神经网络、模糊系统及其在运动控制中的应用

丛　　爽　著

中国科学技术大学出版社

2001·合肥

内 容 简 介

这是一本关于人工神经网络、模糊逻辑系统，以及模糊神经网络研究成果的专著。作者在简要介绍神经网络理论与模糊集合理论的基础上，对人工神经网络与模糊逻辑控制系统的设计及其应用，以及两者之间的相互关系和相互结合，进行了较深层次的理论分析与综合，并结合其在运动控制中的建模与控制的各种实际应用，使读者能够从中了解和掌握运用模糊神经系统的理论与技术进行实际的系统设计和灵活应用的方法。

本书内容新颖，材料翔实，自成体系。既有理论分析与综合，又有实际系统的设计与应用。本书既可作为高校自动化专业、机械专业、电力电气专业、计算机应用专业及其他相关专业的高年级本科生和研究生教学用书，也可供从事智能科学、系统科学、计算机科学、应用数学、自动控制等领域研究的广大科技人员阅读。

图书在版编目(CIP)数据

神经网络、模糊系统及其在运动控制中的应用/丛爽著. -合肥:中国科学技术大学出版社，2001.5

ISBN7-312-01270-1

Ⅰ. 神… Ⅱ. 丛… Ⅲ. ①人工神经元网络-应用-运动控制②模糊工程-应用-运动控制 Ⅳ. TP24

中国版本图书馆 CIP 数据核字（ 2001）第 08878 号

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路 96 号, 230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本：787×1092/16 **印张：**15.625 **字数：**400 千

2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

印数：1—3 000 册

ISBN7-312-01270-1/TP·270 **定价：**23.00 元

(凡图书出现印装质量问题，请向承印厂要求调换)

前　言

智能控制是一门新兴的交叉学科，模糊逻辑和神经网络正是该学科发展和研究的关键技术内容。在最近的几年中，我们已经看到模糊逻辑和神经网络以各自的优势进行相互渗透，所形成的模糊神经系统在各种优化技术的配合下，其应用在数量和种类上都得到迅速的增长，其范围涉及到各行各业。模糊神经系统在智能系统的概念和设计中起到越来越大的重要作用。

运动控制也是近十年来国际上形成的一个研究热点，它是一门综合性多学科的交叉技术。由于运动控制本身并不产生新的理论，它主要是应用先进的智能控制策略达到高质量的运动控制效果，所以，将其作为模糊与神经系统应用对象是非常典型和合适的。本书在前后章节的衔接上强调相互之间的联系和系统性，注重分析各种不同理论和方法之间的相关性，讲究概念和思路的清晰。

笔者根据多年来从事先进控制策略及其在运动控制中应用的研究工作，特别是在模糊神经网络和遗传算法的研究中所取得的研究成果，撰写了这本关于人工神经网络、模糊逻辑系统以及模糊神经网络理论与应用的专著。全书共分 16 章，系统地介绍了模糊逻辑与神经网络的基本概念和理论、各种高效率的神经网络的学习算法；近年来最新发展的模糊神经网络的理论，包括模糊神经系统的构成、原理以及与其他函数之间的相互关系；介绍了神经网络、模糊逻辑、模糊神经，结合遗传算法在运动控制中的应用，给出了许多代表性的应用实例，在对人工神经网络与模糊逻辑控制系统的设计及其应用，以及两者之间的相互关系和相互结合上，进行了较深层次的理论分析与综合，对在校高年级本科生和研究生以及相关专业的科技人员了解和掌握智能策略与方法、解决复杂控制的实际问题具有指导意义。

本书取材新颖，内容深入浅出，便于自学与应用，既可作为高校自动化专业、机械专业、电力电气专业、计算机应用专业及其他相关专业的高年级本科生和研究生教学用书，也可供从事智能科学、系统科学、计算机科学、应用数学、自动控制等领域研究的广大科技人员阅读。

本书的部分内容由已发表的学术论文改编而成，在此对论文的合作者高雪鹏、钱镇、胡晟、林航、向微表示感谢。

本人才疏学浅，对于疏漏不妥之处，恳请读者指教。

丛　爽

2001 年 5 月

于中国科学技术大学

目　　次

前　言 …… (Ⅰ)

第 1 章 典型前向神经网络 …… (1)

1.1 感知器网络 …… (4)

1.1.1 感知器的网络结构及其功能 …… (5)

1.1.2 感知器权值的学习规则与训练 …… (6)

1.2 自适应线性元件 …… (11)

1.2.1 自适应线性神经元模型和结构 …… (11)

1.2.2 W－H 学习规则及其网络的训练 …… (12)

1.3 反向传播网络 …… (13)

1.3.1 反向传播网络模型与结构 …… (13)

1.3.2 BP 算法 …… (15)

1.3.3 BP 网络的设计 …… (17)

1.3.4 BP 网络的限制与不足 …… (19)

第 2 章 网络训练优化算法 …… (20)

2.1 基于标准梯度下降的方法 …… (21)

2.1.1 附加动量法 …… (21)

2.1.2 自适应学习速率 …… (22)

2.1.3 弹性 BP 算法 …… (23)

2.2 基于数值优化方法的网络训练算法 …… (23)

2.2.1 拟牛顿法 …… (24)

2.2.2 共轭梯度法 …… (25)

2.2.3 Levenberg－Marquardt 法 …… (26)

2.3 数值实例对比 …… (27)

2.3.1 非线性函数的逼近 …… (27)

2.3.2 逼近非线性直流电机的输入/输出特性 …… (29)

2.4 小　结 …… (31)

第 3 章 BP 网络在智能系统中的建模与控制 …… (32)

3.1 直接正向模型建立 …… (32)

3.2 逆模型建立 …… (33)

3.3 系统中的控制 …… (35)

3.3.1 监督式控制 …… (35)

3.3.2 直接逆控制 …… (35)

3.3.3 模型参考控制 …… (36)

3.3.4 内模控制 …… (36)
第 4 章 反馈网络 …… (37)
4.1 霍普菲尔德网络模型 …… (38)
4.2 DHNN 的学习规则 …… (38)
4.2.1 海布学习规则 …… (38)
4.2.2 正交化的权值设计 …… (40)
4.3 离散型反馈网络的稳定点与稳定域 …… (43)
4.3.1 两个输入神经元的情况 …… (45)
4.3.2 网络输入神经元为三个时的情况分析 …… (49)
4.3.3 小结 …… (51)
4.4 连续型霍普菲尔德网络 …… (51)
4.4.1 对应于电子电路的网络结构 …… (52)
4.4.2 霍普菲尔德能量函数及其稳定性分析 …… (54)
4.4.3 能量函数与优化计算 …… (57)
4.5 用 CHNN 求解 TSP 问题 …… (58)
4.5.1 网络设计 …… (59)
4.5.2 对终态时系统的输出向量 V 的解释 …… (61)
4.5.3 用 CHNN 算法实现 TSP 的问题探讨 …… (62)
4.5.4 各参数的影响 …… (64)
4.5.5 小结 …… (67)
第 5 章 自组织竞争人工神经网络 …… (68)
5.1 几种联想学习规则 …… (68)
5.1.1 内星学习规则 …… (69)
5.1.2 外星学习规则 …… (70)
5.1.3 科荷伦学习规则 …… (71)
5.2 自组织竞争网络 …… (71)
5.2.1 网络结构 …… (71)
5.2.2 竞争学习规则 …… (73)
5.2.3 竞争网络的训练过程 …… (74)
5.3 科荷伦自组织映射网络 …… (76)
5.3.1 科荷伦网络的拓扑结构 …… (77)
5.3.2 网络的训练过程 …… (78)
5.3.3 科荷伦网络的应用 …… (78)
5.4 小 结 …… (79)
第 6 章 径向基函数网络 …… (80)
6.1 径向基函数及其网络分析 …… (80)
6.2 网络的训练与设计 …… (82)
6.3 广义径向基网络 …… (83)

6.4 数字应用对比及性能分析 …… (84)
6.5 小 结 …… (85)
第 7 章 模糊理论基础 …… (86)
7.1 引 言 …… (86)
7.2 模糊集合及其隶属函数 …… (88)
7.2.1 模糊集合的定义 …… (88)
7.2.2 模糊集合的表示方法 …… (89)
7.2.3 模糊集合的并、交、补运算 …… (90)
7.2.4 模糊集合的隶属函数 …… (90)
7.3 模糊逻辑 …… (92)
7.3.1 二值逻辑、多值逻辑和模糊逻辑 …… (92)
7.3.2 模糊逻辑的基本运算 …… (92)
7.3.3 模糊关系和模糊矩阵 …… (94)
7.3.4 模糊语言及其算子 …… (96)
7.4 模糊规则与模糊推理 …… (99)
7.4.1 模糊‘如果—那么’规则 …… (99)
7.4.2 模糊逻辑推理 …… (100)
第 8 章 模糊控制器的设计方法 …… (110)
8.1 精确与模糊控制的事例 …… (110)
8.1.1 采用精确的非模糊求解方法 …… (110)
8.1.2 模糊方法 …… (113)
8.2 模糊逻辑控制过程 …… (114)
8.3 输入变量和输出变量的确定 …… (115)
8.4 论域的确定 …… (116)
8.5 确定模糊化和解模糊化方法 …… (117)
8.5.1 模糊化方法 …… (118)
8.5.2 解模糊判决方法 …… (120)
8.6 模糊控制规则 …… (122)
8.7 模糊逻辑推理 …… (123)
8.7.1 合成模糊推理法 …… (123)
8.7.2 结论是线性函数的模糊推理方法 …… (132)
8.7.3 量化因子及比例因子的选择 …… (133)
第 9 章 运动控制中的摩擦力补偿及其建模技术 …… (135)
9.1 引言 …… (135)
9.2 摩擦学及实验上提出的非线性摩擦力模型 …… (136)
9.3 机械控制工程上采用的分析与补偿法 …… (137)
9.4 运动控制 …… (139)
9.5 运动控制中库仑摩擦力的结构分析 …… (140)
9.6 基于模型的摩擦力前向补偿器的设计 …… (143)

9.7 线性模型的参数辨识 …… (144)
9.7.1 基于不同原理的辨识方法 …… (144)
9.7.2 MATLAB 中系统模型辨识的描述方法 …… (147)
9.7.3 库仑摩擦力参数的辨识 …… (148)
9.8 运动控制中的机电控制系统 …… (149)
9.8.1 机械系统 …… (149)
9.8.2 电子学技术 …… (149)
9.8.3 先进的计算机控制 …… (150)
9.8.4 自适应运动控制应用的例子 …… (151)
9.9 非线性直流电机仿真模型系统的建立 …… (153)
9.9.1 被控过程线性段模型的参数辨识 …… (153)
9.9.2 非线性模型的建立及仿真系统的实现 …… (155)
第 10 章 模糊控制系统的应用 …… (158)
10.1 速度模糊控制器的设计 …… (158)
10.2 三种控制器的设计与性能比较 …… (163)
10.2.1 控制算法的描述 …… (163)
10.2.2 结果的对比 …… (165)
10.3 变参数双模糊控制器 …… (167)
10.3.1 参数的设定 …… (167)
10.3.2 仿真实验验证 …… (169)
10.3.3 小结 …… (171)
第 11 章 神经网络的应用 …… (172)
11.1 BP 网络结构、参数及训练方法的设计与选择 …… (172)
11.1.1 BP 网络的设计 …… (172)
11.1.2 采用自适应学习速率与固定学习速率的比较 …… (176)
11.1.3 改进算法的性能比较 …… (177)
11.2 神经网络在电机非线性补偿中的设计与实现 …… (178)
11.2.1 问题的提出 …… (178)
11.2.2 伺服电机神经网络仿真器的设计 …… (178)
11.2.3 神经网络补偿器的设计 …… (180)
11.2.4 神经网络控制系统 …… (181)
11.2.5 实验测试结果 …… (182)
11.2.6 小结 …… (183)
第 12 章 模糊神经网络 …… (184)
12.1 引 言 …… (184)
12.2 模糊系统的关系式 …… (184)
12.3 用神经网络直接实现的模糊系统 …… (186)
12.4 Sugeno 模糊推理法 …… (187)

12.5 B样条模糊神经网络 …… (188)
12.5.1 B样条函数及其网络 …… (188)
12.5.2 B样条模糊神经网络控制器的设计 …… (191)
12.6 径向基函数神经网络 …… (192)
12.7 小 结 …… (195)
第13章 模糊神经系统的应用 …… (196)
13.1 基于ANFIS的非线性电机系统的建模 …… (196)
13.1.1 ANFIS的结构 …… (196)
13.1.2 混合学习算法 …… (198)
13.1.3 非线性电机系统建模 …… (198)
13.1.4 基于ANFIS的建模 …… (199)
13.1.5 辨识模型的验证 …… (200)
13.1.6 小结 …… (202)
13.2 用自组织竞争网络优化模糊神经网络的结构 …… (202)
13.2.1 自组织竞争神经网络 …… (203)
13.2.2 具有最佳结构与参数的模糊神经网络控制器的设计 …… (204)
13.2.3 小结 …… (207)
第14章 遗传算法 …… (208)
14.1 遗传算法的基本特点 …… (209)
14.2 遗传算法的基本操作 …… (210)
14.3 遗传算法的设计步骤 …… (212)
14.4 遗传算法的实质 …… (213)
14.5 小 结 …… (214)
第15章 遗传算法的应用 …… (215)
15.1 采用遗传算法提高神经网络模型辨识的精度 …… (215)
15.1.1 引言 …… (215)
15.1.2 改进的遗传算法 …… (216)
15.1.3 实例验证 …… (217)
15.1.4 小结 …… (218)
15.2 模糊神经网络和遗传算法相结合的控制策略 …… (218)
15.2.1 引言 …… (218)
15.2.2 优化控制系统的结构 …… (219)
15.2.3 优化仿真结果的对比和分析 …… (222)
15.2.4 小结 …… (223)
第16章 模拟退火算法及其应用 …… (224)
16.1 Metropolis准则和模拟退火算法 …… (224)
16.2 模拟退火算法的设计步骤 …… (225)
16.3 应用模拟退火算法求解TSP问题 …… (226)

16.3.1 TSP 问题的求解步骤 …… (227)
16.3.2 冷却进度表的选取 …… (228)
16.3.3 求解 TSP 问题的程序实现 …… (230)
16.3.4 模拟退火算法的性能对比 …… (232)
16.4 模拟退火算法的改进 …… (235)
参考文献 …… (236)

第1章 典型前向神经网络

人工神经网络(Artificial Neural Network, 简称 ANN）是由大量简单的处理单元组成的非线性、自适应、自组织系统，它是在现代神经科学研究成果的基础上，试图通过模拟人类神经系统对信息进行加工、记忆和处理的方式，设计出的一种具有人脑风格的信息处理系统。人脑是迄今为止我们所知道的最完善最复杂的智能系统，它具有感知识别、学习、联想、记忆、推理等智能，人类智能的产生和发展经历了漫长的进化过程，而人类对智能处理的新方法的认识主要来自神经科学。虽然人类对自身脑神经系统的认识还非常有限，但已设计出像人工神经网路这样具有相当实用价值和较高智能水平的信息处理系统。

按其信息流向来分类，人工神经网络可以被分成前向网络和反馈网络。本章将对典型前向网络的结构、功能及其性能予以介绍，其中包括感知器、自适应线性元件和反向传播网络。

人工神经网络是通过计算机软件或电子线路硬件构成的。它是由最简单的人工神经元并联，或者再加上串联组成。人们通常用图形来表示网络系统的输入到输出的转化关系。单个神经元可以表示为如图 1.1 所示的模型结构，其中，神经元输入矢量用矩阵形式可以表示为：

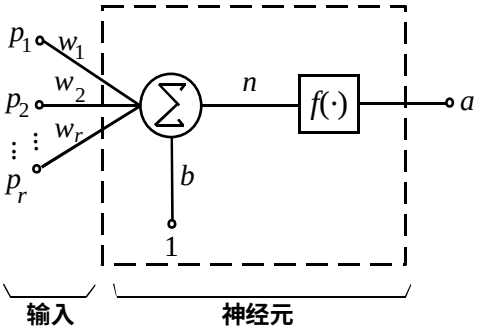


图 1.1 单个神经元模型结构

$$P = [p_1\ p_2 \Lambda\ p_r]^T, \quad \text{权矩阵 } W = [w_1\ w_2 \Lambda\ w_r]$$

单个神经元输出 A 与输入 P 之间的对应关系可表示为：

$$A = f\left(\sum_{j=1}^{r} w_j p_j + b\right) = f(W*P + b) \qquad (1.1)$$

其中，b 称为阀值，或偏差，f 代表某种激活函数关系式。

可以说单个神经元就是一个多输入/单输出的系统。两个或更多的单神经元相并联则构成单层神经网络，如图 1.2 所示。

两个及其以上单层神经网络相级联则构成多层神经网络，如图 1.3 所示。

由于在网络输入/输出关系式中对所使用的变量采用了矩阵形式来表达，所以对于单层多输出神经网络，其表达形式与单个神经元时完全相同。如图 1.2 所表示的神经元个数为 s 的单层神经网络的输入/输出关系式可写为：

$$A_{s\times 1} = F(W_{s\times r} * P_{r\times 1} + B_{s\times 1}) \qquad (1.2)$$

一个多层神经网络的输入/输出关系，可以以单层神经网络的关系进行递推。将前一层的输出作为后一层的输入，并采用与单层的神经网络同样的书写方式即可方便地写出。如

图 1.3 所示的一个具有 3 层神经网络的输入/输出关系可写为：

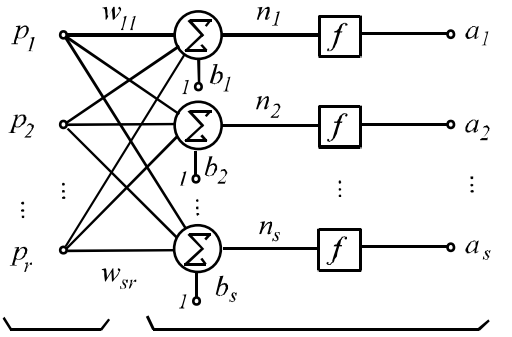
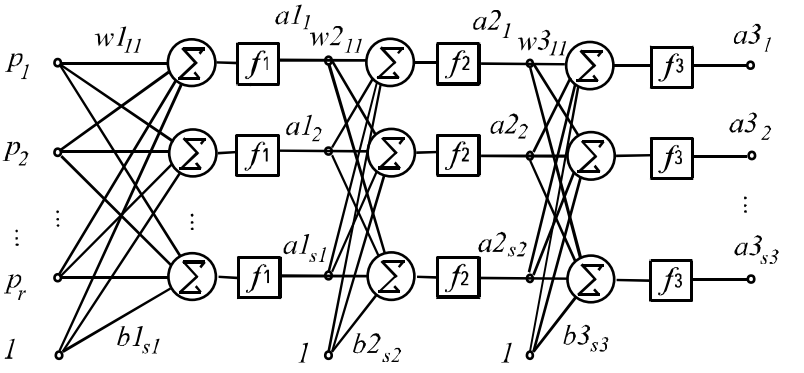
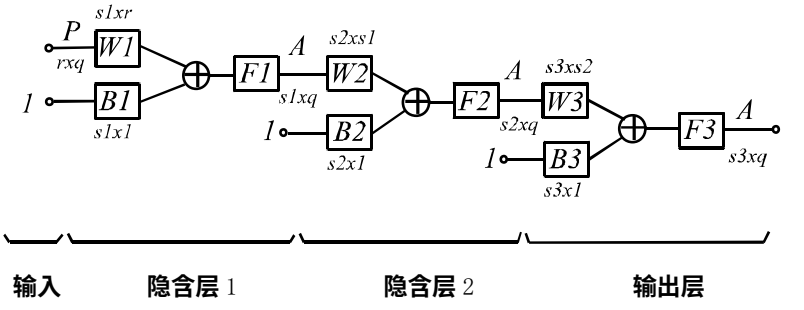


图 1.2 单层神经元网络模型结构图



（a） 三层的神经网络结构图



（b）（a）所示神经网络结构的简化图

图 1.3 多层神经网络

$$
\begin{aligned}
A1 &= F1(W1*P+B1)\\
A2 &= F2(W2*A1+B2)\\
A3 &= F3(W3*A2+B3)\\
&= F3\{W3*F2[W2*F1(W1*P+B1)+B2]+B3\}
\end{aligned}
\tag{1.3}
$$

对于这种标准全联接的多层神经网络，更一般的简便作图是仅画出输入节点和一组隐

含层节点外加输出节点及其连线来示意表示，如图 1.4 所示。图中只标出输入、输出和权矢量，完全省去激活函数的符号。完整的网络结构则是通过具体的文字描述来实现的，如：网络具有一个隐含层，隐含层中具有 5 个神经元并采用 S 型激活函数，输出层采用线性函

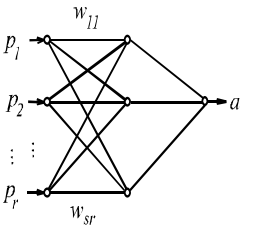


图 1. 4 神经网络结构示意图

数，或者更简明的可采用‘ 网络采用 2-5-1 结构 ”来描述，其中，2 表示输入节点数；5 表示隐含层节点数；1 为输出节点数。如果不对激活函数作进一步的说明，则意味着隐含层采用 S 型函数，而输出层采用线性函数。

特别值得强调的是，在设计多层网络时，隐含层的激活网络应采用非线性的，否则多层网络的计算能力并不比单层网络更强。因为在采用线性激活函数的情况下，由于线性激活函数的输出就等于网络的加权输入和，即 $A = W * P + B$，如果将偏差作为一组权值归为 W 中统一处理，两层网络的输出 $A2$ 则可以写为：

$$\begin{aligned} A2 &= F2(W2 * A1) \\ &= W2 * F1(W1 * P) \\ &= W2 * W1 * P \\ &= W * P \end{aligned} \tag{1.4}$$

其中，$F1 = F2 = F$ 为线性激活函数。上式表明两层线性网络的输出计算等效于具有权矢量为 $W = W2*W1$ 的单层线性网络的输出。

由此可见，人工神经网络工作时，所表现出的就是一种计算，利用人工神经网络求解问题时所利用的也正是网络输入到网络输出的某种关系式。与其他求输入/输出关系式方法不同的是，神经网络的输入/输出关系式是根据网络结构写出的，并且网络权值的设计往往是通过训练而不是根据某种性能指标计算出来的。所以，应用神经网络解决实际问题的关键在于设计网络，而网络的设计主要包括两方面的内容，一是网络结构，另一个是网络权值的确定。第一个方面涉及到对不同网络结构所具有的功能及其本质的认识，第二个方面涉及到对不同网络权值训练所用的学习规则的掌握。所以无论是作为学习、应用，还是作为更深层次的理论研究，这两步都是最重要的。正因为如此，人们对人工神经网络进行分类时，也是有两种方法，一种是根据网络结构分为前向网络和反馈网络两大类，另一种是按训练权值方法分为监督式（ 或有教师）网络和无监督式（ 无教师）网络两种。

在神经网络的输入与输出之间所存在的关系函数，类似于控制系统中的转移函数，称为激活函数，由它决定网络的线性和非线性，它是网络特性及功能的关键所在。在控制系统中，一般对控制器的设计是根据某个性能指标来设计控制器的结构或参数，与此不同，在神经网络信息处理系统中，设计者是根据期望网络完成的目标（ 或任务）以及某种调整网络权值所采用的学习规则来确定网络的结构与权值。由于网络的权值一般不是计算出而是训练出的，所以利用神经网络解决问题时，对于采用监督式学习的网络，网络的设计要

经过两个阶段，首先是网络权值的训练（即学习并修正）阶段，然后才是网络的运行工作阶段。而对于采用非监督式学习的网络，则将网络权值的训练与工作合二为一，即在网络工作的同时，调整网络的权值，所以神经网络具有自学习、自适应的特性。

一般而言，由人工神经网络组成的信息处理系统所具有的特性为：

1）非线性

神经网络对非线性控制领域问题的解决带来了希望，这主要是由神经网络理论上能够任意逼近非线性连续有理函数的能力所决定的。神经网络还能够比其他逼近方法得到更加易得的模型。

2）并行分布处理

神经网络具有一个使其自身进行并行实施的高度并行结构，如此的实现结构使其能够达到比常规方法所获得结果具有更高的容错性。另外，虽然一个神经网络中的基本处理单元是非常简单的结构，然而将其进行并行实现的连接，则产生了极快的整体处理效果。

3）硬件实现

神经网络不仅能够进行并行处理,还可以通过引入大规模集成电路将其进行硬件实现，这将带来附加的速度，并且可以增加应用网络的规模。

4）学习与自适应

神经网络是通过采用被研究系统的数据记录进行训练而获得的。对于一个训练好的网络，对其输入训练中未知的数据时，具有很好的泛化能力。

5）数据融合

神经网络能够同时操作定量和定性的数据，这一特性使神经网络可以处理处在传统的系统工程（定量数据）与人工智能领域的处理技术（符号数据）之间的问题。

6）多变量系统

神经网络本身就是一个多输入/多输出系统，这个系统为解决复杂的多变量系统的建模和控制等问题开辟了一条新的途径。

1.1 感知器网络

我们已经知道，人工神经网络是在人类对其大脑神经网络认识理解的基础上人工构造的能够实现某种功能的神经网络。它是理论化的人脑神经网络的数学模型，是基于模仿大脑神经网络结构和功能而建立的一种信息处理系统。人工神经网络吸取了生物神经网络的许多优点，它具有高度的并行性、非线性的全局作用，以及良好的容错性与联想记忆功能，并且具有很强的自适应、自学习能力。随着人工神经网络技术的不断发展，其应用领域也在不断拓展，主要应用于模式信息处理、函数逼近和模式识别，以及联想记忆、最优化问题计算和自适应控制等方面。在前向网络中，最典型的网络是反向传播网络，不过，它也是在最早的人工神经网络——感知器基础上发展起来的，并且在线性分类问题中，感知器仍然发挥着重要的作用。

1.1.1 感知器的网络结构及其功能

最早用数学模型对神经系统中的神经元进行理论建模的是美国心理学家麦卡洛克(W. McCulloch)和数学家皮茨（ W. Pitts）。他们于 1943 年在分析和研究了人脑细胞神经元后用电路构成了简单的神经网络数学模型（ 简称 MP 模型）。感知器（ Perceptron）是由美国计算机科学家罗森布拉特（ F. Rosenblatt）于 1957 年提出的。感知器是在 MP 模型的基础上，加上学习功能，使其权值可以连续调节的产物。它是一个具有一层神经元、采用阀值激活函数的前向网络。

感知器的网络结构是由单层 s 个感知神经元，通过一组权值 $\{w_{ij}\}$ (i = 1, 2, ...,s ； j = 1, 2, ... , r)与 r 个输入相连组成。对于具有输入矢量 $P_{r\times q}$ 和目标矢量 $T_{s\times q}$ 的感知器网络的简化结构如图 1.5 所示。阀值激活函数如图 1.6 所示。

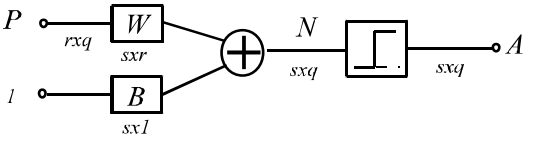


图 1.5 感知器简化结构图

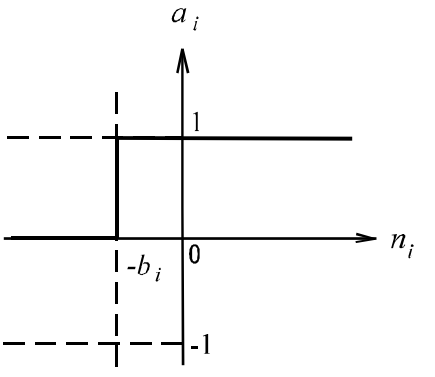


图 1.6 阀值激活函数

感知器输入/输出函数关系式为：

$$A=\begin{cases}1 & \sum W*P+b\geq 0\\ 0 & \sum W*P+b<0\end{cases} \tag{1.5}$$

由上式可知，通过对网络权值的训练，可以使感知器对一组输入矢量的响应达到元素为 0 或 1 的目标输出，从而实现对输入矢量分类的目的。

感知器的这一功能可以通过在输入矢量空间里的作图来加以解释。以输入矢量 r = 2 为例，对于选定的权值 w_1、w_2 和 b，可以在以输入矢量 p_1 和 p_2 分别作为横、纵坐标的输入平面内画出 $W*P + b = 0$，即 $w_1p_1+w_2p_2+ b = 0$ 的轨迹，它是一条直线，此直线上的以及线以上部分的所有 p_1、p_2 值均使 $w_1p_1+w_2p_2+b \geq 0$，这些点若通过由 w_1、w_2 和 b 构成的感知器则使其输出为 1；该直线以下部分的点则使感知器的输出为 0，如图 1.7 所示。

所以当采用感知器对不同的输入矢量进行期望输出为 0 或 1 的分类时，其问题则转化为：对于已知输入矢量所处输入平面的不同点的位置，设计感知器的权值 W 和 b，将由 $W*P + b = 0$ 的直线放置在适当的位置上使输入矢量按期望输出值进行上下分类。推而广之，阀值函数通过将输入矢量的 r 维空间分成若干区域而使感知器具有将输入矢量分类的能力。对于不同的输入神经元 r 和输出神经元 s 组成的感知器，当采用输入矢量空间的作

图法来解释网络功能时，其分类的一般情况可以总结为：

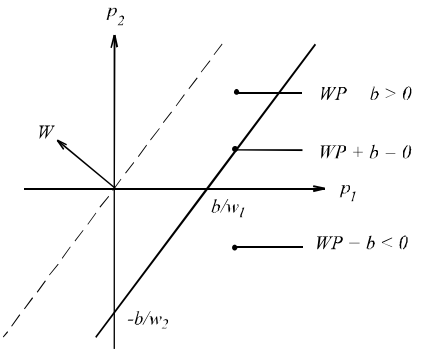


图 1.7 感知器的图形解释

①当网络输入为单个节点，输出也为单个神经元，即 $r = 1$，$s = 1$ 时，感知器是以点作为输入矢量轴线上的分割点；

②当网络输入为两个节点，即 $r = 2$ 时，感知器是以线对输入矢量平面进行分类；其中：当 $s = 1$，分类线为一条；$s = 2$，分类线为两条，依此类推。输出神经元个数 s 决定分类的直线数，可分成的种类数为 2^s；

③当网络输入为三个节点，即 $r = 3$ 时，感知器是以平面来分割输入矢量空间，而且用来进行空间分割的平面个数等于输出神经元个数 s。

1.1.2 感知器权值的学习规则与训练

学习规则是用来计算新的权值矩阵 W 及新的偏差 B 的算法。感知器利用其学习规则来调整网络的权值，以便使网络对输入矢量的响应达到数值为 0 或 1 的目标输出。

对于输入矢量 P、输出矢量 A、目标矢量为 T 的感知器网络，感知器的学习规则是根据以下输出矢量可能出现的三种情况来进行参数调整的。

①如果第 i 个神经元的输出是正确的，即有 $a_i = t_i$，那么与第 i 个神经元联接的权值 w_{ij} 和偏差 b_i 保持不变；

②如果第 i 个神经元的输出是 0，但期望输出为 1，即有 $a_i = 0$，而 $t_i = 1$，此时权值修正算法为：新的权值 w_{ij} 为旧的权值 w_{ij} 加上输入矢量 p_j；类似地，新的偏差 b_i 为旧偏差 b_i 加上它的输入 1；

③如果第 i 个神经元的输出为 1，但期望输出为 0，即有 $a_i = 0$，而 $t_i = 1$，此时权值修正算法为：新的权值 w_i 等于旧的权值 w_i 减去输入矢量 p_j；类似地，新的偏差 b_i 为旧偏差 b_i 减去 1。

由上面分析可以看出，感知器学习规则的实质为：权值的变化量等于正负输入矢量。具体算法总结如下：

对于所有的 i 和 j, $i = 1, 2, \cdots, s$； $j = 1, 2, \cdots, r$，感知器修正权值公式为：

$$\begin{aligned} \Delta w_{ij} &= (t_i - a_i) \times p_j \\ \Delta b_i &= (t_i - a_i) \times 1 \end{aligned} \tag{1.6}$$

用矢量矩阵来表示为：

$$\begin{aligned} W &= W + EP^{\mathrm{T}} \\ B &= B + E \end{aligned} \tag{1.7}$$

此处，E 为误差矢量，有 $E = T - A$ 。

感知器的学习规则属于梯度下降法。已被证明：如果解存在，则算法在有限次的循环迭代后可以收敛到正确的目标矢量。

要使前向神经网络模型实现某种功能，必须对它进行训练，让它逐步学会要做的事情，并把所学到的知识记忆在网络的权值中。人工神经网络权值的确定不是通过计算，而是通过网络的自身训练来完成的。这也是人工神经网络在解决问题的方式上与其他方法的最大不同点。借助于计算机的帮助，几百次甚至上千次的网络权值的训练与调整过程能够在很短的时间内完成。

感知器的训练过程如下：

在输入矢量 P 的作用下，计算网络的实际输出 A，并与相应的目标矢量 T 进行比较，检查 A 是否等于 T，然后用比较后的误差 E，根据学习规则进行权值和偏差的调整；重新计算网络在新权值作用下的输入，重复权值调整过程，直到网络的输出 A 等于目标矢量 T 或训练次数达到事先设置的最大值时训练结束。

若网络训练成功，那么训练后的网络在网络权值的作用下，对于被训练的每一组输入矢量都能够产生一组对应的期望输出；若在设置的最大训练次数内，网络未能够完成在给定的输入矢量 P 的作用下，使 $A = T$ 的目标，则可以通过改用新的初始权值与偏差，并采用更长训练次数进行训练，或分析一下所要解决的问题是否属于那种由于感知器本身的限制而无法解决的一类。

感知器设计训练的步骤可总结如下：

①对于所要解决的问题，确定输入矢量 P，目标矢量 T，并由此确定各矢量的维数以及确定网络结构大小的神经元数目：r, s 和 q；

②参数初始化：a) 赋给权矢量 W 在(-1, 1)的随机非零初始值；

b) 给出最大训练循环次数；

③网络表达式：根据输入矢量 P 以及最新权矢量 W，计算网络输出矢量 A；

④检查：检查输出矢量 A 与目标矢量 T 是否相同，如果是，或已达最大循环次数，训练结束，否则转入⑤；

⑤学习：根据(1.7)式感知器的学习规则调整权矢量，并返回③。

下面给出例题来进一步了解感知器解决问题的方式，掌握设计训练感知器的过程。

例 1.1 考虑一个简单的分类问题。

设计一个感知器，将二维的四组输入矢量分成两类。

输入矢量为：P = [-0.5 -0.5 0.3 0;
-0.5 0.5 -0.5 1];

目标矢量为：T = [1.0 1.0 0 0],

解： 通过前面对感知器图解的分析可知，感知器对输入矢量的分类实质是在输入矢量空间用 $W*P + B = 0$ 的线性表达式对其进行分割而达到分类的目的。根据这个原理，对此例中二维四组输入矢量的分类问题，可以用下述不等式组来等价表示出：

$$\begin{cases} -0.5w_1 - 0.5w_2 + b \geq 0 & (\text{使} t_1 = 1 \text{成立}) \\ -0.5w_1 + 0.5w_2 + b \geq 0 & (\text{使} t_2 = 1 \text{成立}) \\ 0.3w_1 - 0.5w_2 + b < 0 & (\text{使} t_3 = 0 \text{成立}) \\ w_2 + b < 0 & (\text{使} t_4 = 0 \text{成立}) \end{cases}$$

实际上可以用代数求解法来求出上面不等式中的参数 w_1、w_2 和 b。经过迭代和约简，可得到解的范围为：

$$\begin{cases} w_1 < 0 \\ 0.8w_1 < w_2 < -w_1 \\ w_1 / 3 < b < -w_1 \\ b < -w_2 \end{cases}$$

一组可能解为：

$$\begin{cases} w_1 = -1 \\ w_2 = 0 \\ b = -0.1 \end{cases}$$

而当采用感知器神经网络来对此题进行求解时，意味着采用具有阀值激活函数的神经网络，按照问题的要求设计网络的模型结构，通过训练网络权值 $W = [w_{11}\ w_{12}]$ 和 b，并根据学习算法和训练过程进行程序编程，然后运行程序，让网络自行训练其权矢量，直至达到不等式组的要求。

鉴于输入和输出目标矢量已由问题本身确定，所以所需实现其分类功能的感知器网络结构的输入节点 r，以及输出节点数 s 已被问题所确定而不能任意设置。

根据题意，网络结构图如图 1.8 所示。

由此可见，对于单层网络，网络的输入神经元数 r 和输出神经元数 s 分别由输入矢量 P 和目标矢量 T 唯一确定。网络的权矩阵的维数为：$W_{s\times r}$，$B_{s\times 1}$，权值总数为 $s\times r$ 个，偏差个数为 s 个。

在确定了网络结构，设置了最大循环次数并赋予权值初始值后，设计者便可方便地利用适当软件，根据题意以及感知器的学习、训练过程来编写程序，并通过计算机对权值的反复训练与调整，最终求得网络的结构参数。

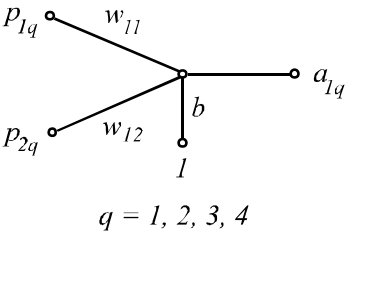


图 1.8 网络结构图

上面的结果似乎表明只要增加输出神经元数 s，就能够解决任意数目的分类问题。事实上，对于输出为 0 或 1 的感知器，其功能可以典型化为对各种逻辑运算的实现。当网络