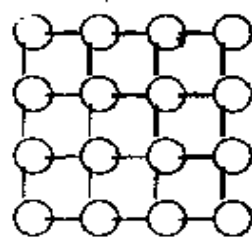


图 7.8 平面域分类结果

这一分类结果符合自然分类的原则, 即两个类别具有相同的点数, 同一类别点之间的联系大于不同类别点之间的联系, 两个类

别之间的联系比任何其它划分形式都少。

下面再对基本竞争型神经网络做以下三点说明：

① 对于竞争型网络，如果学习模式集合本身具有较明显的分类特征，则网络可以对其进行正确的分类；特别是对同一类输入模式，网络具有较稳定的输出响应，即同一类或相似的输入模式始终对应一个输出单元的激活状态。

② 当学习模式集合本身错乱无章，没有一定的分类规律时，网络对输入模式的响应呈现振荡现象，即对应同一输入模式，可能会激活不同的输出单元。

③ 网络的分类性能与网络连接权的初始值，学习系数 η 及提供给网络的学习模式的顺序有关。

7.3 抑制竞争型神经网络及学习规则

7.1 中曾经指出，除了靠竞争手段得到获胜神经元的方法之外，还有一种靠抑制的手段得到获胜神经元的方法。与这种方法对应的神经网络结构如图 7.9 所示。

如图所示，这一网络结构与单纯竞争型网络结构大致相同。不同的是竞争层各神经元之间出现了起相互抑制作用的连接键（注意，连接键只表示抑制关系，并没有具体数值）。这一算法与基本竞争型神经网络的算法的主要区别

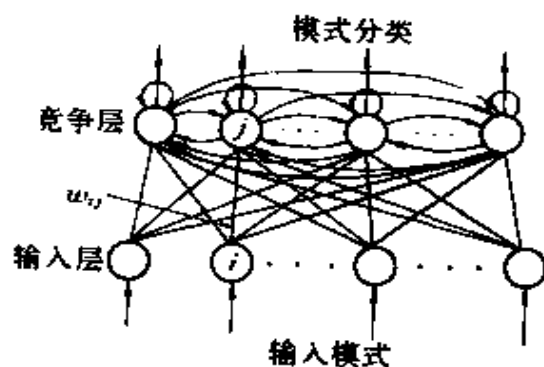


图 7.9 抑制竞争型网络结构

是：当某个竞争层神经元的输入值大于其它所有神经元的输入值时，并不象基本竞争型学习规则那样，将其自身的输出状态置为 1，将其它所有神经元的输出状态置

为 0, 而是保持其本身的输出值不变, 将其它所有神经元的输出值靠抑制作用逐渐减小。这样, 竞争层各神经元的输出就形成连续变化的模拟值。

设网络有 N 个输入层神经元和 M 个竞争层神经元, 网络的 P 个二值输入、学习模式为 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$, 网络的连接权为 (w_{ij}) $i=1, 2, \dots, N; j=1, 2, \dots, M$ 。网络竞争层的输出向量为 $B_k(t) = (b_1^k(t), b_2^k(t), \dots, b_M^k(t))$ $k=1, 2, \dots, p$ 。 $B_k(t)$ 表示网络对 k 个输入、学习模式的第 t 次学习结果。

则网络的学习规则如下:

① 给 $\{w_{ij}\}$ $i=1, 2, \dots, N; j=1, 2, \dots, M$ 赋予 $(0, 1)$ 之间的随机值, 且满足式 7.1 约束条件。

② 任选一个学习模式提供给网络的输入层。

③ 按式 7.6 计算竞争层各神经元的输入 s_j

$$s_j = \sum_{i=1}^N w_{ij} a_i \quad j = 1, 2, \dots, M \quad (7.6)$$

④ 以竞争层各神经元输入值 s_j 作为其输出值的初始值, 即:

$$b_j(0) = s_j \quad j = 1, 2, \dots, M \quad (7.7)$$

⑤ 按式 7.8 计算竞争层各神经元下一时刻输出的值

$$b_j(t) = F\{b_j(t-1) + \frac{\alpha}{Q} b_j(t-1) - \frac{\beta}{Q} [\sum_{i \neq j}^M b_i(t-1)]\} \quad (7.8)$$

$$t = 1, 2, \dots, j = 1, 2, \dots, M, 0 < \alpha \ll 1, 0 < \beta \ll 1$$

$$Q = \sum_{j=1}^M b_j(t-1) \quad (7.9)$$

$$F(x) = \begin{cases} x & 0 \leq x < 1 \\ 0 & x < 0 \\ 1 & x > 1 \end{cases} \quad (7.10)$$

⑥ 返回步骤⑤, 按式 7.8 对输出值 $b_j(t)$ 进行反复更新, 直至第 j 个神经元的输出值远远大于其它 $M-1$ 个神经元的输出值。

⑦ 按式 7.4 对与神经元 j 连接权进行调整。

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

$$\Delta w_{ij} = \eta \left(\frac{a_i^k}{m} - w_{ij} \right)$$

$$i = 1, 2, \dots, N \quad (0 < \eta < 1)$$

⑧ 选择另一个输入、学习模式,返回步骤③,直至所有输入、学习模式全部提供给网络。对上以学习规则作如下说明:

(a) 当一个输入、学习模式提供给网络的输入层之后,对竞争层的每个神经元只进行一次输入值的计算,并把计算结果作为其输出值的初始值。

(b) 式 7.9 中 Q 表示竞争层所有神经元上一时刻输出值的总和。

(c) 式 7.8 中的 α, β 是一个很小的常数,表示自兴奋与侧抑制的反馈强度。

(d) 函数式 7.10 是如图 7.10 所示的一种限幅值为 0 和 1 的非线性放大函数。

(e) 式 7.8 括弧中的第一项是神经元 j 的前一时刻的输出值,后两项则表示对这一输出值的修正量。其中,第二项表示神经元 j 的自兴奋,也可称为是神经元 j 的正反馈,第三项表示另 $M-1$ 个神经元对神经元 j 抑制的负反馈。同时由式 7.8 还可以看出,神经元 j 的输出值越大,对其它神经元的抑制作用就越强,自身兴奋的正反馈也就越大,从而还可引起其它神经元对自己的抑制作用的减弱。这样经过反复迭代学习,最终神经元 j 将“脱颖而出”,成为获胜神经元。

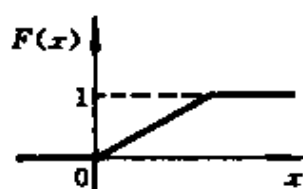


图 7.10 $F(x)$
函数
曲线

图 7.11 表示了有 7 个输出神经元的网络随学习过程其输出值的变化情况。

抑制竞争学习算法的突出优点是:这种学习方式更接近生物神经系统的记忆动力学过程,从而使得网络的非线性记忆性能更

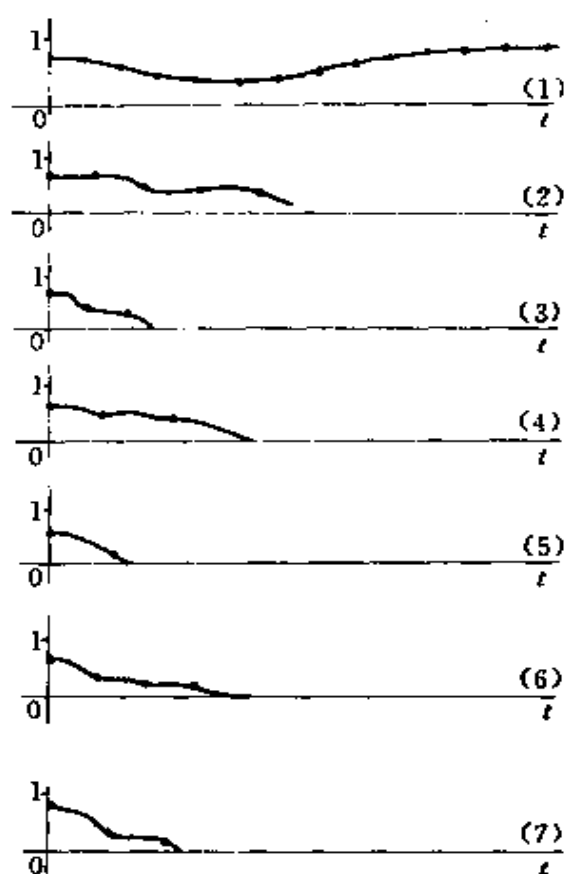


图 7.11 七个输出神经元的变化过程

加优越。为用模拟电路实现网络提供了可能性。

对于侧抑制学习算法,本书不再做详细介绍,读者可参考有关文献^[25]。

7.4 自适应共振理论网络的提出及特点

人类的大脑是由大量神经细胞组成的一个复杂、庞大的神经网络系统,这一系统不但可以记忆来自外界的各种信息,具有可塑性,而且还可以将新、旧信息相互不影响地保存下来,即所谓的稳

定性。但是,对于一般的模拟人类大脑的人工神经网络来说,却不能同时很好地解决可塑性和稳定性问题,即很难做到既能保持已有记忆不受影响,又能对新的信息加以记忆。例如,一个已学习结束的误差逆传播网络(BP 网络),当它学习一个新的模式时,将会打乱已有的连接权,使分布存贮在这些连接权上的已有记忆遭受严重破坏,甚至导致已有记忆完全消失。避免这一问题出现的唯一办法,就是将新、旧学习模式一起重新提供给网络进行记忆,直至达到网络的稳定状态。这样做,显然给神经网络的应用带来了很大的不便。因此,寻找一个能完全模拟大脑可塑性与稳定性的人工神经网络,就成为人们努力的一个目标。Grossberg 博士提出的自适应共振理论(Adaptive Resonance Theory,简称 ART),使这一努力获得了成功。这一理论的显著特点,就是充分利用了生物神经细胞之间自兴奋与侧抑制的动力学原理,让输入模式通过网络双向连接权的识别与比较,最后达到共振来完成对自身的记忆,并以同样的方法实现网络的回想。当提供给网络回想的是一个网络中已记忆的、或是与已记忆的模式十分相似的模式时,网络将会把这个模式回想出来,提出正确的分类。如果提供给网络回想的是一个网络记忆中不存在的模式,则网络将在不影响已有记忆的前提下,将这一模式记忆下来,并将分配一个新的分类单元作为这一记忆模式的分类标志。尤其是当我们了解了这一网络的学习算法之后,还将发现蕴藏在这一网络中的许多深层次的特点。

7.5 自适应共振理论网络结构及学习、工作规则

ART 网络共有三种类型:ART1、ART2 和 ART3,这里主要介绍 ART1 网络。

TRT1 网络结构如图 7.12 所示。网络分输入/输出两层,一般根据各层所具有的功能特征还称输入层为比较层,输出层为识别层。与其它阶层型网络的显著区别是:网络不但有从输入层至输出层的前馈连接权,还有从输出层至输入层的反馈连接权。

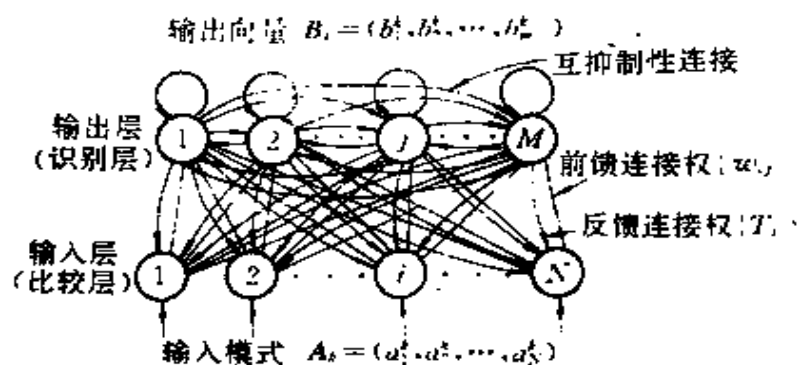


图 7.12 ART1 网络结构

设网络输入/输出层各有 N 个和 M 个神经元,二值输入模式和输出向量分别为: $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 和 $B_k = (b_1^k, b_2^k, \dots, b_M^k)$, $k = 1, 2, \dots, p$, 其中 p 为输入学习模式的个数。前馈连接权和反馈连接权分别为 $\{w_{ij}\}$ 和 $\{t_{ij}\}$ $i = 1, 2, \dots, N$; $j = 1, 2, \dots, M$ 。

ART1 网络的学习及工作过程,是通过反复地将输入学习模式由输入层向输出层自下而上地识别和由输出层向输入层自上而下的比较过程来实现的。当这种自下而上的识别和自上而下的比较达到共振,即输出向量可以正确反映输入学习模式的分类,且网络原有记忆没有受到不良影响时,网络对一个输入学习模式的记忆和分类则告完成。为了更加清楚地阐明 ART1 网络的学习、工作过程,现将这一过程分为初始化阶段、识别阶段、比较阶段和探寻阶段来分别加以分析和介绍。

(1) 初始化阶段

ART1 网络需要初始化的参数主要有三个,即 $\{w_{ij}\}$ 、 $\{t_{ij}\}$ 和

ρ 。其中 ρ 称为网络的警戒参数,其取值范围为 $0 < \rho < 1$,关于 ρ 的作用,后面还将详细说明。前馈连接权 $\{w_{ij}\}$ 与反馈连接权 $\{t_{ij}\}$ 的初始化与其它类型网络连接权初始化不同,它不是用取随机值的方法,而是按一种固定的模式来设定。其中反馈连接权 $\{t_{ij}\}$ 在网络的整个学习工作过程中取 0 或 1 二值形式。这一参数实际上反映了输入/输出层之间反馈比较的范围或强度。由于网络在初始化前没有任何记忆,相当于一张白纸,即没有选择比较范围的余地。因此可将 $\{t_{ij}\}$ 全部设置为 1,即

$$\begin{aligned} \{t_{ij}\} = 1 \quad i = 1, 2, \dots, N \\ j = 1, 2, \dots, M \end{aligned} \quad (7.11)$$

这意味着网络在初始状态时,输入/输出层之间将进行全范围比较,随着学习过程的深入,再按一定规则选择比较范围。

前馈连接权 $\{w_{ij}\}$ 在网络学习结束后,承担着对学习模式的记忆任务。在对 $\{w_{ij}\}$ 初始化时,应该给所有学习模式提供一个平等竞争的机会,然后通过对输入模式的竞争,按一定规则调整 $\{w_{ij}\}$ 。 $\{w_{ij}\}$ 的初始值按式 7.12 设置

$$\begin{aligned} \{w_{ij}\} = \frac{1}{1 + N} \quad i = 1, 2, \dots, N \\ j = 1, 2, \dots, M \end{aligned} \quad (7.12)$$

(2) 识别阶段

网络的识别阶段类似于前两节介绍的基本竞争型神经网络的竞争学习过程。实际上如果去掉网络的反馈连接权 $\{t_{ij}\}$,ART1 网络结构与基本竞争型网络结构完全相同。而 ART1 网络的学习识别过程就发生在输入学习模式由输入层向输出层的传递过程中,这里假设在输出层不存在神经元之间的相互抑制。在这一阶段,首先将一个输入学习模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 提供给网络的输入层,然后把作为输入学习模式存贮媒介的前馈连接权 $\{w_{ij}\}$ 与表示对这一输入学习模式分类结果的输出层的各个神经元进行比较,以寻找代表正确分类结果的神经元 g 。这一比较与寻找过程是通过

寻找输出层神经元最大加权输入值,即神经元之间的竞争过程实现的,如式 7.13、7.14 所示。

$$s_j = \sum_{i=1}^N w_{ij} a_i^k \quad j = 1, 2, \dots, M \quad (7.13)$$

$$\begin{aligned} b_j^k &= 1 & s_j > s & (i = 1, 2, \dots, M \ i \neq j) \\ b_i &= 0 & i &\neq j \end{aligned} \quad (7.14)$$

当然,这一得到获胜神经元的过程也可以按抑制、竞争或侧抑制的方法实现,使网络的分类性能更加完善。

至此,网络的识别过程只是告一段落,并没有最后结束。此时,神经元 $j=g$ 是否真正有资格代表对输入学习模式 A_k 的正确分类,还有待于下面的比较和寻找阶段来进一步确定。一般情况下需要对代表同一输入学习模式的分类结果的神经元进行多次反复识别。

(3) 比较阶段

比较阶段是继识别阶段之后的一个重要阶段,也是 ART1 网络学习工作过程中最具有特色的一个阶段。在介绍 ART1 网络特点时曾经指出:ART1 网络与其它类型网络的最大区别就在于它不但模拟了人脑的可塑性,而且成功地模拟了人脑的稳定性,即对新信息的学习不影响对原有信息的记忆。而比较阶段与寻找阶段正是实现 ART1 网络这一功能的两个主要阶段。可以这样认为,如果 ART1 网络的学习工作过程只到识别阶段为止,那么它与前面介绍的基本竞争型网络则完全一样。在这种情况下,如果网络输出层 M 个神经元已分别代表了输入学习模式的 M 个分类,则当让网络记忆一个未学习过的新模式时,按竞争规则,它总能找到一个输出神经元 j 代表自己的分类。而这样做,原来这个神经元 j 所代表的学习模式的分类则被破坏。也就是说,对这个新信息的记忆破坏了原有信息的记忆。解决这一问题的一个有效办法就是,每当给已学习结束的网络提供一个供识别的输入模式时,首先检查一下这个模式是否是已学习过的模式,如果是,则让网络回想出这个

模式的分类结果;如果不是,则对这个模式加以记忆,并分配一个还没有利用过的输出层神经元来代表这个模式的分类结果。比较阶段的主要职责就是完成这一检查任务。具体过程如下:把由输出层每个神经元反馈到输入层的各个神经元的反馈连接权向量 $T_j = (t_{j1}, t_{j2}, \dots, t_{jN})$ $j=1, 2, \dots, M$ 作为对已学习的输入模式的一条记录,即让向量 $T_j = (t_{j1}, t_{j2}, \dots, t_{jN})$ 与输出层第 j 个神经元所代表的某一学习输入模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 完全相等。当需要网络对某个输入模式进行回想时,这个输入模式经过识别阶段,竞争到神经元 g 作为自己的分类结果后,要检查神经元 g 反馈回来的向量 T_g 是否与这个输入模式相等。如果相等,则说明这是一个已记忆过的模式,神经元 g 代表了这个模式的分类结果,识别与比较产生了共振,网络不需要再经过寻找阶段,直接进入对下一个输入模式的识别阶段;如果不相符,则放弃神经元 g 的分类结果,进入寻找阶段。图 7.13 表示了这一检查比较方式。

这里需要特别指出的是,在比较阶段中,当用向量 T_g 与输入模式 A_k 进行比较时,允许两者之间有一定的差距,差距的大小由警戒参数 ρ 决定,首先计算式 7.15

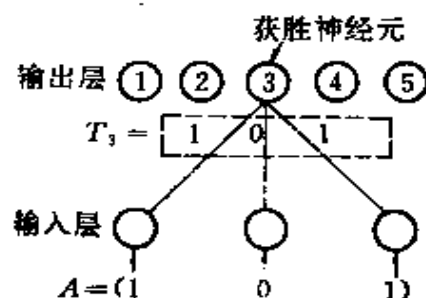


图 7.13 比较阶段示意图

$$C_g = \frac{\sum_{i=1}^N t_{gi} * a_i}{\sum_{i=1}^N a_i} \quad (7.15)$$

式中, $\sum_{i=1}^N t_{gi} * a_i$ 表示向量 $T_g = (t_{g1}, t_{g2}, \dots, t_{gN})$ 与输入模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 的逻辑“与”。由于 T_g 与 A_k 都是二值向量,所以这个

和式实际上就是向量 T_j 与 A_k 中同时为 1 的元素的个数。 $\sum_{i=1}^N a_i$ 表示输入向量 A_k 中元素为 1 的个数。这两个和式的比值,即式 7.15 则表示向量 T_j 与 A_k 的拟合度。当 $T_j = A_k$ 时, $C_g = 1$ 说明两向量完全相等;当 $C_g \geq \rho$ 时,说明拟合度大于要求,没有超过警戒线。以上两种情况均可以承认识别结果。同时当 $C_g \neq 1$ 且 $C_g > \rho$ 时,按式 7.16 和 7.17 将前馈连接权向量 $W_g = (w_{1g}, w_{2g}, \dots, w_{Ng})$ 和反馈连接向量 $T_g(t_{g1}, t_{g2}, \dots, t_{gN})$ 向着与 A_k 更接近的方向调整。

$$w_{ig}(t+1) = \frac{t_{gi}(t) \cdot a_i}{0.5 + \sum_{i=1}^N t_{gi}(t) \cdot a_i} \quad (7.16)$$

$$i = 1, 2, \dots, N$$

$$t_g(t+1) = t_{gi}(t) * a_i \quad (7.17)$$

$$i = 1, 2, \dots, N$$

式 7.17 所表达的调整意图是十分显见的。对于式 7.16,后面还将作详细分析。当 $C_g < \rho$ 时,说明拟合度小于要求,超过了警戒线,则拒绝识别结果,将神经元 g 重新复位为 0,并将这个神经元排除在下次识别范围之外,网络转入寻找阶段。

(4) 寻找阶段

寻找阶段是网络在比较阶段拒绝了识别结果之后转入的一个反复探寻的阶段,在这一阶段中,网络将在余下的输出层神经元中搜索输入模式 A_k 的恰当分类。只要在输出向量 $B_k = (b_1^k, b_2^k, \dots, b_n^k)$ 中含有与这一输入模式 A_k 相对应、或在警戒线以内相对应的分类单元,则网络可以得到与记忆模式相符的分类结果。如果在已记忆的分类结果中找不到与现在输入的模式相对应的分类,但在输出向量中还有未曾使用过的单元,则可以给这个输入模式分配一个新的分类单元。在以上两种情况下,网络的寻找过程总能获得成功,也就是说共振终将发生。

现将 ART1 网络的学习工作规则归纳总结如下:

① 初始化

$$t_{ij}(0) = 1 \quad (7.18)$$

$$w_{ij}(0) = \frac{1}{1 + N} \quad (7.19)$$

$$i = 1, 2, \dots, N$$

$$j = 1, 2, \dots, M \quad 0 < \rho \leq 1$$

② 将输入模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 提供给网络的输入层。

③ 计算输出层各神经元的输入加权和

$$s_j = \sum_{i=1}^N w_{ij} \cdot a_i^k \quad j = 1, 2, \dots, M \quad (7.20)$$

④ 选择 A_k 的最佳分类结果

$$s_x = \text{MAX}_{j=1}^M [s_j] \quad (7.21)$$

令神经元 g 的输出为 1。

⑤ 计算以下三式, 并进行判断

$$|A_k| = \sum_{i=1}^N a_i^k \quad (7.22)$$

$$|T_g \cdot A_k| = \sum_{i=1}^N t_{gi} \cdot a_i^k \quad (7.23)$$

$$\frac{|T_g \cdot A_k|}{|A_k|} > \rho \quad (7.24)$$

当式 7.24 成立时, 转入步骤⑦, 否则转入步骤⑥。

⑥ 取消识别结果, 将输出层神经元 g 的输出值复位为 0, 并将这一神经元排除在下次识别的范围之外, 返回步骤④。当所有已利用过的神经元都无法满足式 7.24 式, 则选择一个新的神经元作为分类结果进入步骤⑦。

⑦ 承认识别结果, 并按下式调整连接权

$$w_{ig}(t+1) = \frac{t_{gi}(t) \cdot a_i}{0.5 + \sum_{i=1}^N t_{gi} \cdot a_i} \quad (7.25)$$

$$i = 1, 2, \dots, N$$

$$t_{gi}(t+1) = t_{gi}(t) * a_i \quad (7.26)$$

$$i = 1, 2, \dots, N$$

⑧ 将步骤⑥复位的所有神经元重新加入识别范围之内,返回步骤②对下一个模式进行识别。

以上是 ART1 网络的学习、工作规则。无论是网络学习还是网络回想,都使用这一规则。只不过在网络回想时,只对那些与未使用过的输出神经元有关的连接权向量 $\{w_{ij}\}$, $\{T_j\}$ 才按式 7.18、7.19 进行初始化。其它连接权向量保持网络学习后的值不变。当输入模式是一个网络已记忆的学习模式时,不再需要按步骤⑦进行网络连接权的调整。这是因为,当输入模式与网络记忆的学习模式完全相等,且按式 7.25、7.26 进行调整时,连接权不会发生任何变化。而当输入模式与网络记忆模式存在一定差距时,如果按式 7.25、7.26 进行连接权调整,则将影响网络原有模式的记忆效果。但是当输入的是一个全新的模式,需要网络对其另加记忆时,则必须按步骤⑦进行连接权调整。

7.6 自适应共振理论网络特性分析

如前所述,ART1 网络的最突出的特点就是将大脑的可塑性与稳定性同时成功地应用于人工神经网络。下面针对 ART1 网络的一些深层次特性作进一步分析。

(1) 在线学习功能

从 ART1 学习、工作规则中可知,ART1 网络对任何一个输

入模式,首先试图将其归入网络已记忆的模式中的某一类别里,当这一努力无法实现时,则将其划入一个新的类别。也就是说,无论输入模式分类是否已记忆在网络中,只要有剩余输出单元,总可以对其进行正确分类。这个过程使网络边学习、边回想,实现了在线学习功能。这种将网络学习与网络工作融为一体的算法,在其它类型的网络中是不多见的。

(2) 快速与慢速学习方式

ART1 网络存在着两种学习方式,即所谓“快速”与“慢速”学习方式。前面介绍的是快速学习方式。由网络的学习、工作规则可知,一旦网络找到了输入模式的恰当分类,则可按式 7.25、7.26 一次性地将这一输入模式存贮于连接权 $\{w_{ij}\}$ 和 $\{t_{ij}\}$ 上。这种学习方式的优点是,学习速度快,同时响应速度也快,适合于实时应用。但是这种学习方式也存在着很大的缺陷。由于将输入模式一次性地存贮于连接权上,故网络把输入模式和模式中所含的噪声(如果存在的话)同等对待,从而使网络的抗干扰能力大大降低。慢速学习方式与快速学习方式最大的不同就是网络连接权的变化不是一次性完成的,而是随着输入模式多次反复地提供给网络,慢慢地将连接权朝存贮这一输入模式的方向变化,即以统计平均的方法,将输入模式的统计特征分布存贮在连接权上,有效地滤除输入模式中所含的噪声。这种学习方式的具体算法一般是采用侧抑制方法。由此可知,当输入模式集合受到噪声污染时,宜采用慢速学习方式。

(3) 可控制的分类细度

在 ART1 网络学习、工作规划中,改变警戒参数 ρ 的大小可以控制网络分类的精细度。当 ρ 较小($\rho < 0.4$)时,网络分类比较粗,即将一些相互差别较大的模式归入同一类;而当 ρ 较大,特别是接近于 1 时,网络分类则非常精细,稍有差别的模式将归入不同的类别。但这时网络的抗干扰能力将明显下降。图 7.14 表示了参数 ρ 对网络分类精度的影响。

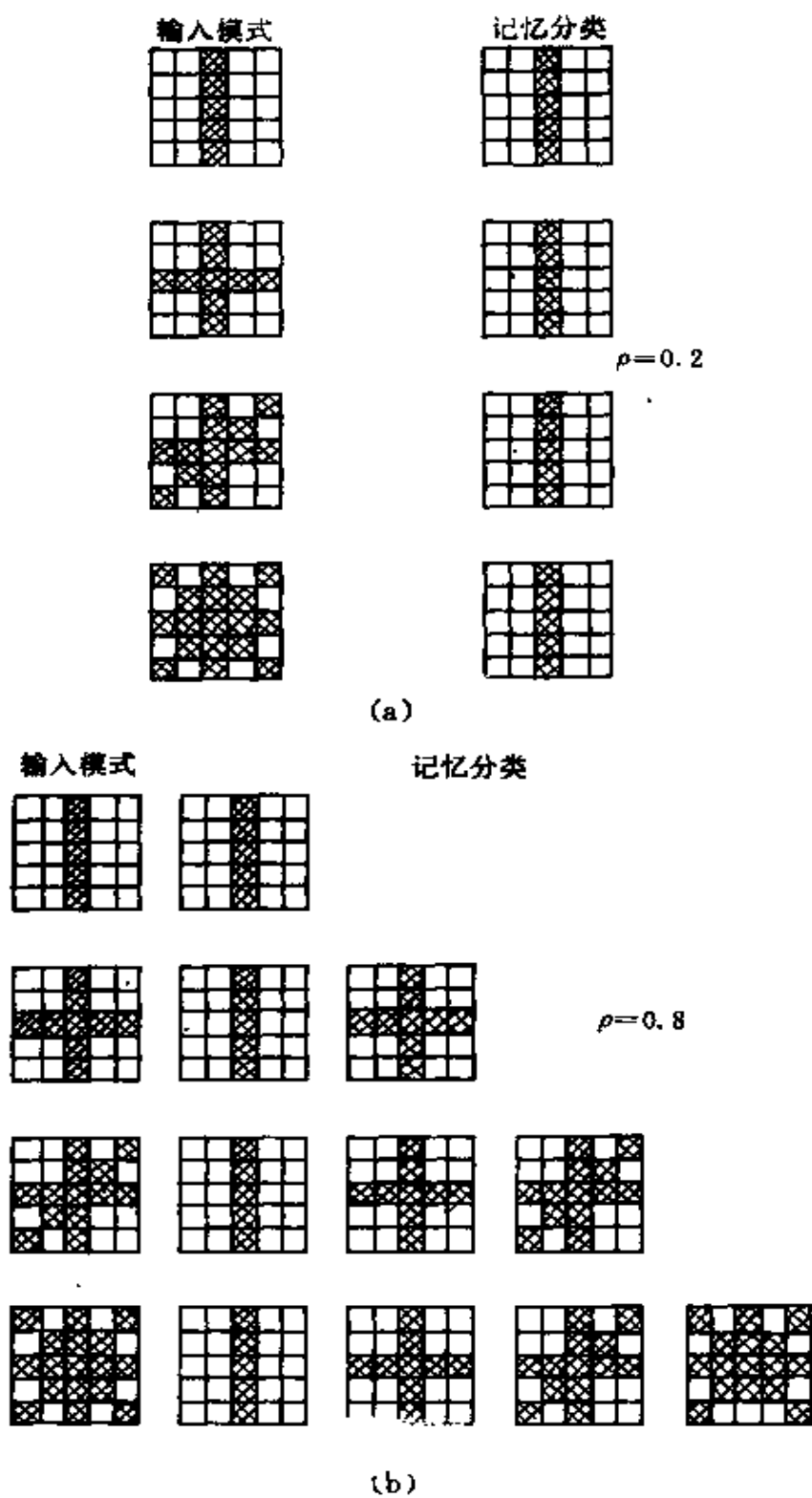


图 7.14 ρ 对 ART1 网络的分类精度影响实例

由图可以看出,当 $\rho=0.2$ 时,网络分类很粗,将四个有明显差别的输入模式归入了一类。当 $\rho=0.8$ 后,则把这四个输入模式归入了不同类别。在学习过程中,可设置动态警戒值,起始时取较小的 ρ ,作粗略分类,然后逐渐增大 ρ ,最后形成精细分类。一种极端的方法是:先取一组有明显差别的输入学习模式,并给 ρ 一个较小的初值。当发现网络分类发生错误时,则不接受这次分类,然后适当增大 ρ ,再进行一次分类,直至分类正确为止。以这时的 ρ 作为以后网络学习过程中的固定警戒值。在这种情况下,ART1 网络则由无教师示教的自组织型网络变为有教师示教网络。

(4) 连接权 $\{w_{ij}\}$ 调整的自规范化(Self-scaling)处理

在 ART1 网络的学习、工作规则中,一个非常隐蔽的、对网络分类能力有很大影响的处理方法,就是式 7.25 所表示的前馈连接权 w_{ij} 的调整方法。现将此式重写如下

$$w_{ij}(t+1) = \frac{t_{gi}(t) \cdot a_i}{0.5 + \sum_{i=1}^N t_{gi} \cdot a_i}$$

根据 ART1 网络学习的基本思想,连接权 $\{w_{ij}\}$ 的调整目的是使它朝着与输入模式 A_k 靠拢的方向修正。那么为什么不直接令 $w_{ij}(t+1)=A_k=(a_1^k, a_2^k, \dots, a_N^k)$,而要按式 7.25 调整呢?我们先分析一下式 7.25 中的分母第二项。反馈连接权 $\{T_{gi}\}$ 与输入模式 A_k 的逻辑“与”表示两个向量对应元素同时为 1 的个数总和,分子项是两个向量中第 i 个元素的逻辑“与”(0 或 1)。如果忽略分母中的常数项 0.5,式 7.25 相当于对 $t_{gi}(t) \cdot a_i$ 进行正规化处理。其结果是 T_{gj} ($j=1, 2, \dots, M$) 与 A_k 的逻辑“与”之后所得的向量中,所含元素值为 1 的个数越多,其对应的连接权 w_{ij} 的值就越小。下面通过一个简单的例子分析这种处理对网络分类性能的影响。如图 7.15 所示,ART1 网络具有六个输入神经元,两个输出神经元。现有两个如图所示的输入模式 X_1 和 X_2 。如果直接令向量 $W_1=X_1, W_2=X_2$,即:

$$\begin{aligned}
 W_1 &= (w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16}) \\
 &= (1, 0, 0, 0, 0, 0) \\
 &= X_1 \\
 W_2 &= (w_{21}, w_{22}, w_{23}, w_{24}, w_{25}, w_{26}) \\
 &= (1, 0, 0, 1, 1, 1) \\
 &= X_2
 \end{aligned}$$

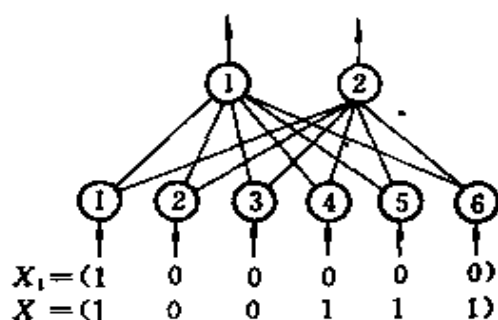


图 7.15 w_{ij} 调整举例

这两个输入模式的特点是前三个元素相同。当将输入模式 X_1 提供给网络回想时,输出神经元 1、2 将有完全

相同的响应,即 $s_1 = \sum_{i=1}^6 w_{1i}x_i$ 与 $s_2 = \sum_{i=1}^6 w_{2i}x_i$ 完全相等。这时将造成分类的错乱。如果按式 7.25 对 w_{ij} 进行调整,得:

$$W_1 = (2/3, 0, 0, 0, 0, 0)$$

$$W_2 = (2/9, 0, 0, 2/9, 2/9, 2/9)$$

当 X_1 再次提供给网络回想时, X_1 与 W_1 的内积大于 X_1 与 W_2 的内积,即只有输出神经元 1 被激活,从而避免了分类混乱。由此例可知,用式 7.25 对连接权 $\{w_{ij}\}$ 进行调整,可以避免因输入模式不完全正交,即模式之间含有相同元素排列时所产生的分类错乱。

(5) 分类的脆弱性

尽管 ART1 网络具有许多其它网络所没有的优点,但是它仅以输出层中单个神经元代表分类结果,而不是象 Hopfield 网络那样,把分类结果分散在各个神经元上来表示。所以,一旦输出层中的某个输出神经元损坏(反映在实际应用上,某个硬件输出单元失效或某个软件中输出变量溢出等),则导致该神经元所代表类别的模式信息全部消失。人们把这个神经元形象地称为“祖母细胞”(Grandmother cell),意思是指,一旦记忆祖母相貌的细胞损坏,一个人一夜之间将不认识自己的祖母。这是 ART1 网络一个很大的缺陷。

7.7 应用举例

现举一简单的例子,以加深对 ART1 网络学习、工作规则的理解。

如图 7.16 所示,设 ART1 网络有五个输入神经元和二十个输出神经元。现有两个输入模式 $A_1=(1,1,0,0,0)$ 和 $A_2=(1,0,0,0,1)$ 。要求用这两个模式训练网络。

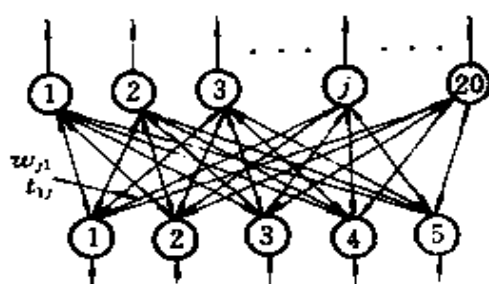


图 7.16 ART1 网络实例

$$\textcircled{1} \text{ 初始化 } \text{ 令 } w_{ij} = \frac{1}{1+N} = \frac{1}{6}$$

$$t_{ji} = 1$$

$$i=1,2,\dots,5; \quad j=1,2,\dots,20。$$

取 $\rho=0.8$

② 首先将输入模式 $A_1=(1,1,0,0,0)$ 加到网络的输入层。

③ 求获胜神经元

因为在网络的初始状态下,所有前馈连接权 $\{w_{ij}\}$ 均取相等值 $1/6$,所以各输入神经元均具有相等的输入加权和 $s_j (j=1,2,\dots,20)$ 。这时可任选一个神经元(如第 1 个)作为 X_1 的分类代表,即获胜神经元,令其输出值为 1。

④ 计算式 7.22、7.23

$$|A_1| = \sum_{i=1}^5 a_i = 2$$

$$\begin{aligned} |T_1 A_1| &= \sum_{i=1}^5 (t_{1i}) \cdot a_i = |(1,1,1,1,1) \wedge (1,1,0,0,0)| \\ &= |(1,1,0,0,0)| \\ &= 2 \end{aligned}$$

⑤ 计算式 7.24

$$\frac{|T_1 \cdot A_1|}{|A_1|} = \frac{2}{2} = 1 > 0.8, \text{接受这次识别结果;}$$

⑥ 按式 7.26、7.26 调整连接权

$$W_1 = (w_{11}, w_{12}, w_{13}, w_{14}, w_{15})$$

$$= \frac{1}{0.5 + 2} (1, 1, 0, 0, 0)$$

$$= (2/5, 2/5, 0, 0, 0)$$

$$T_1 = (t_{11}, t_{21}, t_{31}, t_{41}, t_{51})$$

$$= (1, 1, 0, 0, 0)$$

至此, A_1 已记忆在网络中。

⑦ 将输入模式 $A_2 = (1, 0, 0, 0, 1)$ 加到网络的输入层。

⑧ 求获胜神经元

$$s_1 = \sum_{i=1}^5 w_{1i} \cdot a_i = \frac{2}{5}, s_2 = \sum_{i=1}^5 w_{2i} \cdot a_i = \frac{1}{6}$$

$$s_3 = \dots = s_{20} = 1/6$$

因为 $s_1 > s_2 = s_3 = \dots = s_{20}$, 所以取神经元 1 为获胜神经元。这显然与 A_1 的识别结果相冲突。按式 7.22、7.23、7.24 计算

$$\frac{\sum_{i=1}^5 t_{1i} \cdot a_i}{\sum_{i=1}^5 a_i} = \frac{(1, 1, 0, 0, 0)^T \cdot (1, 0, 0, 0, 1)}{2} = \frac{1}{2} < 0.8$$

所以拒绝这次识别结果, 重新进行识别。因为 $s_2 = s_3 = \dots = s_{20} = 1/6$, 故可以从中任选一个神经元, 如神经元 2 作为 A_2 的分类结果, 即获胜神经元。

⑨ 按式 7.25、7.26 调整连接权

$$W_2 = (w_{21}, w_{22}, w_{23}, w_{24}, w_{25})$$

$$= \frac{1}{0.5 + 2} (1, 0, 0, 0, 1)$$

$$= (2/5, 0, 0, 0, 2/5)$$

$$T_2 = (t_{12}, t_{22}, t_{32}, t_{42}, t_{52})$$

$$= (1, 0, 0, 0, 1)$$

至此, A_2 已记忆在网络中。

以上仅介绍了自适应共振理论中 ART1 型网络。ART2 型与 ART1 型网络的主要区别是, ART2 型网络以模拟量作为输入模式, 同时在算法上做了一些相应改进, 并采用慢速学习方式。其抗干扰能力大大增强。ART3 型网络是由多个 ART1 型网络的复合阶层型网络。

第八章 自组织特征映射神经网络

8.1 自组织特征映射神经网络的基本思想

前面几章介绍的各种类型的神经网络,总是把一个网络当作一个整体,由这个整体来完成一个共同的任务——模式记忆与识别。每个识别结果由一个神经元或若干个神经元的组合来表征。实际上,生物神经网络并非这样简单。正如著名人工智能专家 Minsky 在《思想社会》(Society of mind)书中所指出的那样,如果人脑象人们所想象的那样简单,那它就不称之为人脑了。人们通过大量的生理学实验发现,人的大脑皮层中存在着许多完成特定功能的网络区域。如语言理解、视觉、控制运动等功能,都分别是由不同的神经网络区域完成的。而进一步实验又发现,在完成某一特定功能的网络区域中,不同部位的若干神经元对含有不同特征的外界刺激同时产生响应。这就是说,人脑的神经细胞(神经元)并不是与记忆模式一一对应的,而是一组或一群神经元对应一个模式。其实这一结论很容易由日常生活现象得到解释。生理学实验早已证实,人的大脑神经细胞在人生的最初 20 年中以平均每天 25000 个的惊人速度消失,同时又不断产生大量的新的神经细胞。假设神经细胞与记忆模式一一对应,则一旦某个细胞消失或损坏,则与之对应的记忆信息也随之全部消失,出现第七章所说的“祖母细胞”现象。显然这是与客观事实不符的。另一方面,如第二章所述,大脑神经网络中各神经细胞之间的信息是通过神经键传递的,传递的结果有抑制与兴奋之分。生理学基础实验表明:某一个外界信息

所引起的并不是对一个神经细胞的兴奋性刺激,而是对以某一个细胞为中心的一个区域神经细胞的兴奋刺激,并且这种刺激的强度不是均一的,有强弱之分。如图 8.1 所示,大脑神经的刺激趋势与强度呈墨西哥帽(Mexican hat)的形状。神经元受兴奋刺激的强度,以区域中心为最大,随着区域半径的增大,强度逐渐减弱,远离区域中心的神经元相反要受到抑制性作用。

早在 70 年代,一些学者就曾根据这些生理学规律研究并提出了各种模拟这些规律的人工神经网络和算法。之后于 1981 年由芬兰学者 Kohonen 提出了一个比较完整的、分类性能较好的自组织特征映射(Self-Organizing Feature Map)神经网络(简称 SOM 网络)方案。这种网络有时也称为 Kohonen 特征映射网络。

SOM 网络的结构如图 8.2 所示。由图可知 SOM 网络由输入层和竞争层组成。输入层由 N 个输入神经元组成,竞争层由 $m \times m = M$ 个输出神经元组成,且形成一个二维平面阵列。输入层各神经元与竞争层各神经元之间实现全互连接。有时,竞争层各神经元

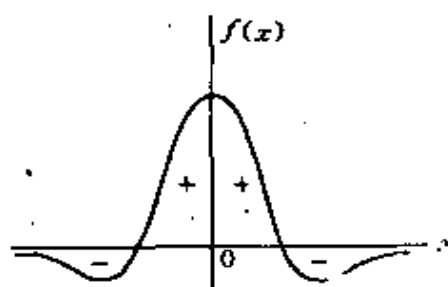


图 8.1 墨西哥帽函数

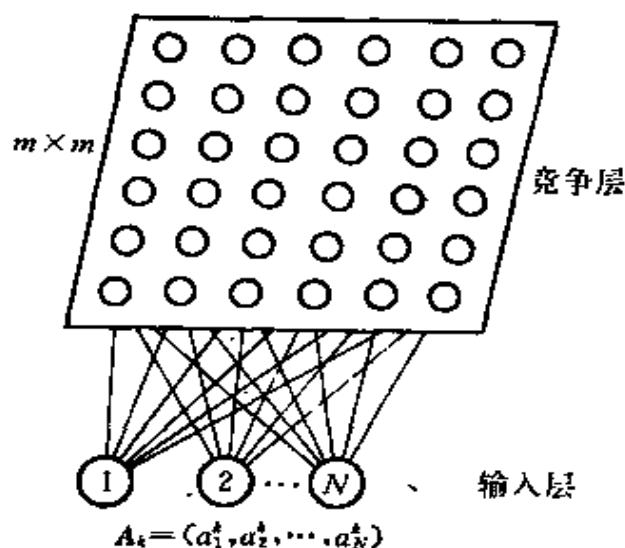


图 8.2 SOM 网络基本结构

之间也实行相邻神经元侧抑制连接。SOM 网络根据其学习规则,对输入模式进行自动分类,即在无教师示教的情况下,通过对输入

模式的反复学习,捕捉住各个输入模式中所含的模式特征,并对其
进行自组织,在竞争层将分类结果表现出来。这种表现方式与其它
类型的网络的区别是:它不是以一个神经元或者一个神经元向量
来反映分类结果,而是以若干神经元同时反映分类结果。与这若干
神经元相连接的连接权向量虽略有差别,但这些神经元的分类作
用基本上是并列的,即其中任何一个神经元都可以代表分类结果
或近似分类结果。一旦由于某种原因,某个神经元受到损害(在实
际应用中,表现为连接权溢出、计算误差超限、硬件故障等)或完全
失效,剩下的神经元仍可以保证所对应的记忆信息不会消失。从这
一点看,SOM 网络完全模拟了大脑神经细胞对外界信息刺激的反
应,克服了所谓“祖母细胞”现象。另外,这种网络之所以被称为特
征映射网络,是因为网络对输入学习模式的记忆不是一次性完成
的,而是通过反复学习,将输入模式的统计特征“溶解”到各个连接
权上的。所以这种网络具有较强的抗干扰能力。

8.2 自组织特征映射网络学习、工作规则

为清楚起见,将图 8.1 所示结构中各输入神经元与竞争层神
经元 j 的连接情况抽出,如图 8.3 所示。

设网络的输入模式为 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ $k=1, 2, \dots, p$; 竞
争层神经元向量为 $B_j = (b_{j1}, b_{j2}, \dots, b_{jm})$, $j=1, 2, \dots, m$; 其中 A_k 为
连续值, B_j 为数字量。网络连接权为 $\{w_{ij}\}$ $i=1, 2, \dots, N$; $j=1, 2,$
 \dots, M 。

网络的学习、工作规则如下:

① 初始化将网络的连接权 $\{w_{ij}\}$ 赋予 $[0, 1]$ 区间内的随机值,
确定学习率 $\eta(t)$ 的初始值 $\eta(0)$ ($0 < \eta(0) < 1$), 确定邻域 $N_x(t)$ 的
初始值 $N_x(0)$ 。所谓邻域 $N_x(t)$ 是指以步骤④确定的获胜神经元 g

为中心、包含若干神经元的区域范围。这个区域可以是任何形状,但一般来说是均匀对称的。最典型的是正方形或圆形区域。 $N_g(t)$ 的值表示在第 t 次学习过程中邻域中所包含的神经元的个数。

确定总学习次数 T 。

② 给网络提供输入模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ 。

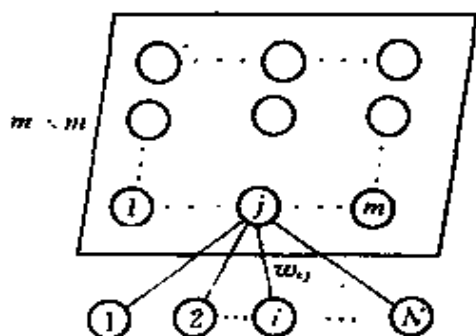


图 8.3 输入神经元与竞争层神经元 j 的连接情况

③ 计算连接权向量 $W_j = (w_{j1}, w_{j2}, \dots, w_{jN})$ 与输入模式 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$ $k=1, 2, \dots, p$ 之间的距离,即计算 Euclid 距离

$$d_j = \left[\sum_{i=1}^N (a_i^k - w_{ij})^2 \right]^{1/2} \quad (8.1)$$

$$j = 1, 2, \dots, M$$

④ 找出最小距离 d_g ,确定获胜神经元 g

$$d_g = \min[d_j] \quad j = 1, 2, \dots, M \quad (8.2)$$

⑤ 进行连接权调整。将从输入神经元到 $N_g(t)$ 范围内的所有竞争层神经元之间的连接权按式 8.3 进行修正

$$w_{ji}(t+1) = w_{ji}(t) + \eta(t) \cdot [a_i^k - w_{ji}(t)] \quad (8.3)$$

$$j \in N_g(t), i = 1, 2, \dots, N \quad 0 < \eta(t) < 1$$

式中, $\eta(t)$ 为 t 时刻的学习率。

⑥ 将下一个输入学习模式提供给网络的输入层,返回步骤③,直至 p 个学习模式全部提供一遍。

⑦ 更新学习率 $\eta(t)$ 及邻域 $N_g(t)$

$$\eta(t) = \eta_0 \left(1 - \frac{t}{T} \right) \quad (8.4)$$

式中, η_0 为学习率的初始值, t 为学习次数, T 为总的学习次数。

设竞争层某神经元 g 在二维阵列中的坐标值为 (x_g, y_g) ,则邻

域的范围是以点 $(x_k + N_k(t), y_k + N_k(t))$ 和点 $(x_k - N_k(t), y_k - N_k(t))$ 为右上角和左下角的正方形。其修正公式为

$$N_k(t) = \text{INT} \left[N_k(0) \left(1 - \frac{t}{T} \right) \right] \quad (8.5)$$

式中, $\text{INT}[x]$ 为取整符号, $N_k(0)$ 为 $N_k(t)$ 的初始值, t 和 T 与式 8.4 定义相同。

⑧ 令 $t=t+1$, 返回步骤②, 直至 $t=T$ 为止。

对 SOM 网络学习、工作规则作以下分析与解释。

(1) SOM 网络学习、工作规则的三个主要学习手段

(a) 寻找与输入模式 A_k 最接近的连接权向量 $W_k = (w_{k1}, w_{k2}, \dots, w_{kN})$ 。

(b) 将连接权向量 W_k 进一步朝与输入模式 A_k 接近的方向调整。

(c) 除调整连接权向量 W_k 外, 还调整邻域内的各个连接权向量 $W_j, j \in N_k(t)$ 。并且随着学习次数的增加, 邻域 $N_k(t)$ 逐渐缩小。

这三点体现了上一节所介绍的大脑神经网络信息传递的规律与效果。

(2) 关于连接权的调整

可以把网络的学习过程分为两个阶段。第一个阶段为粗学习与粗调整阶段。在这一阶段里, 指向各个随机方向的连接权向量朝着输入模式 A_k 的方向进行初步调整, 并大致确定各个输入模式所对应的在竞争层上的映射位置。为保证这一阶段的学习与调整速度, 式 8.3 中的学习率应保持较高值, 一般为 $\eta(t) > 0.5$ 。一旦发现各个输入模式有了相对稳定的映射位置之后, 则转入学习的第二个阶段, 即精学习与细调整阶段。在这一阶段中, 网络的学习集中在对较小范围内的连接权进行调整, 而且连接权的调整也趋于精细。为此, 学习率 $\eta(t)$ 应随着学习的进行不断减小。一般地, 第二阶段所进行的学习次数是第一阶段的 100~1000 倍。所以, 学习率的初值一般取 0.5 左右, 学习总次数在 500~10000 次之间较为合

适。另外,学习率的更新方式也可不限于式 8.4 的形式,如可采用前缓后陡的非线性函数形式等。

(3) 关于连接权 $\{w_{ij}\}$ 的初始化

网络的学习与记忆主要是通过对连接权的调整实现的。因此,连接权的初始状态对网络的学习过程有很大的影响。这里主要说明两点

(a) 连接权向量 $W_j = (w_{j1}, w_{j2}, \dots, w_{jN})$ $j=1, 2, \dots, M$ 的归一化处理。

我们知道,两个向量之间的距离主要取决于两个向量的方向,而与它们的模值大小无关。学习规则步骤③、④所要寻找的是与输入模式 A_k 方向最一致的连接权向量 W_s 。因此,在按式 8.1 计算两向量之间距离 d_j 之前,将连接权向量 W_j 按式 8.6 进行归一化处理

$$\vec{w}_j = \frac{W_j}{|W_j|} = \frac{(w_{j1}, w_{j2}, \dots, w_{jN})}{\sqrt{w_{j1}^2 + w_{j2}^2 + \dots + w_{jN}^2}} \quad (8.6)$$

$$j = 1, 2, \dots, M$$

如此处理之后,只保留了向量 W_j 中的方向因素,有效地缩短了网络的学习时间,能以较快的速度调整 W_j 的方向,使之与 A_k 的方向趋向一致。特别是当把输入模式 A_k 也进行归一化处理之后,可以直接利用 A_k 与 W_j 的内积值的大小,寻找连接权向量 W_s ,即按 $d_j = \sum_{i=1}^N w_{ij}a_i$ 的大小确定连接权向量 W_s 。其中 d_j 值大表示距离短,反之亦然。

(b) 连接权 $\{w_{ij}\}$ 初始值的确定。

学习规则步骤①规定,用 $[0, 1]$ 区间内的随机值初始化 $\{w_{ij}\}$ 。但在实际应用中,按这种方法初始化往往会出现网络学习时间过长,甚至无法收敛的现象。这里只从直观的角度来阐明这一问题。在一般情况下,输入学习模式只处于整个模式空间的有限位置,如果连接权向量 W_j 广泛地分布于各个随机方向上,则一定会有大

量的连接权向量 W_j 与输入模式 A_k 方向差异很大,甚至方向相反。这给寻找输入模式的最佳映射带来很大困难。为达到 W_j 与 A_k 的方向一致,需经过多次反复学习。同时,可被利用的连接权向量 W_j 的数量也会减少,从而大大降低网络的学习容量。连接权向量 W_j 初始状态的最理想分布是其方向与各个输入模式的方向一致。显然,在初始化时做到这一点是不现实的,因为这正是网络通过学习要达到的目的。但是可以设法做到使 W_j 的初始状态与 A_k 处于一种相互容易接近的状态。下面介绍三种常用的方法。第一种方法是,将所有连接权向量 $W_j (j=1,2,\dots,M)$ 赋予相同的初值,这样可以减少输入模式 A_k 在最初阶段对 W_j 的挑选余地,增加每一个连接权向量 W_j 被选中的机会,尽可能快地校正 W_j 与 A_k 之间的方向偏差。另一种方法是当把连接权 $\{w_{ij}\}$ 赋予 $[0,1]$ 区间内随机初值之后,在网络学习的初级阶段对提供给网络的学习模式做一些修正。具体方法是,给原学习模式 $A_k (k=1,2,\dots,p)$ 的每个元素加上一个很小的随机值,形成输入向量 $A'_k (k=1,2,\dots,p)$ 。由于 A'_k 比较容易找到与其方向大致一致的连接权向量 W_j ,所以可以加快网络最初阶段的学习速度。随着学习的进行,逐渐滤掉 A'_k 中的随机值,使 A'_k 复原为 A_k ,而这时 A_k 与 W_j 的方向已基本趋向一致,可以进入较精细的调整阶段。第三种方法是给每一个竞争层神经元增设输出阈值 θ_j ,以 $d_j + \theta_j$ 作为判断两向量距离的依据。在学习过程中,监视每个神经元被选中的次数。当发现某个神经元经常被选中时,暂时提高该神经元的阈值,进而增加其它神经元被选中的机会,提高连接权向量 W_j 的利用率,以此来促进学习的快速进行。

(4) 邻域的作用与更新

在 SOM 网络中,脑神经细胞接受外界信息的刺激产生兴奋与抑制的变化规律是通过邻域的作用来体现的。可以说邻域 $N_c(t)$ 是墨西哥帽函数的一种具体实现。邻域规定了与获胜神经元连接权向量 W_s 进行同样调整的其它神经元向量的范围。在学习的最

初阶段, $N_g(t)$ 所包含的范围较大。学习规则步骤⑦中 $N_g(t)$ 的典型值一般是竞争层神经元阵列幅度的 $1/3 \sim 1/2$, 甚至可以覆盖整个竞争层。随着学习的深入进行, $N_g(t)$ 的区域逐渐缩小, 最后达到预定的范围。为了加深对邻域作用的理解, 现把网络学习过程中各连接权向量及邻域的变化状况用图 8.4 表示出来。

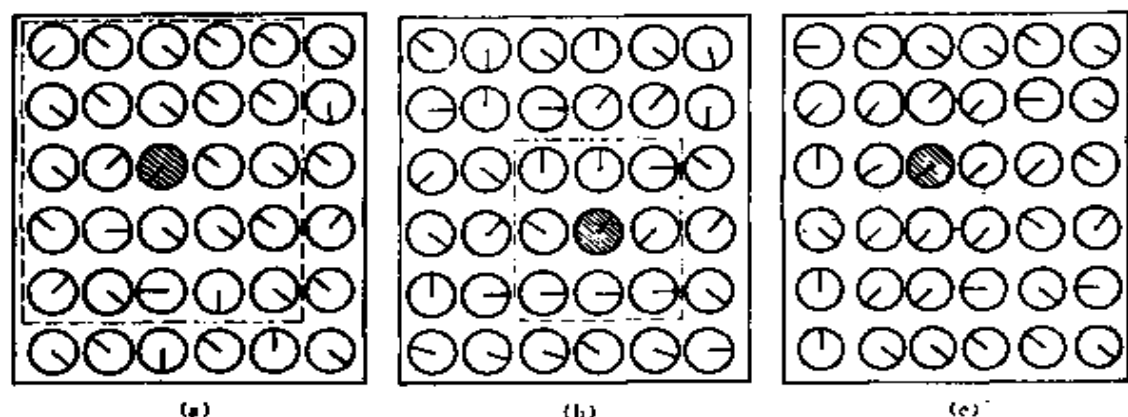


图 8.4 连接权向量及邻域的变化状况

图中的圆圈代表神经元, 圈中的斜杠表示该神经元连接权向量的方向。为清楚起见, 假设网络只在虚线所括的邻域中心形成唯一分类结果。图 8.4(a) 表示网络的初始状态, 连接权向量指向各随机方向, 邻域中包含了 25 个神经元, 其中处于中心位置的为获胜神经元。随着学习的进行, 邻域中各神经元的连接权向量朝着输入模式的方向不断调整, 逐渐形成了如图 8.4(b) 所示的一个最佳特征映射区域, 此时邻域已缩小至包含 9 个神经元。网络的最终学习结果如图 8.4(c) 所示, 连接权向量在这 9 个神经元范围内经过多次调整之后, 形成了与输入模式基本一致的方向。当再向网络提供与学习模式相同或近似的输入模式时, 邻域最后范围内的这 9 个神经元都有可能得到最大响应, 最终哪个神经元获胜, 要看哪个神经元的连接权向量与输入模式更接近。如果最近的那个神经元消失或损坏, 则在剩下的神经元中接近程度处于第二位的神经元将得到相对最大响应, 从而保证了记忆模式不会轻易地随某个神

经元的损坏而消失。

关于邻域的范围最终究竟缩小到多大范围最为合适,目前还没有明确的理论指导方法,一般在应用中凭经验确定。而且邻域的变化方式也不见得按式 8.5 所示进行。还有一点应注意到,按式 8.5 进行的邻域更新,并不是每一次学习 $N_g(t)$ 都发生变化, $N_g(t)$ 的变化具有一定的稳定性和跳跃性。邻域的形状除正方形外,也可以采用圆形或六角形,这样可以减小边界效应的影响。当然,无论采用哪种形状,区域的范围总是随着学习的进行逐渐缩小的。

(5) 网络的回想

SOM 网络经学习后按式 8.7 进行回想

$$\begin{aligned} b_g &= 1 \quad \text{当 } d_g = \min_{j=1}^M [d_j] \\ b_i &= 0 \quad i = 1, 2, \dots, M \quad i \neq g \end{aligned} \quad (8.7)$$

$$d_j = \left[\sum_{i=1}^N (a_i - w_{ji})^2 \right]^{1/2} \quad (8.8)$$

将需要分类的输入模式 $A = (a_1, a_2, \dots, a_N)$ 提供给网络的输入层,按式 8.7 寻找出竞争层中连接权向量与 A 最接近的神经元 g ,此时神经元 g 有最大的激活值 1,而其他神经元被抑制而取 0 值。神经元 g 表示对输入模式 A 的分类结果。

8.3 SOM 网络的自组织特性

SOM 网络的自组织能力表现在,经过网络的学习,竞争层各神经元的连接权向量的空间分布能够正确反映输入模式的空间概率分布。我们知道,模式的概率分布函数表示了在相应的模式空间模式分布的统计规律,即对于模式空间中的任意一点,模式的概率分布函数定义为在这一点上输入模式出现的可能性的。大小。如果

给定了由网络输入层 N 个神经元所形成的模式空间的模式概率分布函数,即预先知道了输入模式的分布情况,则通过对按给定的概率分布函数产生的输入模式的学习,网络竞争层各神经元连接权向量的空间分布密度将与给定的输入模式的概率分布趋于一致。换句话说,学习后的网络连接权向量的空间分布将符合输入模式的空间概率分布,这些连接权向量可作为这类输入模式的最佳参考向量。作为网络这一特性的逆应用,当我们不清楚一组输入模式的概率分布情况时,可以通过让网络对这组输入模式进行学习,最后由网络的连接权向量的空间分布把这组输入模式的概率分布情况表现出来。所以有时也称 SOM 网络为学习矢量量化器。为了更加清楚地说明 SOM 网络的这一自组织特性,下面举一个二维模式空间的例子。

如图 8.5(a)所示为一个由两个输入层神经元, 8×8 个竞争层神经元组成的 SOM 网络。显然,这一网络的输入模式空间为二维模式空间,输入模式在二维空间的分布情况如图 8.5(b)所示,输入模式在图中矩形内均匀分布。每次独立且随机地从中选取一点作为输入模式提供给网络学习。

首先对网络的连接权向量初始化。设 $W_j = (w_{j1}, w_{j2}) \quad j=1, 2, \dots, 64$ 。初始化的方法为,给每一个连接权赋予 $0.5 + (10\% \text{ 以内的随机值})$ 。连接权向量如图 8.6 所示。如果将竞争层各神经元所对应的二维连接权向量 (w_{j1}, w_{j2}) 决定的几何点描绘在边长为 1 的正方形之内,并将相邻神经元所对应的几何点逐个用直线连接起来,则网络的初始状态如图 8.7(a)所示。

当网络经过 1000 次学习之后(每次学习按前述方法给网络提供一个输入模式),网络连接权向量的几何点变为如图 8.7(b)所示形状。由图可以看出,网络连接权向量的均匀排列形状已大致显现出来。当网络经过 6000 次学习之后,网络连接权向量几何点的排列如图 8.7(c)所示。此时,网络连接权向量几何点向整个模式空间扩展开来,但点与点之间的距离还不完全均等。当网络经过

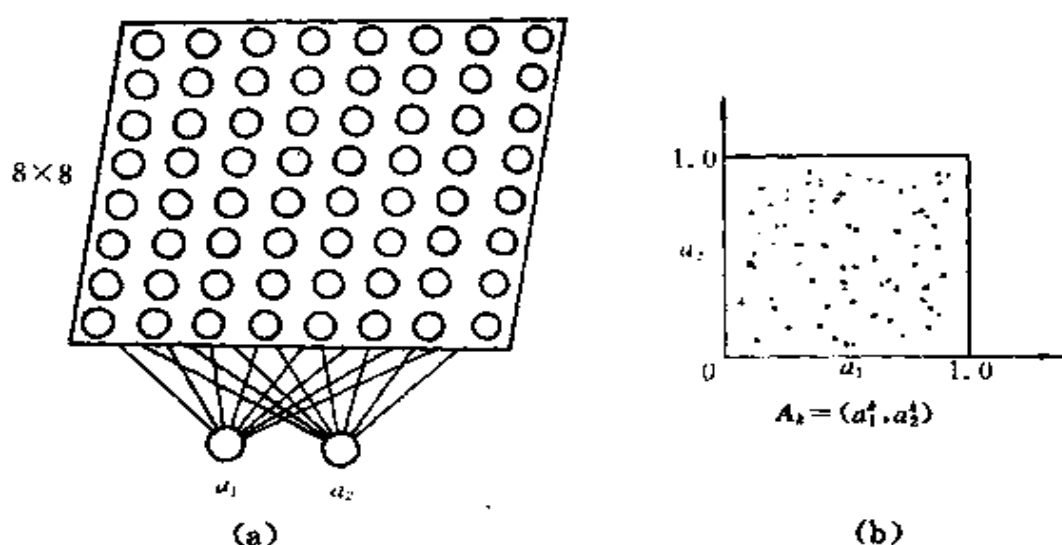


图 8.5 网络结构与输入模式分布

20000 次学习之后,网络连接权向量几何点的排列与网络竞争层各神经元的自然排列基本上趋于一致,其状况如图 8.7(d)所示。此时网络连接权向量的空间分布已反映出输入模式的二维均匀分布状况。在以上的学习过程中,学习率 $\eta(t)$ 的初值 $\eta(0)$ 为 0.2,邻域值 $N_s(t)$ 的初值 $N_s(0)$ 为 4,最后缩小至 1。

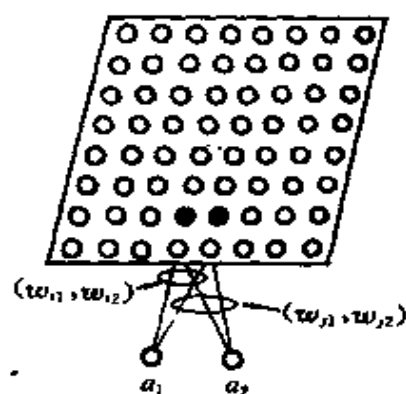


图 8.6 网络连接权向量

下面再看一下学习后的网络的回想情况。当给学习后的网络分别提供两个输入模式 $A_1 = (0.23, 0.19)$ 和 $A_2 = (0.58, 0.69)$ 后,网络的两个激活神经元如图 8.8 所示。这两个神经元所对应的连接权向量与输入模式 A_1, A_2 具有最小距离。同理,如果给网络提供二维均匀分布中的其它输入模式,则将激活与之最接近的连接权向量所对应的神经元。

还有一点值得注意的是网络的边界效应。如图 8.7(d)所示,

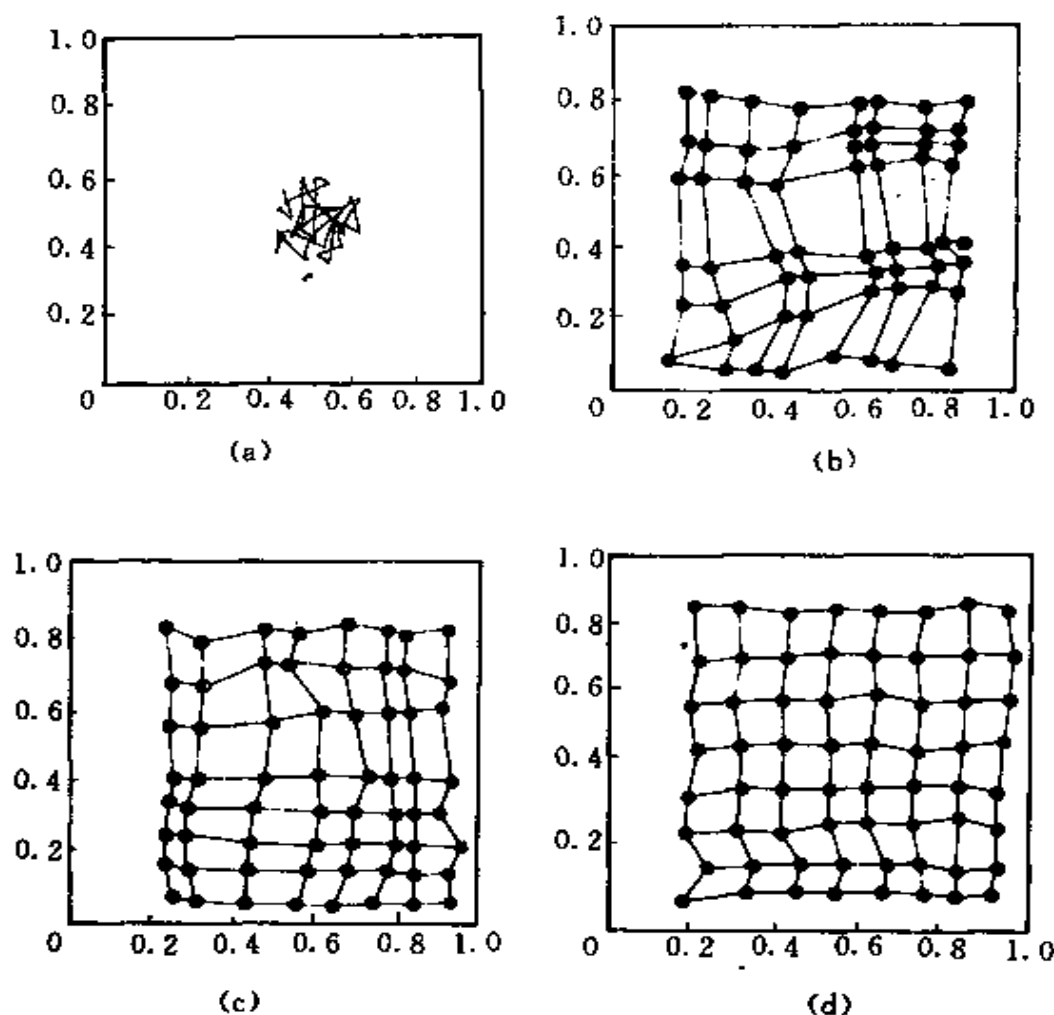


图 8.7 网络连接权几何点变化状况

网络学习之后,网络连接权向量的几何点总是离开正方形边界而偏向内侧。这是因为,在正方形边界的外侧没有输入模式提供给网络学习,处于边界附近的神经元得不到向外扩展的校正作用。

8.4 网络的分类精度分析

SOM 网络的分类精度取决于提供给网络学习的输入模式的概率分布函数。当学习模式空间某一区域内的模式多于另一区域

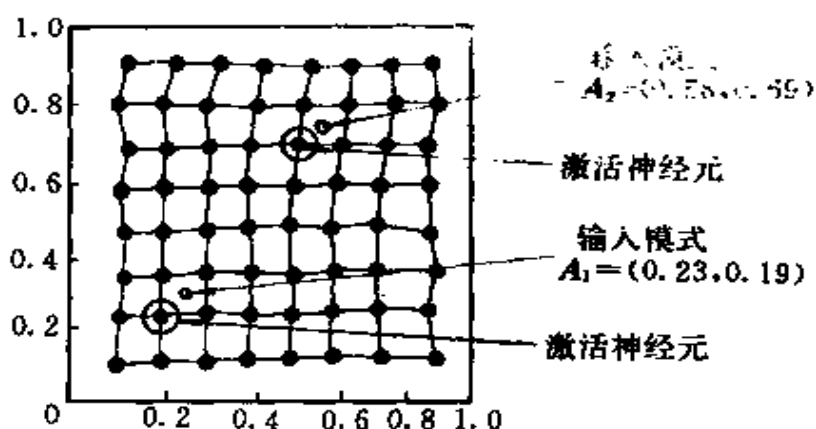


图 8.8 网络回想情况

内的模式时,网络学习后连接权向量最终形成的空间分布也将呈现疏密不均的现象。也就是说,非均匀分布的输入学习模式将产生非均匀分布的网络连接权向量空间。在分布密度高的区域,网络的分类精度高;在分布密度低的区域,网络的分类精度也低。下面举例说明。如图 8.5(a)所示的 SOM 网络结构,其学习模式按图 8.9(a)所示方案选取。在二维输入模式空间内,从虚线框内占整个学习模式集合 42% 的模式,另外 58% 学习模式从剩余的大范围内提供。显然,虚线框内学习模式的密度要高于虚线框之外的区域,这样当网络经过 20000 次学习之后($\eta(0)=0.2, N_s(0)=2$),网络连接权向量几何点如图 8.9(b)所示。显而易见,网络在回想时,如果输入模式落在虚线框的区域内,则输入模式之间细小的差别将对应不同的连接权向量,即激活不同的神经元,这些神经元将代表不同的分类结果。如果输入模式落入虚线框以外的区域,则在某些情况下,虽然两个或两个以上的输入模式之间存在差距,但它们将激活同一个神经元,网络的分类精度比虚线框内区域大大降低。

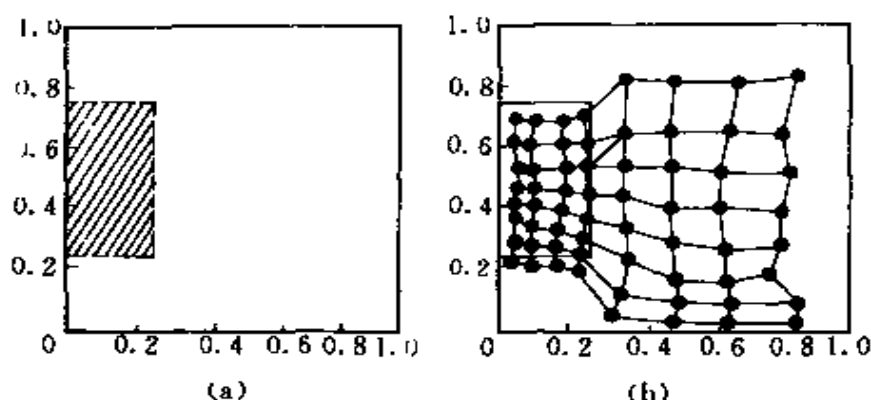


图 8.9 网络输入模式非均匀的学习结果

8.5 SOM 网络的局限性

SOM 网络的最大局限性是,当学习模式较少时,网络的分类效果取决于输入模式的先后次序;且网络连接权向量的初始状态对网络的收敛性能有很大的影响。例如,同样是如图 8.5(a)所示的网络,当提供给网络学习且符合均匀分布的学习模式数量较少时,网络连接权向量的几何点有可能成为如图 8.10 所示形状。在这种状况下,网络的学习是失败的。因为当输入模式落在图中交叉区域附近时,网络将无法对输入模式进行正确分类。

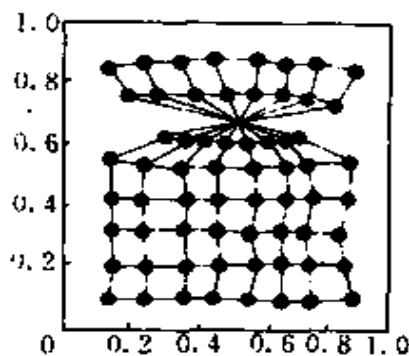


图 8.10 网络学习失败的情况

另外,与 ART1 网络不同,SOM 网络不具有生物神经网络的稳定性,即对新学习模式的记忆将影响网络原有的模式记忆。

8.6 SOM 网络的有教师示教学习规则

SOM 网络也可以用有教师示教的方式进行学习。在这种情况下,每一个输入学习模式 A_i 应归属的类别是预先知道的。当模式 A_i 提供给网络的输入层后,按式 8.2 选出获胜神经元 g 。如果神经元 g 是 A_i 的恰当分类,则将神经元 g 所对应的连接权向量朝 A_i 靠拢(相当于奖励);否则,将神经元 g 所对应的连接权向量朝离开 A_i 的方向引导(相当于惩罚)。其奖惩过程可表示为

$$w_{gi}(t+1) = \begin{cases} w_{gi}(t) + \eta(t)[a_i - w_{gi}(t)] & g \text{ 为恰当分类} \\ w_{gi}(t) - \eta(t)[a_i - w_{gi}(t)] & g \text{ 不是恰当分类} \end{cases} \quad (8.9)$$
$$j = 1, 2, \dots, N$$

有教师示教学习比无教师示教学习需要较少的学习时间,同时也有较高的分类精度。有时为了提高 SOM 网络的分类精度,对于一个已用无教师示教方式学习完毕的网络,追加一些已知分类结果的学习模式,然后让网络按有教师示教的方式学习。这样做对于提高网络分类精度有明显效果。

8.7 SOM 网络应用举例——语音打字机

SOM 网络最成功的应用例子,就是由 Kohonen 本人开发的芬兰语(Kohonen 本人的母语)语音打字系统。他把自组织特征映射神经网络巧妙地应用于语音识别这一古老而又复杂的模式识别

问题,取得了令世人瞩目的成绩。这一应用的基本原理是,将大量不同的音素(即区别声音的最基本单位)提供给含有 SOM 网络的语音识别系统,在网络充分学习之后,当向系统输入发音时,系统会自动识别声音,并将其转换为文字通过打印机输出。该系统对输入声音的正确识别率高达 90% 以上。

从信号处理的观点来看,语音波形是一种不规则且十分复杂的实变模拟信号。同一音素,不同的发音者具有不同的波形与强度。即使是同一发音者,其发音音素也会因语句中前后单词关系的不同而不同。在某些情况下,即使音素之间有差别,但往往因其功率谱信号重迭而给音素的识别造成困难。因此,用传统的方法进行语音识别需要十分复杂的技术。而应用 SOM 网络,可使语音识别的处理过程大大减化,比传统技术明显优越。但是必须指出,仅仅靠人工神经网络一项技术是无法完成如此复杂的模式识别任务的。正如 Kohonen 早就指出的那样,神经网络并不是万能的钥匙,它不能解决信号、数据处理的所有问题。Kononen 开发的语音处理系统是以 SOM 网络为核心,包括前、后处理子系统在内的一个复合系统,其结构如图 8.11 所示。由图可以看出,SOM 网络承担着系统中音素的分类任务,即对语音频谱信号进行矢量化处理。

由麦克风输入的信号首先通过截止频率为 5.3kHz 的滤波器,滤除语音中所含的噪声,然后以 13.03kHz 的采样频率,用 12 位 A/D 将语音模拟信号转换为数字信号。接着将这一数字信号进行 256 点的 FFT(Fast Fourier Transform——快速傅里叶变换)处理,得到分辨率为 9.83ms 的语音信号的频谱。将这些频谱进行平滑处理并取对数后,在 200Hz~5kHz 范围内将其分成 15 个元素,这 15 个元素所组成的向量,代表了输入语音信号的模式。SOM 网络将这些语音模式进行矢量化处理,即寻找这些音素所代表的标准值的最佳映射,尽管这些音素可能已偏离其标准值,但它仍然是围绕在标准值附近的值。由于 SOM 网络的特征映射功能,可以找到与输入模式最接近的分类结果(即最邻近分类),从而得

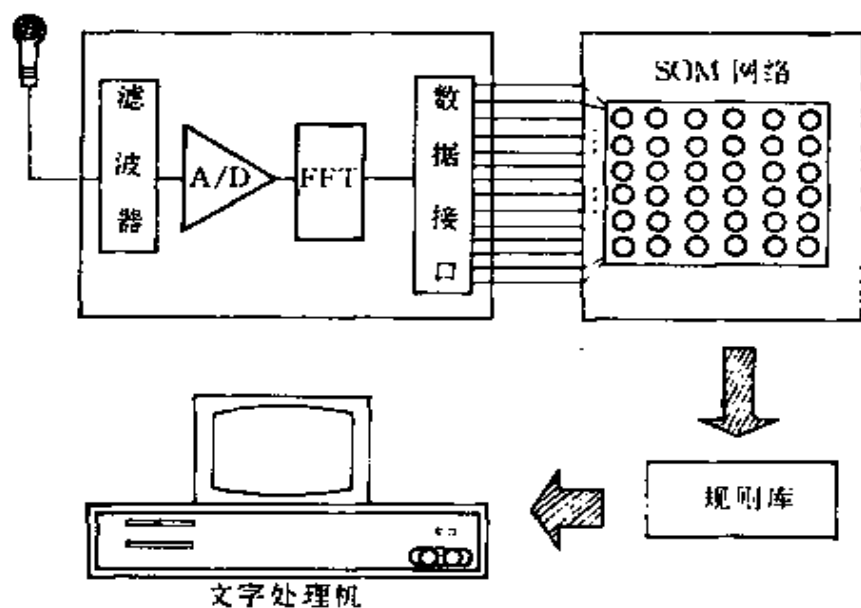


图 8.11 语音处理系统框图

到输入语音信号的标准值。Kohonen 所使用的是有 15 个输入神经元, 96 个输出神经元的双层 SOM 网络。他曾选用了 50 个实验语音(共含有 21 个不同音素)对网络进行训练, 训练结果——竞争层的分类状况如图 8.12 所示。

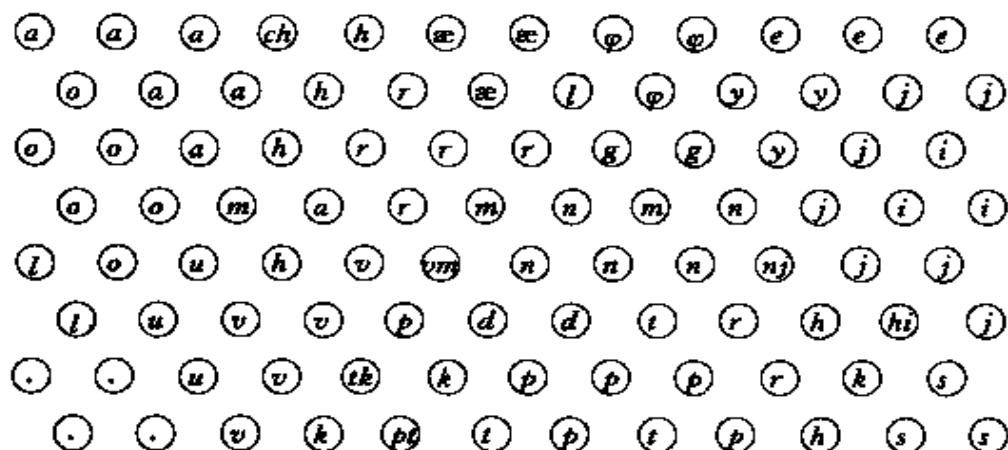


图 8.12 音素特征映射

当给系统输入一个单词的发音, 如“hummpila”后, 在 SOM 网络

的竞争层可以映射出这个单词所顺序对应的音素。如果把这一识别过程描绘下来,则可以得到如图 8.13 所示的轨迹。所得到的识别结果还必须经过最后处理,即将音素序列传入规则库进行语音规则分析,对识别结果进行确认与修正。在这个规则库中,大约存贮了 15000~20000 条规则。经确认和修正后的结果,输入文字处理机,将其显示或打印出来。这一系统的硬件是由一台 IBM PC/AT 计算机和两个信号处理单元组成。信号处理单元为 TMS32010,其中一个负责输入信号的前期处理,另一个用于特征映射后的语音分类。信号的后期处理由计算机完成。系统可将按正常速度发音的语音识别、分辨出来。

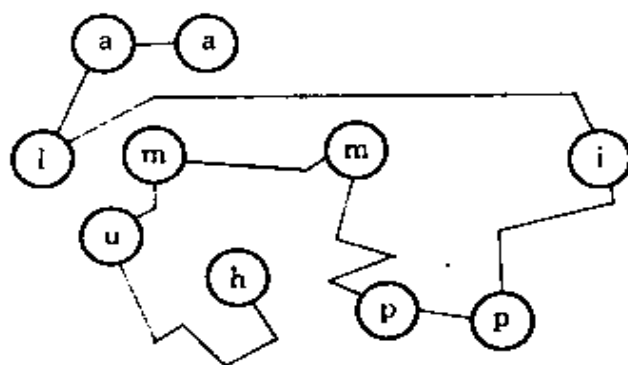


图 8.13 芬兰语“humppila”的语音特征映射轨迹

第九章 对向传播神经网络

9.1 对向传播神经网络的基本思想

对向传播(Counter-Propagation)神经网络(简称 CP 网络),是将 Kohonen 特征映射网络与 Grossberg 基本竞争型网络巧妙结合,发挥各自特长的一种新型特征映射网络。这一网络是美国神经计算机专家 Robert Hecht-Nielsen 于 1987 年提出的^[27]。这一网络被有效地应用于模式分类、函数近似、统计分析和数据压缩等领域。网络结构如图 9.1 所示。网络分输入、竞争、输出三层。输入

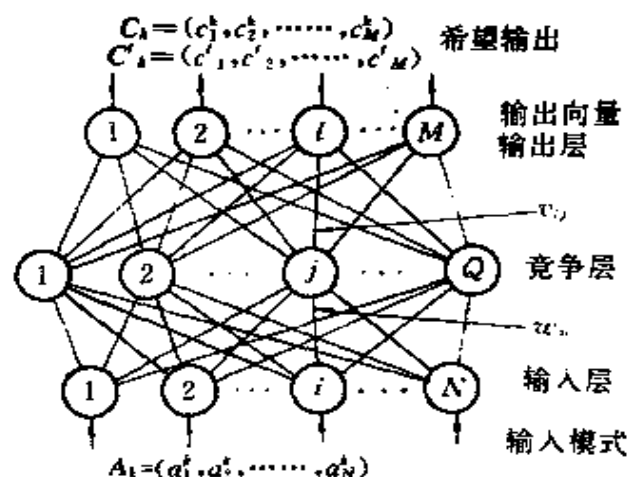


图 9.1 CP 网络结构

层与竞争层构成 SOM 网络,竞争与输出层构成基本竞争型网络。网络从整体上看属于有教师示教型网络,而由输入层和竞争层构成的 SOM 网络又是一种典型的无教师示教型网络。因此,这一网

络既汲取了无教师示教型网络分类灵活、算法简练的优点,又采纳了有教师示教型网络分类精细、准确的长处,使两种不同类型的网络有机地结合起来。由输入层至竞争层,网络按 SOM 学习规则产生竞争层的获胜神经元,并按这一规则调整相应的输入层至竞争层的连接权;由竞争层至输出层,网络按基本竞争型网络学习规则,得到各输出神经元的实际输出值,并按有教师示教的误差校正方法,调整由竞争层至输出层的连接权。经过这样反复地学习,可以将任意输入模式映射为输出模式。由 CP 网络的这一基本思想我们可以发现,处于网络中间位置的竞争层获胜神经元及与其相关的连接权向量,既反映了输入模式的统计特征(由 SOM 网络的特性决定),又反映了输出模式的统计特征(由误差校正原理决定)。因此,可以认为输入、输出模式通过竞争层实现了相互映射,即网络具有双向记忆功能。如果输入、输出选取相同的模式(即输入模式等于希望输出模式)对网络进行训练,则由输入模式至竞争层的映射可以认为是对输入模式的压缩;而由竞争层至输出层的映射可以认为是对输入模式的复原。利用这一特性,可以有效地解决图象处理及通讯中的数据压缩与复原问题,并能得到较高的压缩比。

9.2 CP 网络学习、工作规则

如图 9.1 所示的 CP 网络结构。设网络输入层有 N 个神经元, P 个连续值输入模式为 $A_k = (a_1^k, a_2^k, \dots, a_N^k)$; 竞争层有 Q 个神经元,所对应的二值输出向量为 $B_k = (b_1^k, b_2^k, \dots, b_Q^k)$; 输出层有 M 个神经元,其连续值输出向量为 $C'_k = (c_1^k, c_2^k, \dots, c_M^k)$; 希望输出向量为 $C_k = (c_1^k, c_2^k, \dots, c_M^k)$ 。以上 $k=1, 2, \dots, p$ 。由输入层至竞争层连接权向量为 $W_j = (w_{j1}, w_{j2}, \dots, w_{jN})$ $j=1, 2, \dots, Q$; 由竞争层至输出

层连接权向量为 $V_l = (v_{l1}, v_{l2}, \dots, v_{lQ}) \quad l=1, 2, \dots, M$ 。

网络学习、工作规则如下。

① 初始化 将连接权向量 $W_j = (w_{j1}, w_{j2}, \dots, w_{jN}) \quad j=1, 2, \dots, Q$ 和 $V_l = (v_{l1}, v_{l2}, \dots, v_{lQ}) \quad l=1, 2, \dots, M$ 赋予 $[0, 1]$ 区域间的随机值。

将所有输入模式 $A_k (k=1, 2, \dots, p)$ 按式 9.1 进行归一化处理

$$a_i^k = \frac{a_i^k}{\|A_k\|} \quad i=1, 2, \dots, N \quad (9.1)$$

$$\|A_k\| = \sqrt{\sum_{i=1}^N a_i^2} \quad (9.2)$$

② 将第 k 个输入模式 A_k 提供给网络的输入层。

③ 将连接权向量 $W_j (j=1, 2, \dots, Q)$ 按式 9.3 进行归一化处理

$$w_{ji} = \frac{w_{ji}}{\|w_j\|} \quad i=1, 2, \dots, N \quad (9.3)$$

$$\|w_j\| = \sqrt{\sum_{i=1}^N w_{ji}^2} \quad (9.4)$$

④ 求每一个竞争层神经元加权输入和 $s_j (j=1, 2, \dots, Q)$

$$s_j = \sum_{i=1}^N a_i^k w_{ji} \quad (9.5)$$

⑤ 求连接权向量 $W_j (j=1, 2, \dots, Q)$ 中与 A_k 距离最近的向量 W_g

$$W_g = \max_{j=1}^Q \sum_{i=1}^N a_i^k \cdot w_{ji} = \max_{j=1}^Q s_j \quad (9.6)$$

将神经元 g 的输出 b_g 置为 1, 其余竞争层神经元输出置为 0

$$b_j = \begin{cases} 1 & j = g \\ 0 & j \neq g \end{cases} \quad (9.7)$$

⑥ 将连接权向量 W_g 按式 9.8 进行调整

$$w_{gi}(t+1) = w_{gi}(t) + \alpha[a_i^k - w_{gi}(t)] \quad (9.8)$$

$$i = 1, 2, \dots, N$$

式中, α 为学习率 ($0 < \alpha < 1$)。

⑦ 将连接权向量 W_k 重新归一化

$$w_{gi} = \frac{w_{gi}}{\|W_k\|} \quad i = 1, 2, \dots, N \quad (9.9)$$

⑧ 按式 9.10 调整由竞争层至输出层的连接向量 V_l ($l=1, 2, \dots, M$)

$$v_{lj}(t+1) = v_{lj}(t) + \beta \cdot b_j \cdot (c_l - c'_l) \quad (9.10)$$

$$l = 1, 2, \dots, M; j = 1, 2, \dots, Q$$

由式 9.7 可将式 9.10 简化为

$$v_{lg}(t+1) = v_{lg}(t) + \beta \cdot (c_l - c'_l) \quad (9.11)$$

$$l = 1, 2, \dots, M$$

式中, β 为学习率 ($0 < \beta < 1$)。

式 9.11 表示只需调整竞争层获胜神经元 g 到各输出层神经元的连接权向量 V_k 。其它连接权向量保持不变。

⑨ 求输出层各神经元的加权输入, 并将其作为输出神经元的实际输出值 c'_l ($l=1, 2, \dots, M$)

$$c'_l = \sum_{j=1}^Q b_j \cdot v_{lj} \quad l = 1, 2, \dots, M \quad (9.12)$$

由式 9.7 可将式 9.12 简化为

$$c'_l = v_{lg} \quad l = 1, 2, \dots, M \quad (9.13)$$

⑩ 返回步骤②, 直至 P 个输入模式全部提供给网络。

⑪ 令 $t=t+1$, 将输入模式 A_t 重新提供给网络学习。直至 $t=T$ 。一般情况下, $500 \leq T \leq 10,000$ 。 T 为总学习次数。

现对学习、工作规则作以下分析与解释。

(1) 关于归一化处理

在学习、工作规则中, 需对输入模式 A_i 和连接权向量 W_i 进行归一化处理。所谓向量的归一化, 就是使向量 X 满足式 9.14

$$\|X\| = \sqrt{\left(\sum_{i=1}^N x_i^2\right)} = 1 \quad (9.14)$$

在这里,规一化的目的,是为了剔除输入模式 A_k 与连接权向量 W_j 中模值大小的因素,而只保留方向因素。其原因与第八章介绍的 SOM 网络学习规则时解释的一样。网络所要寻找的是与输入模式方向最一致的连接权向量 W_x ,而并不考虑它们模值的大小。由下面的分析我们还将知道,正是因为对 A_k 和 W_j 进行了规一化处理,才可以利用式 9.6 求得与 A_k 最接近的连接权向量 W_x 。

(2) 关于 W_x 的寻找方法

在 SOM 网络的学习、工作规则中,是按式 8.1 直接求得输入模式 A_k 与连接权向量 W_j 之间的距离,经过比较求得距离最短,即与 A_k 最接近的连接权向量 W_x 。而在 CP 网络学习、工作规则中, A_k 与 W_j 之间的距离是用式 9.6 表示的。实际上,这两种方法本质上是等价的。我们知道,两个向量之间的距离既可以用两个向量之间的垂直距离(即按式 8.2 表示),也可以用两个向量之间的夹角表示。式 9.6 实际上是 A_k 与 W_j 的内积(也称点积)表达式

$$S_j = A_k \cdot W_j \quad (9.15)$$

将式 9.15 进一步展开,得到式 9.16

$$S_j = \|A_k\| \|W_j\| \cos\theta_j \quad (9.16)$$

式中 θ_j 是 A_k 与 W_j 之间的夹角。因为已对 A_k 、 W_j 进行了规一化处理,即 $\|A_k\| = \|W_j\| = 1$,所以,式 9.16 可简化为 $S_j = \cos\theta_j$,于是,寻找竞争层获胜神经元则可转化为式 9.17、式 9.18

$$\cos\theta_x = \max_{j=1}^M (\cos\theta_j) \quad (9.17)$$

$$\theta_x = \min_{j=1}^M \theta_j \quad (9.18)$$

(3) 关于连接权的调整

连接权的调整是网络学习的关键。同 SOM 网络连接权的调整原理一样,CP 网络输入层至竞争层连接权的调整就是使获胜神

经元所对应的连接权向量朝着与输入模式 A_i 方向一致的目标变化。式 9.8 的几何表示如图 9.2 所示。当把 P 个按均匀分布的学习模式提供给网络学习时,网络将根据学习规则对连接权向量 W_j 进行自组织,即按式 9.8 进行连接权调整,使各个连接权向量在网络学习结束时分别处于各个相似类别输入学习模式空间的分布中心,即此时的连接权向量

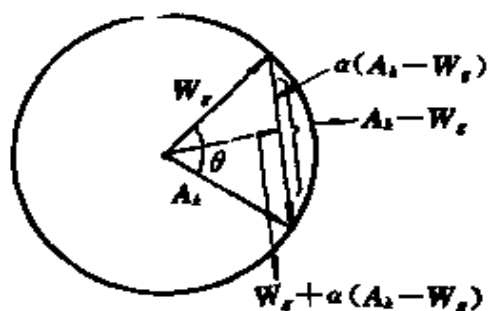


图 9.2 连接权调整的几何表示

代表了它周围学习模式的平均方向。如果设 CP 网络有三个输入神经元,即输入模式 A_i 和连接权向量 W_j 都是三维向量,则连接权向量与输入模式之间的相对位置的演变过程可用图 9.3 表示。这时经归一化后的模式空间为一个单位球体(如果输入模式超过三维,模式空间为一个单位超球体)。图 9.3(a)表示输入模式的空间分布,如图所示,大致有五组聚类模式群。图 9.3(b)表示输入模式与连接权向量空间分布相对位置的初始状态,它们之间存在着一定距离。图 9.3(c)表示网络学习结束后,输入模式与连接权向量的相对位置。各个连接权分别落入各个聚类模式群的中心位置。图 9.3(d)表示某个连接权向量朝某个聚类模式群调整的动态过程。此图是图 9.2 的一种立体表示。

(4) 网络各层神经元状态激活规律

CP 网络各层神经元有着不同的状态激活规律。其中,输入层是由连续值输入模式 A_i 激活的,其激活值也是连续值;竞争层神经元是通过网络竞争获胜神经元而被激活的。整个竞争层只有一个神经元被激活,激活值为 1,其余神经元输出均为 0;输出层神经元的激活值是由竞争层各神经元的加权和决定的,为连续激活值。图 9.4 是 CP 网络各层神经元激活的例子。

(5) 连接权调整的单一性

CP 网络连接权的调整量与同等规模其它类型网络相比是最

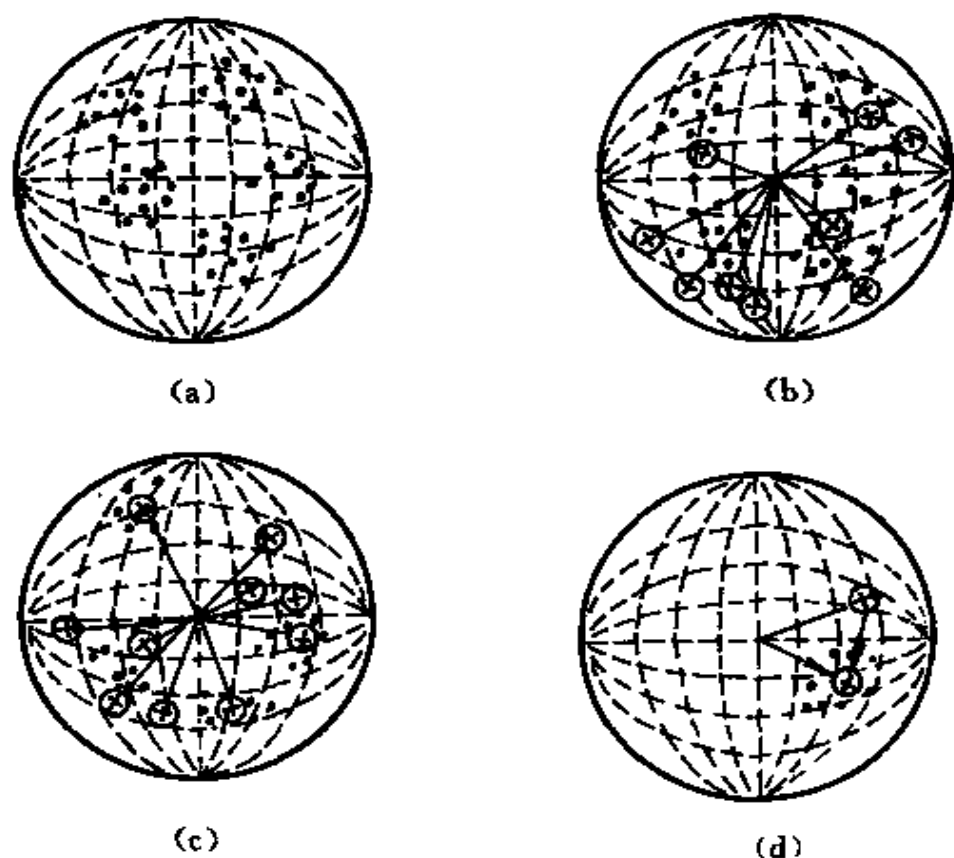


图 9.3 连接权的演变过程

少的。它既不象 BP 网络那样在每一个学习循环阶段内对所有连接权向量进行全面调整,也不象 SOM 网络那样每次对一个区域内的神经元所对应的连接权向量进行调整。CP 网络每次只对一个有关的神经元所对应的连接向量进行调整,即由输入层至竞争层连接权的调整只发生在竞争层获胜神经元所对应的连接权向量上。由此可知,CP 网络比 BP 和 SOM 网络有更短的学习时间。图 9.5 所示的是 CP 网络连接权调整的单一性例子。

(6) 网络的回想过程

CP 网络学习结束后,按以下步骤进行网络回想

- (a) 将输入模式 A 提供给网络的输入层。
- (b) 按式 9.19 求出竞争层的获胜神经元 g

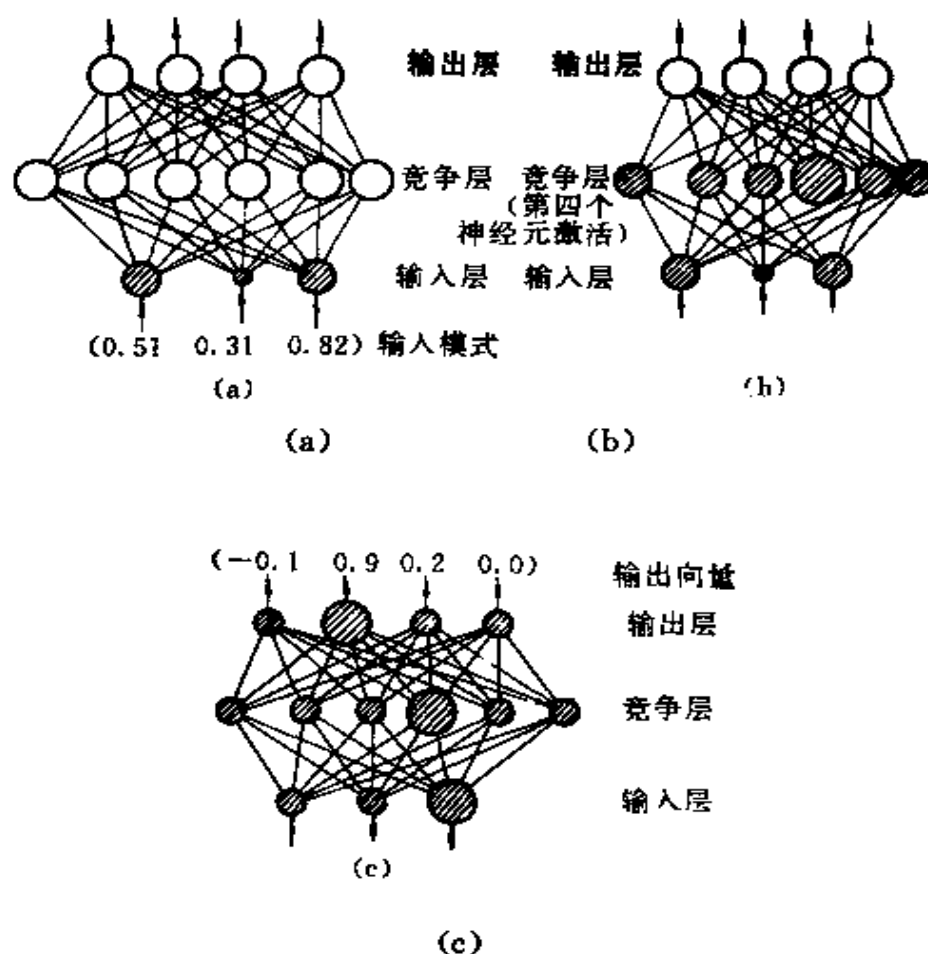


图 9.4 CP 网络各层神经元激活规律举例

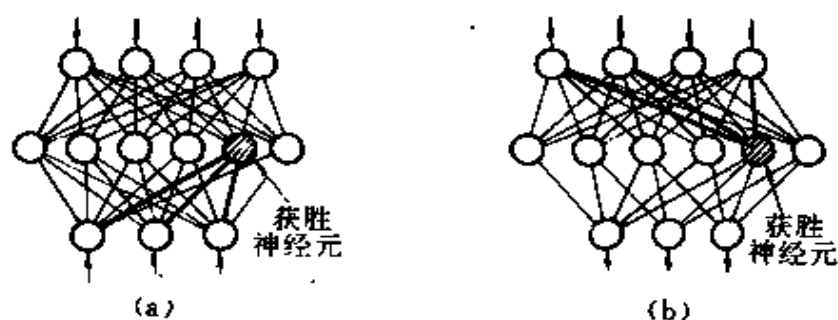


图 9.5 CP 网络连接权调整的单一性例子

$$b_r = \max_{j=1}^Q \left(\sum_{i=1}^N w_{ji} a_i \right) \quad (9.19)$$

Q ——竞争层神经元个数； N ——输入层神经元个数。

(c) 令 $b_g = 1, b_j = 0 \quad j \neq g$, 按式 9.20 求得输出层各神经元的输出值

$$c_j = v_{jk} \cdot b_g \quad j = 1, 2, \dots, M \quad (9.20)$$

M ——输出神经元个数。

由此产生输出模式 $C = (c_1, c_2, \dots, c_M)$, 从而得到输入模式 A 的分类结果。

9.3 网络应用举例

这里举一个简单、有趣的例子, 说明 CP 网络的应用方法。如图 9.6 所示是一个由两个输入神经元、18 个竞争层神经元和 5 个

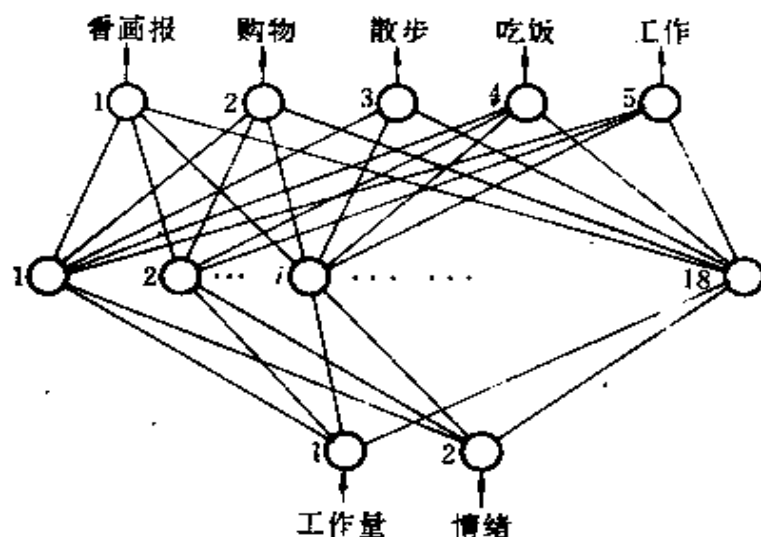


图 9.6 星期日午后活动安排辅助决策系统

输出神经元组成的辅助决策系统。系统的任务是在已知一个人本星期应完成的工作量和此人当时的思想情绪状况的情况下,对此人星期日下午的活动安排提出建议。图 9.7 是这一系统的学习模式集合,它表示了在相应工作量和思想情绪下的最佳活动安排。

工作量		情 绪		活动安排	希望输出模式
没有	(0.0)	低	(0.0)	看画报	(1.0,0,0,0)
有一些	(0.5)	低	(0.5)	看画报	(1.0,0,0,0)
没有	(0.0)	一般	(0.5)	购物	(0,1,0,0,0)
很多	(1.0)	高	(1.0)	公园散步	(0,0,1,0,0)
有一些	(0.5)	高	(1.0)	吃饭	(0,0,0,1,0)
很多	(1.0)	一般	(0.5)	工作	(0,0,0,0,1)

图 9.7 网络学习模式集合

如图所示,把工作量分为三个档次,即没有、有一些和很多,所对应的量化值分别为 0.0、0.5 和 1.0;把思想情绪也分为三个水平,即低、一般和高,所对应的量化值分别为 0.0、0.5 和 1.0。可选择的活动有 5 个:在家里看画报、去商场购物、到公园散步、与朋友一起吃饭和做工作。工作量与思想情绪共有 6 种组合,这 6 种组合分别对应各自的最佳活动选择。当把这组学习模式提供给系统进行充分学习之后,网络具有了一种所谓“内插”功能,即当给网络输入一对在(0,1)区间内的反映工作量和思想情绪的量化值后,网络将自动根据原有记忆,找出对应于这对量化值的最佳活动选择,以输出模式的形式提供给用户作为决策参数。图 9.8 所示的是把思想情绪固定在量化值 1 的水平上,当工作量的量化值在 0.5~1.0 之间变化时系统相应的输出。由图可知,系统从推荐与朋友一起吃饭,变化到去公园散步所对应的工作量的量化值区间为 0.8 至 1.0。

事实上,不但 CP 网络具有这种所谓“内插”功能,BP、SOM 等

网络也具有这种功能。从模式识别的角度上讲,这些网络具有对输入模式进行最邻近分类的功能。

9.4 网络的改进与完善

对 CP 网络的学习与工作规则,人们陆续提出了一些改进方案与设想,目的在于使网络的性能更加完善。

(1) 分阶段学习方案

这一学习方案的基本思想是,把网络的学习分为两个阶段进行。第一阶段只对由输入层至竞争层的网络连接权进行调整,即学习只发生在网络这两层之间。当网络在这一阶段进行充分学习之后,才转入第二阶段学习,即只对由竞争层至输出层的网络连接权进行调整。这一改进方案的理论依据是,在网络的前部连接权没有充分学习并对输入模式形成正确的特征映射前,网络后部的连接权调整不具有任何实际意义。这种改进方案的优点是,使整个网络的学习思路更加清晰。另外,还可取动态变化的学习率 α 和 β ,使其随学习进程的深入,由大至小变化。这样做,既可以加快网络在初始阶段的学习速度,又可以在网络学习的后期阶段使连接权的调整更加精细。

工作量	情绪	推荐活动安排
0.5	1.0	吃饭
0.6	1.0	吃饭
0.7	1.0	吃饭
0.8	1.0	吃饭
0.9	1.0	公园散步
1.0	1.0	公园散步

图 9.8 学习后的系统应用实例

(2) 具有线性叠加映射能力的网络回想方案

由 CP 网络的学习、工作规则可知,网络无论在学习还是在回想时,只允许竞争层出现一个获胜神经元,而当网络学习结束,进行网络回想时,允许竞争层出现两个获胜神经元,其具体做法是,

给网络提供一输入模式后,通过网络竞争选取处于前两位的输入加权后的神经元作为两个获胜神经元。之后对与这两个获胜神经元相关的连接权向量进行调整,并产生输出模式。下面通过一个具体例子说明这一方案的实施方法。

图 9.9(a)所示是一 CP 网络的三对输入与希望输出模式,将这三对模式提供给如图 9.9(b)所示的网络进行学习。网络学习结束后,把如图 9.9(c)左边所示的三个输入模式提供给网络进行回

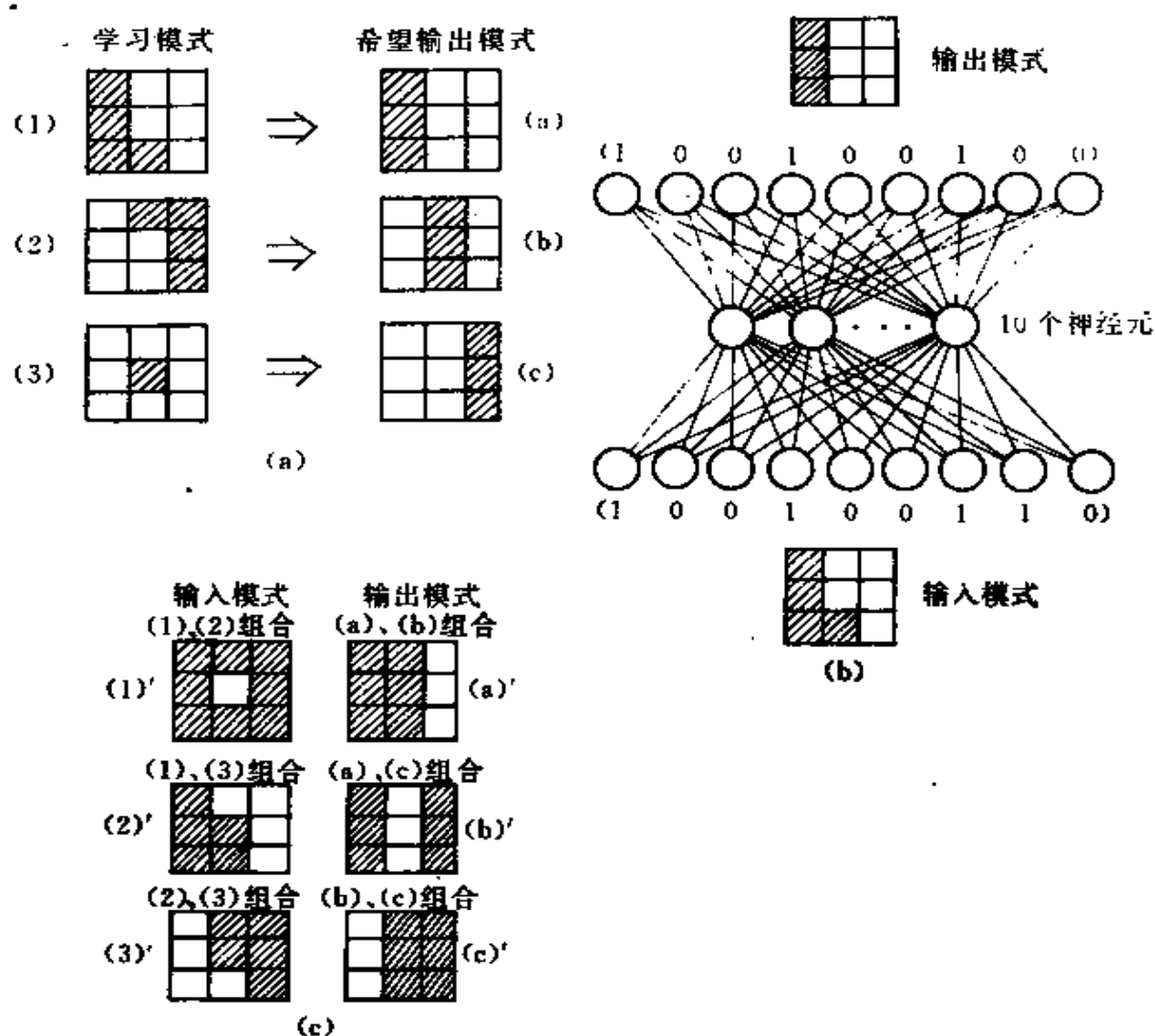


图 9.9 改进回想方案举例

想,其结果如图 9.9(c)右边所示。仔细观察这三个输入模式与网络回想输出模式可以发现,三个输入模式分别是图 9.9(a)所示的三个学习模式的两两线性组合,而三个回想输出模式正是图 9.9(a)所示的输入组合模式所对应的希望输出模式的线性组合。这说明,按这一方案回想的网络,具有对输入模式的线性叠加映射能力。

(3) 无需规一化处理的学习、工作规则

如果想免去 CP 网络学习、工作规则中对输入模式 A_k 和连接权向量 W_j 的规一化处理,可采用 SOM 网络中寻找与 A_k 最接近的 W_j 的方法,代替学习步骤④、⑤中的式 9.5 和式 9.6

$$\|A_k - W_j\| = \min_j \|A_k - W_j\| \quad (9.21)$$

其它学习步骤不变。

(4) CP 网络联想映射特性的应用

CP 网络具有一种特殊的联想映射的功能。如图 9.10 所示,取输入学习模式和希望输出模式为同一模式 A_k ,且令 $A_k = (X_k, Y_k)$,即将模式分为 X_k 和 Y_k 两部分。当给网络提供 P 对这样的模式进行学习之后,则可用这一网络进行模式间的联想映射。具体

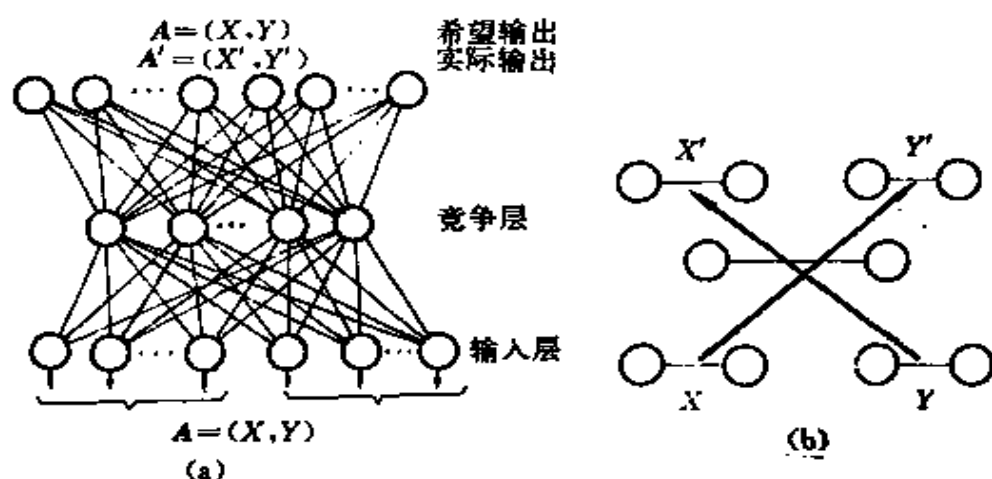


图 9.10 网络联想映射示意图

做法是,给网络提供一输入模式 $A = (X, 0)$,则在网络的输出层将

产生一个与模式 (X, Y) 非常接近的输出模式 (X', Y') ,即由模式 X 联想起模式 Y (或 Y')。同理,当给网络提供一输入模式 $(0, Y)$ 时,则在网络的输出层同样将产生一个与模式 (X, Y) 非常接近的输出模式 (X', Y') ,即由模式 Y 联想起模式 X (或 X'),这也是网络称为“对向传播”的缘故。

特别是当 Y 与 X 之间存在函数关系 f 时,即 $Y=f(X)$,则网络可以由函数的自变量联想起对应函数 Y 。同样也可以由函数 Y 联想起其自变量 X ,即网络能同时记忆这个函数及函数的逆函数。这一特性在图象数据压缩中有着重要的意义,这时 $f(X)$ 为压缩函数,而其逆函数 $f^{-1}(X)$ 为恢复函数。

应该注意,为使CP网络有较准确的模式识别性能,提供给网络训练的学习模式应尽量符合均匀分布。同时,输入模式的类别必须不重叠,即其本身是可分的。而且要求网络有足够的竞争层神经元,以保证网络有效地对学习模式进行自组织。

第十章 人工神经网络应用

10.1 文字识别

10.1.1 文字识别的生理特点

文字识别是模式识别领域中具有典型意义的一类问题,也是人工神经网络应用研究中一个比较活跃的课题。对于一个智力健全的人来说,通过学习获得认识文字的能力,从而能识别各种复杂的字符是一件不太困难的事情。即使对那些字体变形很大的手写体文字,人们一般也很容易将其识别出来。但是若要阐明人脑识别文字的神经活动机理,并应用这一机理提出机器识别的方法,却是非常困难的。这也是按传统的识别方法对文字识别,特别是对变形文字的识别至今未能获得满意效果的原因之一。现代医学研究表明,在大脑的视觉感受野中存在着对方向线段和轮廓图形的局部特征产生反应的细胞,进一步的基础医学实验还表明,在大脑视野的中枢神经系统内,还存在着对圆、三角形及四边形等更复杂模式的选择反应细胞。正如第二章中所指出的那样,这种分工负责的神经细胞的组成方式,构成了大脑皮层的组织结构特征。根据大脑视觉神经系统的组成方式,可以这样理解大脑对文字识别的神经活动机理,即视觉系统首先从输入的文字模式中抽取出简单的模式特征,然后有机地组合这些简单特征,并在此基础上抽取出更复杂的特征。大脑的这种文字识别过程,实际上是一种把模式的外部表示向神经网络的内部表示的转换过程,并在转换的过程中不断地抽取模式特征,且随着网络阶层的增加,所获得的模式特征中包含

的输入模式的信息量越加丰富,从而最终得到对输入模式的准确响应。

从信息处理角度来看,最简单的文字识别过程可以看作输入文字与预先贮存的文字代表符号进行比较、检索的过程。具体方法是,将某一文字的所有模式(即这一文字的所有变形形式)事先存入计算机,通过检索处理程序获得识别结果。但是这种识别方法的实际操作过程几乎是无法实现的。以 10×10 点阵的文字二值模式为例,某个文字的手写体,最多可能有 2^{100} 种。显然,面对如此庞大的存贮模式,即使用高速计算机也将无能为力。一种可行的识别方案是只将每个文字的代表模式存入计算机,这一代表模式包含了此文字的主要特征,如对称性、重心位置、笔划数量及走向等。识别程序的主要任务是从含有大量噪声的输入文字模式中抽取这一文字的主要特征,并将这些特征与计算机内存贮的特征进行比较,得到对输入文字模式的正确响应。这种对模式特征抽取的功能,正是人工神经网络所具有的特长。

对于文字识别的两类主要对象,即印刷体和手写体来说,印刷体文字具有形状规范、文字线段数量及走向稳定等特点;而手写体文字具有文字线段的相对关系和相对走向稳定的特点。这些特点给机器识别文字提供了实施基础。

10.1.2 印刷体文字识别

最简单的印刷体文字识别神经网络可由如图 10.1 所示的三层型网络构成。被识别的印刷体文字首先被分割成 10×10 的网格阵列,每个网格的实、空状态,决定了它所对应的神经网络输入层单元的“1”、“0”值。显然,网络的输入单元与网格阵列的网格数相等,即 $10 \times 10 = 100$ 个;网络的输出层单元数应与被识别文字的数量相等。网络的中间层单元数一般需要根据经验或通过反复实验决定,目前还没有一个理论上的准确确定方法。网络的结构确定之

后,还需要给网络提供与输入文字模式相对应的教师示教模式。其具体方法是, m 维输出向量中的 m 个元素分别对应 m 个输入文

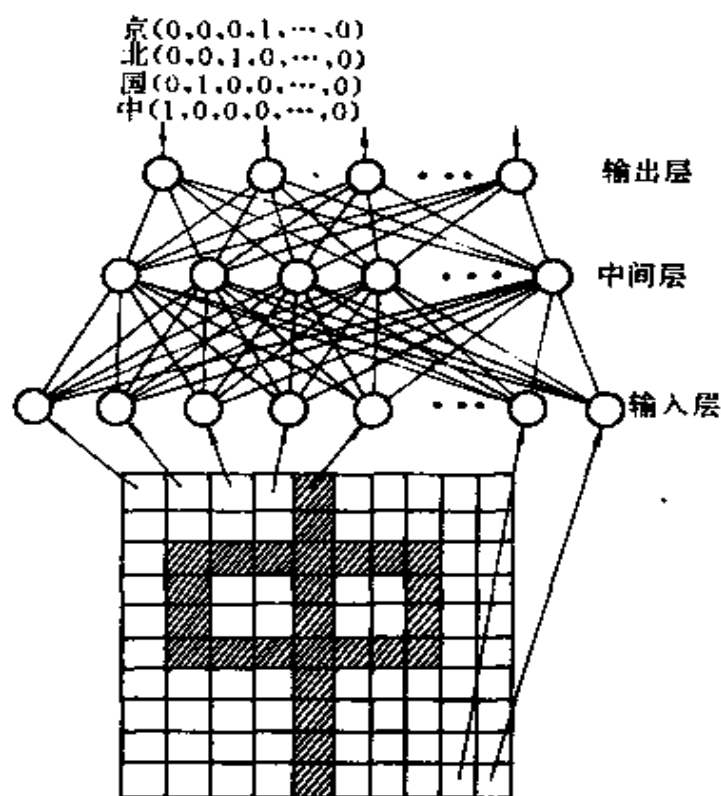


图 10.1 简单的文字识别神经网络

字,对于某个文字,只有向量中与之对应的元素为 1,其余均为 0。用输入模式与教师示教模式按误差逆传播算法对网络进行反复训练,直至网络误差收敛到小于所要求的值为止。学习后的网络则可用于文字识别。在识别过程中,当给网络提供某一文字的模式后,按网络的工作规则可以得到与此输入模式相对应的输出模式,根据输出模式向量中“1”元素的位置,则可判断出所识别文字。应该指出的是,即使被识别文字与网络学习时所使用的文字字体不同,网络仍可对这一文字作出判断。此时网络的模拟值输出向量,并不是只有某个元素为 1,其余为 0 或远远小于 1 的理想状态;而很可能是 m 个元素均为非 0 的模拟值向量。这种情况下应根据向量中

相对最大值元素的位置对输入文字作出近似判断。

以上介绍的文字识别方法虽然非常简单、直观,但是这种方法很粗糙,且识别率也不高。其主要原因有两点,一是文字分割密度太低,仅为 $10 \times 10 = 100$,使得文字分割后所得模式本身很难将相近文字加以区别。如果单纯地增大分割密度,会造成网络输入单元剧增,给网络的学习和计算带来困难。二是文字没有经过任何预处理就直接提供给网络训练,而网络在学习过程中,由于未经任何预处理的输入模式,其特征隐蔽较深,而且信息量分散,所以大大增加了模式特征抽取的难度,降低了网络的学习效率。为了解决这种简单方法的缺陷,人们提出了一些对输入模式事先进行各种预处理的改进方案。下面介绍一种以文字浓度特征作为文字输入模式的网络学习、识别方法。

所谓文字的**浓度特征**,就是按特定规则求得的文字标本点浓度值的集合。

文字标本点及浓度值的设定、计算方面如下。

设某文字在二维点阵 (x, y) 坐标处的状态值为 $f(x, y)$,

$$\text{则} \quad f(x, y) = \begin{cases} 1 & \text{实点} \\ 0 & \text{空点} \end{cases}$$

在网格点阵中按等间隔设置若干个标本点,设网格点阵中坐标为 (i, j) 的掩模值为 $M(i, j)$ 。则标本点 (xs, ys) 的浓度值 $d(xs, ys)$ 为

$$d(xs, ys) = \sum_{i=1}^n \sum_{j=1}^m M(i, j) \cdot f(xs + i, ys + j) \quad (10.1)$$

式中, n, m 为以标本点 (xs, ys) 为中心的二维掩模式计算范围,即 $n \times m$ 。图 10.2 为 5×5 掩模值域计算表。由式 10.1 和图 10.2 可知,如果在 5×5 点阵范围内文字模式的点阵值均为 1,则处于这个 5×5 点阵中心的标本点的浓度值就是图 10.2 中各数值之和,即 982。某文字所有标本点浓度值的集合就构成了这个文字模式的浓度特征。下面以较简单的印刷体阿拉伯数字及英文字体为例,说明以浓度特征值为输入模式的神经网络识别方法。

5

5

0	24	30	24	0
24	60	80	60	24
30	80	110	80	30
24	60	80	60	24
0	24	30	24	0

图 10.2 5×5 掩模值域计算表

如图 10.3 所示为 35×29 点阵列构成的文字二值模式图。横向和纵向每隔两个点设置一个标本点共有 11×9 个点。由图中所示的数字“2”的模式及标本点浓度值的计算方法,可以得到如图 10.4 所示的浓度值图。由此可以看出,这种模式预处理方法有以下两个优点:① 文字模式的浓度特征淡化了字体边界,即对于同一个文字,即使字体有些变化,轮廓有些变形,文字边界含有噪声,



图 10.3 35×29 文字模式点阵

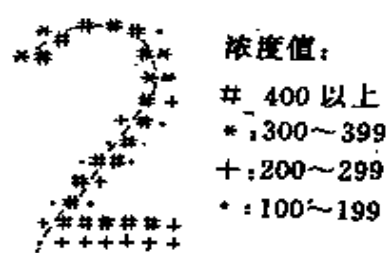


图 10.4 文字浓度特征举例

但其浓度特征的变化却不大。也就是说,这一预处理将文字模式的特征进行了预抽取,提高了提供给网络输入层的文字模式的相对

稳定性,同时也提高了网络学习和运行的效率。② 文字模式的浓度特征弥补了由于增加文字分割密度而造成的网络输入单元数剧增的缺陷。以图 10.3 为例,采用这一方法,输入单元数可由 $35 \times 29 = 1015$ 个减少到 $11 \times 9 = 99$ 个,缩小了一个数量级。因此,文字浓度特征的预处理方法在字体形状变化不是很剧烈的印刷体文字识别中经常使用。

图 10.5 为文字识别的过程框图。首先用图象扫描装置获得

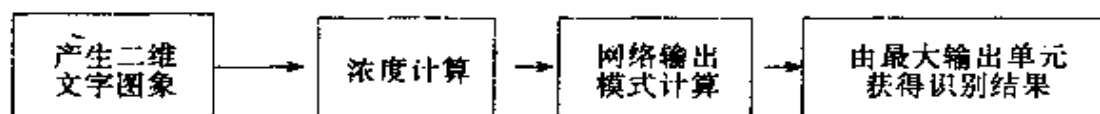


图 10.5 文字识别过程

被识别文字列的二值图象,并根据文字之间的空白间隔,将各个文字从文字列中分离开。对于每个模式化的文字图象,通过这一外接矩形,求得图象中心位置,进而如图 10.3 所示那样,确定图象中各标本点的坐标位置。然后按式 10.1 计算出各个标本点的浓度值。为计算方便起见,将所有浓度值除以 1000,得到 $(0,1)$ 区间内的浓度值。把这些浓度值作为网络的学习输入模式,连同相应的教师示教输出模式一起,按误差逆传播算法对网络进行训练,最后得到对输入文字具有识别能力的网络。

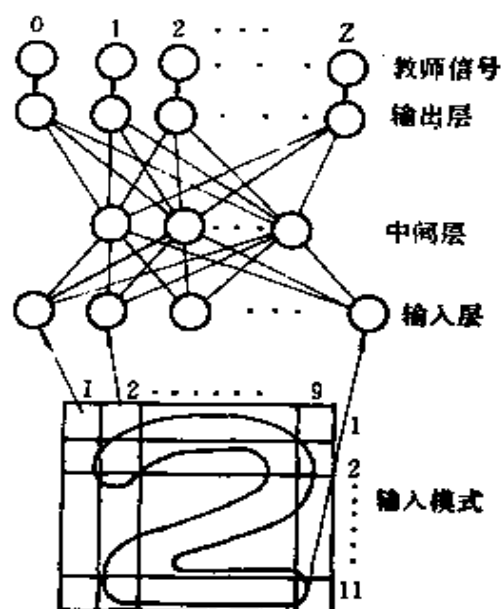


图 10.6 文字识别网络结构

文字识别的网络结构如图 10.6 所示。为进一步提高网络的记忆容量和记忆效率,可以用输出元素值的组合表示识别结果,并以组合方式构成网络希望输出模式。同时,为保证训练后的网络具有较高的识别

率,还需要对网络的识别效果进行评价。通常的做法是,选择若干种类型的印刷字体,以其中 60% 左右字体模式作为学习模式,另外 40% 作为对训练结束后网络识别效果的评价输入模式。

网络的文字识别能力除了与这类网络结构及算法本身的性质有关之外,在很大程度上还与网络本身的各种参数及网络的学习次数有关。这里仍以上面介绍的识别 36 个阿拉伯数字及英文字母的三阶层神经网络为例,分析误差逆传播算法中连接权学习率 α 、神经元输出阈值的学习率 β 以及 S 函数的倾斜率 μ_0 和网络学习次数与网络识别率的关系。第三章中曾经指出,增大学习率 α 和 β ,可以加大修正强度,从而加快学习速度;但高学习率也容易造成网络不适当的修正,形成收敛振荡,从而降低网络识别率。反之,如果 α 、 β 过小,则会大大降低网络学习速度,影响网络的实际应用。

图 10.7 表示了网络参数 α 和 β 比例值与网络识别率的关系。

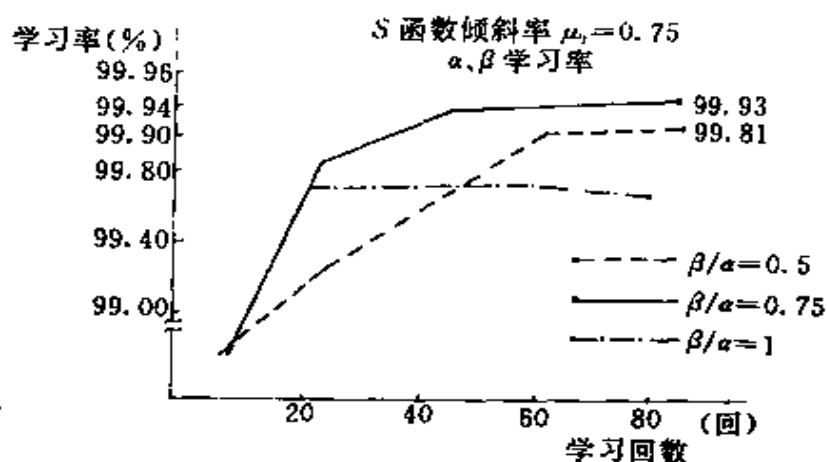


图 10.7 识别率与网络参数 α 、 β 的关系

由图可以看出,当 $\beta > \alpha$ 时,网络的最终识别率将会明显降低,也就是说当神经元阈值的修正强度大于网络连接权的修正强度时,网络的识别率将会降低。当 $\beta/\alpha=0.75$ 时,网络具有最佳识别率,即当网络经过 40 回左右的学习之后,可达到 99.93% 的识别率。

图 10.8 表示了网络输出函数——S 函数中倾斜率 μ_0 ($f(\mu) = \frac{1}{1+e^{-\frac{\mu}{\mu_0}}}$) 与网络识别率的关系。

如图所示, μ_0 越小, 即 S 函数的倾斜越大, 则 S 函数的特性越接近于阶跃函数, 因而网络学习进展困难, 且网络的识别率将急剧下降; 反之, μ_0 趋近于 1, 即 S 函数的倾斜越小, 网络各神经元输出的变化越加缓慢, 为达到预定的网络误差要求, 只能不断增加网络学习次数, 即便如此, 网络识别率反而有所下降或越加缓慢。

图 10.9 表示了学习次数为 80 回的情况下, 网络中间层单元数与网络识别率的关系, 随着中间层单元数的增加, 网络识别率有提高的趋势; 但当中间层单元数超过 40 个以后, 网络识别率的提高趋于饱和。

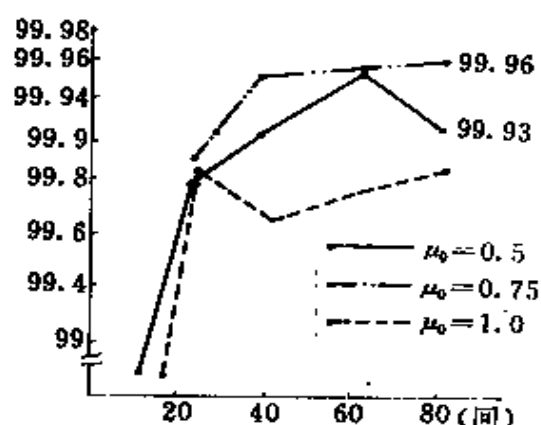


图 10.8 参数 μ_0 与网络识别率的关系

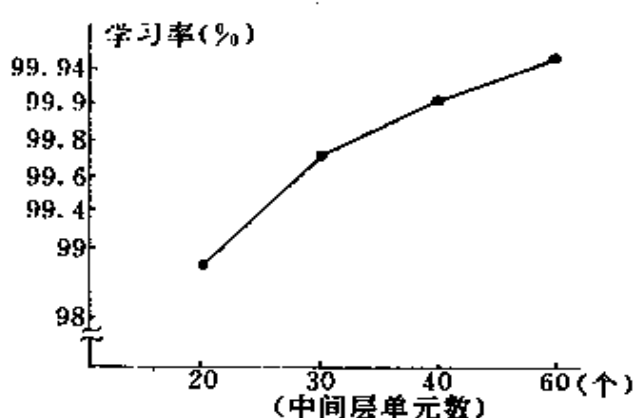


图 10.9 中间层单元数与网络识别率的关系

10.1.3 手写体文字识别

手写体文字最突出的特点是字形变化大, 字体种类多。严格地讲, 同一文字可以有无穷多种变体。这就使得对手写体文字的识别比印刷体困难得多。但是, 正如前面分析的那样, 尽管手写体文字

变化繁多,但从整体来看,文字笔划的相对关系及走向是相对稳定的。如图 10.10 所示,是一个 7 个英文手写体字母的例子。对于手写体文字同样可以运用文字浓度特征的数据预处理方法,但因手写体文字的不规则性,在同样的网络参数条件下,网络的识别率要比对印刷体的识别率低得多。



图 10.10 手写体文字举例

图 10.11 表示了含有 99 个输入层神经元与 10 个输出层神经元的三层神经网络对手写体 0~9 10 个阿拉伯数字的识别率。由图表可知,随着网络中间层单元个数的增加,并没有出现象印刷体状态时网络识别率有明显增加的趋势。对网络识别效果的实验还表明,学习结束后的网络,对学习过的输入模式识别率可达到

中间层 单元数	识别率 (%)	总连接 权数
20	94.4	2180
40	94.7	4360
60	95.2	6540

图 10.11 手写体文字识别率

99.5%,但对网络的测试输入模式的识别率却比这一识别率低 4%~5%。其主要原因是由于测试模式中含有许多未学习过的输入模式。为了提高网络手写体文字的识别率,人们提出了一些改进方案,其中一种行之有效的是改变网络输入层单元与网络中间层单元的连接关系。深入分析网络对手写体文字的学习过程可知,由于网络输入层各单元与担负特征抽取任务的中间层单元实行全连接,学习模式的整体相关特性被分布记忆在网络各连接权之中。

而手写体文字的特点是其模式整体相关性变化大,而各笔划之间相对关系较稳定,所以当网络运行时,当遇到未学习过的输入模式时,由于整体相关性的差别造成了网络输出模式的振荡,降低了网络的识别率。针对这一问题,如图 10.12 所示,将输入层单元分为若干区域,每个区域的单元仅与部分中间层单元连接,目的是使网络对学习模式的特征分解记忆,而将其整体特征剔除在记忆之外。

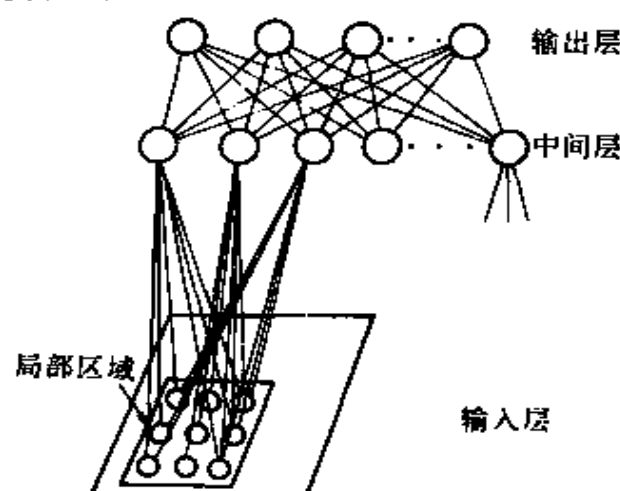


图 10.12 局部连接型神经网络

这一改进方案,可以有效地提高网络的手写文字识别率。图 10.13 是改进后神经网络手写体的识别率。与图 10.11 相比,局部连接网络比全连接网络的识别率提高了 1%~1.5%。若要进一步提高网络的识别率,还必须适当地解决诸如输入层单元区域的划分方法、各区域之间的重叠程度、每个区域所对应连接的中间层单元的数量等问题。

中间层 单元数	识别率 (%)	网络连接 权总个数
48	95.3	1680
96	96.3	3360
192	96.2	6720

图 10.13 局部连接型网络识别率

除以上介绍的这种改进方案之外,这里再简单介绍两种方案。一是对文字的大小尺寸做规整化预处理,即在网络学习、运行之前,将大小不一的各个手写体文字按比例规整为基本相等尺寸。这样做可以缩小同一文字的各种字体的模式差别,进而提高网络的识别率。另一种方案是以手写体文字的固有特征作为网络的学习、输入模式。如将手写体英文文字的轮廓和曲率当作其固有特征提供给网络。

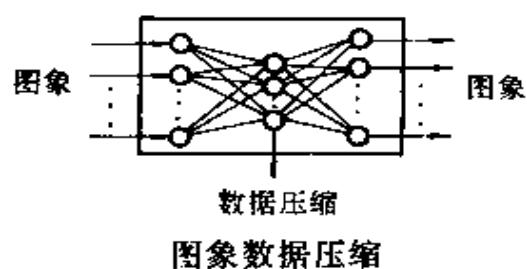
10.2 图象处理

10.2.1 图象处理概述

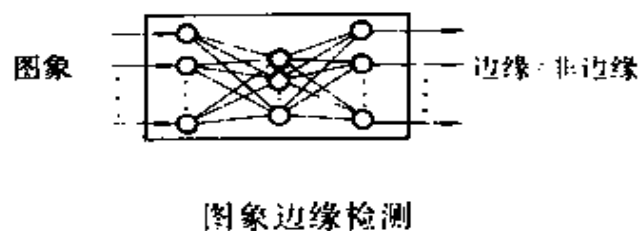
图象处理是包含丰富内容和具有广阔应用领域的研究学科。近几年来作为具有自组织、自学习和联想功能的人工神经网络理论,已成功地应用于图象处理的许多方面。概括地讲,神经网络应用于图象处理的主要思路是,把原始图象或经过适当预处理的图象当作网络的输入信号,在网络的输出端对输入图象进行分类或识别。图象预处理的主要手段有:傅里叶变换、特殊函数转换、专用的图象特征抽取算法以及其它一些数字图象处理技术。图 10.14 列举了人工神经网络在图象处理应用中的几个主要方面。

下面详细说明有关神经网络在图象数据压缩方面的应用技术,并对其它几方面的应用做一般性介绍。

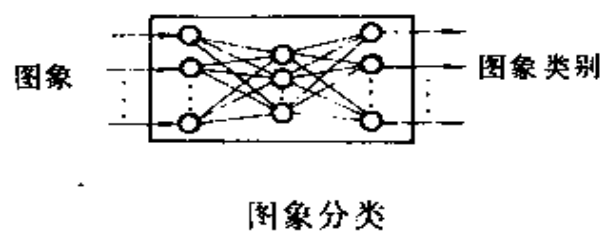
①



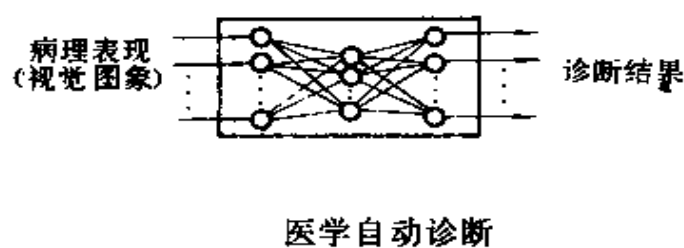
②



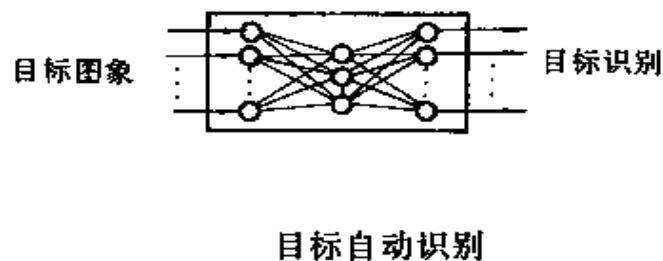
③



④



⑤



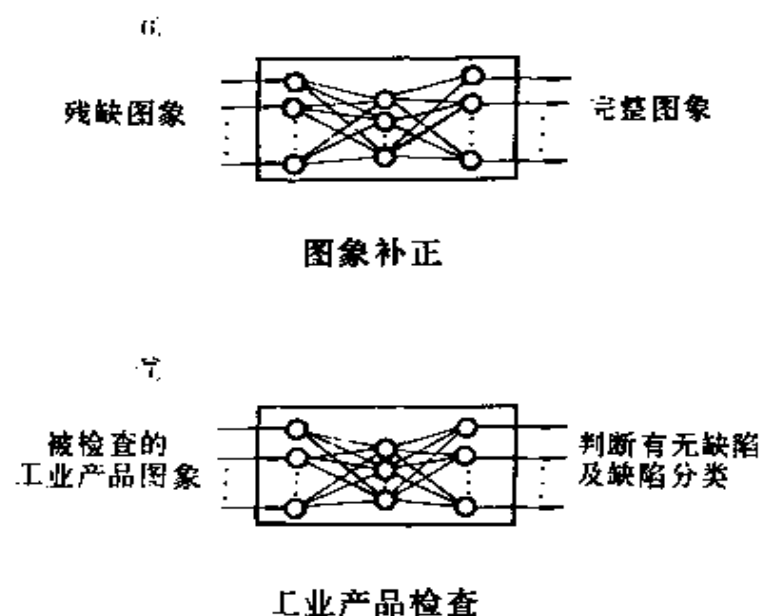


图 10.14 人工神经网络在图象处理中的几个应用

10.2.2 图象数据压缩

随着信息技术的不断发展,图象的远程传输和磁介质存贮日趋广泛。但是,一幅画面复杂、幅面较大的图象的信号量将占据上兆甚至更多的存贮字节,这给图象的存贮和传输带来了很大不便。因此,人们希望通过某种变换手段,将数量巨大的图象信号进行压缩,以压缩后的、大大减少了的图象数据信号进行存贮和传输。在需要提取图象或传输的终端接收图象时,按一定规则将图象恢复出来。这就是所谓的图象数据压缩。应用神经网络进行图象数据压缩的基本方法如图 10.15 所示。

由图可知,神经网络图象数据压缩系统主要由压缩网络(输入层——中间层)、传输通道(中间层——中间层)和再生网络(中间层——输出层)组成。这一系统的基本原理是,神经网络的输入层和输出层由相同的 N 个单元组成。在网络的学习过程中,网络输

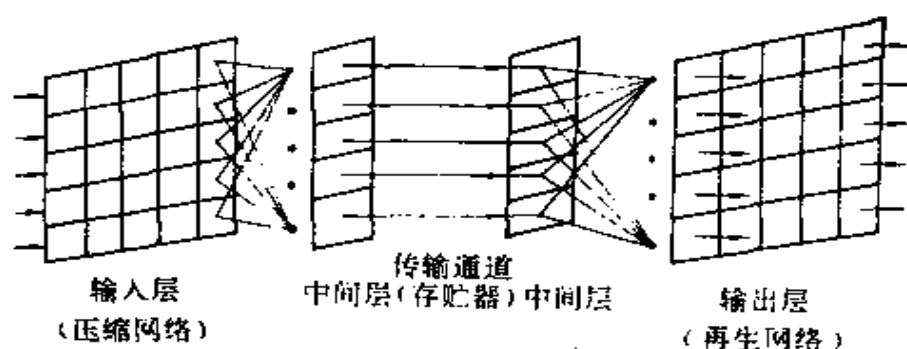


图 10.15 神经网络图象数据压缩系统

入层的学习模式及输出层的教师模式使用同一图象数字信号。由于网络中间层的单元数可以大大少于网络输入、输出层的单元数，则学习后的网络将以较少的中间层单元有效地表示输入图象模式并以此进行存贮或传输。由于网络的输入和输出层提供的是同一图象模式，因此网络的输出层可很容易地将输入图象模式再现出来。在这一过程中，输入层中间层的变换可以看作压缩编码过程，中间层至输出层的变换可看作解码过程。

图象的数据压缩处理通常采用两类方法，即逐次扫描法和并行处理法。逐次扫描法是将一幅图象划分成若干个图象块，然后通过对这些图象块的扫描逐次将各个图象块的数据输入同一个神经网络进行学习或压缩。并行处理是将所分割的各个图象块分别对应不同的神经网络进行学习或压缩，即网络个数与图象块个数相等。因此可以说，逐次扫描法网络所记忆的是整幅图象的平均特征，而并行处理法网络所记忆的是各个图象块的本身特征。

用误差逆传播网络按逐次扫描法进行图象数据压缩处理的具体步骤是^[30]：设需被学习或编码的输入图象幅面为 $M \times N$ 个象素点。将这 $M \times N$ 个象素点划分为若干个由 $n \times m$ 个象素点组成的图象块。网络的输入层和输出层均由 $n \times m$ 个单元构成，各个单元对应于每个图象块中的一个象素。网络中间层的单元数根据要实

现的压缩比以及误差要求而取不同的数目。网络的结构如图 10.16 所示。

网络的输入学习模式和希望输出模式按如下方法构成：由于图象中每个象素点的灰度级有 2^k 个取值，故将 2^k 个灰度值按线性关系规范为 -1 至 $+1$ 区间内的模拟值。把每个图象块中各个象素点的灰度值作为学习模式对依次提供给网络学习，直至误差收敛到符合要求为止。

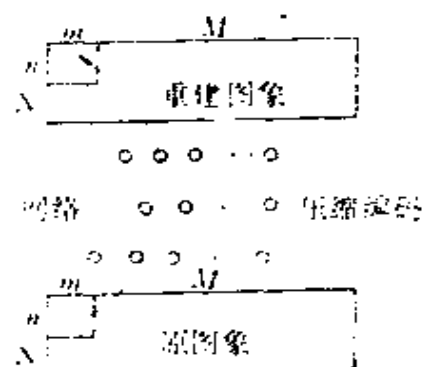


图 10.16 图象数据压缩

实验中，一般 n, m 在 $8 \sim 16$ 范围内取值，中间层单元数为 16, k 值取 8。网络先以 0.25 的学习系数进行 10 万次学习迭代，然后再以 0.01 的学习系数进行 5 万次学习迭代。学习过程中，网络各单元的输入/输出函数为式 10.2 所示的取值范围在 $-1 \sim +1$ 之间的 S 型函数

$$y = \frac{2}{1 + e^{-x}} - 1 \quad (10.2)$$

一幅图象的压缩质量是由图象的压缩比和图象的正规化二乘误差衡量的。但需要指出的是，网络中间层对输入层图象信号的响应是 $(-1, +1)$ 区间内的连续值，不能直接对其进行传输或存贮，必须将其转化为数字量。因此，无论是图象的压缩比还是图象的正规化二乘误差，都应输入、输出的数字量作为比较基准和误差计算依据，不能简单地以网络输入、输出层的单元数与中间层单元数之比作为图象压缩比。正确的方法是，首先把中间层的连续量按一定的量化等级将其转化为数字量（作为存贮或传输的数据），然后将这一数字量变换为相应的连续值（注意：由于量化误差，变换前后的连续量是不完全一致的），通过网络中间层和输出层的连接权传送到输出层，从而得到重建图象，并以此为依据，计算出相应的网络压缩比及误差。例如在上述网络中， $n=8, m=8, k=8$ ，则每个图象

块的数据量为 $8 \times 8 \times 8 = 512$ 。网络中间层单元数为 16, 将中间单元的模拟输出量量化为 8 位数字量, 其传输或存贮数字量位数为 $16 \times 8 = 128$, 这时图象的压缩比为 $512/128 = 2$ 。

设网络的原输入图象块为 f_{ij} , 经网络中间层模拟→数字→模拟变换后在输出层得到的重建图象块为 g_{ij} , 则整幅图象的正规化二乘误差可由式 10.3 计算得到

$$E = \frac{\sum_{i,j=1}^s (f_{ij} - g_{ij})^2}{\sum_{i,j=1}^s f_{ij}^2} \quad (10.3)$$

式中, i, j 表示划分图象块的行、列, s 表示图象块总数。上例的正规化二乘误差为 0.413%。一般来说, 压缩比和网络误差是两个很难在保持中间层单元数或量化等级不变的情况下同时满足的指标。增加中间层单元数目或减少中间层单元的输出生量化等级都可以提高网络的压缩比, 但网络误差也随之增加。因此, 在设计网络时应兼顾考虑两个指标的要求。经验表明, 在保持相同的压缩比的情况下, 减小网络误差的有效方法是适当减少中间层的单元数, 增加量化等级。这样处理比增加中间层单元数而减少量化等级的网络误差要小。如上例中, 对于 $8 \times 8 \times 8$ 位数字输入, 16 个中间层单元、8 位量化等级的网络误差为 0.413%; 而当压缩比仍为 2, 而中间层单元的数增至 32、量化等级减至 4 位时, 网络误差为 0.625%。其原因这是由于误差逆传播网络对中间层单元数的敏感度比在网络数据精度的敏感度低。

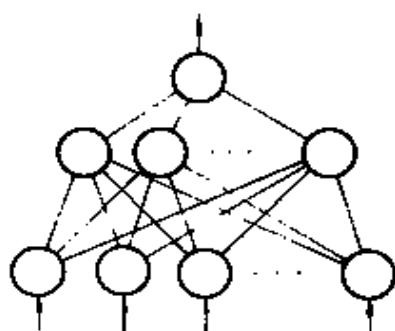
还有一点需要说明的是, 由于网络在对样本图象进行自联想学习的过程中已记忆了图象在输入、输出层之间的灰度对应特征, 因此对于未曾学习过的输入图象, 网络仍然可以对其进行压缩和重建, 但网络误差要比学习过的图象大得多。若要减小误差, 则应尽可能在网络的学习模式中包含丰富的图象灰度样本, 使网络通过训练掌握较高的图象灰度的分辨能力。

为检验神经网络在图象数据压缩中的抗干扰能力,有人曾做了以下实验:将一幅 32×32 点像素的图象作为学习模式,提供给输入、输出层单元数为 1024、中间层单元数为 8 的三层 BP 网络学习。对学习结束后的网络分别提供在原图象基础上加入随机噪声、模糊图象边缘、图象像素移位、图象部分重叠等受损图象,检查网络输出层重建图象的质量。实验结果表明,网络对加入噪声、模糊边缘和部分重叠的图象具有较强的抗干扰能力,可得到较好的重建图象,而对像素移位的图象适应性较差。

10.2.3 图象边缘检测

图象识别一个关键性问题是准确地确定图象边缘,从而得到清晰的图象轮廓。图象边缘的特点是其像素的灰度值具有突变的性质,这也正是边缘检测的出发点。当沿某一特定方向、在某一范围内的图象像素灰度值发生突变时,则可判定此处为图象的边缘。但图象边缘具有多样性,很难找到一个精确的通用算法对各种各样的图象边缘作出正确的判断。由于神经网络具有记忆及联想功能,多层网络还可以完成非线性映射,所以将神经网络应用于图象边缘检测是十分有效的。其具体做法是^[31],选择一组包含各种方向、各种强度的无噪声图象作为网络的学习模式,用尽可能精确的边缘检测器作为检测标准器。在被测图象上移动图象扫描仪,其输出同时提供给检测网络的输入层和边缘检测器的输入端,以此作为网络的学习模式和希望输出模式。图象每被扫描一遍作为一次学习,网络的结构如图 10.17 所示。图 10.17 边缘检测网络结构

这是一个多输入、单输出的三层网络,网络按 BP 算法学习,为保证网络有较好的收敛性能,在学习可采用变学习系数的方法,即按式 10.4 取学习系数 α



$$\alpha_K = \alpha_0 * (0.9)^K + 0.1 \quad (10.4)$$

式中, K 为学习回数, $\alpha_0=0.3$ 。网络各连接权的初始值为 $(-0.5, +0.5)$ 区间内的随机值。

同图象数据压缩一样, 对于一个学习结束后的边缘检测网络, 尽管被测试的图象是未经学习过的, 但由于用误差逆传播算法学习过的网络具有自联想功能, 可以按相似输入得到相似的响应, 所以只要在学习模式中包含着丰富的边缘样本, 网络对那些未曾学习过的图象边缘或对加有噪声的学习模式仍能作出准确的判断。

10.2.4 图象自动分类

神经网络的一大特点就是具有自组织能力。因此, 当一个网络在进行充分学习之后, 很容易用来对一组图象进行自动分类。其网络结构及网络学习方法如图 10.18 所示。

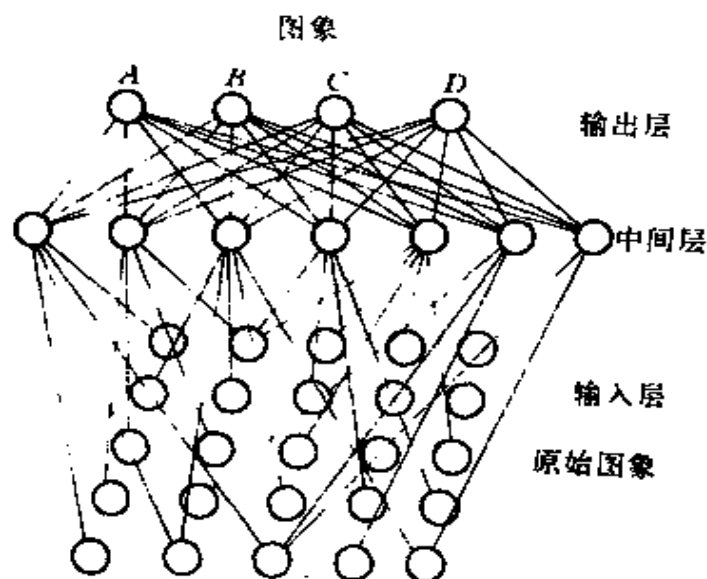


图 10.18 图象分类网络结构

10.2.5 医学自动诊断

根据各种医疗仪器的检查结果对病人的病情进行诊断是医学上最常用的手段。如各种病理化验,X光检查,B超、CT检查等等。目前基本上是医生凭借病理学知识和自己的经验,对检查结果进行分析和判断。因此,诊断结果往往会因人而异,误差较大。人们一直希望能有一种自动的诊断方法,可以排除各种人为因素,得到准确、客观的结果。神经网络自然成为最有希望的自动判断手段之一。如在病理化验时,当产生某种病理化学反应时,在被化验中将产生均匀度不一的块状物;反之,被化验物的形状比较平滑。这就给运用神经网络通过图象处理进行医学诊断提供了有利条件。运用这种方法可以区别出正常细胞和病变细胞。又如,通过对大量的CT图象样本的学习,神经网络可以对直接从CT仪上获得的图象进行自动诊断。对心电图、X光片等也有类似的自动诊断方法。

10.2.6 目标自动识别

如图10.19所示,所谓目标识别就是要在很短的时间内,对探测器(如:卫星、红外线摄影机等)获取的图象作出迅速、准确的判断,以确定被寻找目标是否处于图象的中心位置。这一问题的复杂性在于,被寻找的目标常常隐藏在某些障碍物之中,如草丛、树林、楼群等。而神经网络对噪声的高抗干扰能力和联想能力则可以对此发挥其应有的作用。

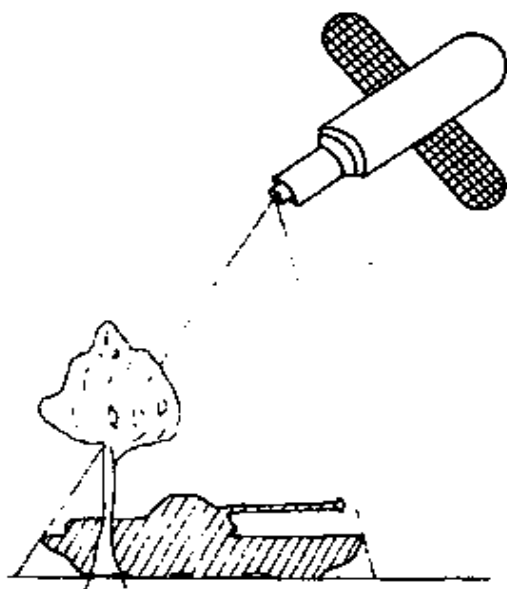


图 10.19 目标自动识别

10.2.7 图象补正

所谓图象补正是指把因某种原因而出现残缺的图象修补完整。图 10.20 所示就是图象补正的一个例子。被检测目标由于背景物的遮挡而使其图象出现了一条空缺,在以军事需要为目的的图象处理中,需要知道图象的全貌,以判断出目标物的各种参数,如型号、重量等。运用神经网络,以其自联想的功能则可以达到这一目的。当然这需要网络在应用前经过充分的学习、训练,即对各种可能遇到的目标,将其完整的图象作为学习样本模式提供给网络记忆。学习后的网络不但对样本模式中的图象具有再生功能,而且对一些与样本模式相近似的目标图象也可以再生出来。

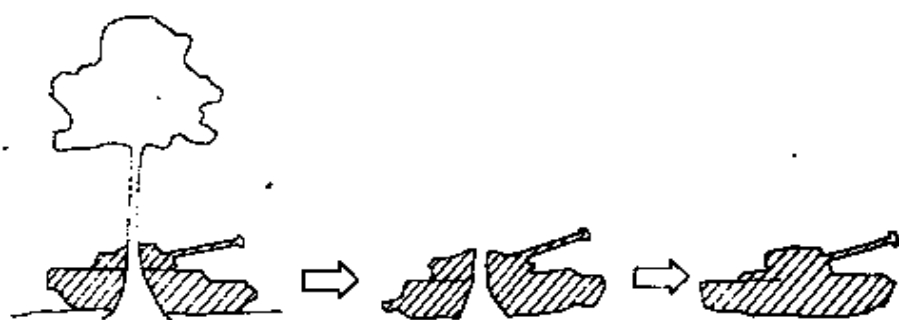


图 10.20 图象补正举例

10.2.8 工业产品检查

工业产品检查是神经网络应用的一个重要领域。工业产品检查包括产品分类、产品的质量检测、产品的缺陷鉴别等。Glover. D 在 1988 年对神经网络在工业产品的检查中的应用做了许多成功的实验^[28],这些实验都是以工业产品的图象作为检测对象的。他还用 Fischer 线性判别函数法、误差逆传播算法、对向传播算法等三种不同的方法进行实验比较,得到了满意的结果。图 10.21 是他

在实验中使用的各种糖浆瓶的图象示意图。



图 10.21 糖浆瓶图象

10.3 优化组合应用——有价证券的选择

优化组合是一个内容丰富、涉及范围广泛、具有重要应用价值的研究领域。运用人工神经网络尤其是随机型神经网络,可以比较准确地得到线性或非线性优化组合问题的最优解。

在第五章介绍 Hopfield 网络时曾经指出,运用神经网络解决优化组合问题的关键是建立适合于问题求解的网络能量函数。这个能量函数一方面必须反映问题的约束条件,另一方面其最小值要对应于问题的最优解。网络按照其自身的运行方程进行状态更迭,最后达到稳定状态,即网络能量最小状态,此时的网络状态就应是问题的最优解。这里举一个有关经济问题的实际例子——有价证券的投资选择问题,从中可以了解如何应用神经网络解决优化组合问题。

有价证券是一个对投资者来说具有很大诱惑力、但又有很大风险性的投资领域。投资对象选择正确可以获得可观的利润,反之

则会蒙受巨大经济损失,甚至导致投资者的破产。因此,如何根据市场的变化情况,根据对各种证券本身效益及证券之间相互关系的分析选择最佳投资方案,就成为摆在投资者面前的一个最实际的优化组合问题。所谓最佳选择方案,是指通过对投资分散性的控制,在保证投资效益不变或尽量大的前提下,使投资风险减小到最低程度。关于这一问题,早在 30 多年前 Markowitz 就提出过一种根据二次规划法得出的求解公式^[29]。但是,当求解对象和约束条件很多,即求解问题较复杂时找不到一个快捷、有效的方法求解这一公式,使得这一求解方法失去了使用价值。随着神经网络理论的逐渐成熟,为求解 Markowitz 公式开辟了新的途径。Markowitz 所提出的有关有价证券的选择方法作为投资基本理论再次受到重视。下面以这一方法为基础,导出有关数学模式,并用神经网络方法求解这一模型^[32]。

设第 i 种证券单位时间的利润率为 R_i , 对此种证券的投资率为 V_i 。

则有价证券总利润率 R 为

$$R = \sum_{i=1}^N R_i v_i \quad (10.4)$$

式中, N 为投资的有价证券的总数。投资率 v_i 的约束条件为

$$\sum_{i=1}^N v_i = 1 \quad (10.5)$$

若要减小投资的风险性,就需知道证券将来的变化趋势,即要预测证券的变化。同许多预测问题一样,证券的发展变化具有一定的随机性,属于随机预测的问题。传统的方法通常是用证券过去一段时间内的利润率推测出利润率的标准误差,以此作为伴随投资而来的风险。用神经网络理论解决这一问题时,把投资的风险性定义为投资的分散程度,即

$$X = \sum_{i=1}^N \sum_{j=1}^N \sigma_{ij} v_i v_j \quad (10.6)$$

式中, σ_{ij} 为不同证券之间的相关系数, 即

$$\sigma_{ij} = \langle R_i, R_j \rangle \quad (10.7)$$

它表示证券 i 和证券 j 在过去一段时间内的相关程度 ($0 < \sigma_{ij} \leq 1$)。

现在的问题是应使式 10.4 尽量大, 而使式 10.6 尽量小, 从而达到投资的效益最好而风险最小。由此可得到网络的能量函数

$$E = \frac{a}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sigma_{ij} v_i v_j - \frac{b}{2} \sum_{i=1}^N R_i v_i + \frac{c}{2} \left(\sum_{i=1}^N v_i - 1 \right)^2 \quad (10.8)$$

式中, $\frac{c}{2} \left(\sum_{i=1}^N v_i - 1 \right)^2$ 为“惩罚”项, N 为所要选择对象的总数目。

系数 a, b, c 分别表示所对应项的权重。

我们已知 Hopfield 网络的能量函数和运行方程分别为

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N w_{ij} v_i v_j - \sum_{i=1}^N v_i I_i + \sum_{i=1}^N \frac{1}{\tau} \int_0^{v_i} g^{-1}(t) dt \quad (10.9)$$

$$\begin{cases} \frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_{j=1}^N w_{ij} v_j + I_i \\ v_i = g(u_i) \end{cases} \quad (10.10)$$

式中, $g(x)$ 为 S 函数。

已知网络能量函数 E 与网络神经元内部状态 u_i 之间的关系

$$\frac{dE}{dv_i} = -\frac{du_i}{dt} \quad (10.11)$$

由式 10.8、10.9、10.10 和 10.11 可得

$$\frac{dE}{dv_i} = a \sum_{\substack{j=1 \\ j \neq i}}^N \sigma_{ij} v_j - b R_i + c \left(\sum_{i=1}^N v_i - 1 \right) + \frac{u_i}{\tau} \quad (10.12)$$

比较式 10.10、10.13 可得到网络连接权的表达式为

$$w_{ij} = -a \sigma_{ij} (1 - \delta_{ij}) - c \quad (10.14)$$

式中,

$$\sigma_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (10.15)$$

神经元 i 的偏置为

$$I_i = bR_i + c \quad (10.16)$$

神经元 i 的输出状态为

$$v_i = g(u_i) = \frac{1}{2} \left[1 + \tanh \left(\frac{u_i}{u_0} \right) \right] \quad (10.17)$$

当设 $\tau=1$ 时,可得到网络内部状态的迭代方程

$$u_i(t+1) = u_i(t) + \Delta u_i = -a \sum_{j=1}^N \sigma_{ij} v_j + bR_i - c \left(\sum_{i=1}^N v_i - 1 \right) \quad (10.18)$$

这一问题可以用 Hopfield 网络算法或模拟退火算法(SA 算法)来进行求解,在求解之前应特别注意以下两个问题。

(1) 网络参数的选择

网络的运行过程和运行结果对式 10.17、10.18 中的参数 a 、 b 、 c 、 u_0 等的变化十分敏感,往往由于这些参数的不稳健性而影响网络的收敛过程,所以原则上这些参数是不能随意改变的。这些参数反映了目标项和惩罚项在优化组合过程中所占的权重。一般说来, a 值较小时,网络容易收敛;而 a 值较大时,则可以使证券项目的选择更趋于优化。 u_0 是放大器增益, u_0 太小时,阈值函数接近符号函数,不能获得较满意的解; u_0 太大时,阈值函数曲线过于平坦,神经元状态不易收敛到 0 或 1,从而使获得最佳项目选择的概率下降。综合考虑以上因素,参数选择如下: $a=b=0.500$, $c=0.200$, $u_0=0.019$ 。

(2) 网络初始状态的选择

同网络参数一样,网络的初始状态也具有不稳健性,即因各神经元初值大小和分布的不同,计算结果将产生差异。因此,初始状态的选择应具有随机均匀分布特性。可按以下方法设置网络初值

$$u_i(t_0) = u_{00} + \delta_i \quad (10.19)$$

式中
$$u_{00} = \frac{1}{2} u_0 \ln(N-1) \quad (10.20)$$

N 为项目总数, δ_n 为随机扰动项 ($-0.1u_0 \leq \delta_n \leq 0.1u_0$), 加此项的目的是使各神经元初值不完全相同而引起竞争求解。

这一问题的 Hopfield 解法程序流程如图 10.22 所示。

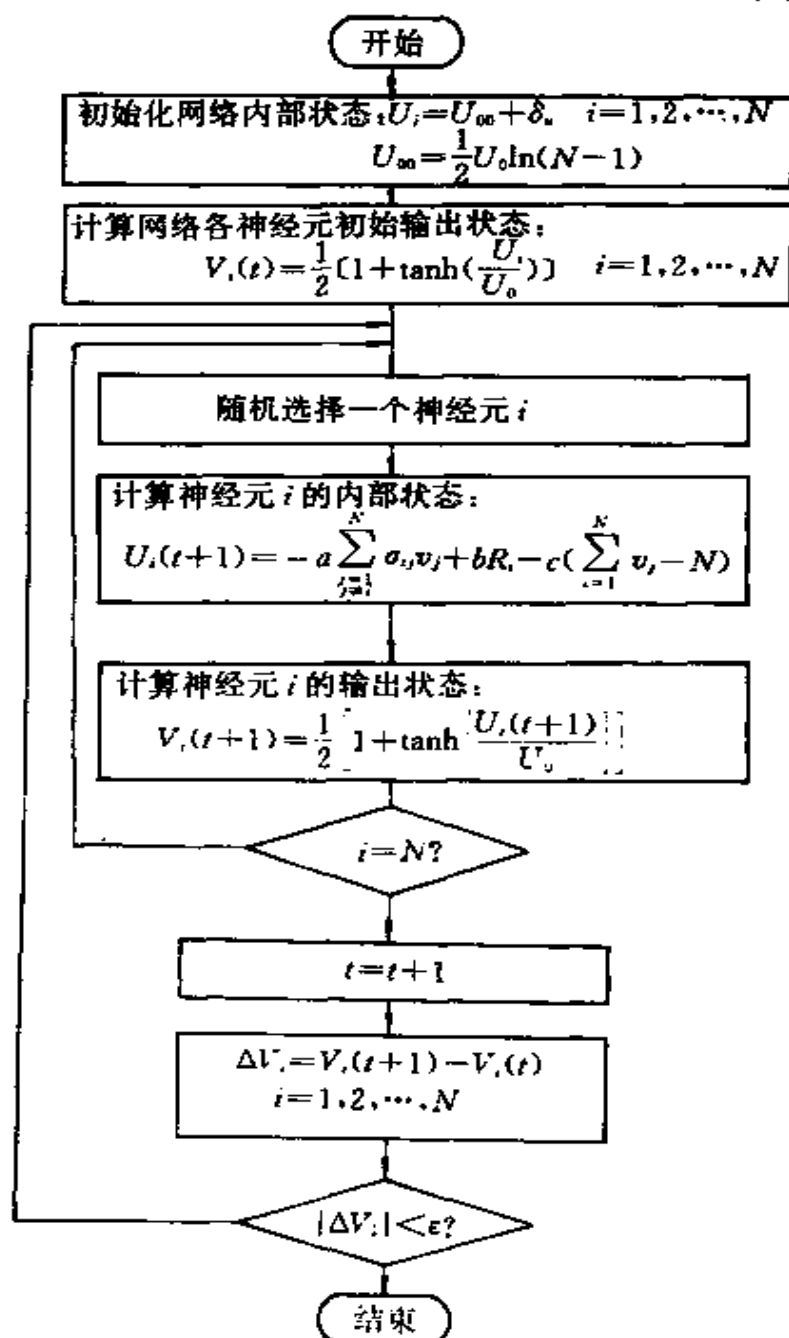


图 10.22 求解有价值证券问题 Hopfield 网络运行程序框图 213

最后介绍一个仿真实例。以 100 个有价值证券项目作为选择对象,要求找出最佳投资方案。其中包括投资具体项目、项目总数及各项的投资率分配。如图 10.23 所示,每一个代表神经元的圆点直径大小与该神经元输出状态的大小成比例。在网络运行初期,各神经元输出状态的大小由初值决定。随着网络迭代次数的增加,各神经元输出状态产生了差异,各圆点的面积发生了明显的变化。图 10.24 是迭代 7 次时的图形。显然,此时的网络状态还不符合约束条件。当迭代次数为 500 次时,如图 10.25 所示,图中只有 5 个圆点的面积突出出来,并且这 5 个神经元输出状态之和基本上趋近于 1,符合了约束条件,可以认为此时网络的能量已趋于最小,问题的最优解由此而得出。

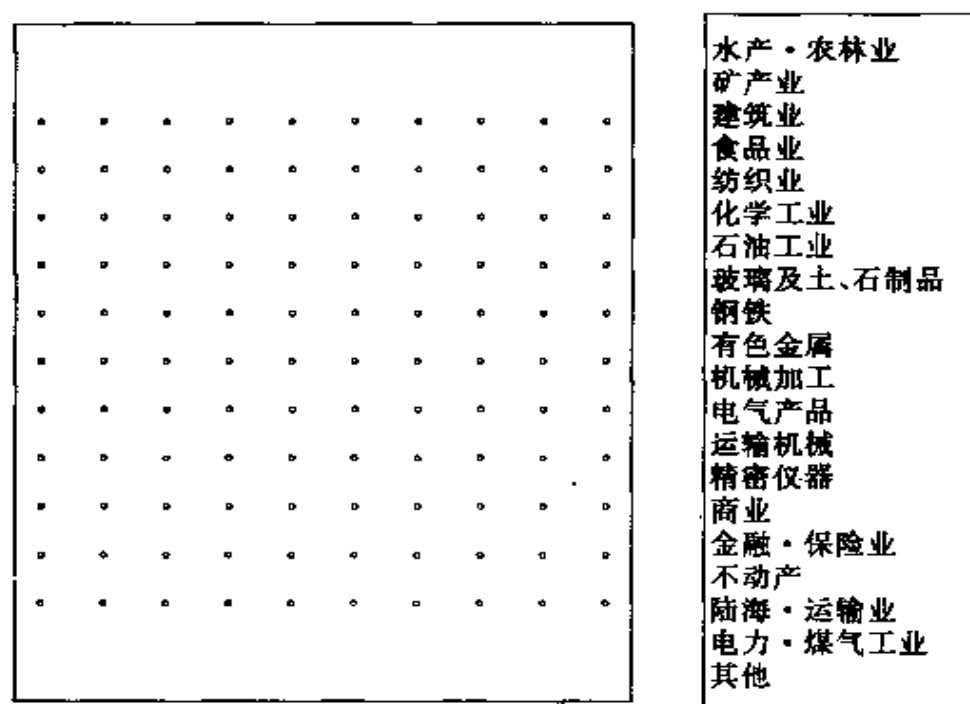


图 10.23 网络初始状态

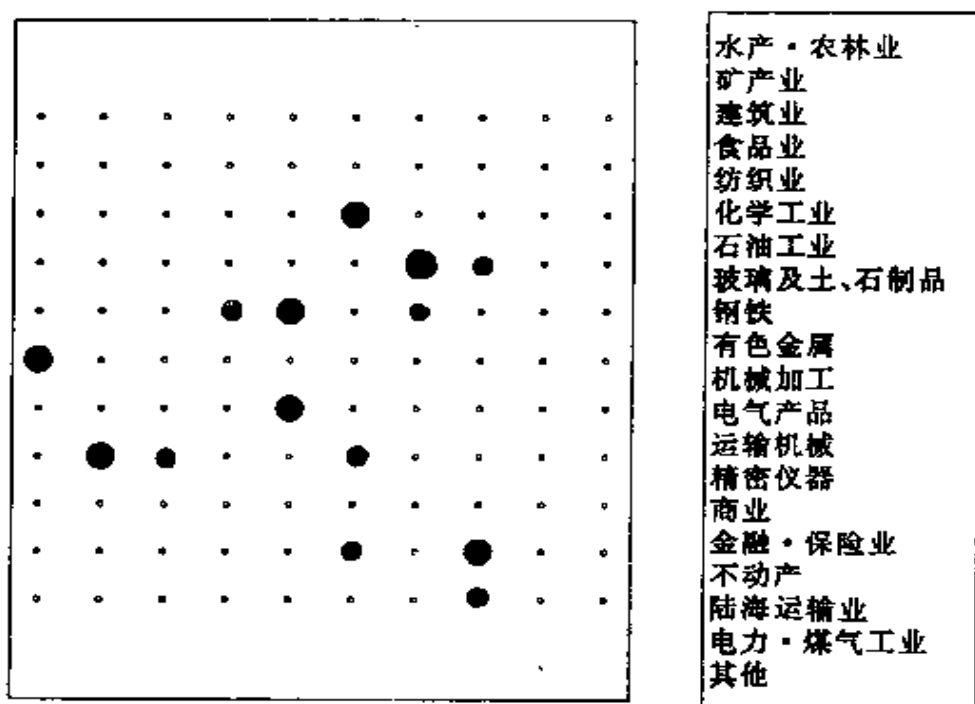


图 10.24 7 次迭代后的网络状态

10.4 神经网络专家系统

10.4.1 神经网络专家系统的基本原理与结构

专家系统是人工智能理论应用获得成功的一个重要研究领域。从本世纪 70 年代开始,得益于人工智能理论的专家系统得到了迅速发展和广泛应用。但是,由于人工智能的传统理论是以符号操作为基础、以 Von Neumann 计算机作为实现工具的,所以一般的专家系统在实现类似人脑的学习、联想等方面遇到了极大的困难,主要表现在以下几个方面。

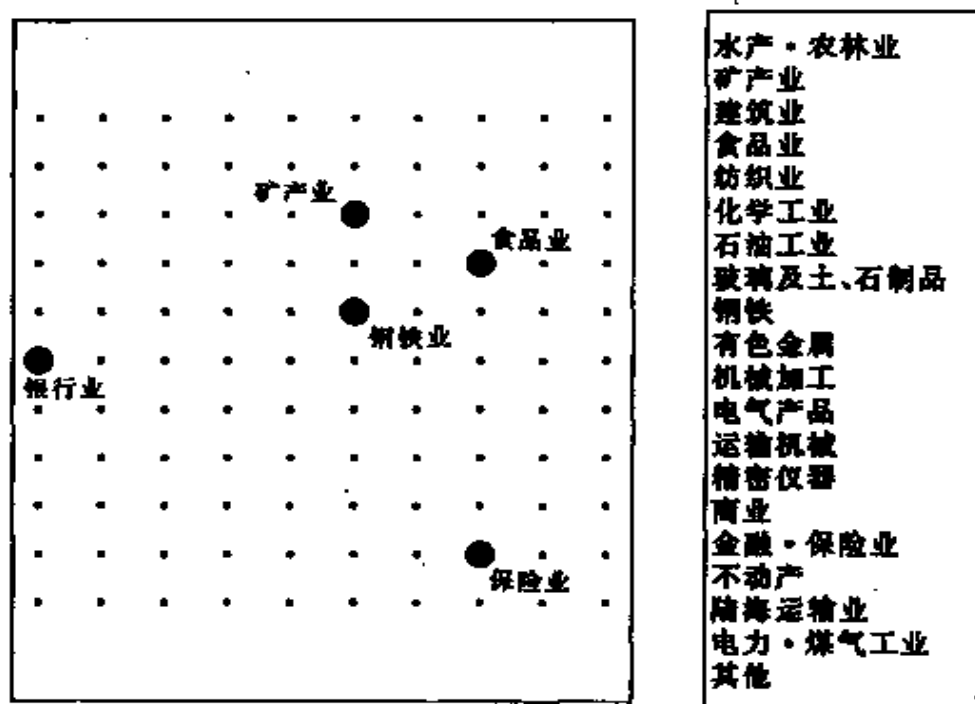


图 10.25 500 次迭代后的网络状态

(1) 知识获取的“瓶颈”问题

一般专家系统的知识获取依靠人工移植,即把有关领域的专家知识以规则的形式存入计算机中,这就引发了两个问题:一是某些专家的知识很难以一些规则来描述,甚至很难用简单的语言加以描述;二是这种移植方式的知识获取是间接的,既费时、效率又低,而且还常常因系统知识库的不良结构造成知识组合爆炸。

(2) 缺少深层的理性知识和强有力的推理功能

一般专家系统依靠人工加工、移植所获取的往往只是浅层的、表面的、经验性的知识,缺少深层的、本质的、理性的知识。系统不具备联想记忆和自学习功能,特别是对相近领域的边缘性问题求解能力很差,即系统适应环境的柔软性差,不能随环境更新知识、自我完善,也不能用联想的方法进行推理。系统的智能水平只是知识的总和。

(3) 推理速度慢、实时性差

一般的专家系统知识表达与知识存贮是一一对应的、且没有冗余性,计算和推理完全集中在 CPU 中,而完成计算和推理所需要的知识则放在存贮器中,即信息的处理与存贮是分开的,信息的提取和处理过程是串行的。这样,知识越多,推理速度就越慢,效率也就越低。因此系统基本都是在“离线”与“非实时”条件下工作的。

人工神经网络理论的发展,为人工智能和专家系统的研究开辟了一条新的有效途径。

神经网络的主要特征是信息的巨量并行处理、信息的分布式存贮、信息的处理与存贮的合二为一性、自组织学习以及实时处理功能等。这些特征给专家系统注入了新的活力。神经网络专家系统在知识获取、知识表达、知识存贮、知识推理等方面具有以下主要特征。

① 知识不是通过人为的加工、移植,而是由专家提供范例(输入学习模式)及相应的解(教师示教模式),通过特定的学习算法对样本进行学习并不断修改网络各神经元之间的连接权而获得的,因而知识的获取是一种自学习行为。

② 知识的表达形式由显式变为隐式。知识不是以产生式规则、语义网络、谓词逻辑等显式表示,而是把专家求解实际问题的启发式知识和经验分布到网络的连接权上,通过这种分布形式表示特定的概念或知识。在这里任何知识规则都变换成数字形式,便于知识库的组织和管理,通用性强。

③ 知识的推理从对知识库内贮存的知识的串行处理和验证过程变为网络中隐含模式对输入的竞争,这种竞争是并行的,是针对特定特征而言的。对于某输入模式,经训练后的网络按特定运行规则进行前向计算,产生一输出模式,且每个输出节点有其自身的特定解释。神经网络允许输入模式偏离学习样本,只要输入模式接近于某一学习样本模式,则网络将按相似输入产生相似输出的原则提供一联想近似输出模式,即实现联想推理。

原则上,几乎所有形式的神经网络都可以用来组成专家系统。神经网络的基本结构如图 10.26 所示,它由知识获取、知识库、推理机、I/O 接口等四大系统组成。知识获取系统包括网络结构、输入学习样本模式的组织形式(将特定领域输入模式中各个抽象概

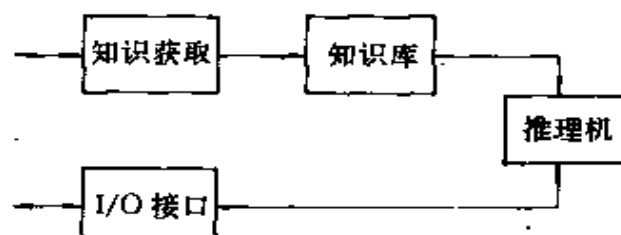


图 10.26 神经网络专家系统的基本结构

念转化为神经网络需要的模式形式的规则)、网络学习算法等。这一系统的主要职责是通过对样本的学习,获取知识并将其分布存贮在网络的连接权上。知识库系统主要表现为网络的连接权矩阵,矩阵元素值以知识获取系统取得。知识库可以不断刷新,表现为矩阵元素值的更新。这一系统的主要职能是为推理过程提供知识支持。推理机系统包括针对特定网络结构的网络运行算法。其主要职能是进行网络前向计算,完成由输入模式到输出模式的非线性映射,并对输出模式进行解释,将输出模式的数学表示转换为认识逻辑概念。I/O 接口是任何形式的专家系统所必备的常用子系统,是用户与系统对话的界面。其主要职能是完成系统参数的设定、数据的输入/输出等。

下面介绍一些神经网络专家系统的实际例子。

10.4.2 高炉炉温预测专家系统

高炉是炼铁的主要设备,铁矿石在高炉内通过冶炼还原成铁水,其主要生产过程如图 10.27 所示。在高炉内,原料铁矿石与燃料焦炭分层放置,在高炉的底部设有风口,由这个风口向高炉内吹入 1000°C 左右的热风,使焦炭燃烧。燃烧所产生的热量与一氧化碳促使铁矿石发生还原反应。这一反应过程随着含有一氧化碳的高温气体的逐渐上升和铁矿石的逐渐下降而不断进行。为了保证铁水质量的稳定,在高炉的生产过程中,必须通过控制吹入高炉热风量将炉内的温度保持在一定的范围内。由于炉温的变化是一个大惯性动态过程,因此若要精确地控制炉温,必须提前预测出炉温的变化,及时确定控制策略。但是高炉冶炼是一个很复杂的还原反应过程,影响炉温的因素也很复杂,这些都造成对高炉炉温预测的困难。日本川崎制铁(株)千叶制铁所运用实时神经网络专家系统进行炉温预测,得到了满意的预测结果^[33]。

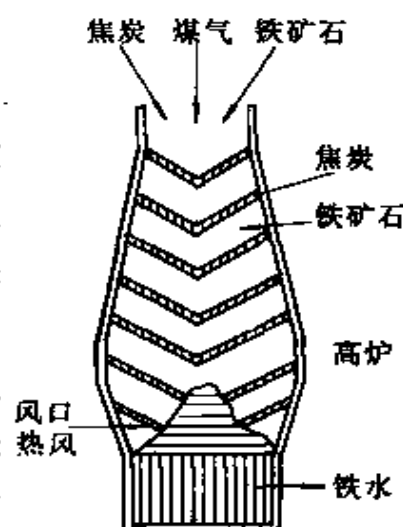


图 10.27 高炉生产工艺过程

实时神经网络炉温预测专家系统的基本思想是,以高炉炉内原料下降量、炉内一氧化碳的变化量、炉壁温度的变化量等动态数据作为网络的输入模式,利用神经网络的非线性映射与联想推理能力,得到有关炉温在将来 2~3 小时内的变化趋势信息。网络的结构如图 10.28 所示。网络由三层组成,其中输入层 12 个单元,输出层 3 个单元,中间层的单元数通过实验最后确定。系统采用误差逆传播算法进行网络学习与运行。

对于神经网络专家系统来说,网络的结构与算法确定之后,网络训练模式样本的选取成为系统成功与否的关键。从理论上讲,系统投入运行后面临的输入模式是无穷多的,但在网络的训练过程中不可能把所有这些模式都作为样本模式来学习,所以如何选择学习样本是一个很值得研究的问题。考虑到神经网络具有按相似输入产生相似输出的联想推理功能,炉温预测专家系统选择了一些包括炉内各参数上、下极限在内的典型数据作为网络的学习样

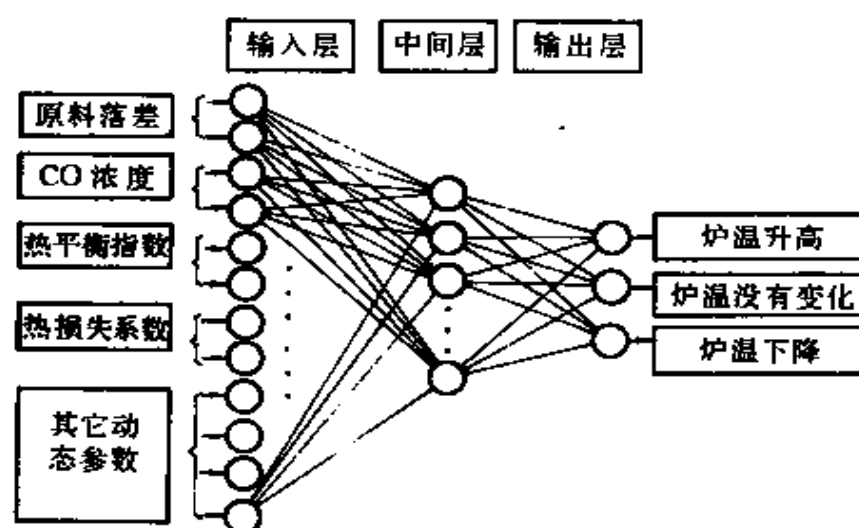


图 10.28 神经网络炉温预测专家系统

本。图 10.29 所选择样本中三个代表性的模式。图中 η_{CO} 为一氧化碳利用率, N_2 为炉内氮含量, 如图所示三个样本代表了炉内参数上、中、下三个典型数据。实践证明, 学习样本选择的是否合适将在很大程度上影响网络的工作性能。

	输入数据			示教输出		
	η_{CO}	N_2	其它	炉温高	不变	炉温低
样本 1	0.9	0.9	0.0	0.9	0.1	0.1
样本 2	-0.9	-0.9	0.0	0.1	0.1	0.9
样本 3	-0.1	0.1	0.1 ~ -0.1	0.1	0.9	0.1

图 10.29 学习样本实例

除了学习样本的选择之外,网络本身各种参数的选定也将对

系统的工作性能产生影响。图 10.30 为学习率 η 与网络学习回数的关系曲线。由图可以看出,由于学习率的选择不同,网络的学习回数有很大的差异。当然,针对不同的学习样本,网络最佳参数和选择也将有所不同,但变化不会很大,只需做一些小的调整。经过实验最后确定的网络参数如图 10.31 所示。这一系统经过对 21 组样本模式学习后在实际应用中收到了预期效果。

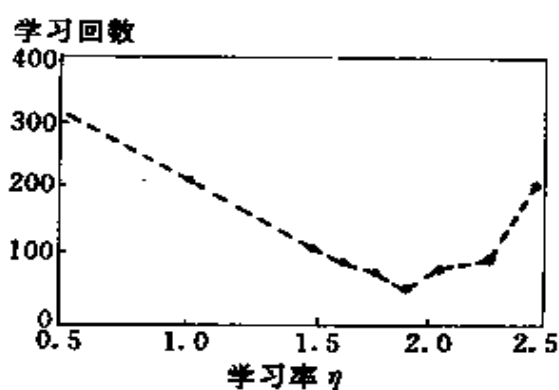


图 10.30 学习率 η 与学习回数的关系

参数项	参数值
中间层单元数	10
学习率 η	1.9
修正率 α	0.9
阈值函数参数 w_0	1.0

图 10.31 网络参数

10.4.3 财务评价专家系统

目的在于增加利润,减少损失的神经网络的财务分析专家系统目前已得到了成功的应用。其中主要形式有财务评价、财务预测及财务分析等。这里介绍一种用于评价抵押贷款对象的财务评价专家系统^[23]。

所谓对抵押贷款对象的评价是指金融部门对申请贷款或贷款保险的单位或个人的还贷可靠性进行财务评估,同时也是一种对借贷风险性的预测。预测的主要依据是借贷对象的资信地位、收入状况、债务负担、抵押金额、固有财产数等有关 100 条以上的信息。面对如此庞大的输入信息以及信息之间错综复杂的关系,单凭人

的经验是很难做出准确、无误的判断的,特别是贷款金额很大时,这种评价本身就具有一定的风险性,即使是很有经验的专家也不敢轻易下结论。应用专家系统可以帮助金融部门按照以往成功的贷款经验,从客观的角度对贷款对象做出正确的评价。

这一专家系统所用的神经网络由多层组成,总神经元数达6000个。图10.32所示是这一系统输入模式的主要项目及输出模式,即是否承诺贷款和贷款的风险性指标。系统的学习样本来自过去对各种复杂贷款对象做出的正确决断,即同意贷款或拒绝贷款。网络经过对学习样本进行充分学习之后成功地应用于实际。

输入模式项目

借贷方的“文化”地位

- { 信用审查结论
- { 扶养人数
- { 工作年限
- { 工作外职业
- { 固定资产状况

借贷方的“财务”地位

- { 当前年收入
- { 工作以外收入
- { 负债额

抵押证书

- { 贷款价额比
- { 抵押种类
- { 收入与抵押额之比
- { 贷款额

财产状况

- { 财产状况
- { 单元数
- { 评价额
- { 财产所在地

输出模式

贷款承诺/拒绝

风险性评价

图 10.32 贷款对象评价专家系统的输入、输出模式

10.4.4 医疗诊断专家系统

有关医疗诊断的神经网络的例子很多,这里举一个皮肤病诊断神经网络专家系统^[35]。此系统由三层前馈阶层网络构成,系统可以根据病人的症状诊断出10种皮肤病。网络的输入层有96个单元,分别对应病人的18种症状及各症状持续的时间,其中每一个症状都对应病人或检查者的一个可能回答。网络输出层有10个单元,每对单元对应于一种皮肤病。网络中间层有20个单元,作为输入症状模式的内部表示。网络的结构如图10.33所示。

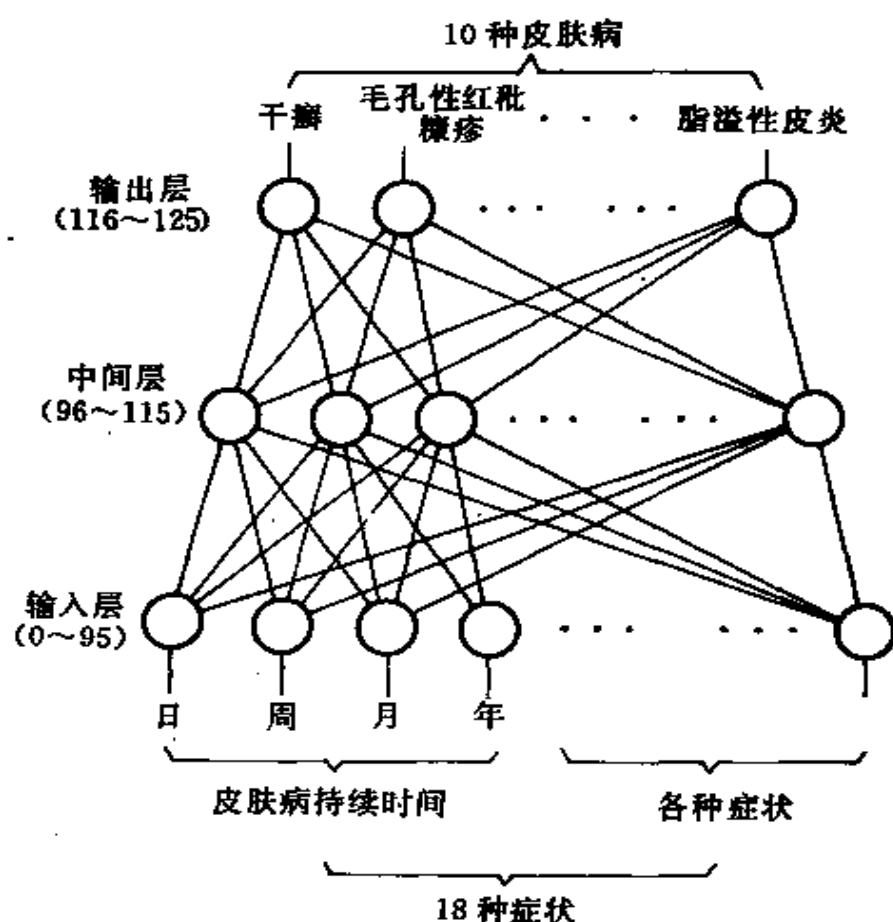


图 10.33 皮肤病诊断专家系统

系统用 500 位病人的病历记录,即 500 位由大夫做出正确诊断的病例组成网络的学习样本,其输入及希望输出模式分别对应病历中的病人症状记录及大夫诊断结果。网络用误差逆传播算法进行学习。经过充分学习后的网络系统获取了包含大夫成功的诊断经验和知识,并将这些经验和知识以数字的形式分布存贮在网络连接权中,从而构成了系统的知识库。当按运行规则对已学习过程样本进行诊断时,网络均可以得到正确的结果。而当把未学习过病例症状提供给网络时,网络的诊断准确率仍可达到 70% 以上。可以设想,当进一步扩大网络学习样本的规模和增加网络中间层单元后,系统的诊断准确率会有所提高。

神经网络医疗诊断专家系统与一般的医疗诊断系统相比,其最大的特点就是自联想推理功能。实际上大夫在对病人进行诊断时,一方面靠的是各种基本病理知识(这也是前人经验的积累),另一个很重要的方面是靠他在过去的医疗实践中总结、积累的大量临床诊断经验,而且无论是在运用理论知识还是自己的经验时,并不是机械地将所观察到的病人的症状与记忆中的诊断规则一一对号,直至丝丝入扣才能做出最后诊断,而是由相似性原理进行模糊推理获得诊断结果的。显然这种按相似输入产生相似输出的诊断方法比一般专家系统的那种严格按知识库中存贮的知识规则进行诊断的方法优越得多。

特别值得指出的是,对于系统通过学习获得的大夫的诊断经验与知识还可以从该系统中提取出来,即可通过 IF—THEN 规则解释推理结论。实际上,这些推理结论反映了症状与疾病之间的依赖关系,这些依赖关系有些甚至是至今大夫本人尚未注意到的。图 10.34 表示的是这种推理解释的一个例子。

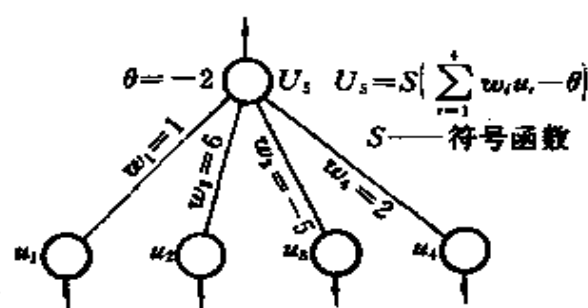


图 10.34 规则提取举例

如图所示,对推理输出 U_5 ,先置网络所有输入状态 u_1, u_2, u_3, u_4 为 0,即所有症状不出现,则此时 U_5 为 0($\because 0-2<0$),即无病症诊断输出。当输出模式取

$$U = (u_1, u_2, u_3, u_4) = (0, 1, 0, 0)$$

则因

$$\begin{aligned} & (0 \times 1 + 6 \times 1 + 0 \times (-5) + 2 \times 0) - 2 \\ & = 4 > 0 \end{aligned}$$

所以 $U_5 > 0$

由此可产生一条肯定性规则

IF u_2 为真 THEN U_5 肯定为真,

即有症状 u_2 必是皮肤病 U_5 。

如果保持 $u_2=1$ 不变,而增加 $u_3=1$,即

$$U = (0, 1, 1, 0)$$

则有

$$\begin{aligned} & (0 \times 1 + 6 \times 1 + 1 \times (-5) + 2 \times 0) - 2 \\ & = -1 < 0 \end{aligned}$$

即 $U_5 < 0$

此时一条否定性规则可由此产生

IF u_5 为假 THEN u_2 AND u_3 为真

即如果不是皮肤病 U_5 ,则肯定会出现症状 u_2 和 u_3 。

从原理上讲,对于一个神经网络专家系统,可以用类似方法将所有肯定性和否定性规则提取出来,这些规则构成了一般专家系统中的知识库。但依靠这种提取组成的一般知识库只有在神经网络的规模非常小时才有可能。即使对于只有一个输出单元的网络,随着输入单元的增多,相应的提取规则将成指数增长。

总之,神经网络医疗诊断系统通过病例学习获取诊断知识与经验,并按自联想推理产生诊断输出,这些都只是一般的医疗诊断专家系统所做不到的。

10.5 神经网络在智能控制中的应用

10.5.1 神经网络控制系统的特点

古典控制理论和现代控制理论都有一个共同的局限性,即被控制对象或过程的数学模型必须预先知道。但实际上有很多被控对象的模型十分复杂,以致无法建立精确的模型,特别是那些不确定性的系统或过程,尽管近年来系统辨识的理论与方法受到人们的重视,得到了很大发展,但在实际应用中特别是对实时控制系统来说,仍然存在一定的困难,往往不能满足时间和精度的要求。为了有效地解决这一问题,人们把人工智能、模糊数学以及专家系统引入控制系统中,逐渐形成了智能控制这一新的学科。各种自校正、自适应的智能控制方法不断出现并得到实际应用,作为人工智能中具有鲜明特色和广泛应用前景的人工神经网络理论也理所当然地被应用到工程控制领域中。实际上,从控制论的角度可以把神经网络理解为一高维非线性动力学系统,而基于神经网络的控制系统则可以看作是一种特殊形式的自适应控制系统。神经网络作为控制系统中的一个环节,常常被用作前馈和反馈补偿环节、自适应控制参数估计器和模糊控制器等。无论用作哪一种环节,由神经网络构成的控制系统都具有以下特点。

(1) 适应性

由于神经网络具有独特的非线性映射能力,所以适应能力很强。对于具有非线性特性的控制对象,用一般控制方法很难得到满意的控制效果,特别是当对象的非线性特性模型不清楚时,就更难选择有效的控制方法。而应用神经网络可以通过对网络的训练、掌握控制对象的非线性函数关系,从而做出恰当的控制决策。神经网络

络还可以用自联想推理功能解决时变系统的参数自适应问题。另外,由于神经网络本身的结构特点,还可以很容易组成多输入、多输出系统。

(2) 柔软性

由神经网络构成的控制系统具有很强的环境适应能力。对于那些不确定的系统,神经网络可以通过在线学习,及时跟随系统的变化,并且不需要改变网络的结构和算法。

(3) 鲁棒性

神经网络是由许多神经元构成的,它对系统特性的记忆表现为各个神经元之间的连接权值,且单个神经元在整个系统中不起决定性作用。一个经过训练的神经网络可以按相似输入模式产生相似的输出模式。因此,当系统因干扰或其它因素造成参数漂移或输入模式变形时,网络仍可以保证稳定的输出。

(4) 实时性

神经网络无论是学习还是运行都采用并行计算。虽然网络训练时比较费时,但在运用时因其采用单巡回并行计算,使响应速度非常快,很适于实时控制。

10.5.2 神经网络控制系统的几种结构形式

(1) 网络作为控制补偿环节的系统结构

这种系统的基本结构如图 10.35 所示。图中 d 是期望输出, y' 是实测输出, y 是实际控制输出, $u = u_1 + u_2$ 为控制输入。其中前馈补偿环节和反馈补偿环节可以用多层神经网络构成。通过网络的学习逐渐改变网络的连接权,使动力学系统的期望响应和实际响应之间的误差信号最小。由于误差信号是输入到反馈控制器中的,所以当误差信号变小时,网络的训练学习将会使网络慢慢地由反馈过程过渡到前馈过程,这说明网络已通过学习获取了原先不清楚或未考虑的动力学系统特性,提高了系统的控制精度。

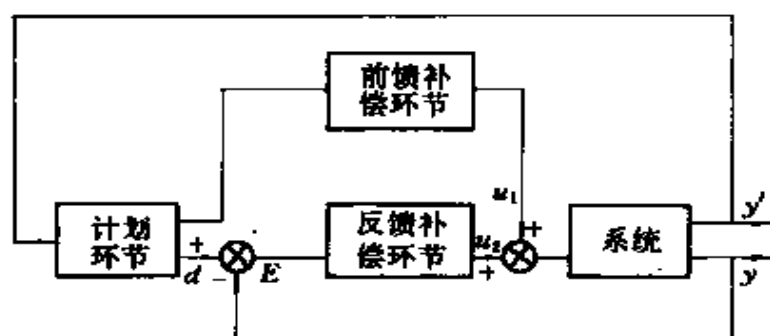


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

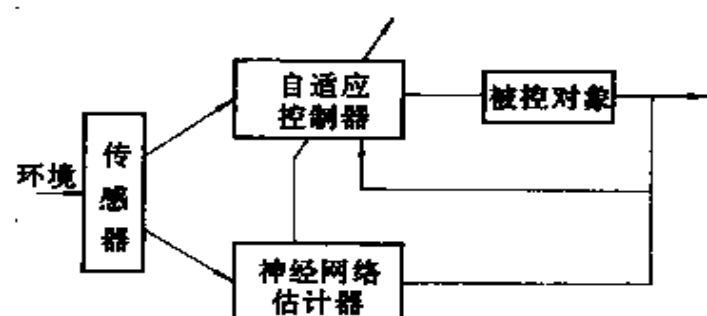


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

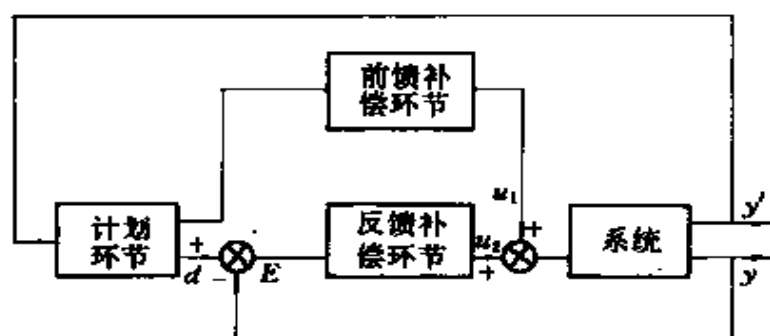


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

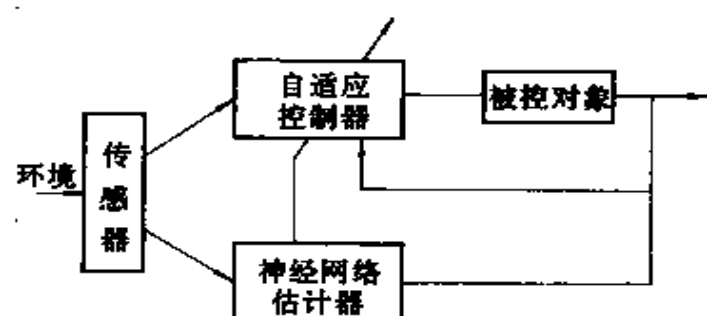


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

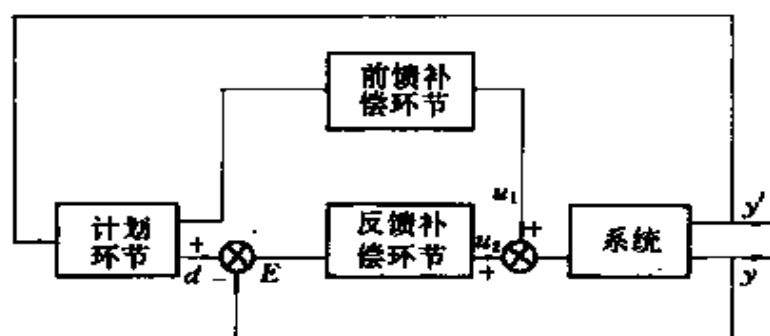


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

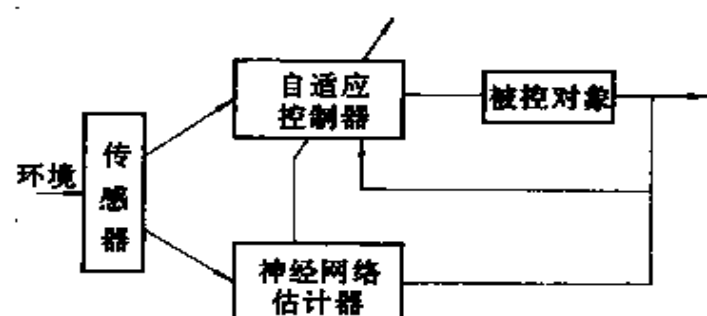


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

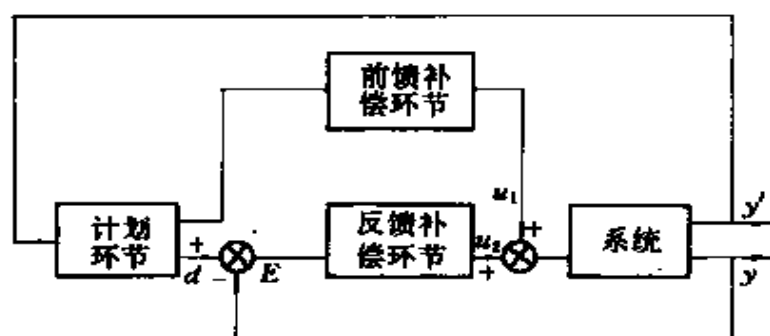


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

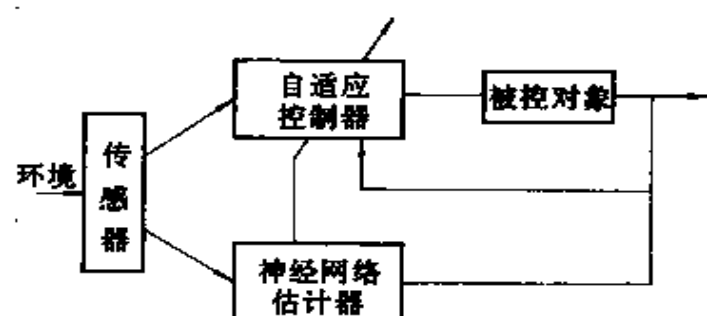


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

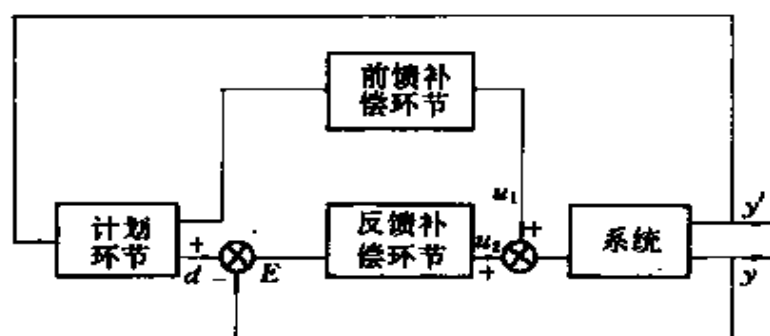


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

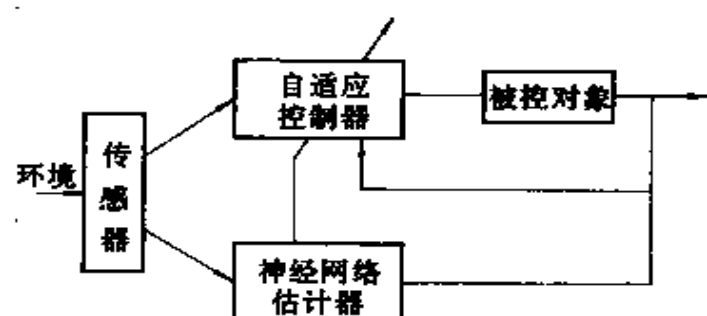


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

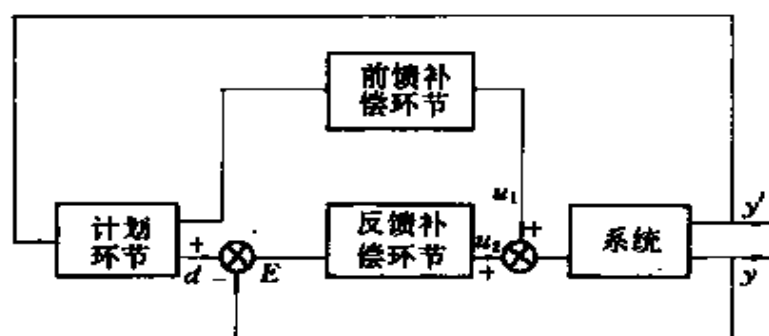


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

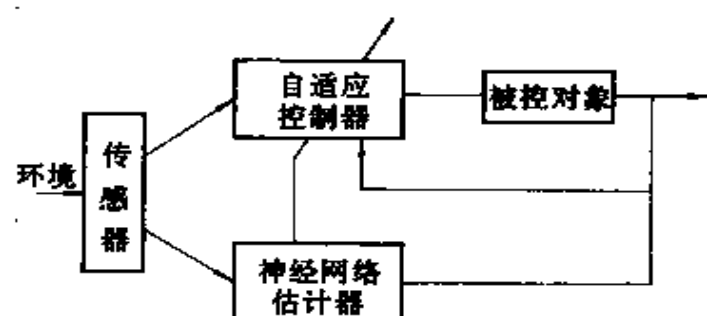


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是,以外界输入信号(环境变化)及系统的控制输出作为神经网络估计器的输入,以系统的控制规律作为约束条件,以对系统的控制特性要求作为目标函数建立网络能量函数,并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数,并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好(因为网络实行并行计算),且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

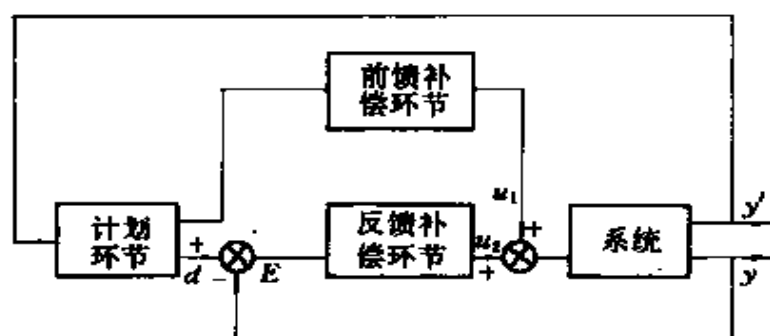


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

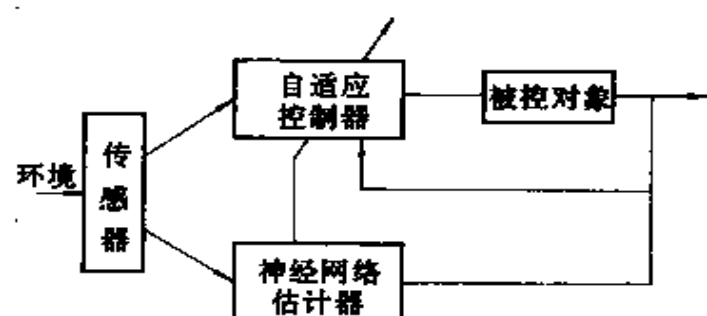


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

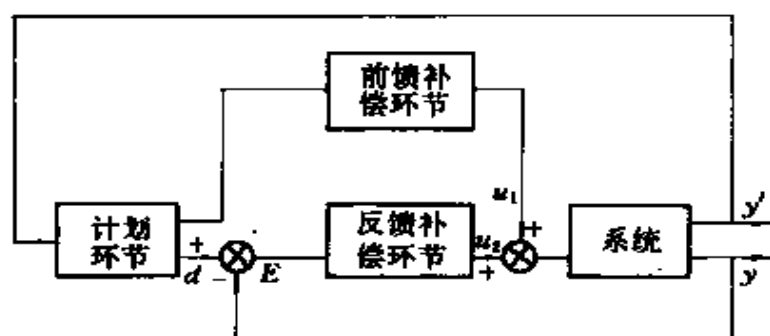


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

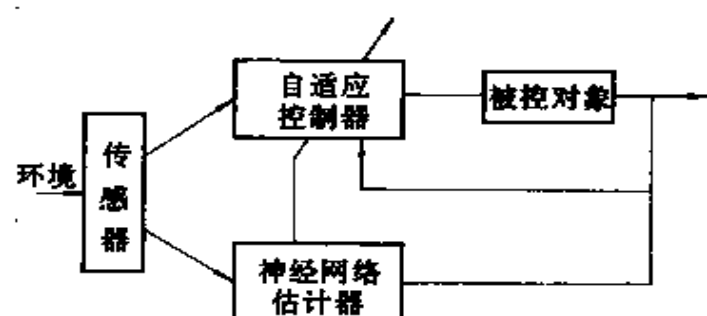


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

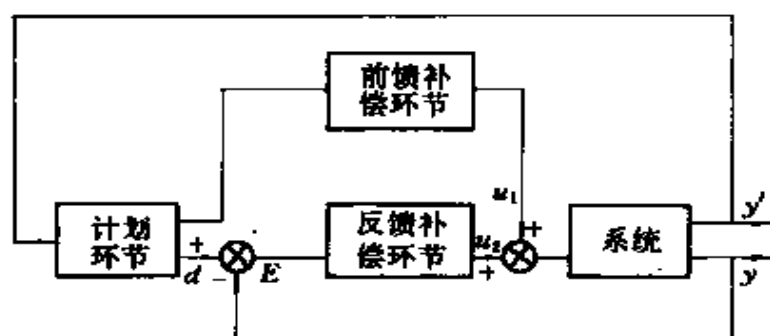


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

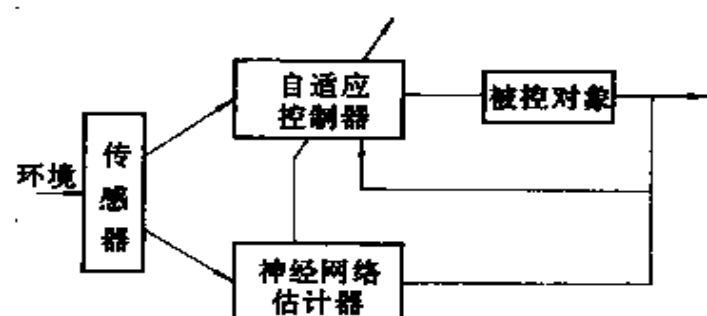


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

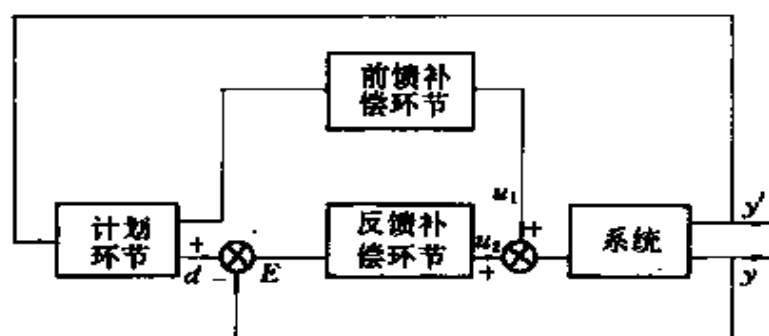


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

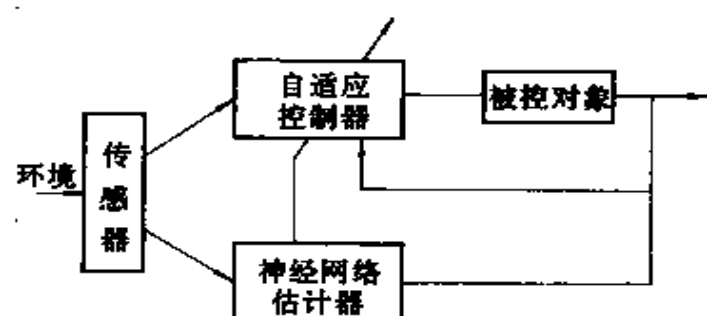


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是,以外界输入信号(环境变化)及系统的控制输出作为神经网络估计器的输入,以系统的控制规律作为约束条件,以对系统的控制特性要求作为目标函数建立网络能量函数,并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数,并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好(因为网络实行并行计算),且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

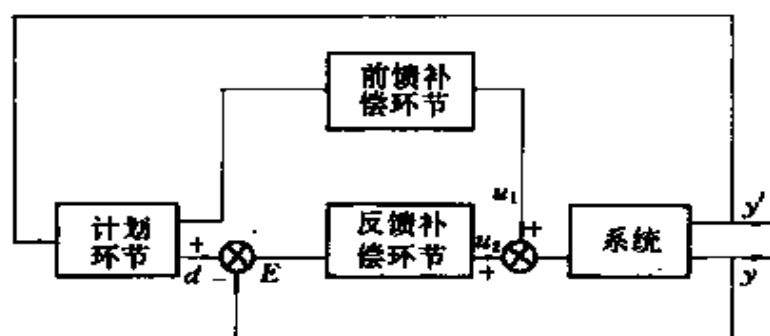


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

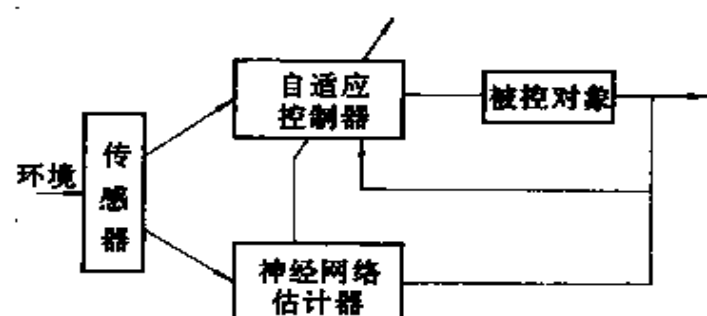


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

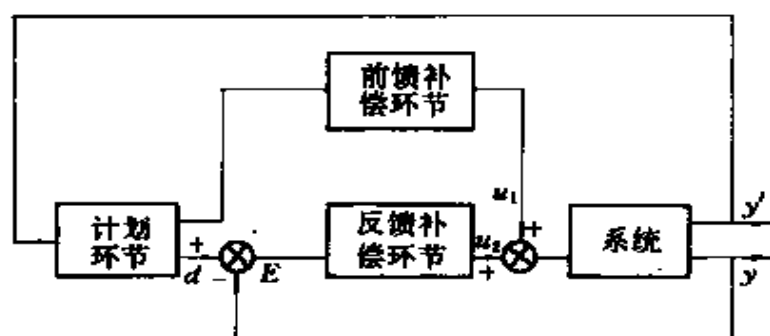


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

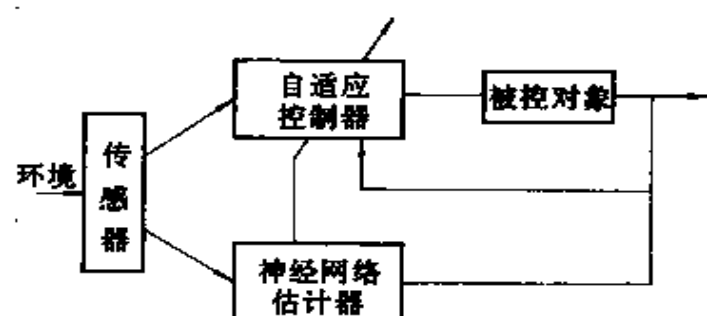


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

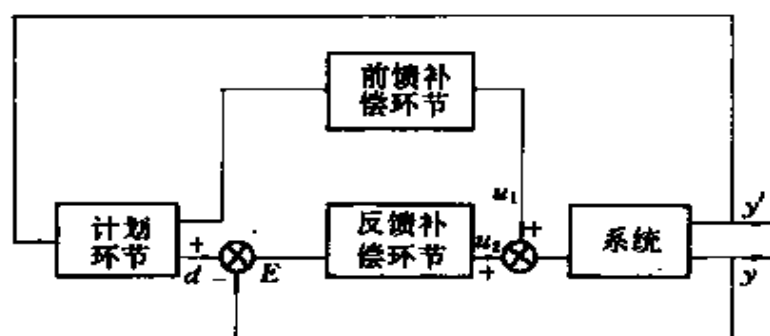


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

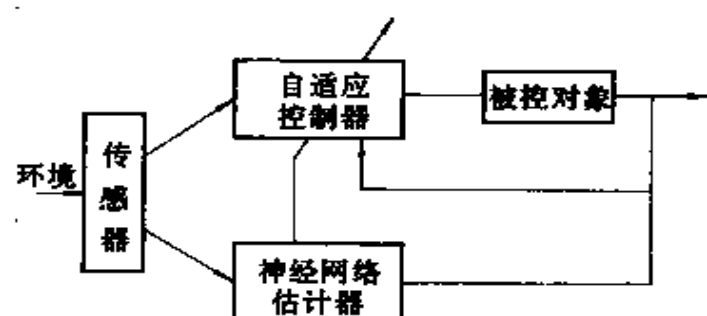


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

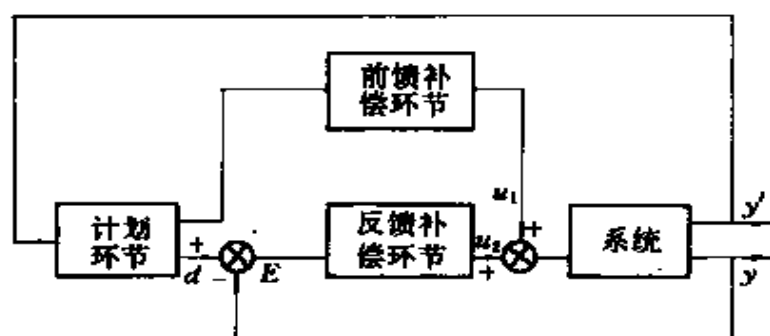


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

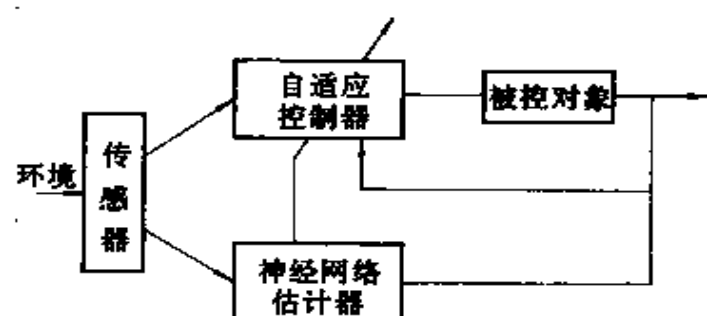


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

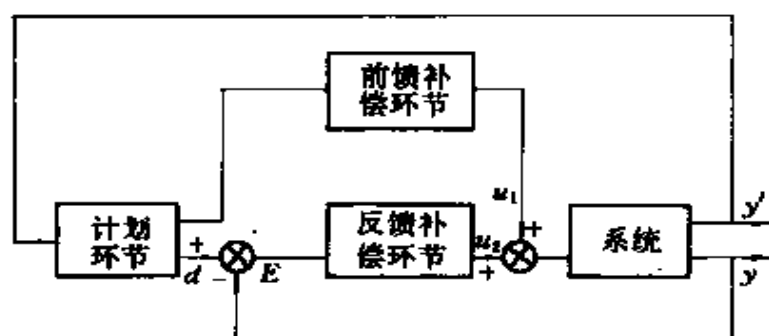


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

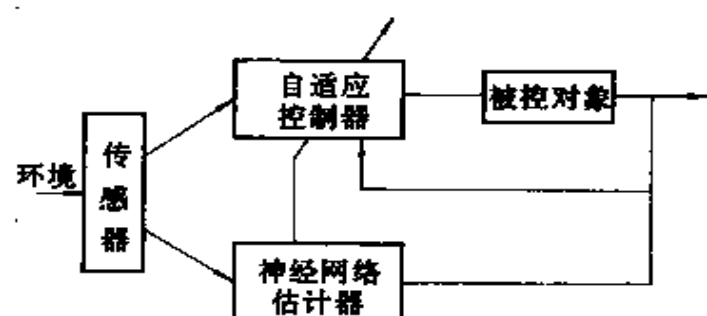


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是,以外界输入信号(环境变化)及系统的控制输出作为神经网络估计器的输入,以系统的控制规律作为约束条件,以对系统的控制特性要求作为目标函数建立网络能量函数,并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数,并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好(因为网络实行并行计算),且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

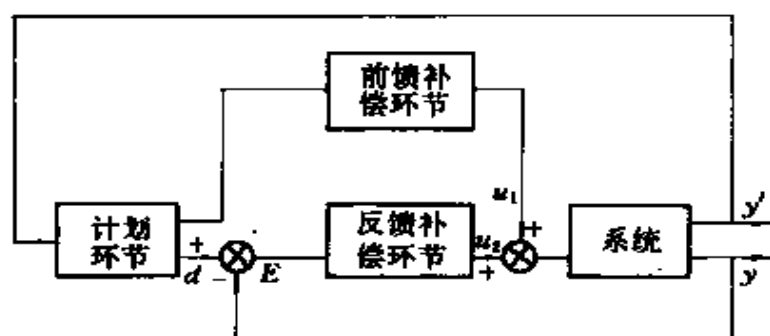


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

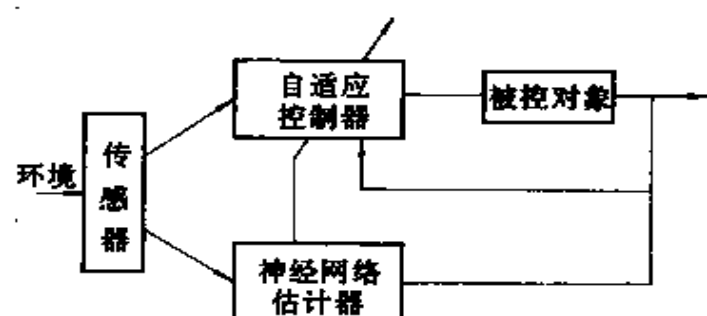


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

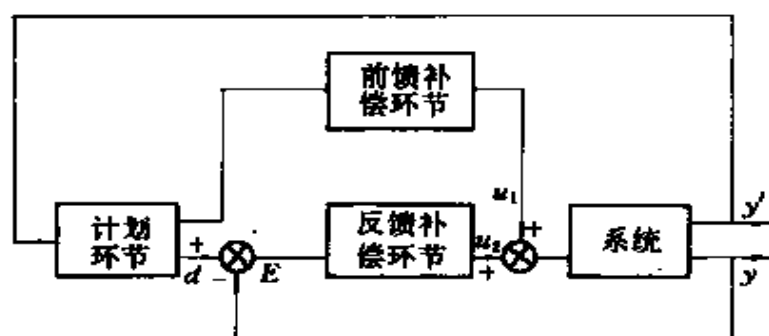


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

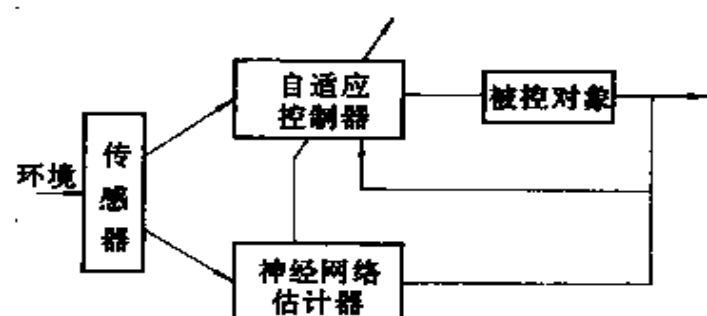


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是,以外界输入信号(环境变化)及系统的控制输出作为神经网络估计器的输入,以系统的控制规律作为约束条件,以对系统的控制特性要求作为目标函数建立网络能量函数,并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数,并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好(因为网络实行并行计算),且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

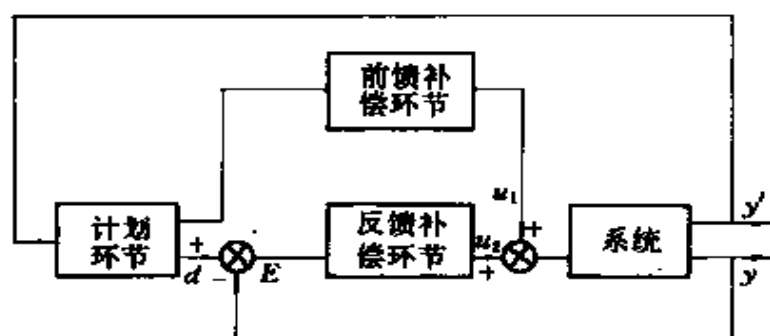


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

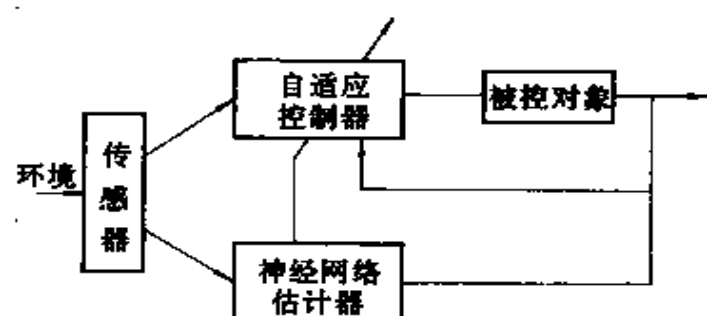


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是，以外界输入信号（环境变化）及系统的控制输出作为神经网络估计器的输入，以系统的控制规律作为约束条件，以对系统的控制特性要求作为目标函数建立网络能量函数，并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数，并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好（因为网络实行并行计算），且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

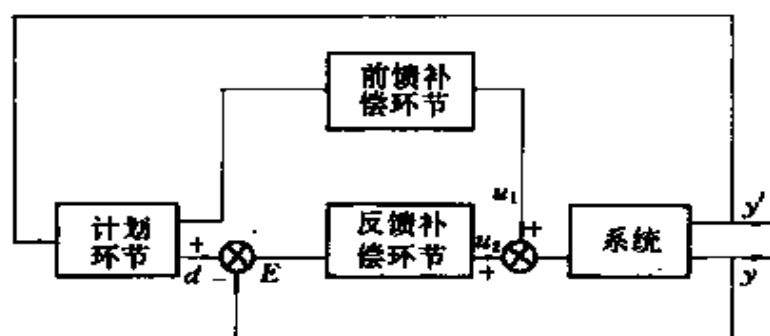


图 10.35 神经网络补偿控制系统框图

(2) 网络作为自适应控制参数估计器的系统结构

这种系统的基本结构如图 10.36 所示。

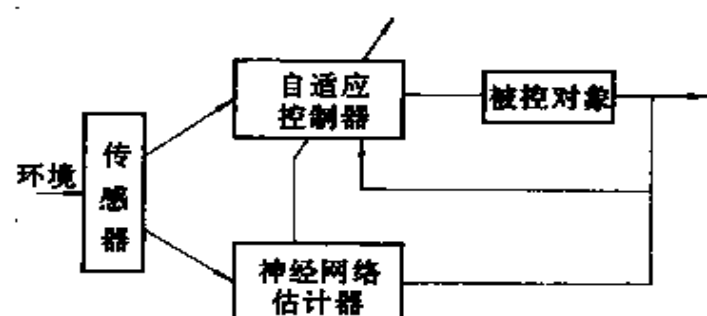


图 10.36 具有神经估计器的自适应控制系统框图

这一系统的基本思想是,以外界输入信号(环境变化)及系统的控制输出作为神经网络估计器的输入,以系统的控制规律作为约束条件,以对系统的控制特性要求作为目标函数建立网络能量函数,并设计出 Hopfield 网络拓扑结构。按 Hopfield 网络算法找到系统的最优参数,并提供给系统的自适应控制器。这种参数估计器的最大特点是寻优速度快、效果好(因为网络实行并行计算),且几乎与参数个数无关。

(3) 网络作为模糊控制器的系统结构

这种系统的基本结构如图 10.37 所示。

