

摘 要

(神经网络是当前发展极为迅速的一门边缘学科, 由于它具有强的容错能力, 鲁棒性和自适应性, 从而决定了它在不确定性模式识别中具有重要的应用价值。)本文系统地研究了神经网络的基本理论, 并以两种典型的神经网络: BP 前馈神经网络模型和 Kohonen 自组织神经网络模型为基础, 对遥感图像分类问题、文字倾斜的自动矫正问题和纹理图像的分类问题进行了研究。

BP 网络模型是一种典型的前馈网络, 采用多层感知机模型, 其分类能力和模式识别能力一般都强于其它网络, 是目前研究最多的网络形式之一。它是在普通的单层感知机模型无法解决非线性可分的问题(例如, 著名的“异或”逻辑问题)上, 通过加入若干中间层(隐含层)而引入的。本文深入分析了多层神经网络的理论, 详细推导了它的 BP 学习规则。提出了用一种新的层次神经网络模型解决遥感图像分类中最困难的问题之一, 波谱混迭问题(即“同谱异构”问题), 其中第一级神经网络主要用于波谱非混迭像元的分类, 第二级和后续层次的神经网络则用于波谱混迭像元的分离。我们用这种模型对葡萄牙里斯本地区 TM 图像的地面实况数据进行了分类实验, 得到了高度精确的实验结果。

Kohonen 网络模型是根据人脑的自组织特征映射能力提出的, 具有良好的拓扑特征保持特性和自组织学习能力。本文仔细研究了 Kohonen 网络的自组织过程、自组织的数学分析、自组织特征学习算法和 Kohonen 网络的拓扑结构, 并讨论了它的推广模型: 将输出节点阵列由二维平面模型推广到封闭的多维超曲面模型。常规的文字倾斜角自动测量方法, 是通过对逐度旋转的文字图像作灰度投影, 检测出某一特征量取极值时的旋转角度, 这种方法存在旋转角度步长的量化问题。我们利用 Kohonen 模型的自组织特性, 提出了一种用二级 Kohonen 网络对文字进行聚类处理, 通过对这些聚类中心的曲线拟合计算, 直接提取出文字倾斜角度的新方法。实验结果表明, 用这种方法计算出来的拟合直线斜率体现了文字的倾斜角度, 可以用于倾斜文字的自动矫正。

模式识别的图像分类问题中, 除了分类器设计外, 还包括特征提取问题。没有典型的模式特征, 神经网络就无法有效地进行分类处理。因此在讨论纹理图像分类时, 本文就纹理特征的提取问题, 详细讨论了分形理论。我们认为, 分形理论是一种度量纹理的有效手段。由于 Hausdorff 维数的局限性, 我们进一步讨论了多重分形理论和广义维数谱的概念。最后, 我们提出用图像的分形维、广义维数谱和空隙特征等作为图像的纹理特征, 结合 BP 神经网络分类器, 进行纹理图像分类的方法。采用这种方法, 我们分别对一组 Brodatz 图像、一幅灰度自然图像、一幅彩色自然图像和一幅彩色航空照片进行了纹理分类实验, 都取得了非常好的分类结果。

本文的部分工作是安徽省“九五”重点科技攻关项目“安徽省资源环境遥感动态服务体系”的组成部分之一。

Abstract

Artificial neural networks, as a borderline subject, have been developed quickly now. They are of fault-tolerant, robust and adaptive, which make them very applicable for the indefinite pattern recognition. Here, the primary principles of neural networks are systematically discussed. Based on two kinds of representative neural network models, BP Feedforward Network model and Kohonen Self-Organizing Network model, the classification of remote sensing image, the automatic rectification for the slant Chinese Characters and the segmentation of texture image are studied in the detail.

BP network, using the multilayer Perceptron model, is a typical feedforward network. Its *classification and pattern recognition capability is usually stronger than other network's*, which make it more popular in the application research compared to other network forms. It was raised to solve the nonlinear undivided problems for the single-layer perceptron classifier, such as the famous XOR logic problem, improved on adding several middle layers known as hide layers. The multilayer neural networks are analyzed in depth, and their BP learning rule is deduced detailedly. A new hierarchical neural network model is presented here to resolve one of the most difficult problems in the classification of remote sensing images, Spectral Overlap Pixels Separating problem (just as the "Different Land Covers with Same Spectrum" problem). The new model consists of two or more levels of cascaded neural networks, the first-level neural network is mainly used for classifying spectral non-mixed pixels, while the second or the higher levels of neural networks are used for separating the spectral overlap pixels. It can separate the spectral mixed pixels with very high accuracy on the classification experiment for the LANDSAT image data in Lisbon, Portugal.

Kohonen network is established according to the ability of the feature map of the brain and possesses good topological invariance property and the ability of learning by self-organization. By studying the self-organizing process, the learning algorithm and the topology of Kohonen network, we generalize its output model: using spherical surface instead of plane surface as output layer of Kohonen network and using multilayer network instead of single layer network. The ordinary method for the Chinese Character slant angle automatically detected is to rotate the image by degrees and observe in which degree some eigenvalue is got an extremum. But there are some *quantificational problems in the rotatable angle step*. Taking advance of the self-organizing feature of Kohonen model, a new method with a two-level Kohonen network model is presented. The method uses the two-level Kohonen networks to make a convergent effect on the slant Chinese characters and then the slant angle of the Chinese characters can then be directly extracted by using a curve fitting method on these convergent center points. The experiment shows that it can get the slope of the curve-fitting line reflects the slant angle of Chinese characters and can be used

in rectifying the slant Chinese characters automatically.

In the field of the image segmentation of the pattern recognition, there are feature extraction problems more than the issue of classifier design. Without characteristic pattern features, the neural networks can't be effectively used in classifying. Thus in order to extract the input signal of network, here just for the image texture feature, the Fractal Approach is presented. In our point of view, fractal dimension is regard as an effective measurement of texture. For the limit of the definition for the Hausdorff dimension, the Multifractal Approach and Generalized Dimension Spectrum Definition are also introduced. In the end, a new method of the texture image segmentation is presented by using all of the fractal dimension, generalized dimension spectrum and lacunarity to stand for the texture feature and combined with the BP neural network classifier. The perfect segmentation results are all achieved on a group of Brodatz images, a gray building image, a color scene image and a color aerial picture.

A proportion of this work is part of the ANHUI province 9th five-year-plan Research Program "Dynamic Research and Service of Resources and Environment by Remote Sensing in ANHUI Province".

致 谢

本文的工作自始至终是在导师刘政凯教授的悉心指导下完成的。他严谨的治学态度，勤奋的工作精神和正直热情的为人对我教益匪浅，给我留下了极其深刻的印象。在科大学习期间，刘老师对我的学习、工作和生活给予了无微不至的关怀和教诲，特此表示衷心的感谢和敬意。在这里的学习和生活将是我终生难忘。

在此，我还要感谢信息处理中心的肖锦玉老师，俞能海老师，陈学詮老师，史杏荣老师，张荣老师，张迁老师的帮助和支持。感谢本实验室的博士生岑曙炜同学，李厚强同学，何佳同学，硕士生赵志刚同学，杨金钟同学，李学龙同学，刘斌同学，邵肖伟同学，陈豪同学以及其他同学对我的鼓励和支持，是他们真挚的友谊和无私的帮助才使得本论文能够顺利完成。此外，还要特别感谢李厚强师兄在我论文期间给予我的大力帮助和许多启发性指导。

感谢信息中心这个集体，感谢科大给予我的八年培养。她活跃的学术气氛与和睦的学习环境，使得本人能够顺利、愉快的完成学业。

作者于 2000.5.27 谨志

第一章 导 论

§ 1.1 模式识别的基本概念

“模式”是人们生活的客观世界的总称。人们生活在客观世界中，为了改造客观世界，首先必须认识它，这就是对模式的识别。人通过感觉器官从外界获取信息，尔后通过思维建立对客观世界的真正认识；通过视觉通道获取有关信息，如色彩、形状、空间等各种形式的图象；通过听觉通道取得各种声音信息；通过触觉传感通道取得如温度、材料强度、表面光洁度等信息，最后将各种信息进行综合思维和判断，以做出决策。

模式识别技术是用计算机来模拟人的各种识别能力，当前主要是对视觉能力和听觉能力进行模拟。模拟人的视觉能力就是用计算机来做图像的识别和理解工作，而模拟人的听觉能力则与计算机辅助语音识别等技术密切相关。目前就人工智能的范畴而言，“识别”和“理解”代表不同的涵义：“识别”是指对客观事物按其物理特征进行分类；而“理解”则是对客观事物内涵的了解。

当前模式识别技术所能做到的基本上是一种分类工作，距离理解工作还有一定的差距。因此，就模式识别技术所希望解决问题的领域和目标而言，它还处于初级阶段。尽管模式识别技术的理论还很不完善，但它作为人的能力的辅助和延伸而起着相当重要的作用。到目前为止，模式识别技术已在语音识别、文字识别、语音合成、目标识别和分类、图像分析与识别等领域得到广泛应用。

作为模式分类方法，所涉及的问题主要有：

1. 特征提取；
2. 学习训练模式样本以得到决策规则；
3. 利用前述得到的决策规则对实际数据进行分类。

模式识别方法可分为两种：即统计模式识别方法和结构（句法）模式识别方法。相应的模式识别系统包括设计和实现两部分，设计是指用一定数量的样本（成为训练集合或学习集合）进行分类器设计，而实现则是指用所设计的分类器对待识别的样本进行分类决策。一般的模式识别系统如图 1.1 所示，包括以下几个部分：

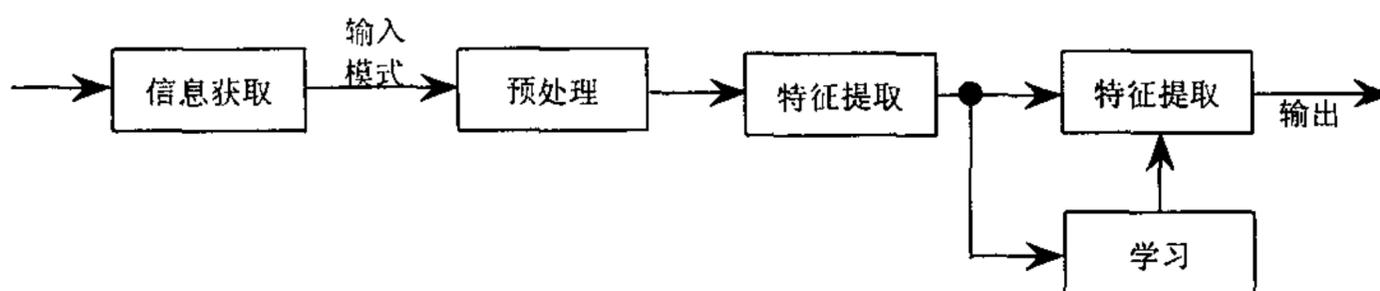


图 1.1 模式识别系统

1. 信息获取 为了是计算机能对各种现象进行分类识别，必须将所研究的对象用计算机所能接受的形式表示。

2. 预处理 预处理的目的是去除噪声，增强有用信息，并对相应的退化现象进行复原。

3. 特征提取与选择 所谓特征提取就是对模式的某些物理性质进行数学描述。具体而言就是对原始数据进行变换,得到最能反映模式分类的本质特征。一般我们把原始数据构成的空间叫测量空间,把分类识别赖以进行的空间叫作特征空间。通过变换可以把维数较高的测量空间中表示的模式变为维数较低的特征空间中表示的模式。常用的特征有:统计特征、结构特征、数学变换特征和逻辑特征等。

4. 学习 学习是在建立识别系统时,对以选定的特征完成特征提取之后,对系统进行训练的过程,学习可以分为:有监督(即学习过程中,样本的类别已知)和无监督(要求学习环节对样本集有一个聚类的过程)两种方式。

5. 特征判别 特征判决是在掌握分类规律后,在实现阶段对连续输入的大量模式进行分类。这种判别方法有:

- 判别函数法,利用一个(或多个)判别函数对两类(或多类)模式进行判别;
- 距离方法,用已识别的特征对训练样本特征进行广义的距离运算,求出距离最近的标准特征,常用的有:欧氏距离、加权欧氏距离、汉明距离和马氏距离等。
- 相似度方法,与距离方法相反,是用被识别特征对训练样本特征进行隶属度运算,求出隶属度最大的标准类作为被识别模式。

综上所述,模式识别系统的关键在于特征提取和特征判决模块的设计。如果能选取具有高度准确描述能力的特征,无疑对系统的建立具有重要意义。它可以用较少的存贮,表达较多的物理意义,而特征判别器的正确设计可使系统具有较高的稳定性和准确性。

§ 1.2 神经网络方法与模式识别

针对模式特征的不同选择及其判别决策方法的不同,可对模式识别方法作如下分类:

1. 模板匹配法 模板匹配法是对每个待识别的类别,找出标准的模板作为识别标准,它可以是一类模式,也可以是一个模式集合。

2. 统计特征法 统计特征法对已知类别的模式样本进行各种特征的提取和分析,选取对分类有利的特征,并对其统计均值等按照已知类别分别进行学习,按贝叶斯最小误差准则,根据以上统计特征设计出一个分类误差最小的决策超平面,识别过程就是对未知模式进行相同的特征提取和分析,通过决策平面方程决定该特征相应的模式所属的类别。

3. 句法结构方法 用已知结构信息的模式作为训练样本,先识别出基元和它们之间的连接关系,并用字母符号表示它们,然后用构造句子的方法来描述生成这些场景的过程,并由此推断出生成该场景的一种文法规则,这就是训练过程。在识别过程中,要对未知结构的模式进行基元识别及其相互结构关系的分析,然后利用训练过程中获得的文法对其作句法分析,如果它能被已知结构信息的文法分析出来,则该幅未知图象模式具有与该文法相同的图形结构;否则就可判定不是这种结构。

4. 逻辑特征方法 逻辑特征方法是这样的一种方法,其特征的选择对每一类模式识别问题来说是独一无二的,即在一类问题中只有一个模式具有某一种(或某一组合的)逻辑特征。因此匹配过程也只有“是”或“非”两种结果,即是本类或不是本类。逻辑特征没有“远”、“近”之分,只有“是”、“非”之别。

5. 模糊模式识别 所谓模糊模式识别是在模式识别过程中引入模糊集的概念,模糊集的概念可以在模式识别系统的后几个环节的任何地方引入。若在后两个环节(特征提取和判别决策)的任何一个中引入则在判别决策时,所求的结果就是该模式特征对于每一标准类的隶属度,进而将该模式归属于隶属度最大的标准类。模糊判别决策有以下几种方法:

- 最大隶属原则法;

- 择近原则法;
- 模糊聚类分析法。

6. 神经网络方法 神经网络是由大量基本运算单元经过分层次的相互联系而形成的动态系统。它是模拟人脑神经元的处理过程, 包括对信息的加工、处理、存贮和搜索等过程, 具有如下的基本特点:

- 分布式存贮信息。其存贮信息的方式与传统的计算机思维方式不同, 不是将信息存于一个地方, 而是将其分布在不同的位置。神经网络是用大量神经元之间的联结及对各联结权值的分布来表示特定的信息。因此当局部网络受损时, 仍有可能恢复出原来的信息。

- 并行的信息处理和推理过程。每个神经元都可根据所接收的信息作独立地运算和处理, 并输出结果。对于一个特定的输入模式, 经过前向计算产生一个输出模式, 再通过输出节点的和本身信号的强弱, 得到一个特定解, 体现了神经网络并行推理的特点。

- 自组织、自学习的信息处理能力。神经网络中各神经元之间的联结强度通过权值大小来表示, 这种权值可以事先定出, 也可以根据学习环境作自适应调整。它模拟了人的形象思维方法, 是一种与传统的符号逻辑完全不同的非逻辑语言的方法。

因此神经网络在许多方面更接近于人对信息的处理方法, 具有模拟人的形象思维能力, 反映了人脑的若干基本特性。

采用规范化特征向量进行模式识别的传统方法, 严重地限制了识别系统的灵活性, 使系统对可获取信息的利用率低, 影响了系统的性能。因此, 智能化模式识别系统旨在增强系统的自适应能力、学习能力以及容错性, 神经网络系统理论为此提供了一种可能的途径。

§ 1.3 神经网络的研究简史

神经网络的研究由来已久, 与之有关的神经科学研究(大脑功能的生理学及心理学研究、神经元的电生理研究等)早在 40 年代就已做了不少工作。1943 年, 心理学家 McCulloch 和数学家 Pitts 首先提出了形式神经元模型(简称 MP 模型), 把神经元作为双态开关, 并应用布尔逻辑的数学工具来研究用形式神经网络模拟解决客观事件的问题。1949 年, 心理学家 Hebb 提出了神经元的学习规则, 使神经网络具有了可塑性。到目前为止, 大部分神经网络的学习规则仍采用 Hebb 规则或它的改进型。1958 年, Roseblatt 提出了感知机(Perceptron)模型, 指出感知过程具有统计分离性, 利用有监督信号可以对感知机进行训练, 试图模拟人脑感知能力和学习能力。1962 年, Block 用解析法证明了感知机学习的收敛性。日本的中野在 1969 年提出了一个联想记忆模型, 同年 Marr 提出小脑功能及其学习法则的小脑感知机模型, 这是神经网络与神经生理学相结合的研究成果。这一阶段可以看成是神经网络的初创期。

自从 1969 年 Minsky 及 Paper 发表的著作《感知机》一书, 严格论证了简单线性感知机功能的局限性, 并指出多层感知机还不能找到有效的计算方法后, 神经网络的研究一度处于低潮。1972 年芬兰 Kohonen 发表了一个与感知机不同的线性神经网络模型, 比非线性神经网络模型更容易分析。1975 年福岛提出了一个自组织识别神经网络模型。1980 年, 芬兰 Kohonen 提出了自组织映射理论。1982 年, 美国加州工学院物理学家 Hopfield 对神经网络的动态特性进行了研究, 引入了能量函数的概念, 给出了网络的稳定性判据, 提出了用于联想记忆和优化计算的新途径, 为神经网络的研究注入了新的活力。1984 年, 多伦多大学 Hinton 教授等人把神经元的输入函数与统计力学中的波尔兹曼分布联系起来, 提出了 Boltzmann 机模型。1986 年 Rumelhart 等人提出了误差反向传播神经网络, 简称 BP 网络, 它是一种能朝着满足给定的输入输出关系反向进行自组织的神经网络。这种网络早在 1974

年前后, 已被哈佛大学的 Werbos 博士所发明, 只是当时未得到重视。BP 网络目前被广泛使用, 并以此为基础作了许多改进, 发展了某些快速收敛的学习算法。这一阶段是神经网络的过渡期。

1987 年 6 月在美国圣地亚哥召开了第一届世界神经网络会议, 标志着神经网络研究在世界范围内形成了高潮。1994 年首次将模糊系统、神经网络和进化计算三个方面内容的会议综合在一起召开, 被称为'94IEEE 全球计算智能大会 (WCCI) 在美国奥兰多召开。90 年代 IEEE 神经网络汇刊的问世, 使关于神经网络的各种专著逐年增加。随着近年来各种边缘学科的发展, 国内外学者越来越多地把神经网络理论、模糊理论和混沌理论这三者融合起来研究。在神经网络中, 抓住其“大规模并行计算”和“自适应学习”等特性; 在模糊理论中, 抓住其“模糊性”和“自反性”; 在混沌理论中, 抓住“非周期背后隐藏着有序性”和“对初始条件的敏感依赖性”等, 研究模糊神经网络和混沌神经网络以及这些网络在信息处理中的作用, 以实现更加柔软的智能信息处理。如今, 神经网络的研究已涉及计算机科学、控制论、信息科学、微电子学、心理学、认知学、数学及物理学等学科。随着研究的不断深入, 神经网络理论将为新一代计算机和人工智能的应用开辟一条崭新的途径。

§ 1.4 神经网络的生物基础

§ 1.4.1 脑的宏观结构

人的神经系统包括脑、脑神经、脊髓和脊神经。脑和脊髓又被称为中枢神经。

人脑的结构从解剖学角度可分为: 延脑 (延髓)、后脑、中脑、间脑和端脑。其中延脑是生命中枢。后脑包括脑桥和小脑, 脑桥负责小脑与高级部位的“通信联结”, 小脑则专司运动和平衡。中脑是专门处理感觉信息的最低级脑结构。其顶部有十字交叉的两条沟将顶部分成两对上丘和下丘, 上丘是视觉的低级中枢, 下丘是听觉的原始中枢。间脑包括丘脑和下丘脑, 是传递感觉和运动信息的关键中继区域。端脑是由嗅脑、基底核、胼胝体与大脑皮层等组成的。

与人类智能最直接相关的是中脑水平以上的脑结构, 特别是大脑皮层。大脑由左、右两半球组成, 每个大脑半球又分为前叶、顶叶、枕叶和颞叶四个区域。每个大脑半球与其对侧躯体的感觉和运动过程有神经关联。但大脑两半球并非完全对称, 其功能也不尽相同。左半球主要负责认识和意识功能, 具有逻辑思维能力; 而右半球则负责知觉和直觉功能, 具有形象思维能力。

半球内最重要的是脑皮质内的细胞、皮质面积的大小及构成皮层各层细胞的高度分化及其严密结构, 脑皮质是人类高级神经思维活动的物质基础。

§ 1.4.2 神经网络的结构与功能

在重约 1500g 的大脑里, 包含着约千亿个神经细胞和胶质细胞, 其中 80% 为胶质细胞, 它们主要负责供应脑的营养工作。其余的神经细胞总数达一百几十亿个, 它们是神经系统的结构和功能单元, 因此又称为神经元。神经元负责接收或产生信息, 传递和处理信息。

1. 神经元的结构

神经元是由细胞体、树突和轴突组成，其结构如图 1.2 所示。

细胞体 由细胞核、细胞质和细胞膜组成。其细胞膜由于具有选择性的通透性，使内外液的成分保持差异，形成膜内外的电位差（膜电位），其大小受细胞体输入信号强弱而变化，一般约在 20~100mV。

树突 由细胞体向外伸出的许多树枝状较短的突起，长约 1mm 左右，用于接受周围其它神经细胞传入的神经冲动。

轴突 由细胞体向外伸出的最长的一条神经纤维。其长度由数厘米到 1 米。远离细胞体一侧的轴突端部有许多分支，称为轴突末梢（神经末梢）。轴突通过轴突末梢向其它神经元传出神经冲动。

突触 一个神经元的轴突末梢和另一个神经元的树突或细胞体之间，通过微小间隙相联系，这样的联结称为突触。其直径约为 0.5~2 μ m。一个神经元的树突从突触处接收其它神经元的信号，这些信号可能是激励的，也可能是抑制性的。突触有兴奋性和抑制性两种。

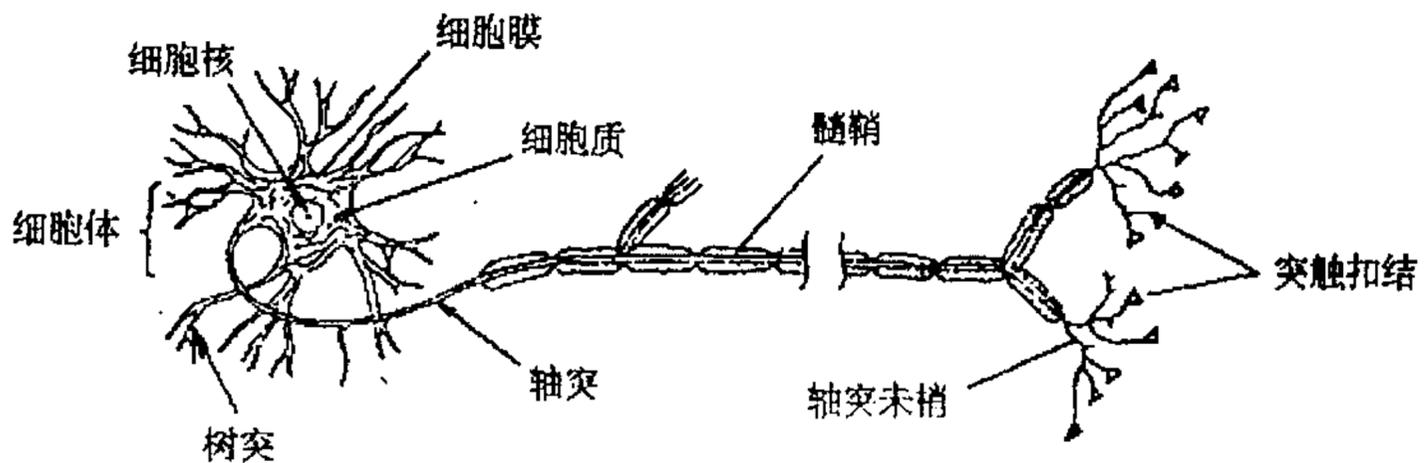


图 1.2 神经细胞的结构

一个神经元约有 $10^3\sim 10^4$ 个突触，人脑中大约有 10^{14} 个突触，神经细胞之间通过突触复杂地结合着，从而形成大脑的神经（网络）系统。

2. 膜电位与神经元的兴奋

细胞膜内的电位与其外的电位差称为膜电位。无信号输入时的膜电位称为静止（静息）膜电位，大约 -70mV。输入信号会引起膜电位的变化，当它的值大过 -55mV 时，神经细胞的活性将被激发，此时细胞内的电位自发地急剧升高，大约比静止时高出 100mV 左右，此后膜电位又急剧下降返回原值，这个过程称为神经细胞的兴奋。兴奋的结果是产生高度为 100mV，宽度为 1ms 的电脉冲，这个脉冲将通过轴突传给其它神经细胞。然而，只要当膜电位的上升不超过 15mV 高的阈值时，神经细胞就不会兴奋，轴突也不会输出任何信号。

由此可见，细胞体具有“阈值作用”。而且脉冲刚发出后，即使再强大的输入信号也不能马上使神经细胞兴奋，这个过程称为“绝对不应期”。在绝对不应期刚结束的一段时间内，兴奋阈值比通常高，因此神经细胞难以兴奋，这段时间称为“相对不应期”，而后变高的阈值慢慢地返回原值。一般来说，绝对不应期为 1 ms，相对不应期为几个 ms。

3. 突触的结构与功能

突触是神经元之间传递信息的重要特殊“接口”，它包括突触前细胞（发出信号的神经

细胞)、突触间隙和突触后细胞(接收信号的神经细胞)三个部分。

由于神经细胞的兴奋而发出的电脉冲,沿轴突以每秒 100 米左右速度传到神经元间的突触,轴突末端也会随之产生某种化学物质,通过化学物质的接触作用使突触内的各细胞膜电位改变。不同类型的神经细胞产生不同的化学物质,如果促使其它细胞的膜电位提高,则称此细胞为兴奋性神经细胞;反之,使其它细胞的膜电位变弱,则此细胞为抑制性神经细胞。这种抑制性效果有以下几种:

突触后前馈抑制:一个神经元经过另一个抑制性中间神经元释放抑制性物质,使第三个神经元产生抑制性电位。

突触后返回抑制:一个中间的抑制性神经元释放物质,对原先的突触前细胞本身产生抑制性作用。

突触后交互抑制:指发生在树突与树突的突触之间存在的某些抑制作用,由于抑制通路很短,所以抑制作用很强。

突触后横向抑制:指相邻神经元之间的相互抑制过程。

§ 1.4.3 神经元模型

实际的神经元是一个多输入/单输出的信息处理单元,它具有如下的几个活动特性:

1. 空间汇总功能 单个神经元在同一时间可以从别的神经元接受多达上千个突触的输入,这些输入对该神经元影响的权重不尽相同,所以单个神经元应具有对它们进行空间汇总的功能。
2. 时间汇总功能 由于单个信号影响会短时间地持续,一定程度上将与后到达的输入信号同时起作用,因此神经元对于不同时间内通过同一突触的输入信号具有时间汇总的功能。
3. 阈值特性 神经元的输入/输出关系具有与阈值 θ 相关的非线性特性。
4. 不应性 阈值 θ 不是一个常数,它随神经元的兴奋而变化。在绝对不应期内,阈值可看作 ∞ 。神经元持续兴奋后其阈值也会慢慢增加。
5. 突触的可塑性 突触结合的强度,即权重 w_{ij} 是不定的,会随着输入/输出信号而可塑性变化。

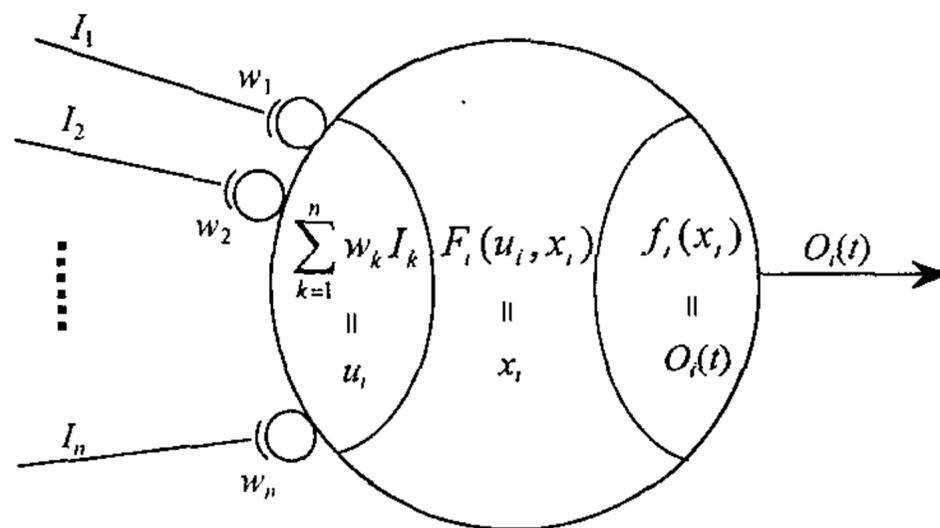


图 1.3 神经元模型

考虑到神经元的结构及其模型特性,从工程角度上,可用图 1.3 来描述神经元的数学模型。图 1.3 中,对于第 i 个神经元,当不考虑时间汇总而只考虑空间汇总时,其膜电位 u_i 的变化等于输入信号 I_k 和权重 w_k 的线性组合

$$u_i = \sum_{k=1}^n w_k I_k \quad (1.1)$$

第 i 个神经元在时刻 t 的活性度 $x_i(t)$ 由膜电位 u_i 的变化及其此刻的活性度 x_i 按下式求得

$$x_i(t) = F(u_i, x_i) \quad (1.2)$$

其中 $F_i(u_i, x_i)$ 是用于确定第 i 个神经元活性度的函数。

输出 O_i 是第 i 个神经元活性度的函数,即

$$O_i(t) = f_i(x_i) \quad (1.3)$$

其中 $f_i(\cdot)$ 是包括阈值的任意函数,又称为输出函数。关于此输出函数常见的有以下几种:

(1) 阈值型:如图 1.4(a), (b)所示,包括阈值函数和双向阈值函数,即

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad \text{或} \quad f(x) = \begin{cases} +1, & x > 0 \\ -1, & x \leq 0 \end{cases} \quad (1.4)$$

(2) 分段型:如图 1.4(c)所示。

(3) S 型:这类曲线能反映了神经元的饱和特性。如图 1.4(d), (e)所示,包括 Sigmoid 函数和双曲正切函数,即

$$f(x) = \frac{1}{1 + \exp(x)} \quad \text{或} \quad f(x) = \frac{1}{2} \left[1 + \tanh\left(\frac{x}{x_0}\right) \right] \quad (1.5)$$

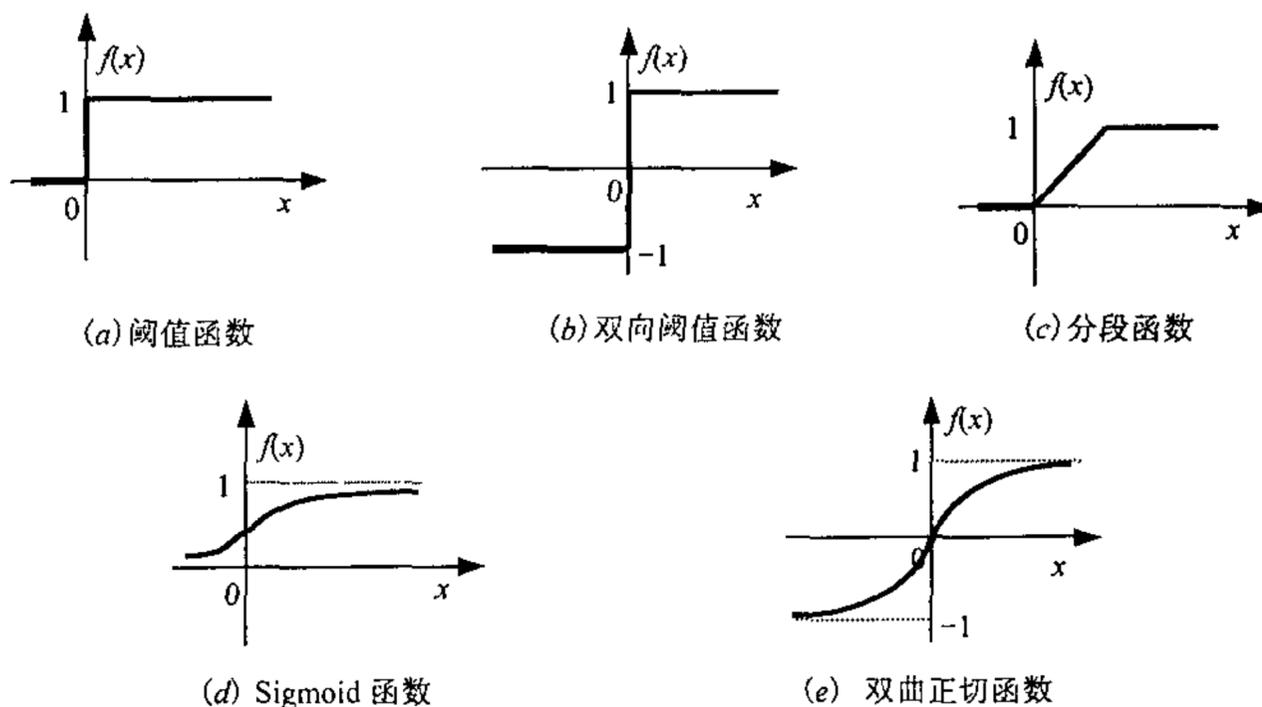


图 1.4 神经元输出特性函数

§ 1.5 神经网络的基本原理

§ 1.5.1 神经网络的结构

通常所说的神经网络结构，主要指它的联接方式。从拓扑结构上考虑，神经网络属于以神经元为节点，以节点间的有向连接为边的一种图。其结构大体上可分为层状和网状两大类：层状结构的神经网络是由若干层组成，每层中有一定数量的神经元，相邻层中的神经元为单向联接，同层内的神经元不能联接；在网状结构的神经网络中，任何两个神经元之间都可能双向联接。

一般常见的神经网络有如下几种网络结构：

1. 前向网络（前馈网络）

前向网络通常包括许多层，如图 1.5 所示为三层网络。这种网络的特点是只有前后相邻两层之间的神经元存在相互联接，各神经元之间没有反馈。每个神经元可以从前一层接收多个输入，并只有一个输入送给下一层的各神经元。

三层前向网络分为输入层、隐含层和输出层。在前向网络中有计算功能的节点称为计算单元，而输入节点无计算功能。

2. 反馈网络

反馈网络从输入层到输出层有反馈，即每一个节点同时接收外来输入和来自其它节点的反馈输入，其中也包括神经元输出信号引回到本身输入构成的自环反馈。如图 1.6 所示。这种反馈网络的每个节点都是一个计算单元。

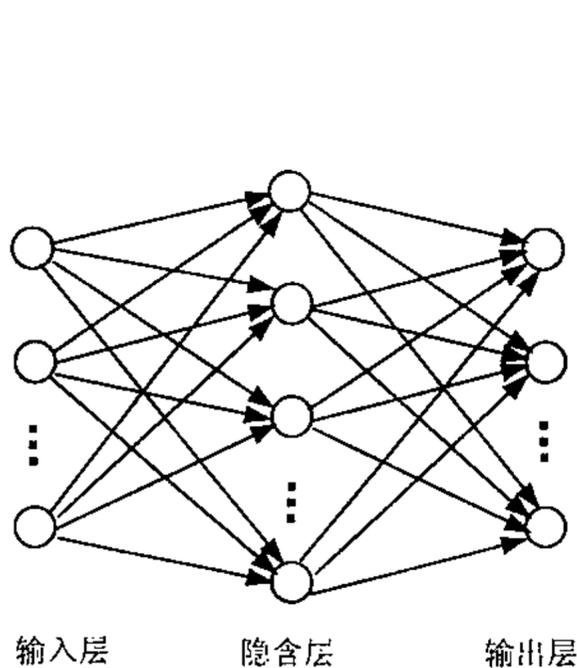


图 1.5 前向网络

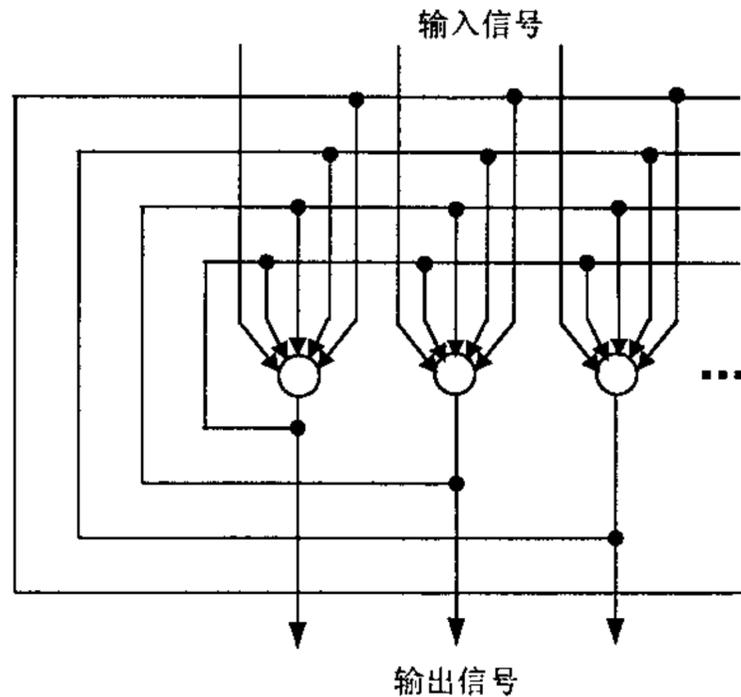


图 1.6 反馈网络

3. 相互结合型网络

相互结合型网络如图 1.7 所示，它是属于网状结构。构成网络的每个神经元都可能相互双向联接，所有的神经元既作输入，同时也用于输出。这种网络对信息处理的方式与前

向网络不同。在前馈网络中,信息处理是从输入层依次通过中间层(隐含层)到输出层后,处理就结束了。而在这种网络中,如果在某一时刻从神经网络外部施加一个输入,各个神经元一边相互作用,一边进行信息处理,直到使网络所有神经元的活性度或输出值收敛于某个平均值为止,才结束信息的处理。

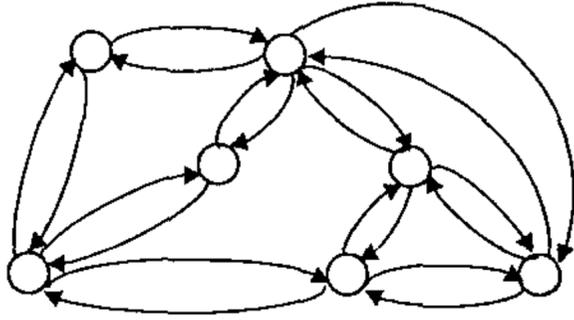


图 1.7 网状结构网络

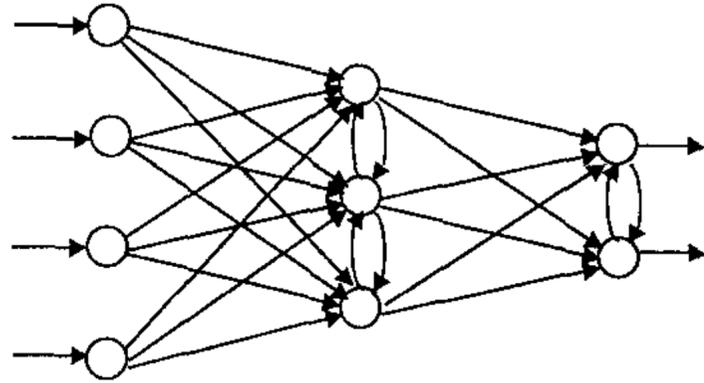


图 1.8 混合型网络

4. 混合型网络

前向网络和相互结合型网络分别是典型的层状结构网络和网状结构网络,介于二者之间的一种联接方式,如图 1.8 所示,在前向网络的同一层间有神经元互联的结构,称为混合型网络。这种在同一层内的互联,目的是为了限制同一层内,同时兴奋或抑制的神经元数目,以完成特定的功能。

§ 1.5.2 神经网络的学习和训练

人脑中一个典型的神经元通过许多树突的精细结构,收集来自其它神经元的信息,当它收到的兴奋输入足够大于它的抑制输入时,神经元将通过轴突发出电活性脉冲,经突触结构影响其它的神经元,这便发生了学习行为。因此,可以认为神经网络学习的本质特征在于神经细胞特殊的突触结构所具有的可塑性连接,而如何调整连接权值就构成了不同的学习算法。

神经网络按学习方式可分为有监督学习和无监督学习两大类。

(1) 有监督学习

为了使神经网络在实际应用中能解决各种问题,必须对它进行训练,就是从应用环境中选出一些样本数据,通过不断地调整权矩阵,直到得到合适的输入输出关系为止,这个过程就是对神经网络的训练过程,这种训练过程需要外界监督,提供训练数据(样本数据)。

权值的调整是基于奖惩式的规则:当网络的输出朝正确方向变化时,调整权值以强化(奖励)此方向的变化;而当网络的输出朝着错误方向变化时,权值将朝着弱化(惩罚)的方向变化。

这种学习方法虽然简单,并能够根据训练数据准确地调整出所需的权矩阵,但它也同时要求训练数据能够代表实际的应用环境。并且为了适应环境的变化,需要重新调整权值,这样当学到新知识时,就容易遗忘旧的知识。

(2) 无监督学习

无监督学习的训练数据集中,只有输入而没有目标输出,训练过程中神经网络将自动地提取各输入数据的特征,并将其分成若干类。经过训练好的网络能够识别训练数据集以外的新的输入类别,并相应获得不同的输出。

无监督学习可使网络具有自组织和自学习的功能,但它同时也带有某种不确定性,其训练和学习的精度有时不如监督学习高。

§ 1.5.3 基本学习规则

神经元是一个可调节的元素。我们可以根据其所收到的输入信号,输出值和相关的监督响应信号来调节它的权值。某些情况下无法获得监督信号,不能使用误差信息,这时神经元将只能根据输入/输出信号来修正它的权值。在学习开始前,权值必须得到恰当地初始。

下面是一些常用的关于神经元权值修正的学习规则。其中,权值向量 $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ 的增量 Δw_i 与输入信号 x 和学习信号 r 成正比,而学习信号 r 是关于 w_i 和 x ,有时还包括监督信号 d_i 的函数。

1. Hebb 学习规则

Hebb 学习规则是根据 Hebb 的假设提出的。Hebb 的假设是:特定模式下的神经元处在活动中时,其间的相互作用所引起的突触系数的变化将强化该模式趋于稳定。一种具体形式是:同时激活一对神经元 i 和 j 将增强其间的权值 w_{ij} ;而只有一个神经元被激活时,其间的权值 w_{ij} 将被减弱。

根据 Hebb 学习规则,学习信号 r 简单地等价于神经元的输出。我们有

$$r = f(\mathbf{w}_i^T \mathbf{x}) \quad (1.6)$$

权值向量 w_i 的增量为 Δw_i ,其中 c 为一正数,被称为学习常数(下同):

$$\Delta \mathbf{w}_i = c f(\mathbf{w}_i^T \mathbf{x}) \mathbf{x} \quad (1.7)$$

此学习规则要求在学习前用 $\mathbf{0}$ 附近的随机小量初始化 w_i 。它是一种纯反馈,无监督学习规则。在某些情况下,Hebb 规则需要作些修正,以防止权值无限制地增长,这种情况在兴奋信号和响应信号的符号保持一致时发生。这就要求权系数在达到某一程度时,应具有相应的饱和值。

2. 感知机学习规则

对于感知机学习规则,学习信号用神经元期望响应和实际响应之差来表示。因此,它是一种有监督学习方式,其学习信号为:

$$r = d_i - o_i \quad (1.8)$$

其中, $o_i = \text{sgn}(\mathbf{w}_i^T \mathbf{x})$, d_i 为该神经元的期望输出。从而,权值向量的增量 Δw_i 为

$$\Delta \mathbf{w}_i = c [d_i - \text{sgn}(\mathbf{w}_i^T \mathbf{x})] \mathbf{x} \quad (1.9)$$

此规则仅适用于二值化的神经元响应。当且仅当 o_i 不正确时,权值才被调整。由于期望的响应仅有 1 和 -1 两种,因此当需要调整权值时,上式又可以简化为:

$$\Delta \mathbf{w}_i = \pm 2c \mathbf{x} \quad (1.10)$$

3. Delta 学习规则

Delta 学习规则仅当激励函数 f 为连续函数时有效,并且处在有监督训练模式下。学习

信号 r 定义如下:

$$r = [d_i - f(\mathbf{w}_i^T \mathbf{x})] f'(\mathbf{w}_i^T \mathbf{x}) \quad (1.11)$$

此学习规则是以 o_i 和 d_i 之间的均方误差最小为条件推导出来的。将与 w_i 有关的平方误差来定义梯度因子如下:

$$E = (d_i - o_i)^2 / 2 \quad (1.12)$$

也就是

$$E = [d_i - f(\mathbf{w}_i^T \mathbf{x})]^2 / 2 \quad (1.13)$$

可得它的梯度向量值为:

$$\nabla E = -(d_i - o_i) f'(\mathbf{w}_i^T \mathbf{x}) \mathbf{x} \quad (1.14)$$

误差最小化要求其权值必须沿着梯度的负方向变化, 从而取

$$\Delta \mathbf{w}_i = -\eta \nabla E \quad (1.15)$$

其中 η 是一个正常数。综合上述结果, 有

$$\Delta \mathbf{w}_i = \eta (d_i - o_i) f'(\mathbf{w}_i^T \mathbf{x}) \mathbf{x} \quad (1.16)$$

开始训练时, 权值可以用任意值进行初始。此规则在多层神经网络中可以被进一步推广。它又被称为连续感知机训练规则。

4. Widrow-Hoff 学习规则

Widrow-Hoff 学习规则被用于神经网络的有监督训练。由于它只是用期望输出 d_i 与神经元的激励值 $\mathbf{w}_i^T \mathbf{x}$ 之间的最小平方误差为条件, 故而与神经元具体的激励函数形式无关。对应的学习信号定义如下:

$$r = [d_i - \mathbf{w}_i^T \mathbf{x}] \quad (1.17)$$

此时的权向量增量为

$$\Delta \mathbf{w}_i = c (d_i - \mathbf{w}_i^T \mathbf{x}) \mathbf{x} \quad (1.18)$$

可知, 此规则是 Delta 学习规则的特例。事实上如果 $f(\mathbf{w}_i^T \mathbf{x}) = \mathbf{w}_i^T \mathbf{x}$, Delta 学习规则就转变 Widrow-Hoff 学习规则。此规则有时被称为 LMS (最小均方) 学习规则。权值向量可以被赋予任意的初始值。

5. 关联学习规则

如果学习信号 $r = d_i$, 我们就可以得到关联学习规则。对应的权向量调整为:

$$\Delta \mathbf{w}_i = c d_i \mathbf{x} \quad (1.19)$$

这个简单的规则表明, 如果 d_i 是输入信号 x_j 的期望响应, 则它对应的权值增量正比于它们之积。此规则用于由二值化神经元组成的联想记忆网络, 以存储有关的记忆数据。它是 Hebb 学习规则的特例, 当且仅当其使用二值化激励函数, 并且 $o_i = d_i$ 时。与 Hebb 学习规则相似, 此学习规则也要求用 $\mathbf{w} = 0$ 初始化。

6. “胜者全得” (Winner-Take-All) 学习规则

本学习规则与前述的规则都不同，是从神经元总体的角度来考虑。它是一种竞争性学习方法，用于无监督网络训练。一般，“胜者全得”训练规则是针对具有统计特征的输入信号。如果某一个神经元（不妨设为第 m 个神经元）对输入信号有最大响应，如图 1.9，则此神经元被称为“胜者”，则其权向量 $w_m = [w_{m1}, w_{m2}, \dots, w_{mn}]^T$ 的增量按如下计算：

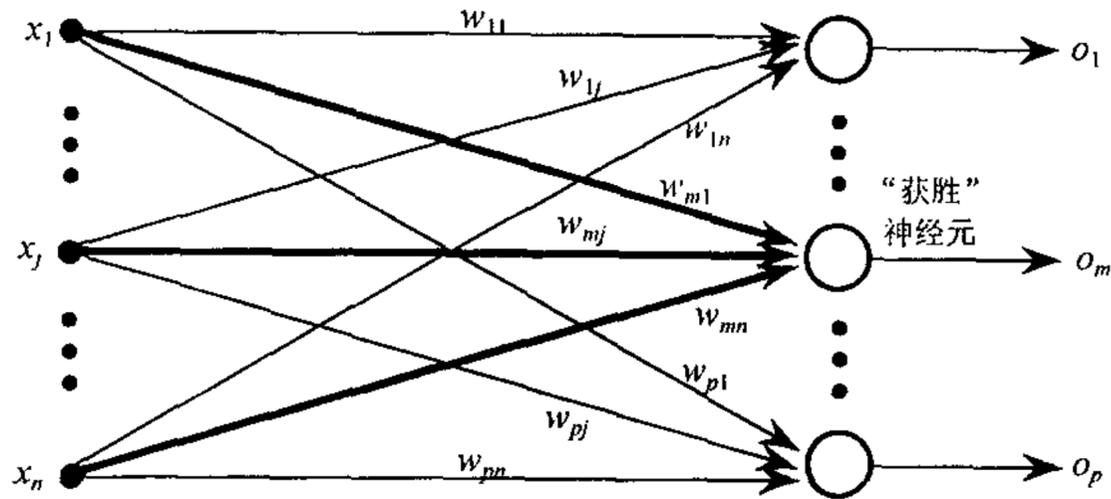


图 1.9 竞争型无监督“胜者全得”学习规则

$$\Delta w_m = \alpha(x - w_m) \quad (1.20)$$

其中， $\alpha > 0$ ，是一个小的学习常数，一般，它将在学习过程中递减。“胜者”的选取是按照如下的标准，从所有的 p 个神经元中以竞争的方式选出具有最大激励的神经元：

$$w_m^T x = \max_{i=1,2,\dots,p} (w_i^T x) \quad (1.21)$$

此标准实际上是找出最接近输入信号 x 的权向量，而且 w_m 调整的增量与 $(x - w_m)$ 成正比。注意到仅有“获胜”神经元的权向量被调整，调整后它将更接近于输入模式。在此方法中，获得调整的神经元有时会从这单个神经元扩展到它的邻域中，即包括它邻近的各个神经元。权向量随机地初始，并且在学习过程中向量的长度要求被归一化。

7. Outstar 学习规则

Outstar 学习规则也是从神经元总体的角度考虑的另一种学习规则。它假设图 1.10 所示的一层共 p 个神经元产生一个 p 维的期望响应向量 d 。本规则用于学习输入/输出关系中

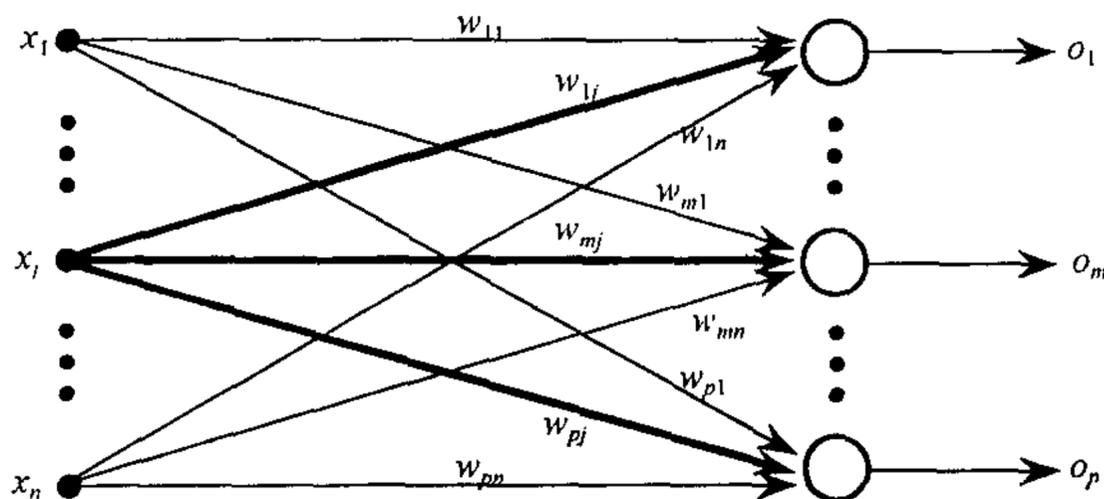


图 1.10 Outstar 学习规则

具有重复性和典型性的特征，它是有监督学习。但先要假定可以从网络中提取输入和输出信号的统计特性。权向量按如下规则调整：

$$\Delta \mathbf{w}_j = \beta(\mathbf{d} - \mathbf{w}_j) \quad (1.22)$$

必须注意，此时被调整的权向量 \mathbf{w}_j 的表示方式与前述的都不同，是所有涉及到输入向量中第 j 个分量的权数组成的向量，被定义为 $\mathbf{w}_j = [w_{1j}, w_{2j}, \dots, w_{pj}]^T$ 。

其中， β 是任意小的正值学习常数，并在训练过程中递减。

§ 1.5.4 典型的神经网络模型

典型的前向神经网络——BP 网络，是目前研究最多的网络形式之一，将在后续章节中详细讨论。这里将讨论一种典型的反馈型神经网络——Hopfield 网络。

由于前向网络是单向连接没有反馈的静态网络，从控制论的观点看，它缺乏系统动态性能。美国物理学家 Hopfield 对神经网络的动态性能进行深入研究，在 1982 年首先提出了一种非线性元件构成的单层反馈网络系统，这就是 Hopfield 网络系统。由于 Hopfield 网络是一个非线性动力学系统，而非线性系统本身就涉及到随机性、稳定性、吸引子以致于混沌现象，使得研究反馈网络要比前向网络复杂得多。

1. 离散 Hopfield 网络模型

如图 1.11 所示，Hopfield 模型是一个单层反馈的神经网络，它假定由 n 个门限为 θ_i 的神经元组成，第 i 个神经元的输出为 x_i ，外部输入信号为 I_i ，第 j 个神经元的输出信号与它连接的权值为 w_{ij} ，则第 i 个神经元的输入总和为：

$$y_i = \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} x_j + I_i - \theta_i \quad (1.23)$$

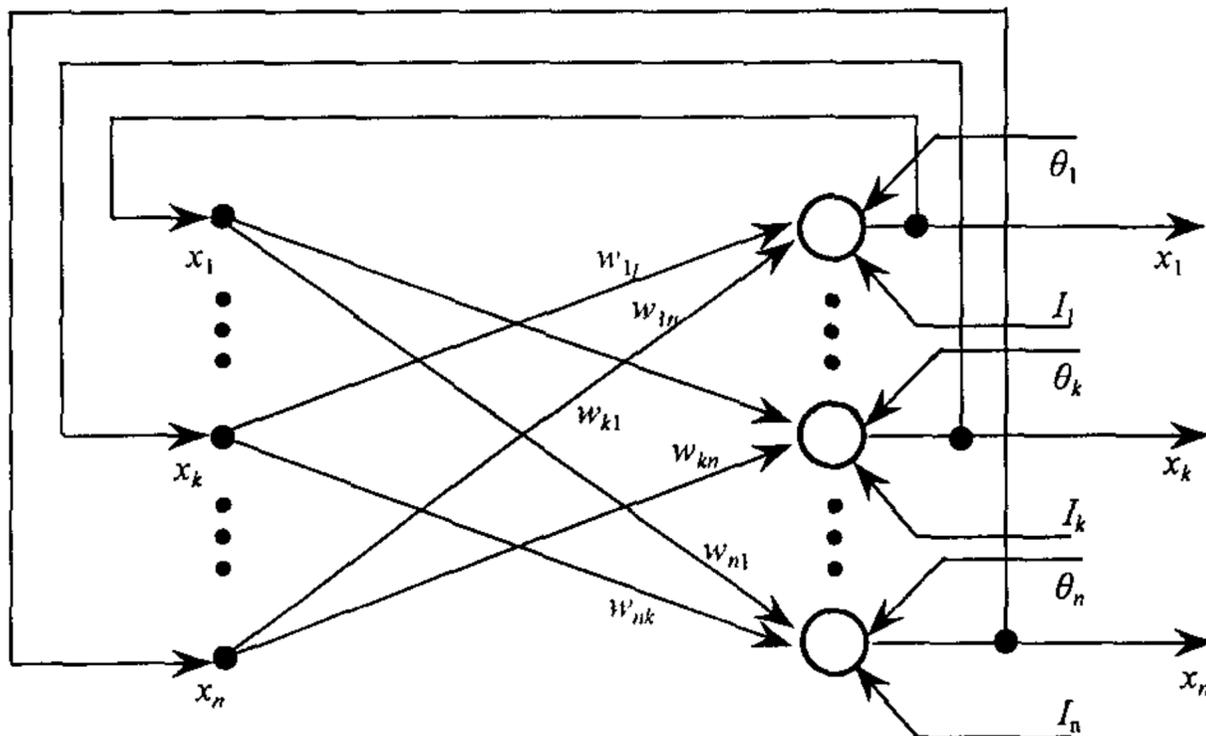


图 1.11 Hopfield 网络模型

进一步，将神经元的激励函数取为 $\text{sgn}(\cdot)$ ，从而对于第 i 个神经元有：

$$\begin{aligned} x_i &\rightarrow -1, \text{如果 } y_i < 0 \\ x_i &\rightarrow +1, \text{如果 } y_i > 0 \end{aligned} \quad (1.24)$$

根据 (1.23) 和 (1.24) 式, 我们可以得出输入信号 x_i 的修正算法:

$$x_i^{k+1} = \text{sgn} \left(\sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} x_j^k + I_i - \theta_i \right), \quad i=1,2,\dots,n \text{ 且 } k=0,1,\dots \quad (1.25)$$

为了衡量动态系统的稳定性, 我们引入如下的能量函数:

$$E = -\frac{1}{2} \sum_{i=1}^n x_i \sum_{j=1}^n w_{ij} x_j - \sum_{i=1}^n x_i I_i + \sum_{i=1}^n x_i \theta_i, (i \neq j) \quad (1.26)$$

从而,

$$\frac{\partial E}{\partial x_k} = -\frac{1}{2} \sum_{i=1}^n x_i (w_{ik} + w_{ki}) - I_k + \theta_k \quad (1.27)$$

如果 $w_{ik} = w_{ki}$, 则

$$\Delta E_k = -\left(\sum_{i=1}^n x_i w_{ik} - I_k + \theta_k \right) \Delta x_k \quad (1.28)$$

将 (1.23) 代入 (1.28) 式中, 即有

$$\Delta E_k = -y_k \Delta x_k \quad (1.29)$$

根据 (1.24) 式所定义的激励函数, 可知

$$\begin{aligned} y_i < 0 \text{ 时, } \Delta x_k &\leq 0 \\ y_i > 0 \text{ 时, } \Delta x_k &\geq 0 \end{aligned} \quad (1.30)$$

故而, 对于 (1.29) 式可得到 $\Delta E_k \leq 0$ 。所以我们可以得出以下结论: 在神经元的输入过程中, 整个神经网络的能量 E 朝着不增的方向变化, 最终将保持稳定。

2. Hopfield 网络用作联想记忆

从动力学的观点考虑, 由于网络的能量 E 是有界的, 对于输入的任意初态 x_0 , 经过有限次调整后, Hopfield 网络最终将稳定于某一稳定状态, 即能量函数的局部最小点。如果网络中有 p 个稳定状态 (吸引子, 记忆模式), 对于某一含有噪声的输入模式, 我们可以用 Hopfield 网络从 p 个记忆模式中, 联想出最接近的模式。用在文字识别上, 就是从已知的文字集中, 根据输入信息, 联想出最接近的文字样本。

关于网络权系数 w_{ij} 的确定, 将采用 Hebb 学习规则。假设有 p 个记忆模式, 它们是二值化的向量 $s^{(m)}$, 其中 $m=1,2,\dots,p$, 则权系数矩阵 W 为:

$$W = \sum_{m=1}^p s^{(m)} s^{(m)T} - pI \quad (1.31a)$$

或者

$$w_{ij} = (1 - \delta_{ij}) \sum_{m=1}^p s_i^{(m)} s_j^{(m)} \quad (1.31b)$$

其中, 当 $i=j$ 时, $\delta_{ij} = 1$; 否则 $\delta_{ij} = 0$ 。可知, 此权系数矩阵 W 为对称阵, 且对角线上的元素为 0。不过, 对于有 n 个神经元组成的网络, 实验表明其能存储的状态数 m 约为 $m = 0.15n$ 。如果过大, 就会在联想过程中产生状态间的交叉干扰, 从而出现联想错误。

3. 评价

(1) 上面所讨论的是双极型神经元模型, 即神经元的激励函数为 $\text{sgn}(\cdot)$ 。对于单极型神经元模型, 即当神经元的激励函数为 $u(\cdot)$ 时, 关于此时的记忆向量 $s^{(m)}$, 其权系数的计算公式应由 (1.31b) 式变为:

$$w_{ij} = (1 - \delta_{ij}) \sum_{m=1}^p (2s_i^{(m)} - 1)(2s_j^{(m)} - 1) \quad (1.32)$$

(2) Hopfield 网络将动力学、统计学的研究成果应用到网络中, 使得 Hopfield 模型是一个可精确求解的模型, 它的性质诸如容量、鲁棒性都可定量的表示出来; 也正因为这样, Hopfield 模型可以被看作是其他神经网络的标准模型。

(3) Hopfield 网络也有局限性: 如假定连接的权系数矩阵是对称的, 这与生物系统中非对称性的突触连接方式不一致; 假定网络是全连接的, 这与生物系统中的泛连接方式不完全一致; 由于 Hopfield 模型中引用的是 Lyapunov 二次型能量函数, 所以该网络不能解决高阶问题等等。

§ 1.6 神经网络的发展方向

脑神经系统是以离子电流机构为基础, 由神经细胞组成的非线性的 (Nonlinear)、可适应的 (Adaptive)、并行的 (Parrel) 和模拟的 (Analog) 网络 (Network), 简称 NAPAN。探索脑的组织结构和工作原理, 最终揭开人脑思维的奥秘, 是人类面临的一项重大挑战, 是整个自然科学研究的前沿领域。对人脑、对思维的认识的根本变化, 将深刻地影响到对人类自身、对整个世界的看法。关于脑科学的研究正以空前的复杂性、交叉性, 在各个科学领域内展开。

近年来, 神经网络的研究取得一定的进展, 构造了许多人工神经网络模型, 推出一些神经计算机的实验系统。但是, 由于脑神经系统的复杂性, 至今还没有可用于分析和设计 NAPAN 的理论, 对于涉及 140 亿个神经细胞所组成的超并行性, 层次和分布式构造所形成的系统本质特征, 人们还知之甚少。目前, 需要从系统论的立场出发来研究复杂的 NAPAN。在网络层次上弄清其功能和信息处理原理, 确定使其体系化的理论。在这种趋势中, 神经网络模型和学习算法的研究独具魅力。这种研究把许多简单的神经细胞模型, 并行地、分层地相互结合成网络模型, 提供了实现新的信息处理的一种手段, 为建立 NAPAN 理论提供了一种途径。

对神经网络的研究, 有助于探明大脑的信息处理方式, 建立脑模型, 进一步弄清楚脑的并行信息处理的基本原理, 并从应用角度寻求其工程实现的方法。目前, 神经网络的理论基础和实现技术的研究侧重于以下几个方面:

- (1) 思维模型的研究, 研究抽象思维、形象思维、灵感思维的机制。
- (2) 非线性动力学巨系统的理论研究, 包括混沌理论、微分动力系统、协同学等。
- (3) 神经网络模型的研究。
- (4) 学习算法的研究。
- (5) 记忆机制的研究。

- (6) 神经计算机体系结构的研究。
- (7) 模拟和数字光学神经网络的研究。
- (8) 光-电神经计算机系统的研究。
- (9) 生物神经计算机系统的研究。

很明显，对神经网络理论及其应用的研究，对整个人工智能的研究与进展都是十分重要的，尤其对智能计算机的研究具有特别的意义。把神经网络的研究成果与当前的计算机理论和技术结合起来，可能会实现更加智能的柔性信息处理系统，为最终实现智能计算机开辟一条可能的途径。

第二章 多层前馈神经网络

前馈网络是相对反馈网络而言的，即在运算过程中不存在任何反馈。从学习的观点看，前馈网络是一种强有力的学习系统，其结构简单而易于编程；从系统观点看，前馈网络是一静态非线性映射，通过简单非线性处理单元的复合映射可获得复杂的非线性处理能力。但是从计算观点看，前馈网络并不是一种强有力的计算系统，它缺乏丰富的动力学行为。大部分前馈网络都是学习网络，并不注重系统的动力学行为，它们的分类能力和模式识别能力一般都强于反馈网络。典型的前向网络是采用多层感知机模型的 BP 网络，它是目前研究最多的网络形式之一。

§ 2.1 感知机

感知机(Perceptron)是由美国学者 F.Rosenblatt 于 1957 年提出的，它是一个具有单层计算单元的神经网络，并由线性阈值元件组成，即单元的输出函数为：

$$y(t) = \begin{cases} 1, & \text{若 } \sum_{i=1}^n w_i(t)x_i - \theta \geq 0 \\ -1 \text{ 或 } 0, & \text{其他} \end{cases} \quad (2.1)$$

其中 x_i 为输入向量 x 第 i 个分量， $w_i(t)$ 为 t 时刻与输入分量 x_i 对应的权值并可调整。原始的 Perceptron 只具有一个输出节点，它相当于单个神经元，结构如图 2.1 所示

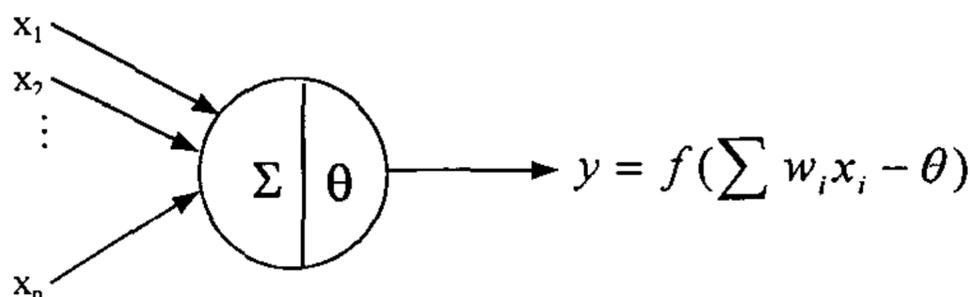


图 2.1 感知机模型

感知机的学习规则为

$$w_i(t+1) = w_i(t) + \eta[d - y(t)]x_i \quad (2.2)$$

其中 η 为学习率 ($0 < \eta < 1$)， d 为期望输出（监督信号）， $y(t)$ 是实际输出。通过不断调整权值，使得 w_i 对一切样本均保持不变，则学习结束。

当用感知机进行多类别模式识别时，几何上相当于在高维空间中，用几个超平面将各类样本分开。超平面的个数与神经元数目相对应。例如对于三类模式识别的问题，可使用两个神经元，其结构与几何解释如图 2.2 所示。

其中，A、B、C 分别为要区分的三类目标在两个分类特征组成的平面中出现的区域；线①和线②分别是神经元 1 和神经元 2 形成的判决函数。

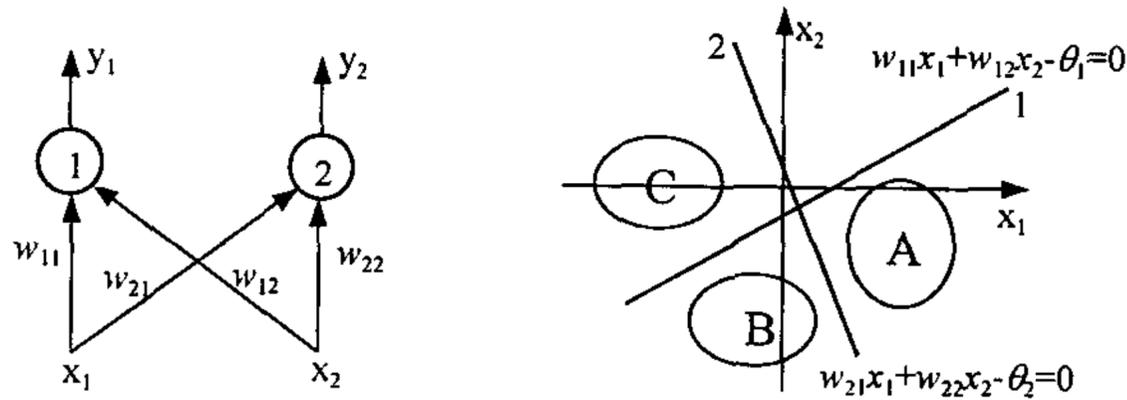


图 2.2 多神经元感知机的结构与其功能的几何解释

Rosenblatt 已经证明：如果两类模式是线性可分的（指存在一个超平面将它们分开），则算法一定收敛，也就是说权矩阵 w 一定存在；否则，判决边界会产生振荡，以致 w 不收敛，所以感知机有它的局限性。例如，如果模式 A, B, C, D 呈图 2.3 分布，其中模式 A

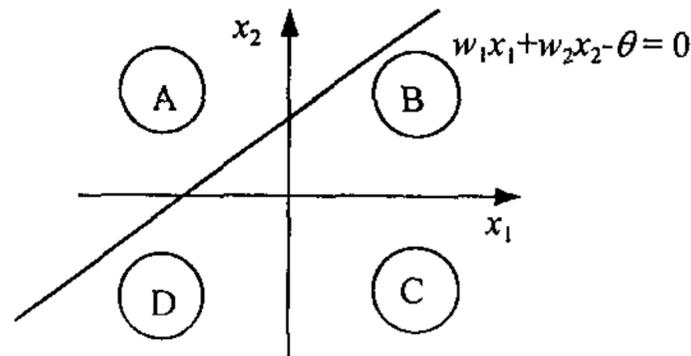


图 2.3 单层感知机模型的局限性

和 C 为一类，模式 B 和 D 为另一类时，采用一个神经元来区分这两类时，可见从图中找不到这样一条直线（感知机或神经元）来划分这两类。典型的的就是所谓的“异或”谓词逻辑问题：A, B, C, D 分别对应于输入向量 $(-1, 1)$ 、 $(1, 1)$ 、 $(-1, -1)$ 和 $(1, -1)$ ，它们的谓词运算如表 2.1 所示。可知，用上述的单层感知机模型无法解决此类问题。

表 2.1 异或运算关系

输入	输出
-1, -1	-1
-1, +1	+1
+1, -1	+1
+1, +1	-1

§ 2.2 多层感知机的结构和性能

多层感知机是一种在输入和输出节点之间含有一层或多层隐含节点的前馈网络。这些附加的中间层含有不直接与输入节点或输出节点相连的隐含单元或隐含节点。多层感知机克服了单层感知机的许多局限。

§ 2.2.1 异或问题的解决

有前述单层感知机不能实现异或运算，但只要在输入和输出之间增加一个隐含层，如图 2.4(a)所示，形成的两层感知机就可以完成异或谓词逻辑，此时第一层感知机的决策域如图 2.4(b)所示。

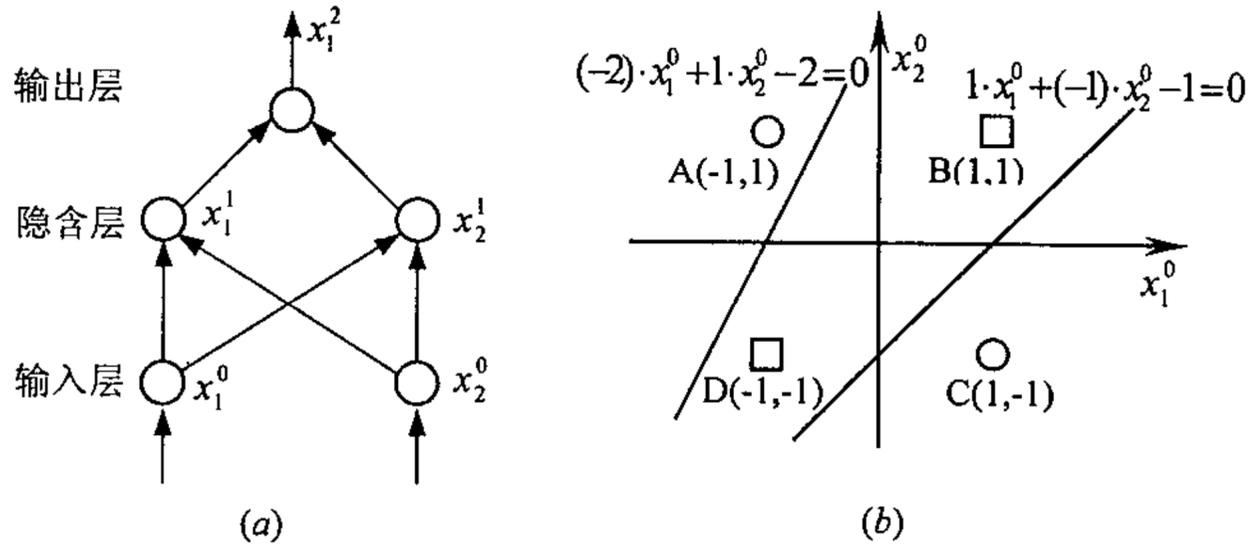


图 2.4 多层感知机及决策域

图 2.4(b)表明，第一层感知机将 x_1^0, x_2^0 组成的模式平面分成了三个部分。模式按如下关系映射到映像空间：

$$\begin{aligned} x_1^1 &= \text{sgn}((-2) \cdot x_1^0 + 1 \cdot x_2^0 - 2) \\ x_2^1 &= \text{sgn}(1 \cdot x_1^0 + (-1) \cdot x_2^0 - 1) \end{aligned} \quad (2.3)$$

映射的具体关系如表 2.2 所示。第一层感知机提供输入模式到映像空间的映射；而第二层实现对映像而不是原始模式的分类功能。与原始模式相对应的映像空间如图 2.5 所示，注

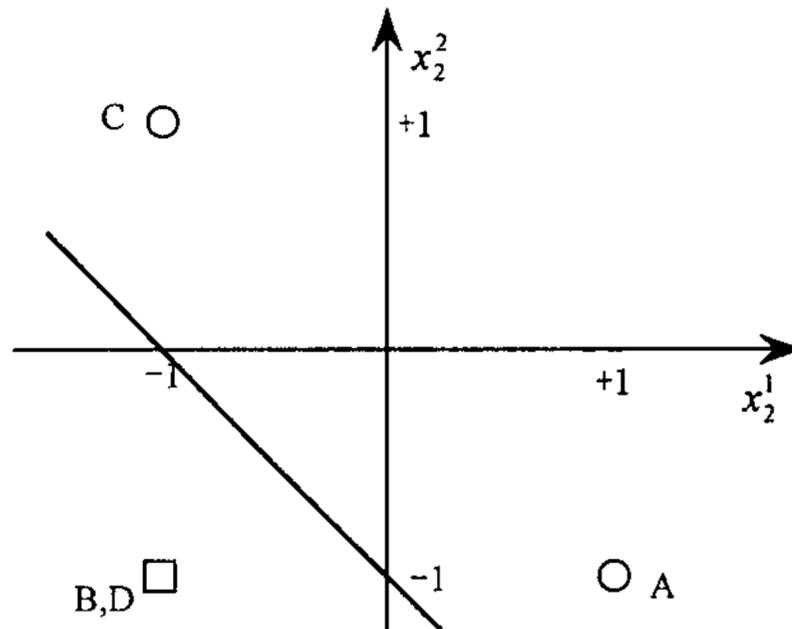


图 2.5 映像空间的决策线

意到此时输入模式 B 和 D 在映像空间中被合并成一个映像点 $(-1, -1)$ 。如下定义一条决策线，以实现期望的分类并分离映像空间中 A, C 点和 B, D 点：

$$x_1^1 + x_2^1 + 1 = 0 \quad (2.4)$$

输出节点相应的输出函数为：

$$x_1^2 = \text{sgn}(1 \cdot x_2^1 + 1 \cdot x_2^1 - (-1)) \quad (2.5)$$

输入模式和输出空间之间完整的映射关系如图 2.6 所示。

表 2.2 分类统计表

符号	模式空间 x_1^0, x_2^0	映像空间 x_1^1, x_2^1	$x_1^1 + x_2^1 + 1$	输出空间 x_1^2	类号
A	-1, +1	+1, -1	+	+1	1
B	+1, +1	-1, -1	-	-1	2
C	+1, -1	-1, +1	+	+1	1
D	-1, -1	-1, -1	-	-1	2

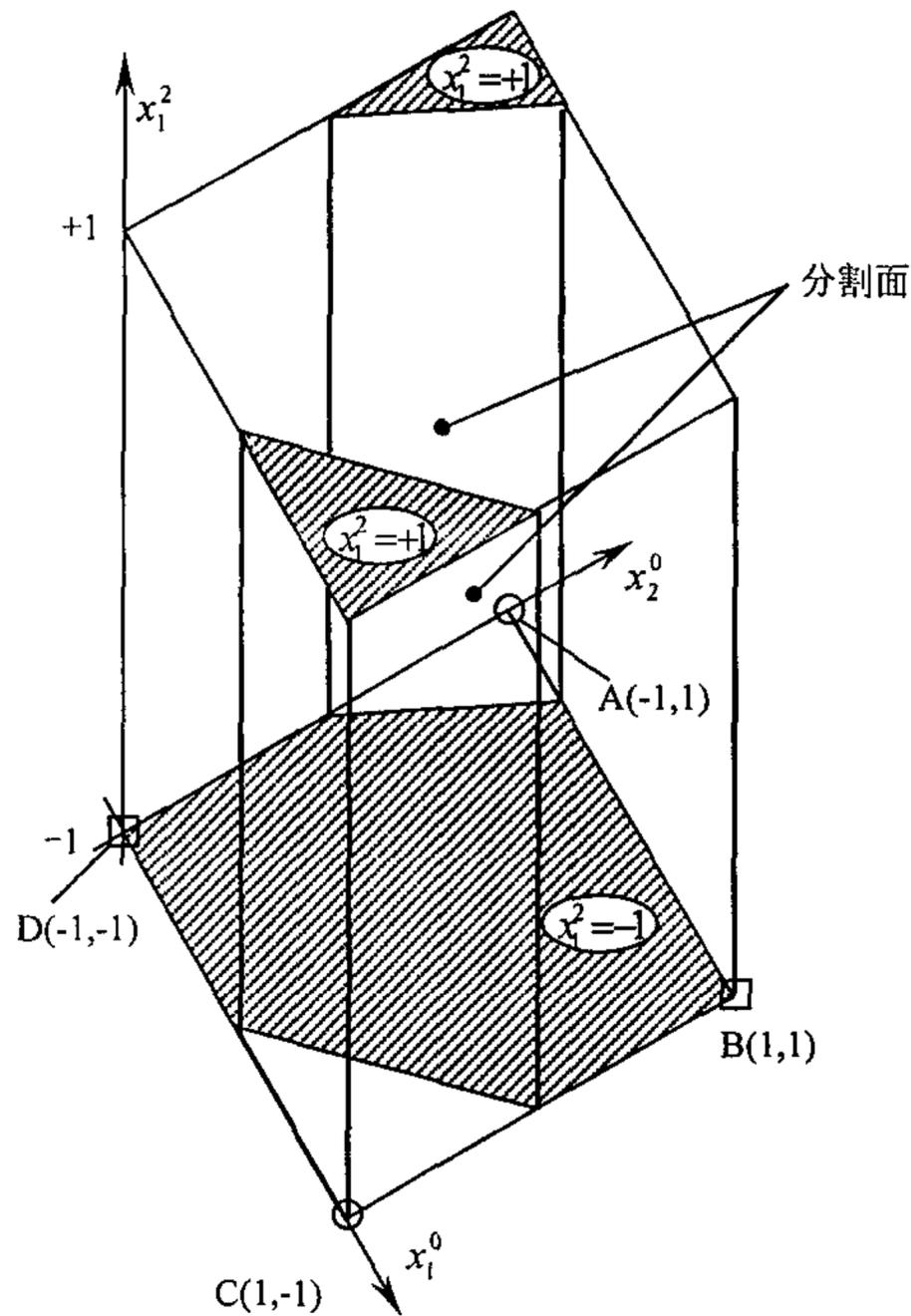


图 2.6 多层感知机中输入到输出的映射关系图

§ 2.2.2 多层感知机和分类区域的关系

多层感知机的能力来自于各节点中的非线性函数 $f(\cdot)$ 。如果节点是线性元件，则只要适当地选择权重就可以用一个单层网去复制多层网的计算行为。当 $f(\cdot) = \text{sgn}(\cdot)$ 时，一、

二、三层感知机的决策域如图 2.7 所示。图中第 2 列显示出各种网络的决策域形式，第 3，4 列是异或问题和网状域问题所形成的决策域例子，第 5 列是能形成的最一般的决策域例子。

有前述可知，单层感知机形成一个半平面的决策域；两层感知机可识别任一凸多边形或无界的凸区域。凸域是由多层感知机中第一层各个节点所形成的半平面相互重叠而成的。第一层中的各个节点就象一个单层感知机一样，当输入点落在由权重和阈值形成的超平面上时，它们对应的输出节点就输出 1，否则为-1。如果从第一层的所有 N_1 个节点到一个输出节点的权重均为 1，并且这个输出节点的阈值为 $N_1 - \epsilon$ ($0 < \epsilon < 1$)，则只有当第一层的各个节点均向其输入 1 时，该输出节点才输出 1。这就相当与一个“逻辑与”运算。最终的决策域是由第一层所形成的一切半平面重叠而得。这些半平面重叠后就形成了上述的凸区域。并且此凸区域的边数不会超过第一层中的节点个数。

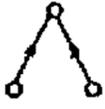
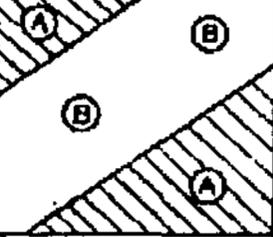
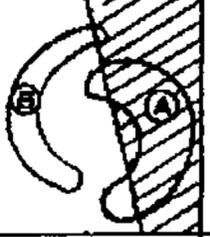
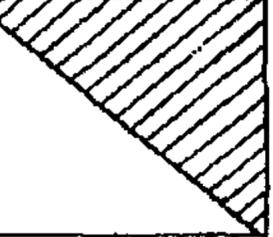
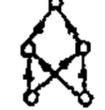
网络结构	分类区域类型	对异或(XOR)分类	网状分类区域	分类区域一般形状
无隐层 	由超平面分成二个区域			
单隐层 	开凸区域或闭凸区域			
2个隐层 	任意形状 (但与节点数之间的关系复杂)			

图 2.7 感知机网络结构与分类区域的关系

以上分析有助于选择两层感知机中的节点个数。首先，节点个数应该多到能形成已给问题所需要的复杂的决策域。但是另一方面，节点数太多，从已知的训练数据中就不能得到所需的众多权值的可靠估计。例如，图 2.7 中第二行显示 2 个节点就足以解决异或问题。但是无论用多少个节点，都不可能用两层感知机将图 2.7 第 2 行中的网状区域类分开。

三层感知机可形成任意复杂区域的决策域，可将图 2.7 中的网状类区域分开。这从其结构上可得到证实。它的思想是：把所希望的决策域分成许多小的超立方体（当只有两个输入时就是正方形），每一个超立方体需要在第一隐含层有 $2N$ 个节点（ N 为输入节点数），每个节点形成超立方体的一个边。第二隐含层的每一个节点对第一隐含层单元的输出作“逻辑与”运算。只有当第一隐含层的输出均在立方体内时，第二隐含层才输出 1。将超立方体按下述方法分给合适的决策域：第二隐含层节点仅于特定的输出层节点相连，这些输出层节点对应于超立方体的决策域，并且每个输出层节点执行“或”运算。如果第二隐含层到输出层的连接权值均为 1，并且输出节点的阈值为 0.5，即可产生逻辑“或”运算。这种结构还可以推广到任意形状的凸区域而不是小的超立方体，并且还能产生如图 2.7 所示的那种互不相连的非凸区域。

以上分析显示三层感知机模型(含两个隐含层的网络)可以产生任意复杂的决策域,因此我们不必采用三层以上的感知机型前馈网络就可以达到所需的目的。从上面分析中还可以了解到在用三层感知机解决具体问题时,应该如何选择节点的个数。如果决策域是互不相连的或是网状并且不能由一个凸面形成,则第二隐含层中的节点个数一定要大于1。在最坏的情况下,第二隐含层所需的节点数等于输入分布中互不相连的区域个数。第一隐含层的节点数必须充分多,使得第二隐含层每个节点所产生的凸区域能从第一隐含层得到3条或更多的边。因此,第一隐含层的节点数至少要比第二隐含层的节点数多三倍。

§ 2.2.3 前馈网络的一般性研究

前馈网络的输入和输出关系可以看作是一种映射关系。从系统观点看这种映射是高度非线性的映射。它的信息处理能力也来自于简单非线性函数的多次复合。前面的分析表明,使用三层感知机可以产生任意复杂的决策域。进一步,对于前馈网络的信息处理能力,它们的识别能力,以及三层网络的构造问题,有下面的Kolmogrov定理:

定理 2.1 (Kolmogrov 定理) m 维单位立方体 $E^m = [0, 1]^m$ 中的任意一个连续函数 $\Phi: E^m \rightarrow R^p$, $\Phi(x) = Y$ 都可以用一个三层神经网络去精确地实现。并且此网络的第一层有 m 个处理单元,中间层有 $2m+1$ 个处理单元,第三层有 n 个处理单元。

定理 2.2 假设 Φ 为有界单调递增的函数, K 为 R^n 上的紧致子集(有界闭子集), $f(x) = f(x_1, \dots, x_n)$ 为 K 上的实值连续函数,则对于任意 $\varepsilon > 0$ 存在整数 N 和实常数 $C_i, \theta_i (i=1, 2, \dots, N)$ 和 $w_{ij} (1 \leq i, j \leq N)$ 使得

$$\hat{f}(x_1, \dots, x_n) = \sum_{i=1}^N C_i \Phi \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right)$$

满足 $\max \left| \hat{f}(x_1, \dots, x_n) - f(x_1, \dots, x_n) \right| < \varepsilon$, 即是说对于任意 $\varepsilon > 0$ 存在一个三层网络,其隐

单元输出函数为 $\Phi(x)$, 输入和输出单元的输出函数是线性的,此网络还满足

$$\max_{x \in K} \left| \hat{f}(x_1, \dots, x_n) - f(x_1, \dots, x_n) \right| < \varepsilon$$

其中 $f(x_1, \dots, x_n)$ 为三层网络的总输入输出关系。

定理 2.3 设 $\varphi(x) = \varphi(x_1, \dots, x_n)$ 为有界单调连续递增函数, K 为 R^n 的紧致子集(有界闭子集),固定层数 $k \geq 3$,则对于任意连续映射 $f: K \rightarrow R^m$ 可用一个 k 层($k-2$ 个隐层)网络的输入和输出映射来逼近。此网络的隐单元输出为 $\varphi(x)$, 输入和输出之间的关系是线性的。即对于任意连续映射 $f: K \rightarrow R^m$ 和任意 $\varepsilon > 0$ 存在一个 $k(k \geq 3)$ 层网络可一致逼近此映射 f 。

§ 2.3 BP 模型及其算法推导

以上讨论是针对感知机进行的,感知机学习算法当函数线性可分时保证收敛,但它存在如下问题:

(1) 由于隐含层监督信号未知,不能推广到一般的前向网络中去,不适于多层网络学习。

(2) 函数不是线性可分时得不到任何结果。

为了克服上述缺点，人们提出了梯度算法（即 LMS 算法），为此需要把单元特性改为可微函数，如 Sigmoid 形式，由此建立了 BP 网络。

上面已经讨论了具有阈值型输出函数的神经元所组成的多层感知机的性质，当把阈值函数换成 S 型输出函数时，网络的形为特征基本不变，这时其判别规则为判输入样本为输出值最大的输出单元所代表的那一类。由于 S 型代替了阈值型，所以划分区域的直线段变为平滑的曲线，因此这种网络的性质相对也更难分析。

§ 2.3.1 BP 算法推导

BP (Error Back-Propagation) 算法最早是由哈佛大学博士 Werbos 在 1974 年发明的，当时未受到重视，直到 1985 年 Rumelhart 和 Parker 再次发现，并将它用于多层神经网络的权值训练后，才得到广泛的推广。BP 算法是一种采用 Delta 学习规则的有监督训练方法，其主要思想是：计算输出层各个节点的实际输出与期望输出之间的误差，并且把误差由网络的输出层等值地向各个隐含层进行反向传播，即保证各个隐含层和输出层的输出误差均相等；再沿输出误差函数的等高线梯度下降方向，连续不断地调整网络的权重，以实现网络的实际输出与期望输出的均方误差最小化。

其具体的推导过程如下：

如图 2.8，网络的输入和输出分别用 y_j 和 o_k 表示，其中 $j=1,2,\dots,J$ 和 $k=1,2,\dots,K$ 分别对应于上一层的第 j 个节点和本层的第 k 个节点；同样，权系数 w_{kj} 用于连接第 j 个节点的输出和第 k 个节点的输入。

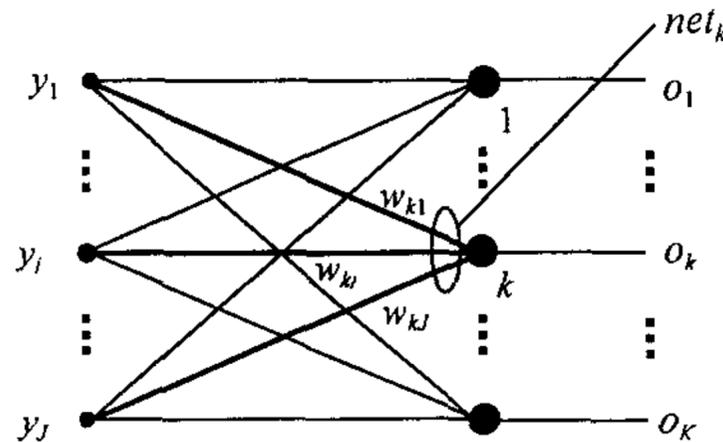


图 2.8 BP 神经网络内部某一层关于输入的连接

对于特定的一个输入模式 p , $p=1,2,\dots,P$, 定义它的输出误差函数为：

$$e_p = \frac{1}{2} \sum_{k=1}^K (d_{pk} - o_{pk})^2 = \frac{1}{2} \|\mathbf{d}_p - \mathbf{o}_p\|^2 \quad (2.6)$$

其中， \mathbf{d}_p 为网络的期望输出向量，而 \mathbf{o}_p 为网络的实际输出向量。

根据 LMS 思想，每个权值的修正方向应为平均误差的梯度方向，即 $\frac{\partial E[e_p]}{\partial w_{kj}}$ 。但这并不实用，因为这样需要对训练集上的所有的样本进行平均计算，所以，一般都采用瞬时梯

度 $\frac{\partial e_p}{\partial w_{kj}}$ 来代替平均梯度 $\frac{\partial E[e_p]}{\partial w_{kj}}$ 。在省略误差 e 的下标 p 后，得到如下的 Delta 学习规则：

$$\Delta w_{kj} = -\eta \frac{\partial e}{\partial w_{kj}} \quad (2.7)$$

对于该层的第 k 个节点, $k=1,2,\dots,K$, 其输入的激励信号 net_k 为:

$$net_k = \sum_{j=1}^J w_{kj} y_j \quad (2.8)$$

则该神经元的输出

$$o_k = f(net_k) \quad (2.9)$$

很明显, 梯度元素 $\frac{\partial e}{\partial w_{kj}}$ 仅与第 k 个神经元的一个中间变量, 即激励信号 net_k 有关。根据链

式法则, 有

$$\frac{\partial e}{\partial w_{kj}} = \frac{\partial e}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{kj}} \quad (2.10)$$

上式右边中的第二个因子根据(2.8)式, 可知

$$\frac{\partial net_k}{\partial w_{kj}} = y_j \quad (2.11)$$

将(2.11)式右边的第一个因子定义为误差信号因子 δ , 即:

$$\delta_k^{(m)} \triangleq -\frac{\partial e}{\partial (net_k)} \quad (2.12)$$

其中, m 为对应的层号, k 是表示关于该层第 k 个神经元的误差信号因子。结合(2.11)式和(2.12)式, 可将(2.10)式中的瞬时误差梯度进一步转变成

$$\frac{\partial e}{\partial w_{kj}} = -\delta_k^{(m)} y_j \quad (2.13)$$

从而, 可将(2.12)式的权值调整方程表示成关于误差信号因子 δ 的关系式:

$$\Delta w_{kj} = \eta \delta_k^{(m)} y_j \quad (2.14)$$

为了能够真正实现权值的调整, 需要根据连续信号的感知机模型, 计算出(2.12)中的误差信号因子 $\delta_k^{(m)}$, 而这是 BP 算法推导的关键。

注意到误差 e 是关于 net_k 的复合函数, 因此可有如下的表达式, 其中 $k=1,2,\dots,K$:

$$e(net_k) = e[o_k(net_k)] \quad (2.15)$$

这样, 我们由(2.12)式可得:

$$\delta_k^{(m)} = -\frac{\partial e}{\partial o_k} \cdot \frac{\partial o_k}{\partial (net_k)} \quad (2.16)$$

其中, 上式右边的第二个因子实际上是神经元激励函数的导数, 即

$$f'_k(net_k) = \frac{\Delta \partial o_k}{\partial (net_k)} \quad (2.17)$$

关于(2.16)式中右边的第一个因子，需要做如下的讨论：

(1) 当 m 层为输出层时，注意到

$$\frac{\partial e}{\partial o_k} = -(d_k - o_k) \quad (2.18)$$

由于是输出层，故上面的监督变量 d_k 和输出变量 o_k 均是可求得的。因此，对于这种情况，(2.16)式中误差信号因子 $\delta_k^{(m)}$ 可以由下面的公式计算出来，其中 $k=1, 2, \dots, K$ ：

$$\delta_k^{(m)} = (d_k - o_k) \cdot f'_k(net_k) \quad (2.19)$$

公式(2.19)表明，输出层的误差信号因子 $\delta_k^{(m)}$ 与本地信号误差 $(d_k - o_k)$ 成正比，且其比例因子为 $f'_k(net_k)$ 。从而可知这种情况下，关于权值最终的调整公式为：

$$\Delta w_{kj} = \eta (d_k - o_k) \cdot f'_k(net_k) y_j \quad (2.20)$$

但是，如果 m 层为隐含层，由于监督变量 d_k 无法获得，所以不能用(2.18)式来计算误差信号因子 $\delta_k^{(m)}$ ，而要从另外的角度来考虑。

(2) 当 m 层为隐含层时，根据 BP 算法中误差反向传递的假设，可知每一层的输出误差均相同，即

$$e^{(m+1)} = e^{(m)} \quad (2.21)$$

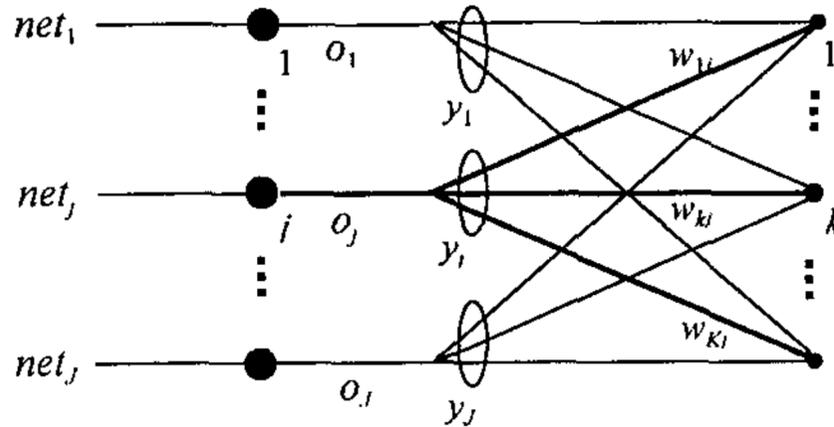


图 2.9 BP 神经网络内部某一层关于输出的连接

又由于的 m 层输出即为的 $m+1$ 层输入，如图 2.9 所示。因此我们有：

$$y_j^{(m+1)} = o_j^{(m)} \quad (2.22)$$

为了将讨论的情况一致，我们在不影响推导正确性的情况下，把(2.16)式中的下标 k 换成下标 j ，从而由(2.21)式和(2.22)式，我们有

$$\frac{\partial e^{(m+1)}}{\partial y_j^{(m+1)}} = \frac{\partial e^{(m)}}{\partial o_j^{(m)}} \quad (2.23)$$

这就是使得我们可以将 m 层的误差信号因子 $\delta_k^{(m)}$ 用 $m+1$ 层的误差信号因子 $\delta_k^{(m+1)}$ 来表示。通过前面已算得的输出层的误差信号因子，反向逐层算出各个隐含层的误差信号因子。具体的推导步骤如下：

因为每一个节点的输出都是其输入的复合函数，即

$$o_k = f_k(\text{net}_k) = f_k(\text{net}_k(\mathbf{Y})) \quad (2.24)$$

其中， $\mathbf{Y} = [y_1, y_2, \dots, y_J]^T$ 。

从而可知，偏微分项 $\frac{\partial e^{(m+1)}}{\partial y_j^{(m+1)}}$ 与 $m+1$ 层的各个神经元的中间变量，即激励信号 net_k 均

有关，根据链式法则，我们有

$$\frac{\partial e^{(m+1)}}{\partial y_j^{(m+1)}} = \sum_{k=1}^K \frac{\partial e^{(m+1)}}{\partial \text{net}_k^{(m+1)}} \frac{\partial \text{net}_k^{(m+1)}}{\partial y_j^{(m+1)}} \quad (2.25)$$

根据 (2.12) 式中误差信号因子 δ 的原始定义，以及 (2.8) 式，我们有 (2.25) 式的如下表示：

$$\frac{\partial e^{(m)}}{\partial o_j^{(m)}} = \frac{\partial e^{(m+1)}}{\partial y_j^{(m+1)}} = -\sum_{k=1}^K \delta_k^{(m+1)} w_{kj} \quad (2.26)$$

从而，当 m 层为隐含层时，(2.16) 式中误差信号因子的计算公式为：

$$\delta_j^{(m)} = f'_i(\text{net}_i) \cdot \sum_{k=1}^K \delta_k^{(m+1)} w_{kj} \quad (2.27)$$

调整后的权值为

$$w'_{kj} = w_{kj} + \Delta w_{kj}, \quad \text{其中 } k=1,2,\dots,K \text{ 以及 } j=1,2,\dots,J \quad (2.28)$$

下面考虑激励函数。为了便于计算推导，此时的感知机输出函数不再采用前述的阶跃型函数，而是采用单极连续，具有一定陡度的 Sigmoid 函数，即

$$f(\text{net}) = \frac{1}{1 + \exp(-\lambda \cdot \text{net})} \quad (2.29)$$

为简单起见，将陡度参数 λ 取为 1。从而有

$$f'(\text{net}) = \frac{\exp(-\text{net})}{[1 + \exp(-\text{net})]^2} \quad (2.30a)$$

又可以将上式写为

$$f'(\text{net}) = \frac{1}{[1 + \exp(-\text{net})]} \cdot \left(1 - \frac{1}{[1 + \exp(-\text{net})]} \right) \quad (2.30b)$$

可将上式写成用输出值表示的更有用的形式：

$$f'(\text{net}) = o \cdot (1 - o) \quad (2.30c)$$

可将激励函数的导数用如此简单地输出值关系式表达出来，这是使用 Sigmoid 函数的作为 BP 神经网络标准激励输出函数的最主要原因。

综上所述, 我们可得学习算法的权值修正公式:

(1) 当节点 i 是输出层节点时,

$$\Delta w_{ij}^{(m)} = \eta f_i'(net_i)(d_i - o_i)y_j = \eta o_i(1 - o_i)(d_i - o_i)y_j \quad (2.31a)$$

(2) 当节点 i 是隐含层节点时,

$$\Delta w_{ij}^{(m)} = \eta f_i'(net_i) \left(\sum_{k=1}^K \delta_k^{(m+1)} w_{ki}^{(m+1)} \right) y_j = \eta h_i(1 - h_i) \left(\sum_{k=1}^K \delta_k^{(m+1)} w_{ki}^{(m+1)} \right) y_j \quad (2.31b)$$

公式(2.31)中, o_i 是输出节点 i 的输出, h_i 是隐含节点 i 的输出, y_j 是节点 j 的输入。

有时为了收敛得快些, 可增加冲量项使得权值的变化更平滑些:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i y_j + \alpha(w_{ij}(t) - w_{ij}(t-1)) \quad (2.32)$$

其中, $0 < \alpha < 1$ 。

§ 2.3.2 BP 学习算法

BP 学习算法的具体过程如下:

(1) 初始化:

置所有权为随机任意小。

(2) 提供训练集:

提供连续的输入向量 x_0, x_1, \dots, x_{N-1} 和所期望的输出 t_0, t_1, \dots, t_{M-1} 。

(3) 计算实际输出:

使用 $f(\alpha) = \frac{1}{1 + e^{-(\alpha - \theta)}}$ 非线性函数和公式 $y_j = f\left(\sum_{i=0}^{N-1} w_{ij} x_i - \theta_j\right)$ 逐级计算输

出值。令最后的输出为 o_0, o_1, \dots, o_{M-1} 。

(4) 调整权值:

用递归的方法从输出节点开始返回到中间隐含层按下式调整权值:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j o_i$$

其中, $w_{ij}(t)$ 是时刻 t 或来自隐节点 i 的权, 或是从输入到节点 j 的权; o_i 或是节点 i 的输出, 或是输入; η 为大于零的增益; δ_j 为节点 j 的误差: 若节点 j 是输出节点, 则 $\delta_j = o_j(1 - o_j)(d - o_j)$, 其中 d_j 是节点 j 的期望输出, o_j 是节点 j

的实际输出; 若节点 j 是内部隐节点, 则 $\delta_j = h_j(1 - h_j) \sum_k \delta_k w_{kj}$, 其中 k 是

高于节点 j 的层次上的所有节点, h_j 是隐含节点 j 的输出。在实际应用中, 更多使用下式调整权值:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j o_i + \alpha(w_{ij}(t) - w_{ij}(t-1)),$$

其中 α 称为平滑因子, 一般取 $0 < \alpha < 1$ 。

(5) 返回第 2 步重复之, 直至误差 e 满意为止。

§ 2.3.3 BP 模型评价

BP 学习算法是目前应用最广泛的学习算法,从本质上讲, BP 算法把对一组样本的输入输出问题变为一个非线性优化问题。它使用了优化技术中最普通的一种梯度下降算法。用迭代运算求解权值相应于学习记忆问题。加入隐含节点可使优化问题的可调参数增加,这样可达到更精确的解。

但是 BP 算法本身尚存有一些不足之处:

(1) 由于它本质上是一种非线性优化问题,这就不可避免地会遇到最常见的局部极小,致使问题得不到最优解。

(2) 学习算法收敛速度很慢,通常要数千步以上。

(3) 生物系统中无此现象:信息向前、向后流过一个 BP 网络,同时还要使用导数。

(4) 网络隐节点的个数的选取尚无理论上的依据,而只能根据经验选取。

(5) BP 在学习新样本时又有遗忘已学过样本的趋势,同时刻划每个样本的特征的数目也要求必须相同。

§ 2.4 BP 算法的改进算法

针对 BP 算法的缺点,国内外进行了不少的改进,下面介绍两种改进算法。

§ 2.4.1 MBP 算法

MBP 算法是通过改变神经元的激励函数 $f(x)$ 来改善 BP 算法的收敛特性,达到加快其收敛速度的目的。标准的 BP 算法,将 (2.29) 式中 Sigmoid 函数的陡度参数 λ 固定为 1,而 MBP 算法则把它也参加训练,随神经元的权值 w_{kj} 和阈值 θ_k 一起发生变化。从而, m 层第 k 个神经元的输出函数由 $o_k = f_k(net_k)$ 变为

$$o_k = f_k(\lambda_k net_k) \quad (2.33)$$

有关参数变量的定义仍与 BP 算法推导时相同,此时的误差信号因子 δ 仍为

$$\delta_k = -\frac{\partial e}{\partial(net_k)} = -\frac{\partial e}{\partial o_k} \cdot \frac{\partial o_k}{\partial(net_k)} \quad (2.34)$$

虽然,此时输出函数的形式改变了,但根据前面 BP 算法的推导知, (2.34) 式中右边的第

一个因子,即偏微分量 $\frac{\partial e}{\partial o_k}$, 不论当 m 层是输出层还是隐含层时,其计算形式仍然不变,

与标准的 BP 算法相同。

但是对于 (2.34) 式右边的第二个因子有

$$\frac{\partial o_k}{\partial(net_k)} = f'_k(\lambda_k net_k) \lambda_k \quad (2.35)$$

从而此时的误差信号因子 δ 的计算公式由 (2.19) 和 (2.27) 式变为

(1) 当节点 k 是输出层节点时,

$$\delta_k^{(m)} = (d_k - o_k) f_k'(\lambda_k net_k) \lambda_k \quad (2.36a)$$

(2) 当节点 k 是隐含层节点时,

$$\delta_k^{(m)} = \left(\sum_{j=1}^J \delta_j^{(m+1)} w_{jk}^{(m+1)} \right) f_k'(\lambda_k net_k) \lambda_k \quad (2.36b)$$

同时, 还可以计算出误差 e 对于陡度参数 λ_k 的梯度, 根据链式法则, 我们有

$$\frac{\partial e}{\partial \lambda_k} = \frac{\partial e}{\partial net_k} / \frac{\partial o_k}{\partial net_k} \cdot \frac{\partial o_k}{\partial \lambda_k} \quad (2.37)$$

其中, (2.37) 式右边第三项的计算公式为

$$\frac{\partial o_k}{\partial \lambda_k} = f_k'(\lambda_k net_k) net_k \quad (2.38)$$

将公式 (2.34)、(2.35) 和 (2.38) 代入 (2.37) 式, 经过计算化简, 可得:

$$\frac{\partial e}{\partial \lambda_k} = -\frac{\delta_k \cdot net_k}{\lambda_k} \quad (2.39)$$

根据 Delta 学习规则, 我们有陡度参数 λ_k 的修正量公式

$$\Delta \lambda_k = -\eta_\lambda \cdot \frac{\partial e}{\partial \lambda_k} = \frac{\eta_\lambda \cdot \delta_k \cdot net_k}{\lambda_k} \quad (2.40)$$

其中, η_λ 是 λ_k 的学习步长, 只要将 η_λ 取为 0, λ_k 的初值取作 1, 其增益量 $\Delta \lambda_k$ 就不再起作用。此时的 BMP 算法就退化为普通的 BP 算法。

有关实验表明, 与 BP 算法相比, 在选用相同的学习训练步数条件下, BMP 算法的精度远远高于 BP 算法的精度 (至少高出三个数量级)。若在相同精度要求的条件下, BMP 算法可在很少步数内达到精度要求, 收敛特性、收敛速度及系统的泛化特性都得到了改善。

§ 2.4.2 前向网络的自构形学习算法

在前向网络的拓扑结构中, 输入节点数与输出节点数由问题本身决定, 而隐含节点数的选取则相对困难得多。如果隐含节点少了, 则学习过程不收敛; 隐含节点多了, 则存在节点冗余, 网络性能下降。为了找到合适的隐含节点数, 最好的办法是在网络的学习过程中, 根据环境要求, 自组织和自学习自己的结构, 这种网络学习方法称为自构形学习算法。它的学习过程分为预估和自构形两个阶段。预估就是根据问题的大小及复杂程度, 设定一个隐含节点数很大的前向网络结构。在自构形阶段, 网络根据学习情况合并无用的冗余节点, 最后得到一个合适的自适应网络。

设 o_{ip} 是隐含节点 i 在学习第 p 个样本时的输出, \bar{o}_i 和 \bar{o}_j 是隐含节点 i 和 j 在学习完 n 个样本后的平均输出, n 为训练样本总数, 则

$$\bar{o}_i = \frac{1}{n} \sum_{p=1}^n o_{ip} \quad (2.41)$$

$$\bar{o}_j = \frac{1}{n} \sum_{p=1}^n o_{jp} \quad (2.42)$$

为了衡量隐含节点的工作情况，如下定义同层隐含节点的相关系数及样本分散度：
同层隐含节点 i 和 j 的相关系数为

$$r_{ij} = \left(\frac{1}{n} \sum_{p=1}^n o_{ip} o_{jp} - \bar{o}_i \bar{o}_j \right) / \sqrt{\left(\frac{1}{n} \sum_{p=1}^n o_{ip}^2 - \bar{o}_i^2 \right) \left(\frac{1}{n} \sum_{p=1}^n o_{jp}^2 - \bar{o}_j^2 \right)} \quad (2.43)$$

其中， r_{ij} 表明隐含节点 i 和 j 输出的相关程度， r_{ij} 过大说明节点功能重复，需要压缩合并。
样本分散度为

$$S_i = \left(\frac{1}{n} \sum_{p=1}^n o_{ip}^2 - \bar{o}_i^2 \right) \quad (2.44)$$

如果 S_i 过小表明隐含节点的输出值变化很小，它对网络的训练没有起什么作用，性能与神经元的阈值差不多，可以将其删减。

由以上的定义，可以给出下面关于隐含节点的动态合并和删减规则。

(1) 合并规则 若 $|r_{ij}| \geq c_1$ ，且 $S_i, S_j \geq c_2$ ，则同层隐含节点 i 和 j 可以合二为一。这 c_1 和 c_2 为规定的下限值。一般 c_1 取 0.8 ~ 0.9， c_2 取 0.001 ~ 0.010，令 $o_j \approx a o_i + b$ ，则

$$a = \left(\frac{1}{n} \sum_{p=1}^n o_{ip} o_{jp} - \bar{o}_i \bar{o}_j \right) / \left(\frac{1}{n} \sum_{p=1}^n o_{ip}^2 - \bar{o}_i^2 \right) \quad (2.45)$$

$$b = \bar{o}_j - a \bar{o}_i$$

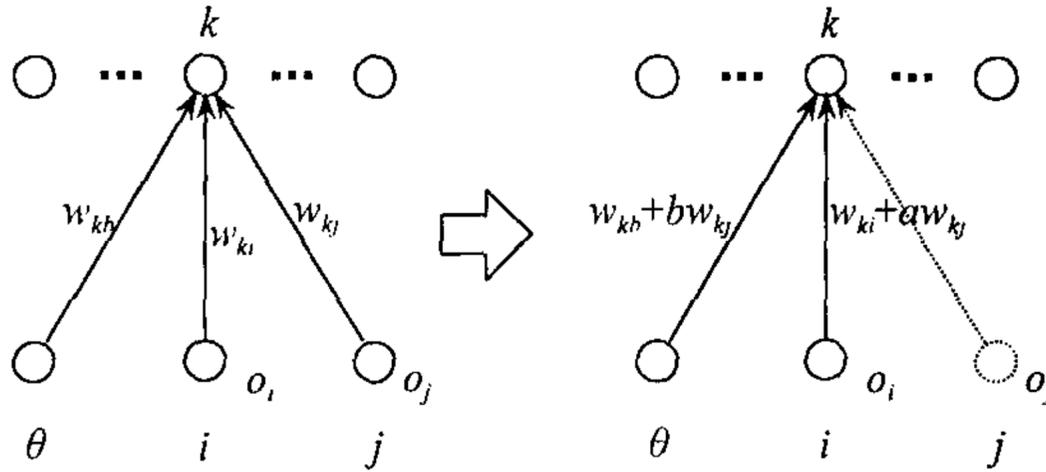


图 2.10 隐含节点 i 和隐含节点 j 的合并

如图 2.10 所示，输出节点 k 的输入为

$$\begin{aligned} net_k &= w_{ki} o_i + w_{kj} o_j + w_{kb} \cdot 1 + \sum_{l \neq i, j} w_{kl} o_l \\ &= (w_{ki} + a w_{kj}) o_i + (w_{kb} + b w_{kj}) \cdot 1 + \sum_{l \neq i, j} w_{kl} o_l \end{aligned} \quad (2.46)$$

从而得合并算法：

$$\begin{aligned} w_{ki} &\rightarrow w_{ki} + a w_{kj} \\ w_{kb} &\rightarrow w_{kb} + b w_{kj} \end{aligned} \quad (2.47)$$

(2) 删减规则 若 $S_i < c_2$ ，则隐含节点 i 可以删除。

令 $o_i \approx \bar{o}_i$ ，则输出节点 k 的输入为

$$\begin{aligned} net_k &= w_{ki} o_i + w_{kb} \cdot 1 + \sum_{l \neq i} w_{kl} o_l \\ &= (w_{kb} + w_{ki} \bar{o}_i) \cdot 1 + \sum_{l \neq i} w_{kl} o_l \end{aligned} \quad (2.48)$$

所以删减算法为

$$w_{kb} \rightarrow w_{kb} + w_{ki} \bar{o}_i \quad (2.49)$$

如图 2.11 所示，实际上是将输出节点和阈值节点合并了。

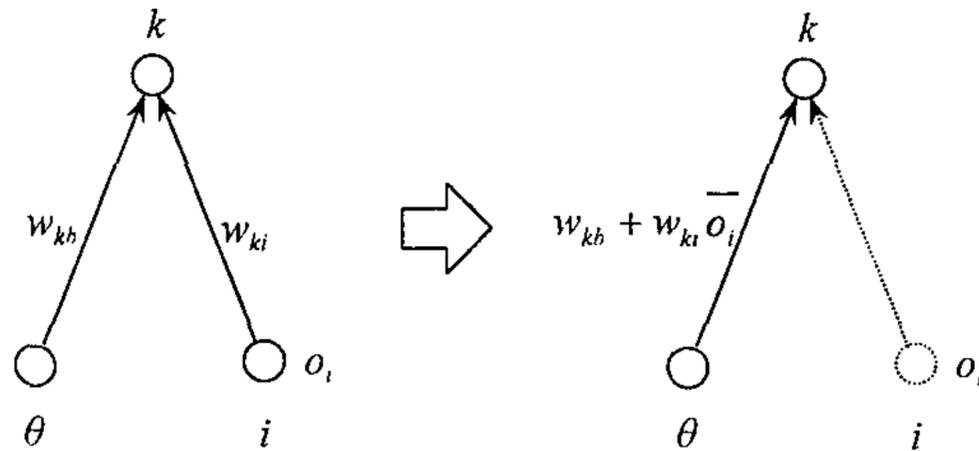


图 2.11 隐含节点 i 的删减

根据上述定义和规则可以得到前向网络的自构形学习算法。有关实验表明，对于三层前向网络，如果用自构形算法，与 BP 算法相比，不仅分类精度高，而且其收敛时间缩短约为 $1/7 \sim 1/5$ 。

§ 2.5 用 BP 网络实现人脸定位

人脸识别是计算机视觉的一大分支，它在安全系统，信用卡认证，罪犯辨认等方面都有许多应用，具有重要的研究价值。我们可以用三层 BP 神经网络实现对人脸位置的检测。大致思路如下：

对于一幅含有人脸的图像，用 20×20 的窗口在图中逐点移动，截取像素，并用这些点集作为神经网络的训练样本，以判断该窗口中是否存在人脸。考虑到获得的各人脸图像尺寸不同，对图像用不同的因子进行抽取（或者说缩小）后，得到一系列不同比例的图像，然后仍用固定的 20×20 的窗口截获像素，通过神经网络训练，得到在已知的不同比例下，该图各个可能的人脸中心。

该 BP 判决网络的结构如图 2.12 所示，仅含有一个输出节点，表示该区域是否是人脸。隐含层有三种。换句话说，用三个不同的 BP 网络并列在一起，在相同输入情况下，共同检测该区域是否存在人脸。之所以有三种隐含层，是因为 20×20 的点阵，可以划分为 4 个 10×10 的子块，或 16 个 5×5 的子块，或者 6 个有交叠的 5×20 的条形子块。它的输入层与隐含层之间不是完全连接，而是一个输入子块与一个隐含层节点完全连接，因此上述不同的子块划分方式分别对应 4, 16 和 6 个隐含节点。这种子块划分的依据是，正方形子块划分的神经网络对人脸的单个眼睛，鼻子和嘴角敏感，而长条形子块划分的神经网络对人脸的嘴和一对眼睛敏感。

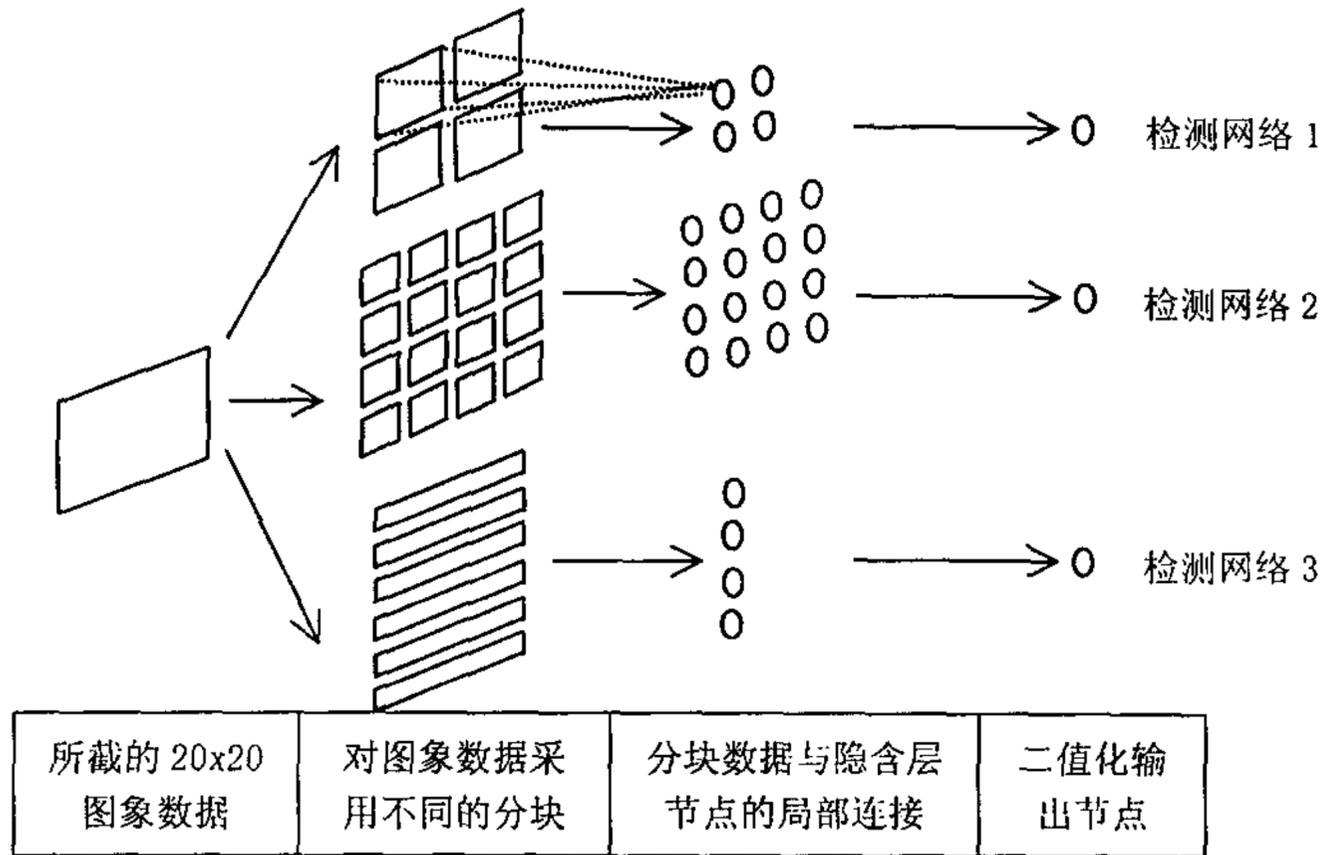


图 2.12 BP 神经网络结构

如果人脸落在 20×20 的窗口中，则该窗口应被神经网络判为存在人脸；由于图像的强相关性，窗口略作偏移，也会被神经网络判为存在人脸，因此该窗口中心的几个邻域点也会被神经网络所肯定。不仅如此，假设此时图像的比例因子是 $1/1.2$ ，则其相邻的比例因子 1 和 $1/(1.2 \times 1.2)$ 对应图像中的对应点也会可能被神经网络所肯定，如图 2.13 所示，对相邻比例的相关区域进行训练样本提取。因此当一个不含人脸的窗口被神经网络错判为存在人脸时，可以通过对其相关窗口作进一步检验，一般它的周围点以及相关抽样比例的点不会同时被 BP 分类器确认为人脸的中心点，从而将此类误判滤除。而且正如前面所述，通过对图像作不同比例的抽样，可以解决原始人脸图像的不同尺寸问题。此外，结合另外两个神经网络的输出，可以显著提高系统的整体判别精度。

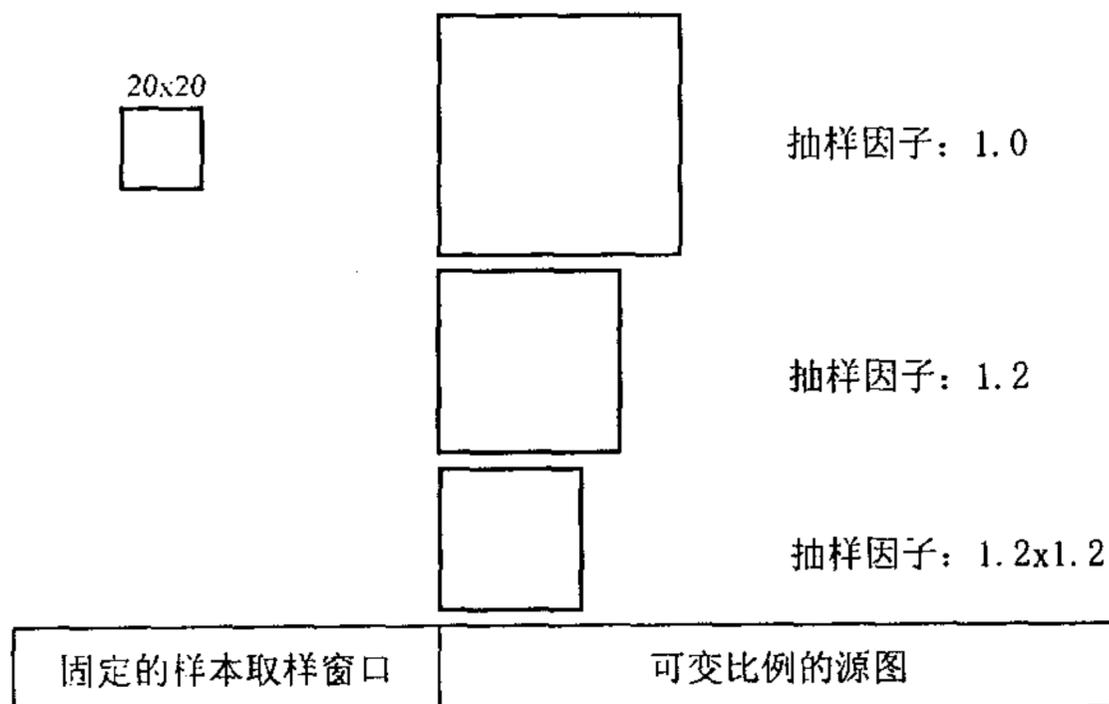


图 2.13 可变比例的样本取样法则

人脸中心点的最终确认。再次用神经网络作为判决器，如图 2.14 所示。图中假设，第一种检测神经网络，在比例因子 1, $1/1.2$, $1/(1.2 \times 1.2)$ 下，某一点的 8 邻域中被确认为人脸中心点依次有 1、2 和 2 个像素点；同理，另两个神经网络的输出依次为 (2, 0, 1), (5, 6, 5)，则将这三组特征值同时送入判别神经网络，判别该点是人脸中心点的最终可能性。

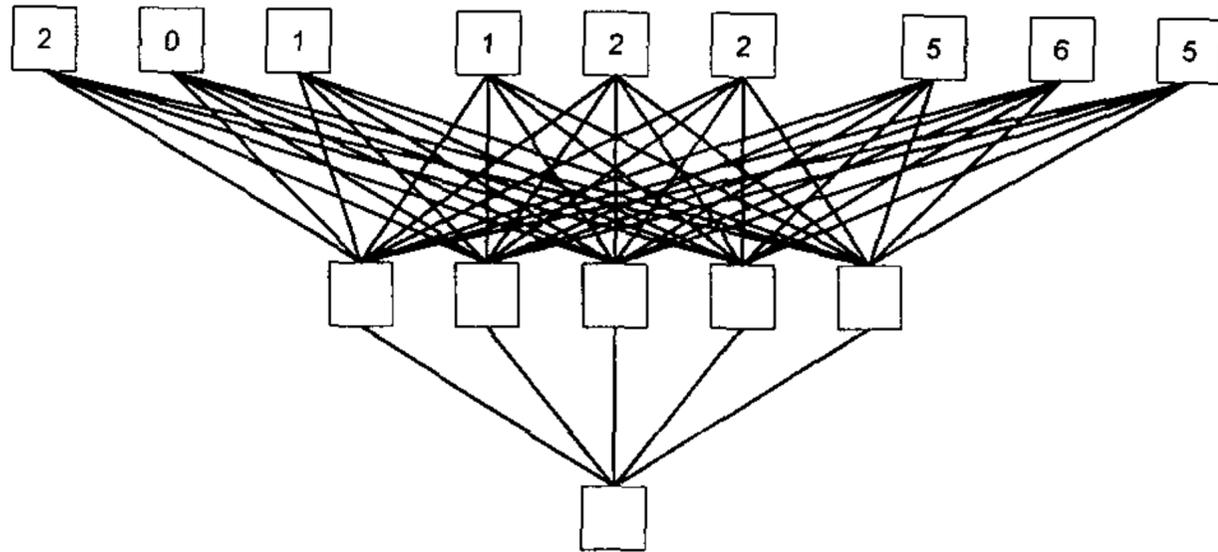


图 2.14 对邻域点进行统计分析的判决网络

为了判断图中的一个像素是否为人脸的中心，采用 BP 神经网络进行双重地判别：第一次用神经网络对图中的各个像素，从各个角度（采用了三种不同敏感结构的 BP 网络）作大致地判断；第二次，根据前面提取出来的像素点，根据图像的强相关性做进一步地判别，以提高系统判别的准确性。而且，这种方法同时也要求，在神经网络训练前，先对训练样本，即含有人脸的图像进行预处理，包括去噪声，图像灰度的对比度均衡等等，并尽量选择典型性训练样本训练 BP 权系数矩阵。实验表明，它能在较复杂的情况下，对一张多人照片，同时、准确地定出各个人的脸部位置。

§ 2.6 利用 BP 网络进行遥感模式识别的实验

§ 2.6.1 “同谱异构”问题的提出

遥感是二十世纪六十年代发展起来的新兴的综合技术学科，它与空间，电子，光学，计算机，地学等科学技术密切相关，是现代科学技术的一个重要组成部分。随着各种资源环境观测卫星不断发射与运行，遥感成为研究地球资源，地球环境的最有力的技术手段之一。但是由于多种原因，目前应用的遥感数据都存在不同程度的波谱混迭。这就是通常所谓的“同谱异构”问题，它是遥感分类中最困难的部分。

“同谱异构”问题是指不同的地物有相同的波谱特征。与之对应的是“同构异谱”问题，即相同的地面覆盖物具有不同的波谱特征。例如，同样的水，由于其深浅或污染的程度不同等原因，呈现出不同的波谱特征；又如同样的植被，由于其所处山的阴阳面不同，也会呈现不同的波谱特征。这类问题相对其前一个问题而言较容易解决。我们用神经网络训练时，可以将“同构异谱”物也视为不同的“类”进行训练/测试，最后再根据已知的映

射表将分类结果中这些不同的“类别”重新又归并为一个“类”。但是对于“同谱异构”问题就没有这么简单。解决它最好的办法是借助地理信息来确定同谱像元的不同归属，但实际应用中往往缺少这方面的知识。

在此，我们认为“同谱异构”本质上是一种波谱混迭问题。我们尝试用一种层次神经网络来分离“同谱异构”的像元。这就要求在一次神经网络分类无法达到要求的情况下，再次用神经网络对发生混迭的像元集合进行细分。这种细分的过程可以不止一次。该层次神经网络中的第一级神经网络主要用于波谱非混迭像元的分类，而第二级及后续层次的神经网络则用于波谱混迭像元的分离。这种解决方法利用了 BP 神经网络对训练样本具有很强的适应性特征，在大范围、多类别训练数据的情况下，它的分辨率低；而在小范围、少类别样本数据的训练下，它具有较高的分辨率。

§ 2.6.2 层次神经网络模型

该模型由目标不同的多层神经网络组成，如图 2.15 所示。图中只画出两级。模型中的第一级神经网络主要用于分出波谱非混迭的像元，而第二级和后续层次的神经网络则用于波谱混迭像元的分离，各层均选用 BP 模型。第一级神经网络仅有一个 BP 网络；而第二级神经网络由许多 BP 网络组成，每一个二级 BP 网络当且仅当第一级神经网络的一个输出节点需要进一步细分时才存在并且级联到该节点上。级联意味着训练样本在第一级神经网络的某一输出神经元获得最大可信度后，被再次用于训练对应的第二级神经网络。

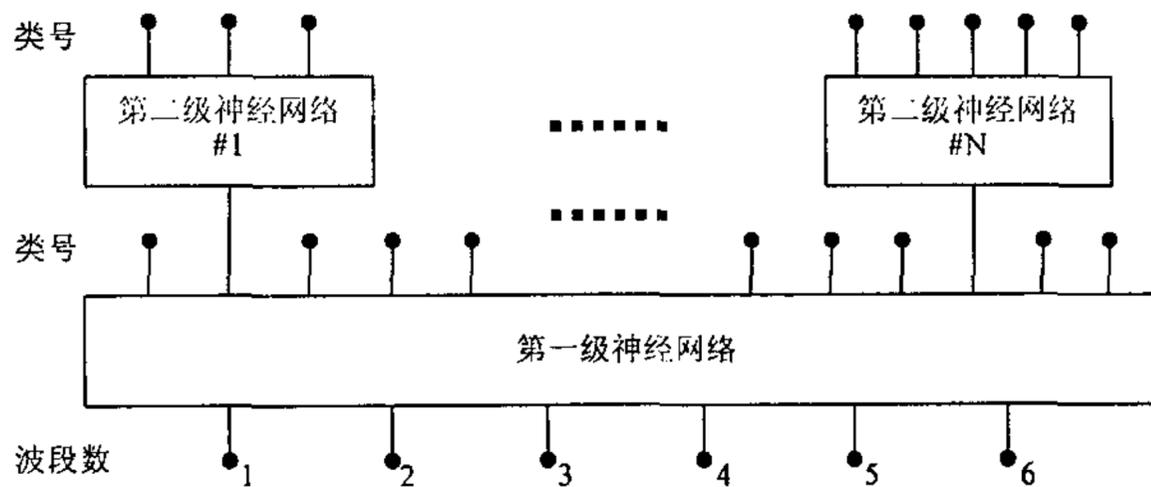


图 2.15 层次神经网络模型

§ 2.6.3 实验结果

我们所用的实际数据集是葡萄牙里斯本地区 TM 图像的地面实况数据，总共有 34985 个像元，16 个类，6 个波段。从每一类中随机抽取 2/3 的像元作为训练数据（文件 Train.dat）来训练神经网络，剩下的像元被用作测试数据（文件 Test.dat）来测试我们模型的性能。

针对我们的实验数据，第一级神经网络采用了一个三层 BP 神经网络，输入神经元数目为 6（即波段数），输出神经元数目为 16（即分类数目），隐含层神经元数目在我们的实验中取 64。初始时，步长因子均取 0.1，平滑因子取 0.7。

使用 Train.dat 训练第一级神经网络并且训练次数为 3000，该训练数据的分类结果如表 2.3 所示。

表 2.3 第一级神经网络关于训练数据的分类结果

类号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	精度(%)
1	808	0	0	0	0	6	0	0	0	0	0	0	0	0	14	0	97.58
2	0	1143	0	0	0	0	0	10	1	0	2	0	0	0	0	0	98.87
3	0	0	848	1	0	0	0	0	0	0	0	0	0	1	0	0	99.76
4	0	0	1	604	5	27	1	0	0	0	0	0	6	0	6	0	92.92
5	0	0	0	2	698	0	20	1	0	0	0	0	0	2	3	0	96.14
6	8	0	0	8	0	3645	17	0	0	0	0	3	0	0	1	0	98.99
7	0	0	0	7	41	3	1112	7	3	0	0	0	0	3	0	0	94.55
8	0	0	0	0	2	0	0	3690	0	0	0	0	0	0	0	0	99.94
9	0	0	0	0	0	0	0	0	1025	0	0	0	0	0	0	0	100.0
10	0	1	0	0	0	0	0	0	0	3938	5	0	0	0	0	0	99.84
11	0	11	0	0	0	0	0	0	0	13	3874	0	0	0	0	0	99.38
12	0	0	5	1	2	59	0	0	0	0	0	125	0	0	2	0	64.43
13	0	0	0	31	0	20	0	0	0	0	0	0	341	0	0	0	86.98
14	0	0	3	5	1	4	0	0	0	0	0	0	2	509	0	0	97.13
15	9	0	0	0	1	0	0	0	0	0	0	0	0	0	494	0	98.01
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	100.0
平均精度: 95.28%																	

由表 2.3 我们可以看出大多数类别的分类精度高于 90%。只有 12、13 类的精度分别为 64.43% 和 86.98%。其中，对于类 12，我们可由图 2.16 和图 2.17（分别对应于 5 波段和 6 波段的直方图，并且为了便于分析说明，两幅图中类 12 的幅度均单独放大了 8 倍）分析可知，它与类 6 在波谱域中发生混迭。

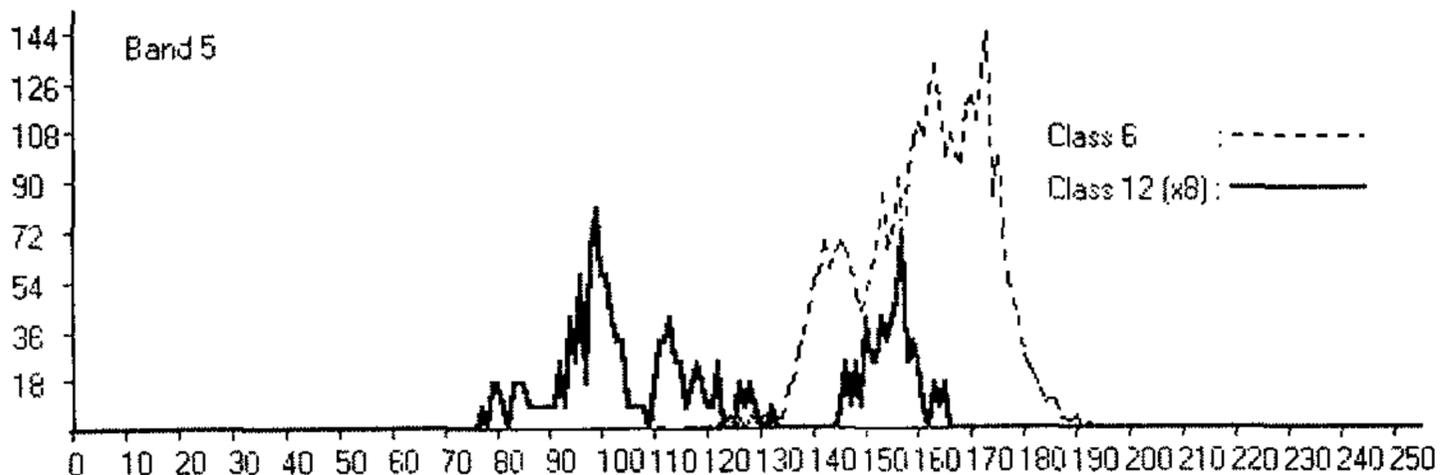


图 2.16 类 12 和类 6 在波段 5 上的直方图

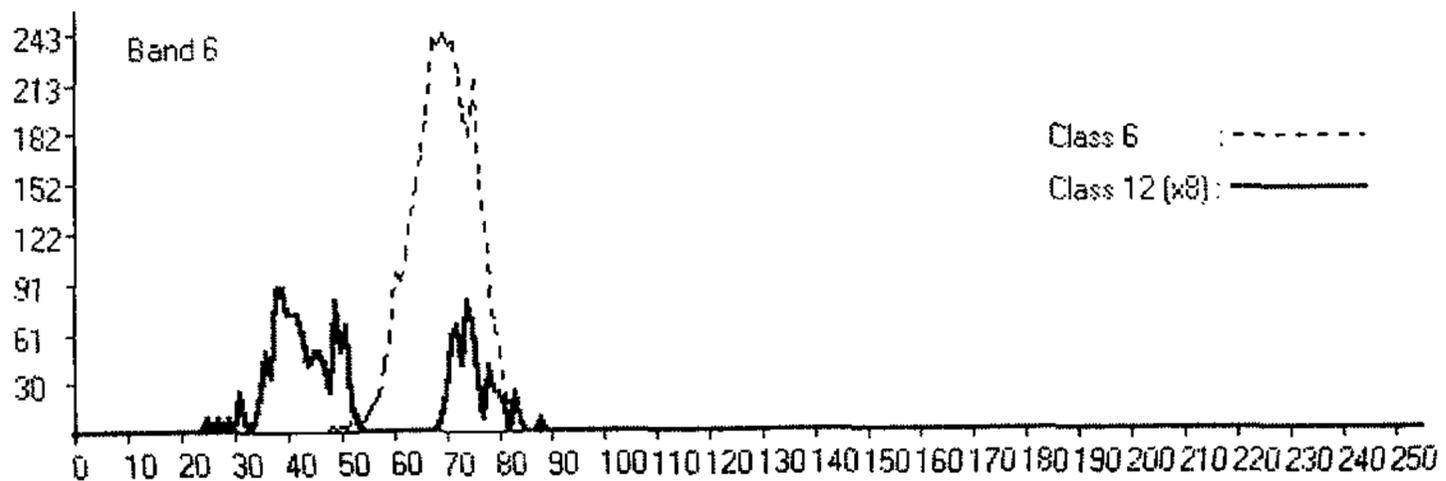


图 2.17 类 12 和类 6 在波段 6 上的直方图

因此,保留从第一级神经网络中得到的第 12 类正确的像元 (125 个),增加一个级联到第一级第 6 节点的二级神经网络,对上述的第 6 类结果进行细分。同样分析方法,考虑类 13 和类 4 也与类 6 发生混迭,该二级神经网络应拥有五个输出神经元,它们依次对应于类 12、13、6、4 和其他(代表因混迭较少而不予考虑的其他类别)。实验中将此二级网络的隐含层神经元数取 32。初始时,步长因子均取 0.1,平滑因子取 0.7。使用 Train6.dat 的训练数据,训练次数为 20000,二级神经网络训练数据的分类结果如表 2.4 所示。

表 2.4 二级神经网络关于训练数据的分类结果

类号	12	13	6	4	其它
12	56	0	3	0	0
13	0	18	2	0	0
6	1	1	3641	2	0
4	0	0	8	19	0
其它	0	0	3	1	9

由表 2.4 可知,类 12 中错分的 59 个像元有 56 个已从类 6 中正确地分离出来。因此,类 12 的精度由表 2.3 中的 64.43% 提高到 93.29%。类 13 中错分的 20 个像元有 18 个从类 6 中正确分出。因此,类 13 的精度由表 2.3 的 86.98% 提高到 91.58%。类 4 中错分的 27 个像元有 19 个从类 6 中正确分出。因此,类 4 的精度由表 2.3 的 92.92% 提高到 95.84%。第一级神经网络中正确分到类 6 的 3645 个像元减少到第二级的 3641 个,其精度也由 98.99% 降到 98.88%。可见类 6 的精度变化不大。最终的分类精度如表 2.5 所示。

由表 2.5,我们可以看出训练数据的平均分类精度是 97.55%,而且每一类的精度都超过 90%。在所有完整地采用这组训练数据进行分类的发表文章中,该分类精度是最高的。

表 2.5 最终分类精度

类号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	精度(%)
1	808	0	0	0	0	6	0	0	0	0	0	0	0	0	14	0	97.58
2	0	1143	0	0	0	0	0	10	1	0	2	0	0	0	0	0	98.87
3	0	0	848	1	0	0	0	0	0	0	0	0	0	1	0	0	99.76
4	0	0	1	623	5	8	1	0	0	0	0	0	6	0	6	0	95.84
5	0	0	0	2	698	0	20	1	0	0	0	0	0	2	3	0	96.14
6	8	0	0	10	0	3641	17	0	0	0	0	4	1	0	1	0	98.88
7	0	0	0	7	41	3	1112	7	3	0	0	0	0	3	0	0	94.55
8	0	0	0	0	2	0	0	3690	0	0	0	0	0	0	0	0	99.94
9	0	0	0	0	0	0	0	0	1025	0	0	0	0	0	0	0	100.0
10	0	1	0	0	0	0	0	0	0	3938	5	0	0	0	0	0	99.84
11	0	11	0	0	0	0	0	0	0	13	3874	0	0	0	0	0	99.38
12	0	0	5	1	2	3	0	0	0	0	0	181	0	0	2	0	93.29
13	0	0	0	31	0	2	0	0	0	0	0	0	359	0	0	0	91.58
14	0	0	3	5	1	4	0	0	0	0	0	0	2	509	0	0	97.13
15	9	0	0	0	1	0	0	0	0	0	0	0	0	0	494	0	98.01
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	100.0
平均精度: 97.55%																	

同样，我们使用测试数据 test.dat 测试了该层次神经网络模型。其中第一级神经网络的分类结果如表 2.6 所示，而第二级神经网络的分类结果如表 2.7 所示，最终测试分类精度如表 2.8 所示。

表 2.6 第一级神经网络关于测试数据的分类结果

类号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	精度(%)
1	402	0	0	0	0	3	0	0	0	0	0	0	0	0	9	0	97.10
2	0	572	0	0	0	0	0	6	0	0	0	0	0	0	0	0	98.96
3	0	0	425	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
4	0	0	1	273	2	33	0	0	0	0	0	0	9	4	3	0	84.00
5	0	0	0	0	349	0	12	0	0	0	0	0	0	1	1	0	96.14
6	8	0	0	0	0	1815	13	0	0	0	0	1	0	0	3	0	98.64
7	0	0	0	4	22	3	550	3	3	0	0	0	0	2	0	0	93.69
8	0	0	0	0	0	0	0	1845	0	0	0	0	0	0	0	0	100.0
9	0	0	0	0	0	0	0	1	511	0	0	0	0	0	0	0	99.80
10	0	0	0	0	0	0	0	0	0	1971	0	0	0	0	0	0	100.0
11	0	4	0	0	0	0	0	0	0	12	1932	0	0	0	0	0	99.17
12	0	0	1	0	2	32	0	0	0	0	0	59	0	0	3	0	60.82
13	0	0	1	9	1	10	0	0	0	0	0	0	174	0	0	0	89.23
14	0	0	6	1	3	0	0	0	0	0	0	0	0	251	0	0	96.16
15	6	0	0	0	0	0	0	0	0	0	0	0	0	0	245	0	97.60
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	100.0
平均精度: 94.44%																	

表 2.7 第二级神经网络关于测试数据的分类结果

类号	12	13	6	4	其它
12	31	0	1	0	0
13	0	7	1	2	0
6	37	17	1755	6	0
4	0	1	9	22	1
其它	3	0	1	1	1

由表 2.8, 我们可以看出测试数据的平均分类精度为 96.88%, 每一类的精度都大于 90%。该分类精度在所有完整地采用这组测试数据进行分类的发表文章中也是最高的。

第二级神经网络的数目可以不止一个。例如, 由表 2.3 我们可以看出类 13 中有 31 个像元错分到类 4。因此我们需要设计一个第二级神经网络从类 4 中分离出混迭的像元: 其 3 个输出神经元分别对应于类 4、类 13 和其它, 并有 32 个隐含神经元和 6 个输入节点, 它级联于第一级神经网络的第 4 输出神经元。初始时, 步长因子均取 0.1, 平滑因子取 0.7, 训练次数为 40000, 其分类的结果如表 2.9 所示。

由表 2.9 我们可以看出类 13 中, 与类 4 混迭的 31 个像元有 15 个已正确地分离出来。因此, 类 13 的训练精度由表 3 中的 91.58% 提高到 95.40%, 而类 4 的训练精度却变化不大。使用测试数据的该二级神经网络分类结果如表 2.10 所示。

表 2.8 最终分类精度

类号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	精度(%)
1	402	0	0	0	0	3	0	0	0	0	0	0	0	0	9	0	97.10
2	0	572	0	0	0	0	0	6	0	0	0	0	0	0	0	0	98.96
3	0	0	425	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
4	0	0	1	295	2	10	0	0	0	0	0	0	10	4	3	0	90.76
5	0	0	0	0	349	0	12	0	0	0	0	0	0	1	1	0	96.14
6	8	0	0	6	0	1755	13	0	0	0	0	38	17	0	3	0	95.38
7	0	0	0	4	22	3	550	3	3	0	0	0	0	2	0	0	93.69
8	0	0	0	0	0	0	0	1845	0	0	0	0	0	0	0	0	100.0
9	0	0	0	0	0	0	0	1	511	0	0	0	0	0	0	0	99.80
10	0	0	0	0	0	0	0	0	0	1971	0	0	0	0	0	0	100.0
11	0	4	0	0	0	0	0	0	0	12	1932	0	0	0	0	0	99.17
12	0	0	1	0	2	1	0	0	0	0	0	90	0	0	3	0	92.78
13	0	0	1	11	1	1	0	0	0	0	0	0	181	0	0	0	92.82
14	0	0	6	1	3	0	0	0	0	0	0	0	0	251	0	0	96.16
15	6	0	0	0	0	0	0	0	0	0	0	0	0	0	245	0	97.60
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	100.0
平均精度: 96.88%																	

表 2.9 级联于第一级神经网络输出节点 4 的第二级神经网络关于训练数据的分类结果

类号	13	4	其它
13	15	16	0
4	1	603	0
其它	0	7	17

由表 2.10 我们可以看出类 13 中, 与类 4 混迭的 9 个像元有 7 个已正确分出。因此, 类 13 的测试精度由表 6 中的 92.82% 提高到 96.41%, 而类 4 的测试精度却没有很大变化。

表 2.10 级联于第一级神经网络输出节点 4 的关于测试数据的第二级神经网络分类结果

类号	13	4	其它
13	7	2	0
4	3	269	1
其它	0	1	4

第三章 Kohonen 自组织特征映射网络

在“学习机制”的研究中我们常忽略了人脑的一个常见但又特别有意义的属性，人脑处理单元的组织顺序。“组序”并不意味着将一个单元移到新的地方。这些单元的结构甚至是一致的，但通过它们之间内在参数的调整，使它们共同承担起处理某些特定过程的角色。于是乎这些特定单元仿佛组成了一个有特定功用的器官组织。

虽然人脑的这种组织结够是由基因决定的，但我们可以从信号统计的角度研究一个空间顺序如何在所接收信息的控制下通过一个简单的自组织过程实现优化。在自组织中形成的各种映射可以通过一维或者二维的表示方式描述输入信号的拓扑关系。

§ 3.1 人脑的特征映射

将数据之间所有的相互关系表示出来是信息科学中最中心的问题之一，而这种能力显然也是人脑所擅长的操作。在思考、潜意识信息处理中，人脑有压缩信息的能力，这是通过减少对大多数相关事物的描述，并且在不损失其关联信息的前提下实现的。智能信息处理的目的是通过不同层次的抽象，根据所接收数据的一个特定子集，创建对实际事物的一个简单映像。

人们很早就知道人脑的各个区域，特别是大脑皮层，是根据不同的感知模型组织的：其各个区域执行不同的任务，例如，语言控制和感觉信号的分析（视觉、听觉、触觉等）。在主要感觉区域之间（仅占整个大脑皮层面积的百分之十），是不太为人所知的关联区域，负责会聚不同形式的信号。近来越来越多的实验研究表明大脑区域中存在着精细结构，视觉、触觉等信号是感觉器官接收而后由大脑皮层通过同样的拓扑顺序获得的。

神经元的轴突与其细胞核的互联方式是复杂的，不能用简单的拓扑关系来描述。例如轴突与神经细胞的连接并不总是一一对应，因为在信号传递途中存在负责将信号混合的处理站（神经细胞核），如图 3.1 所示。当信号模式按更抽象的方式组织时，理解它们相互之间的映射关系就更加困难。

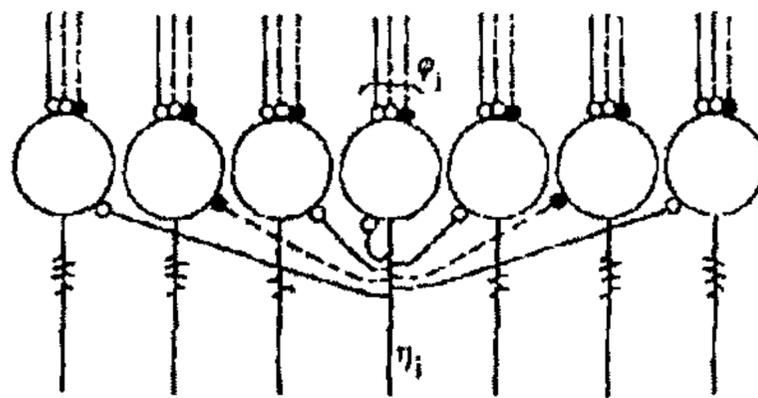


图 3.1 横向互联的神经元

毫无疑问大脑主要的映射网络结构是由先天决定的，但也存在着感知映射被后天经历所影响的事例。例如，在切除感觉器官、脑组织或幼年时被剥夺感觉后，一些感知映射就不再发展，大脑相应的区域就被剩下的映射所占据。这种现象可以用神经可塑性来解释。这些例子都说明了通过感知信息的调控可以实现简单自组织过程。

通常情况下关于某些特定事物类别的知识表述可以看作是一种特征映射的方式, 这种映射几何上是通过大脑对应区域的有序组织来完成。我们可以将自组织方法广泛地用于不同信息的表示。进一步, 我们可以用一个相同单元构成、单连接、单层的模型 (Kohonen 模型) 来描述具有复杂层次关系的数据, 这是通过将模型的不同子区域分配给信息的不同抽象层来实现。

我们将基本的自组织过程看成一种由简单的物理系统中产生的自适应现象。这里, 我们只是讨论由感觉信号触发的理想化神经结构, 它的目的是显示哪种结构类型和系统功能属性适用于自组织过程。近来相关的研究表明某些由互联的自适应单元组成的层状网络具有改变它们响应的能力, 接收响应信号的网络单元排列方式与输入信号中的某些特征有关。如前所述, 网络细胞或单元并不做任何移动, 而是对内部具体定义的参数集做相应的调整。由于这种网络通常是二维的阵列, 这种结果也意味着存在这种映射, 虽然其表现空间的维数减少了, 但其拓扑关系仍然得到保持。

这种显著的性质我们将通过后续的一些基本系统模型加以说明。最有意义的差别在于反馈类型, 这种反馈通常依靠距离, 根据一个典型的横向反馈函数控制 (此过程在神经网络中常见)。

§ 3.2 通过横向反馈形成局部响应

人脑中的大多数神经网络是由处理单元 (细胞或细胞膜) 组成的二维层面, 其中的单元是通过横向反馈紧密地联系在一起。根据一些估计, 大脑皮层中有 10000 个关于各主细胞之间的相互联结。图 3.1 描绘了一个神经元模型阵列, 每一个神经元接收输入信号 ϕ_i 和大量来自其他单元输出的横向联结信号。

此横向耦合可以视作下述的距离函数。在解剖学和心理学中都有证据表明哺乳动物的脑细胞中存在下列类型的横向交互作用: (i) 主细胞中短程兴奋可达 50 至 100 微米; (ii) 兴奋区域被一个半径达 200 至 500 微米的抑制作用半影所包围; (iii) 抑制半影周围还有更弱的兴奋作用, 其半径达几分米。横向交互作用的程度通常用“墨西哥帽”函数形式来描述 (图 3.2)。我们将忽略 (iii) 所描述区域的影响。

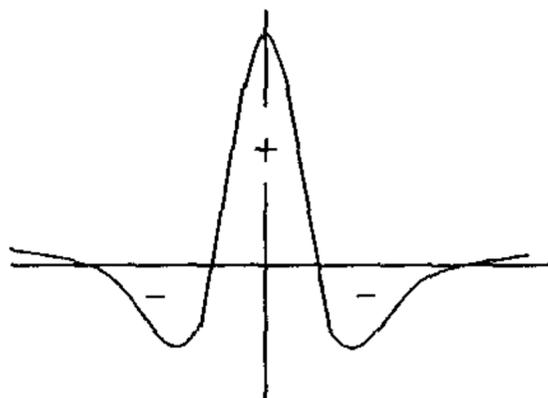


图 3.2 横向交互的“墨西哥帽”函数

模拟的横向交互作用函数如图 3.3 所示。网络中的信号传输假定由下列离散时间序列的表达式所描述:

$$\eta_i(t) = \sigma \left[\phi_i(t) + \sum_{k=-16}^{16} \gamma_k \eta_{i+k}(t-1) \right] \quad (3.1)$$

其中每一个反馈都有一个小的归一化延时, 系数 γ_k 由图 3.3 所定义。 $\sigma[\cdot]$ 为如下的非

线性关系:

$$\begin{cases} \sigma[z] = 0, z < 0 \\ \sigma[z] = z, 0 \leq z \leq 10 \\ \sigma[z] = 10, z > 10 \end{cases}$$

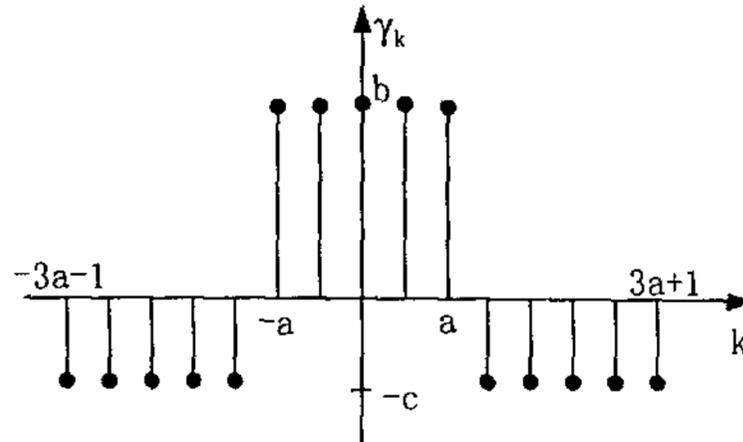


图 3.3 模拟的横向交互函数

假定输入兴奋信号 $\phi_i(t)$ 为某一平滑的函数, 它可以持续一个单位时间的间隔。 $\eta_i = \eta_i(t)$ 随时间的分布情况如图 3.4 所示。

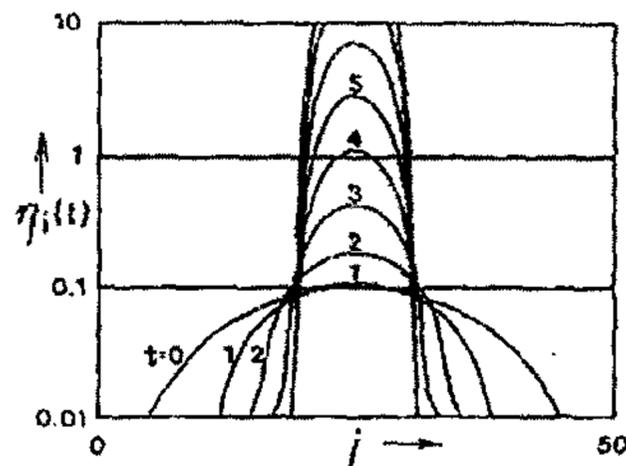


图 3.4 一维阵列的聚类变化

如果 $\phi_i(0) = A \sin^2(\pi i/50)$, 则其变化, 由于横向耦合, 将趋向于一个空间受约束的簇, 并且由于饱和性, 其变化幅度最终将稳定在一个常数上。该簇是以输入兴奋信号 $\phi_i(t)$ 的局部极大值为中心。

在这个简单模型中“簇”是恒稳定的。可以加入一些简单的特性使此模型具有“衰减”表现, 或者在持续变化中暂时减弱联结作用; 这种效果与实际的神经环路相似。在这种情况下, 簇将经历一段时间变化后衰退, 于是网络又可以自由地接受新的兴奋。

图 3.5 表示另一种二维类型的聚类过程, 是在 21×21 的方型处理单元阵列上产生的。横向交互函数是二维的, 参照图 3.3 所示。图中只显示出最终的变化状态。它在最大兴奋点周围产生一个“活动气泡”。该图也说明了另一个重要结论, 气泡的宽度取决于横向耦合的强度。在图 3.5 中正反馈和负反馈的绝对值相等。如果正反馈越强, “气泡”越宽, 反之, 增强负反馈将使“气泡”集中得更尖锐。(当然, 如果网络的负反馈太大, 聚类现象也不会产生。)

上面描述的效果在自组织过程中将起很大的作用。反馈程度可以通过调整相应的反馈参数来控制。聚类过程可能对参数微小的改变不敏感, 但它也不能在任意的参数下工作, 因此需要耐心调整以找到恰当的联结关系。例如, 当反馈增益在相对值也许是 1 到 10 之间

变动时可以实现聚类现象。另一方面, 为了实现最佳控制, 正负增益的比值也许要在 10% 到 20% 之间。

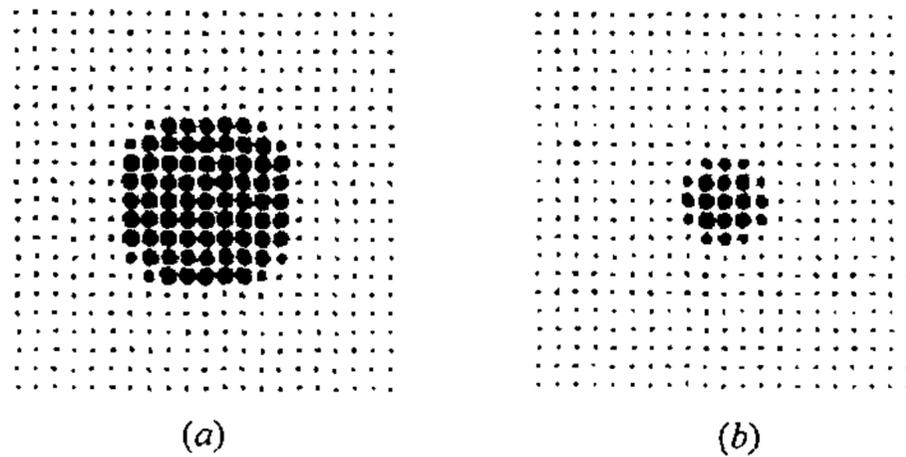


图 3.5 二维阵列的聚类变化。(a) 增强正反馈;
(b) 增强负反馈

如果兴奋函数不是平滑的, 例如, 输入增益处在随机变动中, 则“气泡”将不规则地改变。即使这种情况下, 也有可能 $\phi_i(t)$ 最大值处得到“气泡”。如果传递函数 $\sigma[\cdot]$ 不选择线性函数而是选择某一上凸函数, 则“气泡”会更好。

§ 3.3 自组织过程的简化计算

可以看出某些限定区域中“活动气泡”的形成是由于该区域的自组织行为; 这可以通过数学上的证明。由于“气泡”是在自组织的局部极大值处形成的, 可以按如下步骤定义算法: (i) 寻找极大值; (ii) 定义该极大值(对应于“气泡”)附近阵列的单元子集。可以根据函数的时间变量进一步定义“气泡”半径, 起始时较大, 随着时间变化逐渐减小。“气泡”的形态对最终的排序结果没有很大的影响。

§ 3.3.1 拓扑保持映射的定义

在自组织过程中, 我们期望的映射是将任意维的信号模式转换为一维或二维的阵列。因此, 阵列的输入不可能象图 3.1 所示的直接联结那样简单: 该图只是用来定义每一个单元的初始状态值。实际上, $\phi_i(t)$ 应该被表征基本信号的权值所代替。每一个单元于是就有自己的输入权值, 为了实现自组织变化, 这些权值应该是可调整的。

大多数情况下, 可调单元通过一个中继网络接收输入信号; 这对应于传感途中的各种变换。中继网中的信号允许混合; 重要的是由此阵列输出的信号仍保持相互关联。例如, 三个输出向量 x_1 , x_2 和 x_3 通过某个阵列排序, 由中继网输出的向量 y_1 , y_2 和 y_3 应该仍然同样相关的顺序。

考虑到一个物理系统, 它的特征是输入信号的响应集中在输出平面的某点附近, 对于不同的输入信号集这种输出定位均不同。这种“局部响应”可以代表一个“活动气泡”的中心。

一个最简单的产生局部响应的系统就是功能单元的线性阵列, 每一个单元并行地接收同一个输入信号集。假设单元 i 对不同的输入模式 $\eta_i(x_1)$, $\eta_i(x_2)$ 等产生不同的响应。我们定义具有自组织特征的系统变换, 假定输入模式集 $\{x_i; i=1, 2, \dots\}$ 可以在某个阵列中排列成 $x_1 R x_2 R x_3 \dots$, 其中 R 代表一种简单排序规则。

定义 3.1 系统将按如下规则产生一维拓扑保持映射, 其中 $i_1 > i_2 > i_3 > \dots$

$$\begin{aligned}\eta_{i_1}(x_1) &= \max_i \{\eta_i(x_1) : i = 1, 2, \dots, n\}, \\ \eta_{i_2}(x_2) &= \max_i \{\eta_i(x_2) : i = 1, 2, \dots, n\}, \\ \eta_{i_3}(x_3) &= \max_i \{\eta_i(x_3) : i = 1, 2, \dots, n\}, \\ &\dots\dots\end{aligned}$$

上述定义适用于二维或更高维的处理单元阵列中。这种情况下阵列的拓扑结构由每个单元邻域的定义所确定。如同输入信号, 也可以通过不同特征或属性采用不同的排序规则的方法定义模式 x_k 的拓扑顺序。

§ 3.3.2 简单的二维自组织系统

考虑图 3.6 所示的阵列, 其单元可以按任一平面方式 (如方格或六边形网格) 组合。每

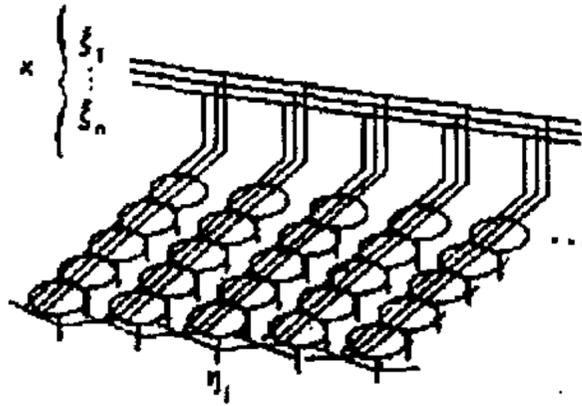


图 3.6 二维自组织系统

一个单元接收同样的信号 $\xi_1, \xi_2, \dots, \xi_n \in R$ 。单元 i 具有最简单的线性处理功能:

$$\eta_i = \sum_{j=1}^n \mu_{ij} \xi_j, \quad (3.2)$$

其中 $\mu_{ij} \in R$ 是可调参数。事实上, 每一个有序集 $(\mu_{i1}, \mu_{i2}, \dots, \mu_{in})$ 可以看作是对应于有序集 $(\xi_1, \xi_2, \dots, \xi_n)$ 的一种映像; 我们的目标是设计一个调整过程: 所有单元相应的参数都将会聚于某些值上, 从而使每一个单元按照某一规律匹配或敏感于某类输入信号。

除了考虑最大输出响应, 我们还可以根据向量 $x = [\xi_1, \xi_2, \dots, \xi_n]^T$ 和 $m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$ 之间的最佳匹配确定最大点的位置。许多自适应系数调整法则都要求将权值向量 m_i 归一化为定长。匹配标准采用向量之间的欧氏距离来衡量。如果定义最佳匹配点在索引值 c 处, 则 c 由以下条件确定:

$$\|x - m_c\| = \min_i \|x - m_i\|. \quad (3.3)$$

使用公式 (3.3) 用于过程定义的优点在于简化数学处理。

在自适应滤波器理论中关于关于滤波器权值向量调整的描述, 一般采用下列的方程:

$$\frac{dm}{dt} \stackrel{\text{def}}{=} \dot{m} = \phi(\cdot)x - \gamma(\cdot)m \quad (3.4)$$

其中 m 为滤波器权值向量, x 为滤波器输入向量, $\phi(\cdot)$ 和 $\gamma(\cdot)$ 为某些标量函数。又令 η 为滤波器的输出 (即 $\eta = m^T x$), 则关于权值向量的调整有如下五种法则:

- (1) $\dot{m} = \alpha x - \beta m$, 其中 α, β 常数, $\beta > 0$ 。

$$(2) \dot{\mathbf{m}} = \alpha \eta \mathbf{x} - \beta \mathbf{m}.$$

$$(3) \dot{\mathbf{m}} = \alpha \mathbf{x} - \beta \eta \mathbf{m} = (\alpha \mathbf{I} - \beta \mathbf{m} \mathbf{m}^T) \mathbf{x}, \text{ 其中 } \alpha, \beta > 0.$$

$$(4) \dot{\mathbf{m}} = \alpha \eta \mathbf{x} - \beta \eta \mathbf{m} = \alpha \mathbf{x} \mathbf{x}^T \mathbf{m} - \beta \mathbf{m} \mathbf{m}^T \mathbf{x}, \text{ 其中 } \alpha, \beta > 0.$$

$$(5) \dot{\mathbf{m}} = \alpha \eta \mathbf{x} - \beta \eta^2 \mathbf{m} = \alpha \mathbf{x} \mathbf{x}^T \mathbf{m} - \beta (\mathbf{m}^T \mathbf{x} \mathbf{x}^T \mathbf{m}) \mathbf{m}, \text{ 其中 } \alpha, \beta > 0.$$

对于阵列的各个处理单元，必须采用某种权向量调整法则。Kohonen 结合法则 (4) 和 (5) 提出如下的调整方程：

$$\frac{d\mu_y}{dt} = \alpha(t) \{ \eta_i(t) \cdot \xi_j(t) - \gamma[\eta_i(t)] \cdot \mu_{ij}(t) \} \quad (3.5)$$

其中 α 是随时间减小的函数，目的是为了实现在规定范围内收敛。此外，考虑为了能引起“聚类”或“气泡”，在引起最大响应的单元 c 附近，定义拓扑邻域 N_c ，它包括以单元 c 为中心一定半径内的所有单元。如果所有单元在 N_c 内，则 $\eta_i(x)=1$ (或其它某个常数值)；反之， $\eta_i(x)=0$ 。不失普遍性，我们进一步假定 $\gamma(0)=0$ ， $\gamma(1)=1$ ，则 (3.5) 式可以简化为如下形式：

$$\begin{cases} \frac{d\mu_{ij}}{dt} = \alpha(t) [\xi_j(t) - \mu_{ij}(t)] & i \in N_c \\ \frac{d\mu_{ij}}{dt} = 0 & i \notin N_c \end{cases} \quad (3.6)$$

最后完整的算法以离散时间的形式表示如下，其中 t_k 代表离散时间的索引值：

相似性匹配

$$\|\mathbf{x}(t_k) - \mathbf{m}_c(t_k)\| = \min_i \{\|\mathbf{x}(t_k) - \mathbf{m}_i(t_k)\|\} \quad (3.7)$$

修正

$$\mathbf{m}_c(t_{k+1}) = \mathbf{m}_c(t_k) + \alpha(t_k) [\mathbf{x}(t_k) - \mathbf{m}_c(t_k)], \quad i \in N_c,$$

$$\mathbf{m}_c(t_{k+1}) = \mathbf{m}_c(t_k), \quad \text{其它.}$$

“拓扑邻域” $N_c = N_c(t_k)$ ，可以有多种定义方式。只要邻域初始时取得足够宽而后随着时移而逐渐缩小，这些模拟方式都能在自组织过程中取得良好的结果 (如图 3.7)。注意如

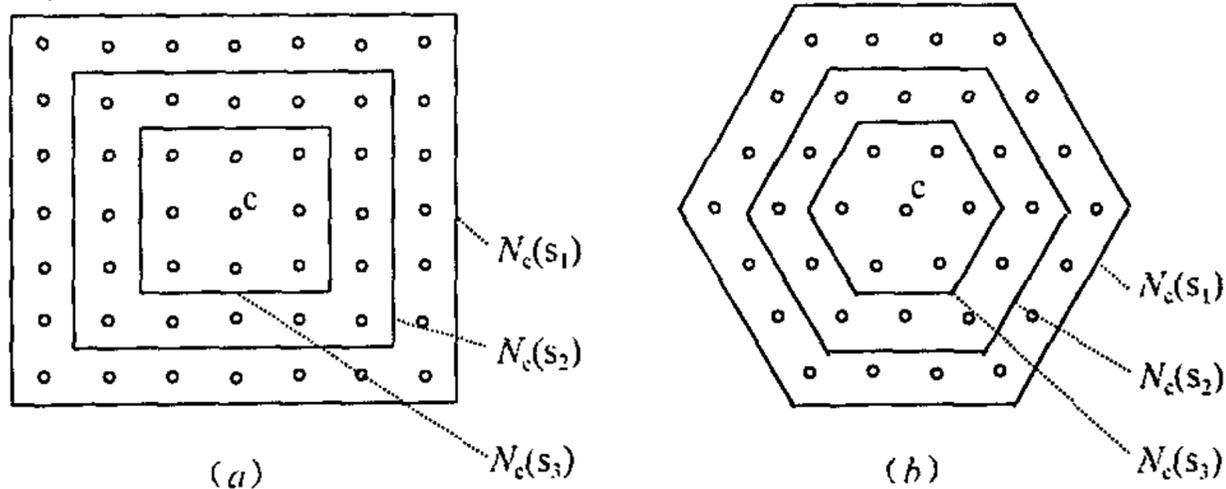


图 3.7 两个关于拓扑邻域的例子 ($s_1 < s_2 < s_3$)

果在阵列的边缘, 所取的邻域 N_c 是不完整的。

增益序列 $\{\alpha(t_k); t_k = 0, 1, \dots; 0 < \alpha(t_k) < 1\}$ 是一个缓慢递减的时间函数。可以假设它满足诸如下面的一些平稳约束条件:

$$\sum_{s=0}^{\infty} \alpha(s) = \infty, \quad \sum_{s=0}^{\infty} \alpha(s)^2 < \infty;$$

事实证明有关取得最优增益序列的问题非常复杂, 特别是排序的初始阶段。实际计算中, $\alpha(t_k)$ 可以取关于 t_k 的线性递减函数, 当 $\alpha(t_k) = 0$ 时过程自动停止。

可以通过经验正确选择 $\{\alpha(t_k); t_k = 0, 1, \dots\}$ 和 $\{N_c(t_k); t_k = 0, 1, \dots\}$ 。注意映射形成过程中有两个阶段与实际有所不同, 即正确排序的最初形成和最终的会聚是一个渐变的产生过程。为了产生好的结果, 后一阶段所需的次数可能是前一阶段的 10 到 100 倍, 其间采用较小的 $\alpha(t_k)$ 值。

排序结果的数学定义

如果输入向量 x 是一个统计概率密度函数为 $p(x)$ 的随机变量, 则处理单元的输入权值 μ_j 将形成关于 $p(x)$ 的有序映像。换句话说, 每一个单元只对一个特定的输入向量最敏感, 这种敏感性有序地出现在对不同的单元中。

命题 3.1 权重向量 m_i 根据其相互之间的相似性而趋于有序排列, 它的局部渐进点分布密度, 从平均的角度说, 是呈 $f(p(x))$ 形式, 其中 f 是一个连续单调递增函数。

§ 3.4 自组织的数学分析

首先我们考虑一维的单元阵列, 每一个单元与一个标量输入 ξ 相连。将每一个单元的标为 $1, 2, \dots, l$ 。每一个单元有一个单一的输入权值 μ_i , 将 ξ 和 μ_i 的相似性归结于它们的差 $|\xi - \mu_i|$; 最佳匹配如下所示:

$$|\xi - \mu_c| = \min_i |\xi - \mu_i|, \quad (3.8)$$

我们将单元集 N_c 的选取按如下定义调整:

$$N_c = \{\max(1, c-1), c, \min(l, c+1)\}, \quad (3.9)$$

换句话说, 每一个单元 i 的具有邻域 $i-1$ 和 $i+1$, 不包括阵列的边缘, 因此单元 1 的邻域为 2, 单元 l 的邻域为 $l-1$ 。于是 N_c 是一个简单的单元集, 包括单元 c 和它的自然邻域。

对于不同的 α ($0 < \alpha < 1$), 其自组织过程都是相似的, 主要是速度不同。在连续时间状态下, 方程为:

$$\begin{cases} \frac{d\mu_i}{dt} = \alpha(t)(\xi - \mu_i), i \in N_c \\ \frac{d\mu_i}{dt} = 0, \text{其它} \end{cases}, \quad (3.10)$$

命题 3.2 随机选取 μ_i 的初始值, 在自组织过程中按照 (3.8-10) 的公式逐渐给这些变量赋以新值, 这样随着 $t \rightarrow \infty$, 变量集 $(\mu_1, \mu_2, \dots, \mu_l)$ 将排成降序或升叙的序列。一旦此集合排序后, 对所有的 i 都将保持。此外, μ_i 的点密度函数将最终接近于函数 $p(\xi)$ 。

以上讨论涉及两部分: μ_i 有序序列的形成和它们最终会聚在某一个“固定点”。

§ 3.4.1 权值排序

如果定义表征 μ_i 乱序程度的量为 D , 则其排序程度可按如下直接得到:

$$D = \sum_{i=2}^l |\mu_i - \mu_{i-1}| - |\mu_l - \mu_1| \quad (3.11)$$

显然, $D \geq 0$, 当且仅当 $\mu_1, \mu_2, \dots, \mu_l$ 完全按升序或降序排列时, 等号成立。如果输入的 ξ 是随机变量则会导致自组织现象的发生, 于是 D 在修正过程中减少的出现机会将大于递增的机会。根据 (3.8-3.11) 计算出来的 dD/dt 可以直接地求得。例如, 如果 $3 < c < l-2$, 则只有 μ_{c-1}, μ_c 和 μ_{c+1} 可以在修正时改变, 影响 D 的只有五种可能。在这些可能中, 可以用关于 $\mu_i - \mu_{i-1}$ 的任一符号连接将 dD/dt 以简单封闭的形式表达出来。我们将看出 D 可以用于指示出整个排序过程。在特定情况下, D 将会聚到 0 并将始终停留在那。

首先我们需要一些用来描述过程瞬间状态的定义, 如 μ_i 部分序列的各种类型。每一个排序步骤仅依赖于 ξ 和有关节点邻域中涉及 μ_i 的值。这些定义的说明参照图 3.8。

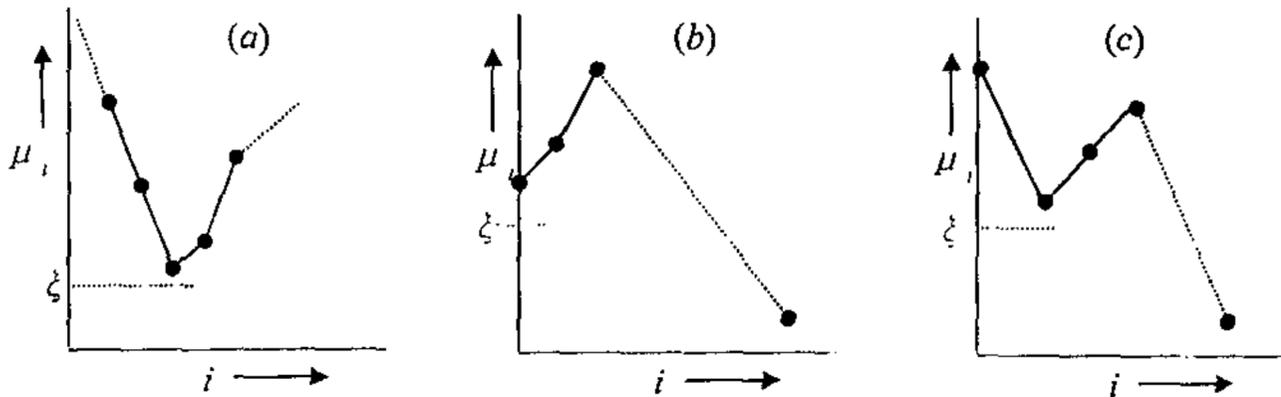


图 3.8 (a) ξ 位于长度为 5 的折叠序列之外; (b) ξ 位于长度为 3 的边缘折叠序列之外; (c) ξ 位于长度为 4 的双边缘折叠序列之外。

定义 3.2 节点 m 处 $2k+1$ 长的折叠序列为序列值 $(\mu_{m-k}, \mu_{m-k+1}, \dots, \mu_m, \mu_{m+1}, \dots, \mu_{m+k-1}, \mu_{m+k})$, 其中部分序列 $(\mu_{m-k}, \dots, \mu_m)$, $(\mu_{m+1}, \dots, \mu_{m+k})$ 中有一个是单调递增, 另一个是单调递减。

定义 3.3 如果值 ξ 位于 $[\mu_{m-k}, \mu_m]$ 和 $[\mu_{m+1}, \mu_{m+k}]$ 的交集中, 则它位于折叠序列内; 否则 ξ 位于折叠序列之外。

定义 3.4 $m=1$ 处长度为 k 的边缘折叠序列为单调递增或递减的序列值 $(\mu_1, \mu_2, \dots, \mu_k)$; 同理, $m=l$ 时为 $(\mu_l, \mu_{l-1}, \dots, \mu_{l-k+1})$ 。

定义 3.5 如果 m 处的边缘折叠序列, 其 $\xi - \mu_1$ 和 $\xi - \mu_l$ 的符号不同, 则 ξ 位于此边缘折叠序列之外。

定义 3.6 $m=2$ 处长度为 $k+1$ 的双边缘折叠序列为序列值 $(\mu_1, \mu_2, \dots, \mu_{k-1}, \mu_l)$, 其部分序列 $(\mu_2, \dots, \mu_{k-1})$ 为单调递增或递减, 且 $\mu_1 - \mu_2$ 和 $\mu_2 - \mu_3$ 的符号不同, 其 $\mu_1 - \mu_2$ 和 $\mu_l - \mu_{l-1}$ 的符号一致; 同理, $m=l-1$ 时, 有序列值 $(\mu_l, \mu_{l-1}, \dots, \mu_{l-k}, \mu_1)$, $\mu_l - \mu_{l-1}$ 和 $\mu_{l-1} - \mu_{l-2}$ 的符号不同, $\mu_l - \mu_{l-1}$ 和 $\mu_l - \mu_1$ 的符号一致。

定义 3.7 如果 $m=2$ 处的双边缘折叠序列, 其 $\xi - \mu_2$ 和 $\xi - \mu_l$ 的符号不同, 则 ξ 位于此双边缘折叠序列之外; 同理对待 $m=l-1$, $\xi - \mu_{l-1}$ 和 $\xi - \mu_1$ 的符号不同时。

定理 3.1 (排序定理) A) 假定 m 表示所选节点。如果 $3 \leq m \leq l-2$, 则 D 在修正中递减, 下述两种情况除外: (A1) 如果 $\mu_{m-2}, \mu_{m-1}, \mu_m, \mu_{m+1}, \mu_{m+2}$ 形成单调递增或递减序列, 则 D 值不变; (A2) 如果 ξ 位于 m 处长度 ≥ 5 的折叠序列之外, 则 D 值减小。

B) 如果 $m=1$ 或 $m=l$ 则 D 值在修正中递减, 下述两种情况除外: (B1) 如果 $\mu_1, \mu_2, \mu_3, \mu_l$ (或, 对应于 $\mu_l, \mu_{l-1}, \mu_{l-2}, \mu_1$) 形成单调递增或递减序列, 则 D 值不变; (B2) 如果 ξ 位于 m 处长度 ≥ 3 的边缘折叠序列之外, 则 D 值递增。

C) 如果 $m=2$ 或 $m=l-1$ 则 D 值在修正中递减, 下述两种情况除外: (C1) 如果 $\mu_1, \mu_2, \mu_3, \mu_4, \mu_l$ (或, 对应于 $\mu_l, \mu_{l-1}, \mu_{l-2}, \mu_{l-3}, \mu_1$) 形成单调递增或递减序列, 则 D 值不变; (C2) 如果 ξ 位于 m 处长度 ≥ 4 的双边缘折叠序列之外, 则 D 值递增。

定理 3.1 的中心思想是只考虑每次 D 值可能改变的情况。改变的方向和大小依靠这些情况下 $\mu_i - \mu_{i-1}$ 各个连续差分的符号组合。下面将分别考虑定理 3.1 的各种情况。

定理 3.1 的证明: A 部分 由于所选节点仅对其最近邻域产生影响, D 中剩余的其它因子保持不变, 除了如下的部分和:

$$S = \sum_{i=m-1}^{m+2} |\mu_i - \mu_{i-1}| \quad (3.12)$$

表达式 S 可以用关于 $\mu_i - \mu_{i-1}$ 符号的一个特定组合进一步简化。另一方面, 在所选节点 m 附近, 根据公式(3.10)有:

$$\begin{cases} \dot{\mu}_{m-2} \equiv 0 \\ \dot{\mu}_{m-1} = \alpha(\xi - \mu_{m-1}) \\ \dot{\mu}_m = \alpha(\xi - \mu_m) \\ \dot{\mu}_{m+1} = \alpha(\xi - \mu_{m+1}) \\ \dot{\mu}_{m+2} \equiv 0 \end{cases} \quad (3.13)$$

在 S 中相关的因子有 16 种可能的符号组合方式, 但其中有 8 种与另外的 8 种成对称关系, 因此我们可以假定 $\mu_{m-1} > \mu_{m+2}$ 。根据表 3.1, 依次将各种可能情况用 a0 到 a7 加以标记:

表 3.1

情况	$\mu_m - \mu_{m-1}$	$\mu_{m+1} - \mu_m$	$\mu_{m+2} - \mu_{m+1}$
a0	>0	>0	>0
a1	>0	>0	<0
a2	>0	<0	>0
a3	>0	<0	<0
a4	<0	>0	>0
a5	<0	>0	<0
a6	<0	<0	>0
a7	<0	<0	<0

如果用标号做 \dot{S} 的下标, 并考虑 μ_m 为最接近 ξ 的值, 则可以作如下的进一步简化:

$$\begin{aligned} \dot{S}_{a0} &\equiv 0, \\ \dot{S}_{a1} &= 2\alpha(\xi - \mu_{m+1}) < 0, \\ \dot{S}_{a2} &= 2\alpha(\mu_{m+1} - \mu_m) < 0, \\ \dot{S}_{a3} &= 2\alpha(\xi - \mu_m) \begin{cases} > 0, \text{当 } \xi > \mu_m \text{ 时 (即 } \xi \text{ 不在 } m \text{ 的折叠序列中)} \\ < 0, \text{当 } \xi < \mu_m \text{ 时} \end{cases} \\ \dot{S}_{a4} &= 2\alpha(\mu_m - \mu_{m-1}) < 0, \\ \dot{S}_{a5} &= 2\alpha(\xi - \mu_{m-1} + \mu_m - \mu_{m+1}) < 0, \\ \dot{S}_{a6} &= 2\alpha(\mu_{m+1} - \mu_{m-1}) < 0, \\ \dot{S}_{a7} &= 2\alpha(\xi - \mu_{m-1}) < 0. \end{aligned} \quad (3.14)$$

B 部分 节点 1 和 l 的证明相似，将仅对 1 证明。由于情况 $\mu_l < \mu_1$ 与另一部分对称，所以我们可以又仅限制于 $\mu_l > \mu_1$ 情况下讨论（注意 D 中的因子为 $|\mu_l - \mu_1|$ ）。现在只有表 3.2 所定义的四种情况需要考虑：

表 3.2

情况	$\mu_2 - \mu_1$	$\mu_3 - \mu_2$
b0	>0	>0
b1	>0	<0
b2	<0	>0
b3	<0	<0

$$\begin{aligned} \dot{D}_{b0} &\equiv 0, \\ \dot{D}_{b1} &= 2\alpha(\xi - \mu_2) < 0, \\ \dot{D}_{b2} &= 2\alpha(\mu_2 - \mu_1) < 0, \\ \dot{D}_{b3} &= 2\alpha(\xi - \mu_1) \begin{cases} > 0, \text{当 } \xi > \mu_1 \text{ 时 (即 } \xi \text{ 不在边缘折叠序列中),} \\ < 0, \text{当 } \xi < \mu_1 \text{ 时.} \end{cases} \end{aligned}$$

C 部分 仅考虑 $\mu_l > \mu_1$ 情况就足够了，将讨论表 3.3 中的各种符号组合：

表 3.3

情况	$\mu_2 - \mu_1$	$\mu_3 - \mu_2$	$\mu_4 - \mu_3$
c0	>0	>0	>0
c1	>0	>0	<0
c2	>0	<0	>0
c3	>0	<0	<0
c4	<0	>0	>0
c5	<0	>0	<0
c6	<0	<0	>0
c7	<0	<0	<0

由前述相似的方法得到如下：

$$\begin{aligned} \dot{D}_{c0} &\equiv 0, \\ \dot{D}_{c1} &= 2\alpha(\xi - \mu_3) < 0, \\ \dot{D}_{c2} &= 2\alpha(\mu_3 - \mu_2) < 0, \\ \dot{D}_{c3} &= 2\alpha(\xi - \mu_2) \begin{cases} > 0, \text{当 } \xi > \mu_2 \text{ 时 (即 } \xi \text{ 不在双折叠序列中)} \\ < 0, \text{当 } \xi < \mu_2 \text{ 时} \end{cases} \\ \dot{D}_{c4} &= 2\alpha(\mu_2 - \mu_1) < 0, \\ \dot{D}_{c5} &= 2\alpha(\xi - \mu_1 + \mu_2 - \mu_3) < 0, \\ \dot{D}_{c6} &= 2\alpha(\mu_3 - \mu_1) < 0, \\ \dot{D}_{c7} &= 2\alpha(\xi - \mu_1) < 0. \end{aligned} \tag{3.15}$$

上述中， D 值递减的情况远多于 D 值保持不变或递增的情况。事实上，这就是能看到自排序效果的原因。定理 3.1 并没有指出排序过程中不同情况出现的次数；它们由整个修正操作的进程所决定。仅在开始时，在随机的初始条件下，可以假定不同的情况按大致相等的概率出现，而且 D 递减的比率可以直接根据 α 估测出来。例如，如果阵列很大，可以忽略涉及其边缘的情况，并且如果初始时各种情况等概率出现，那么参考表 3.1， D 递减的可能为 13/16（注意 ξ 可以小于或大于选中节点的 μ_m 值）。然而，一旦启动排序，我们可

以看到 D 会越来越的可能保持不变, 并最终占据了大多数的排序过程。定理 3.1 有一个非常重要的结果, 作为推论 3.1 表述如下:

推论 3.1 如果所有的数值 $\mu_1, \mu_2, \dots, \mu_l$ 被排序后, 它们在将来的修正过程中不可能再变成乱序。

其证明是根据事实: 如果所有的部分序列被排序, 那么 D 值递增的情况就不可能出现。

更严格的排序证明可以描述如下。假定 $\xi = \xi(t) \in R$ 是一个随机的 (标量) 输入, 在有限时域内的密度函数为 $p(\xi)$, 且 $\xi(t_1)$ 与 $\xi(t_2)$, $t_1 \neq t_2$, 互不相关。

命题 3.3 当 $t \rightarrow \infty$ 时, μ_i 将以 1 的概率按递增或递减排序。

此证明是根据 Markov 过程的基本性质, 尤其是其吸收态的转移概率为 1: 即如果某一状态可以由某一输入序列以正值的转移概率到达, 则用任意的起始输入值 (即, 允许随机输入的序列), 当 $t \rightarrow \infty$ 时, 几乎可以肯定到达此吸收状态。 μ_i 的有序序列与吸收态的性质相同, 可以由此证明命题 3.3。

§ 3.4.2 会聚阶段

假定 $\mu_i, i=1, 2, \dots, l$ 已经完成排序, 并仍然保持进一步地修正过程。这时目标将是计算 μ_i 的收敛值。当 $t \rightarrow \infty$ 时, 选择适当的 $\alpha(t)$, 可以使 μ_i 的变化任意小。

图 3.9 表示排于实数轴上有序的 μ_i 值。并假定 $[\mu_i, \mu_i]$ 是 $[a, b]$ 上的子集, 按 $p(\xi)$ 概率密度分布。

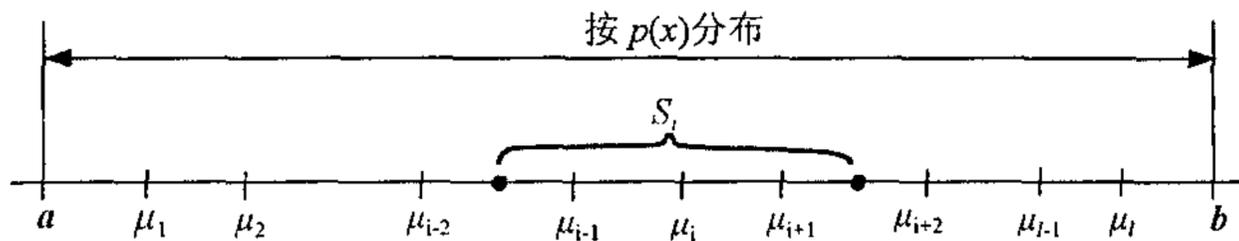


图 3.9 排序后的输入权值

由于已做 μ_i 完成排序的假定, 且所选的节点只能影响它的直接邻域, 由公式 (5.8-10) 很容易看出只有当 ξ 落在 S_i 所定义的间隙内时, 值 μ_i 才受影响: 假定 $l \geq 5$, 我们有

$$\begin{aligned}
 3 \leq i \leq l-2: & \quad S_i = [1/2(\mu_{i-2} + \mu_{i-1}), 1/2(\mu_{i-1} + \mu_{i+2})], \\
 i=1: & \quad S_i = [a, 1/2(\mu_2 + \mu_3)], \\
 i=2: & \quad S_i = [a, 1/2(\mu_3 + \mu_4)], \\
 i=l-1: & \quad S_i = [1/2(\mu_{l-3} + \mu_{l-2}), b], \\
 i=l: & \quad S_i = [1/2(\mu_{l-2} + \mu_{l-1}), b].
 \end{aligned} \tag{3.16}$$

根据 (5.10), μ_1, \dots, μ_l 的条件均值为:

$$\langle \dot{\mu}_i \rangle \stackrel{def}{=} E\{\dot{\mu}_i\} = \alpha(E\{\xi | \xi \in S_i\} - \mu_i). \tag{3.17}$$

其中 $E\{\xi | \xi \in S_i\}$ 是 S_i 的重心, S_i 是 (3.16) 定义的关于 μ_i 的函数。假定 μ_i 的概率密度 $p(\xi)$ 恒为定值, 则上式又可以进一步简化为:

$$\begin{aligned}
 3 \leq i \leq l-2: & \quad \langle \dot{\mu}_i \rangle = a/4(\mu_{i-2} + \mu_{i-1} + \mu_{i+1} + \mu_{i+2} - 4\mu_i), \\
 \langle \dot{\mu}_1 \rangle & = a/4(2a + \mu_2 + \mu_3 - 4\mu_1), \\
 \langle \dot{\mu}_2 \rangle & = a/4(2a + \mu_3 + \mu_4 - 4\mu_2),
 \end{aligned}$$

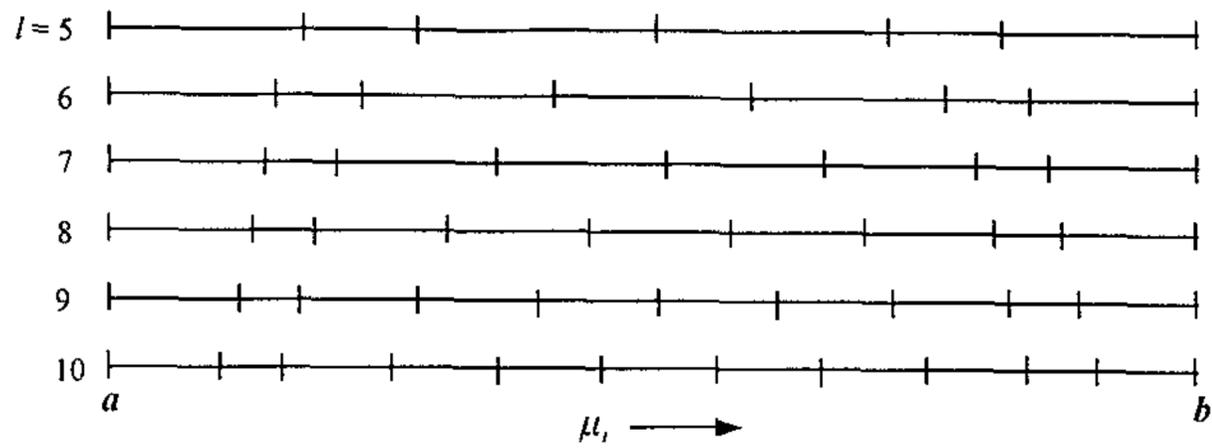


图 3.10 不同阵列长度下输入权重的收敛值

可以看出

- 处在外侧的 μ_1 和 μ_l 各向内侧移动了大约 $1/l$ ；这个结果将随着 l 的增大而消失。
- 从 μ_3 到 μ_{l-2} 看起来将趋于均匀分布。

因此可有结论，当 $p(\xi)$ 恒为定值时， μ_i 将趋于沿着实数轴作均匀分布。

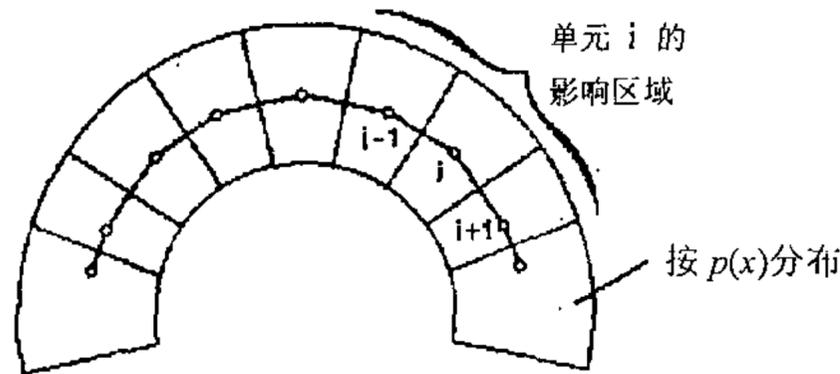


图 3.11 自组织的稳定状态

可以将一维的结论做进一步推广。例如，如果阵列是线性，但是却呈二维分布时， x 一定落在图 3.11 所划分的某一区域内。当 x 位于单元 $i-1$ 、 i 或 $i+1$ 中时，单元 i 将受影响。在稳定状态下，每一个权值向量的收敛值将对应于相关区域的重心。

事实上，收敛的权值向量 m_i 是模式空间的一种量化向量。将一个空间 R^n 等分成若干个区域后， m_i 实际上是每一个区域内关于所有点的最邻近点。两个区域以超平面为边缘，且此超平面正交于 m_i 和 m_j 的连线。

关于 m_i 在自组织过程中的分布问题。我们可以看到 m_i 只有被落在邻域集 N_c 内的点所修正。这个集合是时间的函数；起始时较宽，而最后仅含点 c 自身和其最临近点。 m_i 的渐进分布可以根据 N_c 计算。

考虑一维时的系统方程：

$$\begin{aligned} du_c / dt &= \alpha(t)[\xi(t) - \eta_c(t)], \\ du_i / dt &= 0, i \neq c \end{aligned} \tag{3.22}$$

一维空间 R 中， μ_{i-1} 和 μ_i 之间的分割平面就是线段 $[\mu_{i-1}, \mu_i]$ 的中点。另一方面，单元 μ_i 周围的空间区域 P_i 对应于线段 $[1/2(\mu_{i-1} + \mu_i), 1/2(\mu_i + \mu_{i+1})]$ 。选中值 ξ 的概率为 $p(\xi) dV_\xi$ ，其中 dV_ξ 为 ξ 空间的体积微分。当处在平稳状态时，对于任意的非零 $\alpha(t)$ ，等式 (3.22) 的右边将均为零，从而得到：

$$\int_{P_i} p(\xi)[\xi(t) - \mu_i(t)] dV_x = 0, \quad (3.23)$$

可见 μ_i 的稳定值与空间区域 P_i 的质心一致。

§ 3.5 自组织特征学习算法

采用图 3.7 所定义（一般取 (a) 方式）的邻域拓扑关系，将所有输入向量的长度归一化为常值（一般取 1）后，重复下列过程。当第 (5) 步中的增益项降为 0 时，权值最终收敛并且固定：

Kohonen 自组织特征映射算法

(1) 权值初始化：

将 N 个输入到 M 个输出节点之间的权值以随机数初始，并设定邻域的初始半径。

(2) 提供新输入。

(3) 计算到所有节点的距离：

计算输入到各个输出节点 j 之间的距离 d_j ：

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2$$

其中， $x_i(t)$ 是时刻 t 节点 i 的输入； $w_{ij}(t)$ 是时刻 t 输入节点 i 到输出节点 j 之间的权值。

(4) 选择最小距离的输入节点：

选择节点 j^* 作为最小 d_j 的输出节点。

(5) 更新到节点 j^* 和邻域内节点的权值：

对节点 j^* 和邻域内的所有节点集合 $NE_{j^*}(t)$ 按下式更新权值：

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i(t) - w_{ij}(t))$$

其中， $j \in NE_{j^*}(t)$ ， $0 \leq i \leq N-1$ ； $\eta(t)$ 是随时间 t 减小的增益项， $0 \leq \eta(t) < 1$ 。

(6) 转第 (2) 步，重复之。

§ 3.6 推广的自组织映射模型

对于实际中的高维信号，其特征空间的分布有可能存在某种形式的封闭性。这时如果自组织网络的维数低于特征空间的维数，则其最终得到的权向量将不可避免存在边界效应。这是由于自组织映射中权向量的调整是在网络的一个邻域中进行的。输入信号特征的封闭性要求在调整过程中，网络所有单元在空间位置上具有等价性，但是由网络维数决定的网络拓扑结构有可能破坏这一等价性。例如，线性结构中两端单元的邻域结构与中间单元是不同的；将环状的二维特征空间映射到一维的线性输出阵列上时，输出阵列两端的距离无疑是最远的，但它们在二维环状的特征空间上却是相邻的。因此，如果将输出阵列推广为一个空间闭合的曲面，就可以消除此类边界，使输出节点在拓扑意义上具有完全对等的地位，更好地保持映射中拓扑特征。

综上所述，推广的自组织模型就是，若用维数和特征空间匹配（相等）、拓扑结构于特

征空间相似的原则排列输出单元, 则 Kohonen 自组织映射必然能更好地保持特征空间的拓扑性质。出于实现及显示的目的, 一般采用三维的输出节点, 并按封闭曲面的方式排列(如球面), 同时采用无边界的封闭网络, 使得每个单元在邻域意义上是等价的, 有利于各单元平等地竞争学习。

这种闭合曲面上的自组织映射模型, 可以更好地保持映射的拓扑特征, 改善网络的分类性能。

§ 3.7 应用自组织网络实现汉字倾斜矫正的实验

§ 3.7.1 汉字倾斜自动矫正问题的提出

汉字识别(包括手写体和印刷体的识别)的研究是当前模式识别与人工智能研究中的难点和热点。简单地说, 汉字识别就是使计算机能认识较为规整的汉字。当前随着计算机技术的迅猛发展, 社会不断进步, 信息处理量激增, 考虑到我国汉字信息的原稿多用手写这一习惯, 社会对高速自动输入汉字有强烈的需求, 它在办公自动化、印刷业等方面将大有用武之地, 它的研究成功将彻底解决汉字信息处理系统手动输入速度低这个影响整个系统效率的“瓶颈”问题, 促进国民经济的发展和社会生产水平的提高。

汉字自动识别的理论包括预处理、汉字特征的抽取等几个部分组成。其中预处理部分又包括汉字的倾斜矫正、字符分割等几个部分。在汉字信息的自动录入阶段, 不可避免地存在字符倾斜的问题。由于倾斜汉字的自动矫正问题比较困难, 故而现在成熟的商用汉字自动识别系统一般采用手动矫正的方法。也有人提出, 将输入的汉字逐度旋转, 通过计算某个特征量(如字符的平均间距等), 在其极值处获得汉字的倾斜角度。且不论这种方法的计算量, 由于这逐度旋转的步长存在量化问题, 因此无法精确求解出其倾斜角度。为此, 我们尝试了用自组织网络对输入汉字做聚类处理, 而后对反映汉字分布的各个聚类中心点做一次曲线拟合计算, 通过直线的斜率直接获得汉字的倾斜角度。

§ 3.7.2 自动矫正的方法及实验结果

首先, 将源图中字符像素的 x 和 y 坐标作为二维的特征样本, 训练 Kohonen 自组织网络。在实验中, 使用两级的 Kohonen 网络。第一级用于图像的整体聚类, 得到大致地聚类中心点; 第二级用于对这些聚类中心点进行精细划分, 得到反映图中字符分布的点集。最后根据这些点集进行曲线拟合运算, 求出需要矫正的角度。

如图 3.12 所示, 设置 Kohonen 的网络参数: 输出平面为 15×15 的神经元阵列; 初始的邻域尺寸 r 为 7, 训练过程中邻域尺寸 r 逐次递减到 1; 当 $r=7$ 时, 对应的收敛判决门限为 0.07, 收敛判决门限的递减因子为 0.75, 即对应的门限为 $0.07 \times 0.75^{(7-r)}$; 当 $r=7$ 时, 增益项 $\eta(t)$ 为 0.1, 递减因子为 0.99, 即 $\eta(t)=0.1 \times 0.99^{(7-r)}$; 另外为了防止在某个 r 值处产生无穷次迭代, 系统还设置了最大迭代次数。图中还显示了源图的像素点映射到 Kohonen 输出平面的情况(只显示出一部分), 可知有 30 个源像素点映射到输出平面的神经元 (1, 1) 中, 有 12 个点映射到神经元 (2, 1) 中, 等等。

图 3.13 显示出 Kohonen 输出神经元所代表的几何位置 (即它的特征值)。如果一个输出神经元与一个字符的重心相对应, 则该字符中的大多数像素都将映射到该神经元, 即映射到它的源像素点较多。根据这一自组织原则, 在显示图 3.13 时, 已经作了如下的相对门限处理: 根据 Kohonen 输入/输出映射表, 获得此次训练的最大映射数 M , 据此可以得到一个映射门限 $m=CM$ (如图, 我们取系数 $C=0.1$), 如果某一输出神经元所对应的源像素点数小于 m , 则忽略此神经元。

我们取出图 3.13 所显示的已作阈值处理的神经元特征值送入第二级 Kohonen 网络再次进行自组织处理。该二级 Kohonen 网络的特征参数如图 3.14 所示, 其输出平面为 $6*6$ 。

同样, 如图 3.15 所示, 我们得到经过阈值处理的二级 Kohonen 网络的有效输出神经元。可知, 此时特征点的几何分布大致体现了源图的字符分布 (关于源图, 我们可以参照后面的图 3.16 所示)。

再根据二级 Kohonen 网络提取出来的聚类中心, 我们作关于一次曲线 (直线) 的拟合运算, 实验中我们采用最小均方误差的拟合算法, 得到如图 3.16 所示的直线。图中为了提供参照, 还把具有倾斜字符的源图作为背景显示出来, 可知此条直线的斜率反映了字符的倾斜角度。

图 3.17 是使用上述参数, 对另一幅倾斜文字作的计算, 可以看出其拟合出来的曲线也反映了该图的文字倾角。从而说明了此方法的可行性。

§ 3.7.3 一些讨论

使用 Kohonen 自组织网络分类相当对图像中代表字符的像素集进行聚类。常规的逐度旋转文字图像, 计算灰度投影的某个特征量以求取文字倾斜角的方法, 存在旋转角度步长的量化问题, 无法精确求解倾斜角, 而本文所提出的方法, 不存在步长量化问题, 理论上可以精确求出文字图像的旋转角度。

曾经考虑过将图像中的字符像素全部提取出来, 做曲线拟合, 已算出字符整体的倾斜角度。但是这样做, 要求对上万个 (甚至更多的) 像素点进行拟合计算, 计算量不确定。也许可以将图像缩小一定比例后, 再按上述办法做拟合运算, 以达到减小计算量的效果。按照上述的两种思路尝试后, 效果不是很好。因为存在像素密集的字符 (即比划较多的汉字) 和像素稀疏的字符 (即比划较少的汉字)。它们对曲线拟合的影响就象天平上的左右两个不对称的砝码, 很大程度上影响了拟合曲线的斜率。

而使用 Kohonen 网络进行自组织映射, 虽然也是一种样本抽样运算, 但它受字符密度的影响很小, 尤其是在作了两级 Kohonen 网络的自组织训练后。

第一级 Kohonen 网络要求输出的平面足够大, 以实现字符训练样本的合理聚类, 它要求的迭代次数可以少一些。第二级 Kohonen 网络的输出平面应接近于图像中的字符数的实际大小, 同时它要求的迭代次数可以多一些, 以提高分类精度。

如何根据实际要求, 自动合理地设置这两层 Kohonen 网络的输出平面尺寸及训练迭代次数, 以实现字符倾斜角度完全自动地计算, 是本实验方法需要进一步深入探讨的地方, 因时间关系只能题录到此。

第四章 分形理论与神经网络

神经网络作为一种智能分类器，与输入信号的模式特征空间关系密切。反映不同模式的特征向量越典型，神经网络的分类效果就越显著。纹理分析作为模式识别领域的一个分支，在图像理解和计算机视觉等方面具有重要的应用价值，也一直是它们的研究重点和难点。关于纹理特征，由于它是一种很强的关联模式，如果仅使用单纯的灰度（或彩色，对于彩色纹理图案而言）信息是不够充分的，还需要我们从它周围的相关信息计算出某些特征量，而分形理论在这方面具有独到之处。

§ 4.1 非欧氏几何学

欧氏几何学是一门具有 2000 多年历史的数学分支，它是以规整几何图形学为研究对象。所谓的规整几何图形就是指点、直线与线段；平面与平面中的各种多边形；空间与空间中的各种多面体等；以及由曲线或曲面所组成的各种几何图形。

对于欧氏几何中的线、面和体等图形，我们可以用长度、面积和体积等量纲来测量，而它们分别是长度单位的 1, 2, 3 等整数次方，它们恰好与这些几何图形存在空间的欧氏维数相等。另一方面，这些数值与决定几何形状的变量个数及自由度是一致的，即直线上的任意点可用 1 个实数表示，平面上的任意点可用 2 个实数组来表示，等等。

把自由度作为维数（又称为经验维数），是很自然的定义，看似没有什么可怀疑的地方。但早在 1890 年，就有人对经验维数提出了较深刻的疑问，这是因为可以用一个实数来表示二维正方形上的任意点。用一条曲线即可把平面完全覆盖的最好例子是 Peano 曲线，如图 4.1 所示，Peano 曲线可定义为图中折线的极限。从图中可知，该曲线可把平面完全覆盖。此线属于自相似，处处不可微，是分形的一个例子，被称为非规整几何图形。属于非规整几何图形的还有 Cantor 集，Sierpinski 集等，它们又被称为病态几何图形。研究非规整几何图形的几何学是非欧氏几何学中的一种。

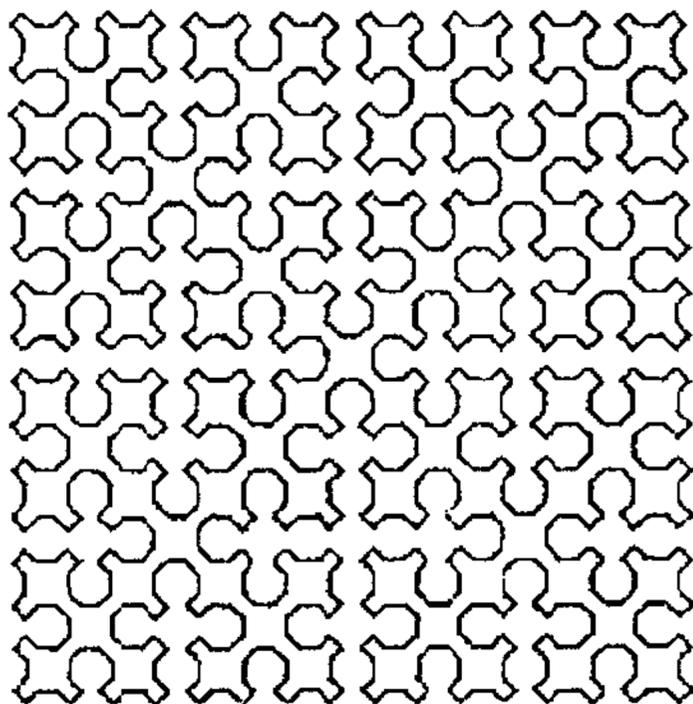


图 4.1 Peano 曲线

Peano 曲线的考虑方法也适用于三维以上, 即可以用一个实数来表示 n 维空间图形中的任意点。也就是说, 如果从自由度来考虑, 也可把 n 维空间看成 1 维, 这就产生了矛盾。为了避免这一矛盾, 必须从根本上重新考虑维数的定义。为此提出了不少有关维数的定义, 其中最著名的是相似维数 (Similarity Dimension)。

如果把线段、正方形和立方体的边 2 等分后, 可以分别得到 2^1 , 2^2 和 2^3 个与原来相似的图形。这里出现的指数 1、2、3 分别与其图形的经验维数一致。更一般地, 如果某图形是由全体缩小 $1/a$ 的 a^D 个相似图形构成, 则此指数 D 就具有维数的意义。此维数被称为相似维数。按此定义, Peano 曲线是由全体缩小 $1/2$ 的 2^2 个图形构成, 所以它的相似维数为 2, 与正方形的欧氏维数一致。前面的矛盾得以解决。

相似维数常用 D_s 表示, 按照其定义, D_s 完全没有是整数的必要。如果某一图形是由全体缩小 $1/a$ 的 b 个相似形所组成, 即 $b=a^{D_s}$, 则相似维数 D_s 为

$$D_s = \frac{\ln b}{\ln a} \quad (4.1)$$

相似维数将经验维数扩大到非整数值, 具有划时代意义, 但它仅适用于具有严格自相似性的规则分形。关于任意图形的维数定义, 最具有代表性的是 Hausdorff 维数。

§ 4.2 Hausdorff 测度和维数

§ 4.2.1 Hausdorff 测度

为了能定量地描述包括非整数值在内的维数, 波恩大学数学家 Felix Hausdorff 在 1919 年从测度的角度引进了 Hausdorff 维数的定义。其中关于 Hausdorff 测度的定义如下:

如果 U 为 n 维欧氏空间 R^n 中任何非空子集, U 的直径定义为 $|U| = \sup\{|x-y| : x, y \in U\}$, 即 U 内任何两点距离的最大值, 式中的 \sup 是上确界的缩写。如果 $\{U_i\}$ 为可数 (或有限)

个直径不超过 δ 的集构成的覆盖 F 的集合类, 即 $F \subset \bigcup_{i=1}^{\infty} U_i$, 且对每一个 i , 都有 $0 < |U_i| \leq$

δ , 则称 $\{U_i\}$ 为 F 的一个 δ -覆盖。

设 F 为 R^n 中的任何子集, s 为一非负数, 对任何 $\delta > 0$, 定义

$$H_\delta^s(F) = \inf \left\{ \sum_{i=1}^{\infty} |U_i|^s : \{U_i\} \text{ 是 } F \text{ 的 } \delta\text{-覆盖} \right\} \quad (4.2)$$

式中 \inf 是下确界的缩写。

显然, 对于给定的 F 和 s , $H_\delta^s(F)$ 是 δ 的一个减函数, 并且当 $\delta \rightarrow 0$ 时, $H_\delta^s(F)$ 趋于一个极限, 这个极限可以是 (通常是) 0 或 ∞ , 也可能是一个非 0 的实数, 记为

$$H^s(F) = \lim_{\delta \rightarrow 0} H_\delta^s(F) \quad (4.3)$$

称 $H^s(F)$ 是 F 的 s 维 Hausdorff 测度。

上述的 Hausdorff 测度满足测度公理; 特别地, $H^s(\Phi) = 0$; 如果集合 $E \subset F$, 则

$H^s(E) \leq F^s(F)$; 若 $\{F_i\}$ 是一个可数不交的波雷尔集序列, 则 $H^s\left(\bigcup_{i=1}^{\infty} F_i\right) = \sum_{i=1}^{\infty} H^s(F_i)$ 。

所谓波雷尔集是 R^n 中满足下列性质的最小集类:

- (a) 每一个开集和每一个闭集都是波雷尔集。
- (b) 每一个有限个 (或可数个) 波雷尔集之交或并都是波雷尔集。

Hausdorff 测度进一步推广了长度、面积和体积等概念。并且它保留了前述的比例性质: 即当比例放大 λ 倍时, s 维 Hausdorff 测度放大 λ^s 倍, 这是分形理论的基础。其数学描述及证明如下:

Hausdorff 测度的比例性 设 $F \subset R^n$, $\lambda > 0$; 令 $\lambda F = \{\lambda x : x \in F\}$, 则

$$H^s(\lambda F) = \lambda^s H^s(F)。$$

证明 设 $\{U_i\}$ 是 F 的一个 δ -覆盖, 则 $\{\lambda U_i\}$ 是 λF 的一个 $\lambda\delta$ -覆盖。所以

$$H_{\lambda\delta}^s(\lambda F) \leq \sum_i |\lambda U_i|^s = \lambda^s \sum_i |U_i|^s$$

由于上式对任意 δ 覆盖 $\{U_i\}$ 都成立, 从而有

$$H_{\lambda\delta}^s(\lambda F) \leq \lambda^s \cdot H_{\delta}^s(F)$$

令 $\delta \rightarrow 0$ 时, 得到 $H^s(\lambda F) \leq \lambda^s \cdot H^s(F)$

$$\text{而上式有可以变量代换为 } H^s\left[\frac{1}{\lambda} \cdot (\lambda F)\right] \leq \left(\frac{1}{\lambda}\right)^s \cdot H^s[(\lambda F)] \quad (4.4)$$

$$\text{即 } \lambda^s \cdot H^s(F) \leq H^s(\lambda F) \quad (4.5)$$

比较 (4.4) 和 (4.5) 式, 命题得证。

§ 4.2.2 Hausdorff 维数

Hausdorff 维度是以 Hausdorff 测度为基础, 是各种分形维数中最基本的一种, 其定义如下:

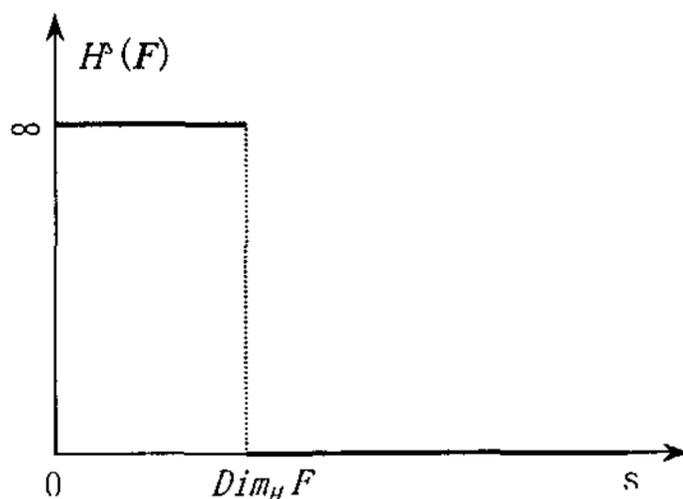


图 4.2 集合 F 的 Hausdorff 测度 $H^s(F)$ 与 s 的关系图。
Hausdorff 维数是由 ∞ 阶跃到 0 的临界值。

对于给定的集合 F , 如果把 Hausdorff 测度 $H^s(F)$ 看成 s 的一个函数, 由数学分析可知 $H^s(F)$ 是关于 s 的不增函数, 其具体关系如图 4.2 所示, 则将 $H^s(F)$ 由 ∞ 阶跃到 0 的临界值定义为 Hausdorff 维度 $Dim_H F$, 有时简写为 D_H , $D_H(F)$ 。

集合 F 的 Hausdorff 维度 $Dim_H F$ 的精确数学描述为

$$D_H = \inf\{s : H^s(F) = 0\} = \sup\{s : H^s(F) = \infty\} \quad (4.6)$$

从而

$$H^s(F) = \begin{cases} \infty, & \text{若 } s < D_H \\ 0, & \text{若 } s > D_H \end{cases} \quad (4.7)$$

如果 $s = D_H$, 则 $H^s(F)$ 可以为 0 或 ∞ 或满足

$$0 < H^s(F) < \infty$$

Hausdorff 维度满足下面的性质:

- (1) 开集 若 $F \subset \mathbf{R}^n$ 为开集, 因 F 包含一个具有正 n 维体积的球, 所以 $D_H = n$ 。
- (2) 光滑集 若 F 为 \mathbf{R}^n 中光滑 (即连续可微) m 维流形 (即 m 维曲面), 则 $D_H = m$ 。
特别地, 光滑曲线维数为 1, 光滑曲面维数为 2。
- (3) 单调性 若 $E \subset F$, 则 $D_H(E) \leq D_H(F)$ 。这可从对每个 s , $H^s(E) \leq H^s(F)$ 这个测度性质得到。
- (4) 可数稳定性 若 F_1, F_2, \dots 为一 (可数) 集序列, 则

$$D_H\left(\bigcup_{i=1}^{\infty} F_i\right) = \sup_{1 \leq i \leq \infty} \{D_H(F_i)\}。 \text{由单调性, 对每个 } j, \text{ 必然有}$$

$$D_H\left(\bigcup_{i=1}^{\infty} F_i\right) \geq D_H(F_j)。 \text{另一方面, 若 } s > D_H(F_j), \text{ 则对所有的 } i, H^s(F_i) = 0。 \text{所}$$

$$\text{以 } H^s\left(\bigcup_{i=1}^{\infty} F_i\right) = 0, \text{ 从而给出反向不等式。}$$

- (5) 可数集 若 F 是可数的, 则 $D_H(F) = 0$ 。若 F_i 是一单点, 有 $H^0(F_i) = 1, D_H(F_i) = 0$,

$$\text{所以由可数稳定性, 得 } D_H\left(\bigcup_{i=1}^{\infty} F_i\right) = 0。$$

大部分维数的定义都基于“用尺度 δ 进行量度”的设想: 忽略尺寸小于 δ 时的不规则性, 并察看 $\delta \rightarrow 0$ 时, 这些测量值的变化。所以也可用同样的方法描述 Hausdorff 维数。

设一条长度 L 的线段, 若用长为 r 的“尺”测量, 结果为 N , 则称该线段有 N 尺, 显然 N 的数值与尺的大小有如下关系:

$$N(r) = L/r \propto r^{-1}$$

同理, 若测量面积为 A 的平面, 则用边长为 r 的单位小正方形去测量, 才能得出确定的 N 值, 其 N 值为:

$$N(r) = A/r^2 \propto r^{-2}$$

一般, 对于任何一个确定维数的几何体, 若用与它相同维数的“尺”去量度, 则可得到一个确定的数值 N ; 若用低于其维数的“尺”去测量, 则结果为 ∞ ; 若用高于其维数的“尺”去测量, 则结果为 0。其数学表达式为

$$N(r) \propto r^{-D_H} \quad (4.8)$$

如果取比例系数为 1, 即将尺度 r 根据上述的长度 L 、面积 A 等归一化后, 则由上式可得:

$$D_H = \ln N(r) / \ln(1/r) \quad (4.9)$$

式中的 D_H 就称为 Hausdorff 维数, 可以为整数, 也可以为分数。Hausdorff 维数是分数的物体被称为分形, 此时的 D_H 值称为该分形的分形维数, 简称为分维, 或分数维数。

§ 4.2.3 分形的有关定义

关于分形, 至今为止尚未有严格的定义。B.B.Manderbrot 在 1982 年曾提出:

如果一个集合在欧氏空间中的 Hausdorff 维数严格大于其拓扑维数, 则称该集合为分形集, 简称为分形。

四年后, 他又提出:

组成部分以某种方式与整体相似的形体称为分形。

但这些定义都不完备。有学者认为, 对“分形”的定义可以用生物学中对“生命”定义的方法处理。生物学中“生命”无严格和明确的定义, 但都可以列出一系列生命物体的特性。同样可以不去寻找分形的精确定义, 而把它看作具有如下性质的集合 F :

- (1) F 具有精细结构, 即有任意小比例的细节。
- (2) F 是如此的不规则, 以至于它的整体与局部都不能用传统的几何语言描述。
- (3) F 通常有某种自相似的形式, 可能是近似的或者是统计的。
- (4) F 的“分形维数”(以某种方式定义的) 一般大于它的拓扑维数。
- (5) 在大多数令人感兴趣的情形下, F 可以用非常简单的方法来定义, 可能由迭代产生。

§ 4.3 广义维数

除了前面讨论的 Hausdorff 维数 D_H 和相似维数 D_S 以外, 分形理论中还有其他的几种维数定义。

由于在 Hausdorff 维数 D_H 的定义中, 只考虑了所需 δ -覆盖的个数 $N(\delta)$, 而不考虑每个覆盖 U_i 中所含分形集元素的多少, 因此引入了如下几种考虑元素分布的维数定义, 其中设 P_i 表示分形集的元素属于覆盖 U_i 中的概率。

(1) 信息维数 D_I

根据分形集中元素关于覆盖集 U_i 的概率分布, 有信息熵 S_I :

$$S_I = -\sum_{i=1}^N P_i \ln P_i \quad (4.10)$$

从而得信息维数 D_I :

$$D_I = -\lim_{\delta \rightarrow 0} \frac{S_I(\delta)}{\ln \delta} \quad (4.11)$$

(2) 关联维数 D_C

若分形中某两点之间的距离为 δ , 引进关联函数 $C(\delta)$:

$$C(\delta) = \frac{1}{N^2} \sum_{i,j=1}^N \theta(\delta - |x_i - x_j|) = \sum_{i=1}^N P_i^2 \quad (4.12)$$

其中

$$\theta(\delta - |x_i - x_j|) = \begin{cases} 1, & \delta \geq |x_i - x_j| \\ 0, & \delta < |x_i - x_j| \end{cases} \quad (4.13)$$

则定义关联维数 D_C :

$$D_C = \lim_{\delta \rightarrow 0} \frac{C(\delta)}{\ln \delta} \quad (4.14)$$

关联维数便于从实验中直接测定, 应用很广, 是由 P. Grassberger 和 I. Procaccia 在 1983 年提出。

(3) 广义维数 D_k

进一步, 如果引进概率论中的 k 阶 R 广义熵

$$S_k(\delta) = \frac{1}{1-k} \ln \left(\sum_{i=1}^N P_i^k \right) \quad (4.15)$$

从而得到更一般的广义维数 D_k :

$$D_k = -\lim_{\delta \rightarrow 0} \frac{S_k(\delta)}{\ln \delta} \quad (4.16)$$

其中 k 可取正负整数。

这几种维数的相互关系如下:

(1) 在等概率的情况下, $D_H = D_I$ 。

事实上, 如果 $P_i = 1/N(\delta)$ ($i=1, 2, \dots, N$), 由 D_I 的定义可知

$$D_I = -\lim_{\delta \rightarrow 0} \frac{-\sum_{i=1}^N \frac{1}{N(\delta)} \ln \frac{1}{N(\delta)}}{\ln \delta} = -\lim_{\delta \rightarrow 0} \frac{\ln N(\delta)}{\ln \delta} = D_H$$

(2) 当 $k=0$ 时, $D_k|_{k=0} = D_H$ 。

$$D_k|_{k=0} = -\lim_{\delta \rightarrow 0} \frac{\ln \sum_{i=1}^N P_i^0}{\ln \delta} = -\lim_{\delta \rightarrow 0} \frac{\ln N(\delta)}{\ln \delta} = D_H$$

(3) 当 $k \rightarrow 1$ 时, $D_k = D_I$ 。

事实上,

$$\begin{aligned}
\lim_{k \rightarrow 1} S_k(\delta) &= \lim_{k \rightarrow 1} \frac{1}{1-k} \ln \left(\sum_{i=1}^N P_i^k \right) = -\lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \ln \left(\sum_{i=1}^N P_i \cdot P_i^\varepsilon \right) \\
&= -\lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \ln \left(\sum_{i=1}^N P_i \cdot \exp(\varepsilon \ln P_i) \right) = -\lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \ln \left(\sum_{i=1}^N P_i (1 + \varepsilon \ln P_i) \right) \\
&= -\lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \ln \left(1 + \varepsilon \cdot \sum_{i=1}^N P_i \cdot \ln P_i \right) = -\sum_{i=1}^N P_i \cdot \ln P_i = S_I(\delta)
\end{aligned}$$

从而得到

$$\lim_{k \rightarrow 1} D_k = \lim_{k \rightarrow 1} \left(-\lim_{\delta \rightarrow 0} \frac{S_k(\delta)}{\ln \delta} \right) = -\lim_{\delta \rightarrow 0} \frac{\lim_{k \rightarrow 1} S_k(\delta)}{\ln \delta} = -\lim_{\delta \rightarrow 0} \frac{S_I(\delta)}{\ln \delta} = D_I$$

(4) 当 $k \rightarrow 2$ 时, $D_2 = D_C$ 。

可见, Hausdorff 维数、信息维数和关联维数等都仅是广义维数的特例。

k 阶广义维数 D_k 具有两个重要性质:

(1) k 值越大则维数值 D 越小, 即通常满足下列不等式

$$D_{k_1} \geq D_{k_2}, (k_1 < k_2) \quad (4.17)$$

特别当 $k=0, 1, 2$ 时, 下列关系成立

$$D_H \geq D_I \geq D_C \quad (4.18)$$

(2) 当两个几何图形之间的变换是根据一个可微分的映射关系产生时, 则变换过程中的广义维数 D_k 保持不变。

§ 4.4 多重分形

多重分形 (multifractal approach) 又称为多标度分形或多重分形测度。对于许多非均匀的分形现象, 一个维数有时无法描述其全部特征。例如, δ -覆盖的单元数分别为 $N_1(\delta)$ 和 $N_2(\delta)$ 的两条分形曲线合并而成的曲线, 其 δ -覆盖的单元数为 $N(\delta)$, 有如下关系:

$$N(\delta) = N_1(\delta) + N_2(\delta) \quad (4.19)$$

如果曲线 1、曲线 2 的分数维分别为 D_1 、 D_2 , 则有

$$N_1(\delta) = \delta^{-D_1}, N_2(\delta) = \delta^{-D_2} \quad (4.20)$$

则根据 Hausdorff 维数定义, 可得合成曲线的分数维 D 为

$$D = -\lim_{\delta \rightarrow 0} \frac{\log(\delta^{-D_1} + \delta^{-D_2})}{\log \delta} = \max\{D_1, D_2\} \quad (4.21)$$

这说明通过计算分形维数无法区分单一分形集 (single fractal) 和多重分形集 (multiple fractal)。为了获得对一个分形集更详细的描述, 需要增加能刻画不同分形子集的参数, 由此引入多重分形理论。

§ 4.4.1 二项多重分形

设 $1/2 < \mu_0 < 1$, $\mu_1 = 1 - \mu_0$, $S = [0, 1]$. 把 S 按二进制划分, 即每次都把一个区间划分成两个相等的子区间。把每一区间上的质量按相对比例 μ_0 和 μ_1 展布于两个二进制子区间上。如此重复 n 次后, S 被分为 $N = 2^n$ 段, 用二进制小数 $t = 0.b_1 b_2 \dots b_n$ 表示每个子区间。若某个子区间的二进制小数 t 中含 1 的个数为 k , 则该子区间所含质量占总质量的比 $\rho_k = (\mu_0)^{n-k} (\mu_1)^k$, $\rho_k \in M$, 其中 $M = \{(\mu_0)^n, (\mu_0)^{n-1} \mu_1, \dots, (\mu_1)^n\}$ 完全刻画了 S 中质量的分布情况。 S 中含有 ρ_k 的子区间数 $N(k)$ 为

$$N(k) = C_n^k, k \in (0, 1, \dots, n) \quad (4.22)$$

定义 $\xi = k/n$, 表示 t 中 1 所含比例, 则上式改为

$$N(\xi) = C_n^{n\xi} \quad (4.23)$$

$$\rho_\xi = (\mu_0^{1-\xi} \mu_1^\xi)^n \quad (4.24)$$

记含有 ρ_k 的子区间所构成的集合为 S_ξ , 现在计算其分数维。用 $\delta = 2^{-n}$ 的区间单元覆盖 S_ξ , 所需的单元数

$$N_n(\xi) = C_n^{n\xi} \quad (4.25)$$

运用 Stirling 公式

$$n! \approx \sqrt{2\pi n} (n/e)^n \quad (4.26)$$

得

$$N_n(\xi) = \frac{1}{\sqrt{2\pi n \xi(1-\xi)}} e^{-n[\xi \log \xi + (1-\xi) \log(1-\xi)]} \quad (4.27)$$

而其分数维 (记为 $D(\xi)$)

$$D(\xi) = \lim_{\delta \rightarrow 0} \frac{\log N_n(\xi)}{\log \delta} = -\frac{\xi \log \xi + (1-\xi) \log(1-\xi)}{\log 2} \quad (4.28)$$

集合 S 是由不同的子集 S_ξ 构成

$$S = \bigcup_{\xi} S_\xi$$

虽然 S 是一线段, 其分数维为 1, 由公式(3-1-7)可知其不同子集 S_ξ 具有不同的分数维 $D(\xi)$, 其介于 0 和 1 之间。由此可以引出多重分形的概念。

§ 4.4.2 分形子集与维数谱

设 x 是拓扑维数为 d 的任一子集, μ 是 x 上的一个概率密度或测度。对测度空间 (x, μ) 作适当的迭代划分, α 是与划分有关的参数。第 n 步划分后具有相同单元测度 μ_α 构成的子集记作 $x_n(\alpha)$ 。若 $x_\alpha = \lim_{n \rightarrow \infty} x_n(\alpha)$ 是一分形, 则 x_α 称为 (x, μ) 的分形子集。一个多重分形可以表示成具有不同维数的分形子集之并。若单元测度 μ_α 与单元尺度 δ 满足幂律关系:

$$\mu_\alpha \sim \delta^\alpha \quad (4.29)$$

则称 α 为 Holder 指数。它控制着测度 μ_α 的奇异性, 因此 α 亦称作奇异性指数。

对于概率密度为 μ_α 的分形子集 x_α 的任意可列 δ -覆盖 $\{U_i\}_{i \in N}$, 定义

$$H'_\delta(x_\alpha, \mu_\alpha) = \inf \left\{ \sum_{i=1}^{\infty} |U_i|^r : \{U_i\}_{i \in N} \text{ 为 } x_\alpha \text{ 的 } \delta\text{-覆盖} \right\} \quad (4.30)$$

则 (x_α, μ_α) 的 r 维 Hausdorff 测度定义为

$$H^r(x_\alpha, \mu_\alpha) = \lim_{\delta \rightarrow 0} H'_\delta(x_\alpha, \mu_\alpha) \quad (4.31)$$

若存在临界指数 $f(\alpha)$ 使得

$$H^r(x_\alpha, \mu_\alpha) = \begin{cases} 0, & r > f(\alpha) \\ \infty, & r < f(\alpha) \\ \text{有限正值}, & r = f(\alpha) \end{cases} \quad (4.32)$$

则 $f(\alpha)$ 称为多重分形的奇异谱。

可见 $f(\alpha)$ 就是分形子集 x_α 的 Hausdorff 维数, 且当 $x_\alpha \subset \bigcup_{i=1}^{\infty} U_i$, $0 < |U_i| \leq \delta$, $i \in N$ 时

$$\begin{aligned} f(\alpha) &= \inf \left\{ r : \liminf_{\delta \rightarrow 0} \sum_{i=1}^{\infty} |U_i|^r = 0 \right\} \\ &= \sup \left\{ r : \liminf_{\delta \rightarrow 0} \sum_{i=1}^{\infty} |U_i|^r = \infty \right\} \end{aligned} \quad (4.33)$$

设 $\{U_i\}_{i \in N}$ 是尺度为 δ 的集合, $N(\alpha)$ 是在 $[\alpha, \alpha + d\alpha]$ 内概率密度为 μ_α 的单元个数, 由于 $H'_\delta(x_\alpha, \mu_\alpha) = N(\alpha)\delta^r$, 因此, 当 $N(\alpha) \sim \delta^{-f(\alpha)}$ 时, $H^r(x_\alpha, \mu_\alpha)$ 有限, 故得

$$f(\alpha) = -\lim_{\epsilon \rightarrow 0} \ln N(\alpha) / \ln \delta \quad (4.34)$$

是描述区域维数的连续谱。

§ 4.5 盒子数和图象分数维特征的计算

§ 4.5.1 盒子维的定义

盒子维又称为计盒维数 (Box Dimension, Box-Counting), 是应用最广泛的维数之一。盒子维的定义如下:

设 $A \subset R^n$, 在欧氏距离下, 用边长为 $1/2^n$ 的小盒子紧邻地去包含 A , 设 $N_n(A)$ 表示 A 所需的最小盒子数, 则:

$$D = \lim_{n \rightarrow \infty} \frac{\ln N_n(A)}{\ln 2^n} \quad (4.35)$$

此即为集合 A 的盒子维。

可以看出, 当 n 增大时, 也就是当测量的尺度减小时, 集合的不规则性迅速地表现出来了。考虑二维平面上的集合, 可以把它看成是一幅二值图象, 对它盒子维的计算可以如下进行: 逐渐增大 n , 分别计算出相应的 $N_n(A)$ 的值, 这样就可以得到一组 $(\ln 2^n, \ln N_n(A))$ 的数据对, 在利用最小均方误差等方法求出 $\ln N_n(A)$ 相对于 $\ln 2^n$ 的斜率, 就是所要求的盒子维。

§ 4.5.2 灰度图象分数维的提取

我们社会生活中, 大量的信息是通过灰度图象获得的, 如果把真实的世界看成是三维的, 视觉计算理论的创始人美国科学家 D. Marr 认为图象反映的仅仅是 2.5 维的信息, 因此如何有效地利用这有效的信息极为重要。一个真实空间中的分形表面映射到图象的灰度表示的过程从数学上说还没有完全解决, 但是图象的灰度值表面最终总可以用相关于分形表面法线上及入射光线之间夹角的一个函数来表示, 所以美国的 A. P. Penland 假定, 如果一个物体的表面是分形的, 则由它产生的图象灰度表面也是分形的性质; 反之亦然。因此我们可以从灰度图象中提取分数维。

灰度图象的盒子维数计算方法, DBC(Differential Box Counting)方法, 是由 N. Sarkar 和 B. B. Chaudhuri 提出的, 其基本思想如下:

对于某个大小为 $M \times M$ 图像 A , $s \times s$ 为一个单元面积 ($M/2 \geq s > 1$, 且 s 为整数), s 可变大小。注意到此时的归一化尺寸为 $r = s/M$ 。则用绝对尺寸 $s \times s \times s$ 的立方体覆盖图像网格点 (x, y) 的灰度所需盒子数为:

$$n_r(x, y) = \left\lceil \frac{\max I_i - \min I_i}{s} \right\rceil + 1 \quad (4.36)$$

如图 4.3 所示, 其中只画出 (x, y) 外围的四个网格点: $I_i (i=1, 2, 3, 4)$ 表示相应点的灰度值大小, $[\cdot]$ 为取整运算。

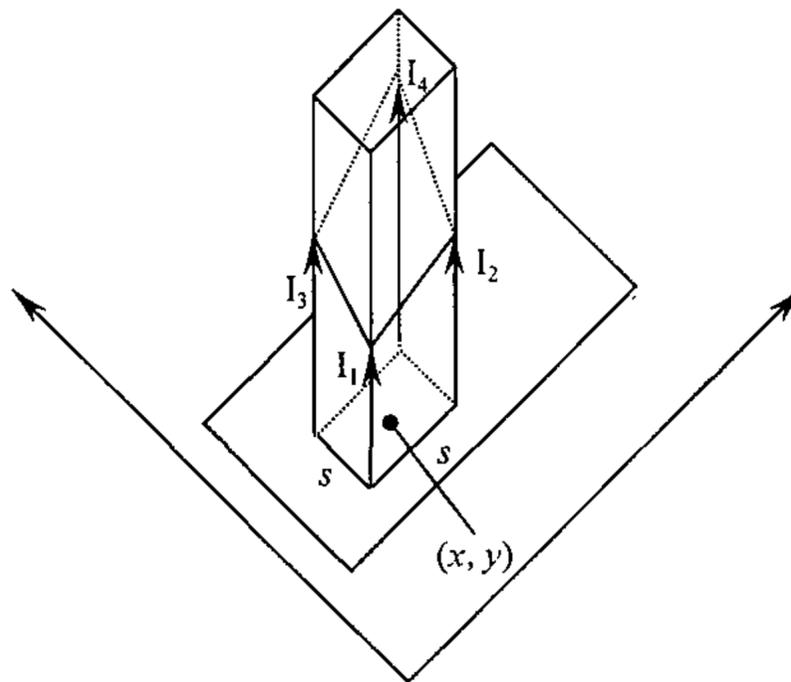


图 4.3 灰度值图象的盒子维求取

对于区域 A 中的每个网格点, 均做同样的计算, 从而可按下列公式近似地计算出覆盖整个区域所需的盒子数 $N(r)$:

$$N(r) = \sum_{(x,y) \in A} n_r(x,y) \quad (4.37)$$

而分维数

$$D = \lim_{r \rightarrow 0} \frac{\log N(r)}{\log r} \quad (4.38)$$

改变 r 的大小, 可以求出相应的各个 $N(r)$, 应用最小二乘法就可以求出该图像区域的分维 D 。

§ 4.5.3 灰度图像的广义维数计算

由于盒子维同样存在没有考虑图像像素点在不同盒子中分布特征的缺点。为了克服此缺点, 在分析数字图像 S 时, 不仅要计算覆盖图像所需的盒子数, 还要统计不同盒子所含的像素点数, 以计算图像的广义维数谱。为此, 针对每个盒子分配一个量

$$\mu_i = \frac{N_i}{N} \quad (4.39)$$

这里 N 是图像 S 包含的总像素数, N_i 是第 i 个盒子所包含的像素数。由此可得集合 M :

$$M = \{\mu_i\}_{i=0}^B \quad (4.40)$$

这里 B 是覆盖图像 S 所需的盒子数, 可知 μ_i 即为集合 M 的测度 (或说概率密度)。

为了分析图像 S 的不同分形子集, 我们有如下的 q 阶广义 Renyi 维数 $D(q)$:

$$D(q) = \begin{cases} \frac{1}{q-1} \lim_{\delta \rightarrow 0} \log[N(q, \delta)] / \log \delta & q \neq 1 \\ - \lim_{\delta \rightarrow 0} \frac{\sum \mu_i \log \mu_i}{\log \delta} & q = 1 \end{cases} \quad (4.41)$$

其中,

$$N(q, \delta) = \sum_{i=1}^{N(\delta)} \mu_i^q \quad (4.42)$$

是概率密度 μ_i 的 q 阶矩。

根据计算分数维的 DBC 方法, 我们可以用一种简单的近似算法求出此广义维数。方法如下:

对于 $M \times M$ 的图像, $s \times s \times s$ 的覆盖盒子, $r = s/M$ 令

$$\mu_r(x,y) = \frac{n_r(x,y)}{N(r)} \quad (4.43)$$

这里 $n_r(x,y)$ 和 $N(r)$ 即为 (4.36), (4.37) 式所定义的。令

$$\chi(q, r) = \sum_{i,j} [\mu_r(i, j)]^q \quad (4.44)$$

则可计算得 q 阶广义维数 $D(q)$:

$$D(q) = \frac{1}{(q-1)} \lim_{r \rightarrow 0} \frac{\ln \chi(q, r)}{\ln r} \quad (4.45)$$

§ 4.5.4 彩色图像维数特征的提取

彩色纹理图像是由 R, G, B 三个基色分量合成的, 但它们之间存在大量相关的冗余信息, 可以用色度学中的 HSV 变换, 消除其相关性, 而仅对 V (亮度信息) 分量进行前述的图像维数特征计算, 并增加两个剩余的 S (饱和度), H (色调) 特征分量来描述彩色图像纹理特征。

实验中, 我们为了简化计算, 使用 NTSC 彩色传输制式的 YIQ 色度模型来代替 HSV 模型。其中 Y 等价于彩色图像的亮度信息, 并对它进行维数特征计算; I, Q 作为表征彩色信息的另外两个独立特征分量。其中 YIQ 模型与 RGB 模型的关系矩阵为:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & 0.523 & 0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (4.46)$$

§ 4.6 纹理图像的空隙特征

Mandelbrot 和 Voss 曾指出不同的分形集可以具有相同的分形维数而具有差异很大的纹理。Mandelbrot 引入“空隙”(lacunarity)这一术语以刻画此现象。他对这一特征提出几种等价的定义, 其中最实用的是

$$\Lambda = E \left[\left(\frac{M}{E(M)} - 1 \right)^2 \right] \quad (4.47)$$

这里 M 是分形集的“质量”, $E(M)$ 是期望“质量”。针对著名的海岸线测量问题, 分形集的“质量”依赖于测量尺度 L , 有以下公式

$$M(L) = KL^D \quad (4.48)$$

这里, K 是常数。此二阶统计量空隙, 用以测量分形集的实际“质量”与期望“质量”之间的差异, 当纹理细密时其值小, 粗糙时其值大。

J.M.Keller 等人在此基础上对空隙特征加以改进。对数字图像 S , 用尺度为 L 的盒子进行覆盖。引入特征 $C(L)$

$$C(L) = \frac{M(L) - N(L)}{M(L) + N(L)} \quad (4.49)$$

这里, $M(L)$ 是盒子平均所含像素点数, $N(L)$ 等于覆盖图像 S 所需盒子数除以 S 所含像素点的总数, 即每个像素点平均所占盒子数。根据这一思想, 在前述 BDC 方法的基础上, 我

们可以建立了一种简单有效的计算空隙特征的公式。设 S 所含象素点数为 P , 则

$$\begin{cases} M(L) = \frac{P}{N_r} \\ N(L) = \frac{N_r}{P} \end{cases} \quad (4.50)$$

将式 (4.50) 代入式 (4.49) 得

$$C(L) = \frac{P^2 - N_r^2}{P^2 + N_r^2} \quad (4.51)$$

§ 4.7 纹理图像特征提取和分类器的构造

纹理图像的分割由两部分组成: 纹理特征的抽取和分类器的设计。我们采用纹理图像的分维 D 、广义维数谱 $D(q)$ 和空隙特征 $C(L)$ 作为纹理特征, 具体如下:

特征 1 纹理图像的分形维数 D , 采用 DBC 方法求得。

特征 2~3 纹理图像的 2 个附加变换图像的分形维数 D_1, D_2 。设原图为 $I(x,y)$, 2 个附加的变换图像如下:

$$I_1(x,y) = \begin{cases} I(x,y) - L_1 & \text{if } I(x,y) > L_1 \\ 0 & \text{otherwise} \end{cases} \quad (4.52)$$

$$I_2(i,j) = \begin{cases} 255 - L_2 & \text{if } I(x,y) > (255 - L_2) \\ I(i,j) & \text{otherwise} \end{cases} \quad (4.53)$$

其中, $L_1 = g_{\min} + av/2$; $L_2 = g_{\max} - av/2$ 。这里 g_{\min} , g_{\max} 和 av 分别表示图像 I 的灰度最小值、最大值和平均值。 I_1 和 I_2 分别称为 I 的高灰度值和低灰度值图像。图像变换的目的是突出图像某些方面的特征, 是图像分形维数估计中的常用方法。如果两幅图像 I_a 和 I_b 有一样的分维, 它们的高灰度图像 I_{a1} 和 I_{b1} 或低灰度图像 I_{a2} 和 I_{b2} 可能具有不同的粗糙度, 所以其分维会不同。

特征 4~5 纹理图像的广义维数谱 $D(q)$ 。根据实验, 我们选取 $q=5$ 和 $q=-5$, 以 $D(5)$ 和 $D(-5)$ 作为两个分类特征。

特征 6 纹理图像的空隙特征 $C(L)$ 。根据实验, 我们选取 $L=4$, 以 $C(4)$ 作为一个分类特征。

特征 7~8 彩色纹理图像中的 I, Q 分量 (YIQ 色度模型), 此类特征针对彩色纹理图像; 与之对应, 此时前述的几个分形维数特征是根据另一分量 Y 计算而得。

对于分类器的设计, 我们采用 BP 神经网络构造有监分类器。由 BP 网络构造的分类器容错能力强, 有良好的鲁棒性和自适应性。同时, BP 网络构造的分类器是非参数的, 对模式隐含的概率分布函数的要求比传统的分类方法弱得多, 适于处理一些分类特征复杂, 背景知识不清, 推理规则不明确的问题, 这些特点对于纹理图像的分类很适用。

§ 4.8 纹理特征提取的有关实验

§ 4.8.1 Brodatz 纹理图像的分割实验

图 4.4(a)是从 Brodatz 纹理图像集中挑选的 8 幅视觉上较接近, 分类较困难的纹理图像构成的 Mosaic, 图像大小为 512×512 , 灰度 8bit。我们采用上述的纹理特征 1~6 作为分类特征, 对图像的每个象素点, 在以此象素为中心的 17×17 的窗内计算这 6 个参数。分类器采用三层 BP 网络, 其各层节点数为 6, 48, 8。训练样本为从原图中目视挑选的 1600 个点, 每类 200 个。训练次数为 2000。图 4.4(b)为采用上述方法产生的分割结果并赋予不同的颜色而生成的图像。图 4.4(b)可见, 8 种纹理被较好地分割。

§ 4.8.2 自然图像的分割实验

图 4.5(a)是一幅房屋的图像, 大小 512×512 , 灰度 8bit。同实验一, 采用纹理特征 1~6。将原图目视分为 5 类: 天空, 屋檐, 玻璃窗, 砖墙, 走道。从原图中目视挑出各类训练样本, 训练样本数为 1000 个 (天空 100 个点, 屋檐 100 个点, 玻璃窗 300 个点, 砖墙 300 个点, 走道 200 个点)。作为分类器的 BP 网络采用二层网络, 各层节点数分别为 6, 30, 5。训练次数为 2000。图 4.5(b)为采用上述方法产生的分类结果并赋予五种颜色而生成的分割图像。从图 4.5(b)可见, 五种物体基本被分割开。

§ 4.8.3 彩色纹理图像的分割实验

图 4.6(a)是一幅自然景物的彩色图像, 大小 512×512 , R,G,B 各为 8bit。将其转化为 YIQ 格式, 采用前述的纹理特征 1, 4~8, 其中 4 个纹理特征由亮度信息 Y 计算而得。将原图目视分为 4 类: 天空, 白云, 树木, 草地。从原图中目视挑出各类训练样本, 训练样本数为 800 个 (天空 200 个点, 白云 100 个点, 树木 250 个点, 草地 250 个点)。作为分类器的 BP 网络采用二层网络, 各层节点数分别为 6, 30, 4。训练次数为 2000。图 4.6(b)为采用上述方法产生的分类结果并赋予 4 种颜色而生成的分割图像。从图 4.6(b)可见, 4 种物体被很好地分割开。

§ 4.8.4 航空图像的分割实验

图 4.7(a)是美国旧金山地区 Golden Gate 上空拍摄的照片, 图像大小 512×512 , R,G,B 各为 8bit。6 个分类特征的计算如同实验一。将原图目视分为 6 类: 大海, 山地, 建筑物, 树林, 平地 (包括沙滩), 大桥。从原图中目视挑出各类训练样本, 训练样本数为 1600 个 (大海 400, 山地 400, 建筑 400, 树林 200, 平地 150, 大桥 50)。作为分类器的 BP 网络采用二层网络, 各层节点数分别为 6, 36, 6。训练次数为 2000。图 4.7(b)为采用上述方法产生的分类结果并赋予 6 种颜色而生成的分割图像。其中蓝色为大海, 黄绿色为山地, 灰色为建筑物, 绿色为树林, 黄色为平地, 白色为大桥。从图 4.7(b)可见, 6 种物体基本被分割开。

结束语

我在研究生期间主要在模式识别领域作了以下几个方面的工作：

- (1) 人脸识别系统的开发（负责系统的整体集成，人眼及嘴的边缘特征提取，图像采集卡接口的编写，图像编辑及入库）；
- (2) GIS 系统中的遥感图像的混合像元分类；
- (3) 纹理图像特征的提取及分类；
- (4) 文字识别的预处理工作（包括二值图像的文字分割、倾斜文字的自动矫正）。

本论文主要对研究工作中理论性较强的神经网络部分及相关的理论进行深入、系统地总结。

神经网络理论是模拟人脑的思维活动而建立起的一整套理论。由于它具有自组织、自学习等特性，在模式识别领域中具有广泛的研究前景和应用价值。作为一种人工智能的分类方法，神经网络主要涉及三个方面的问题：

- (1) 特征提取。这涉及到许多领域的方法。例如在纹理分析中，我们通过分形理论得到反映纹理的特征量。
- (2) 网络结构设计。神经网络除了 BP 网络外，还有 Kohonen 模型，Hopfield 模型等等。就是同一种模型的网络也有不同的结构设计。精巧的神经网络结构设计有时可以解决一些很复杂的问题。这既给予了我们机会，也给予了我们挑战。
- (3) 关于训练样本的选取。神经网络的训练在整个过程中占有很重要的地位，由它确定的神经网络参数将直接决定样本分类精度的好坏。所以样本的选取非常重要，好的训练样本必须能够反映该类数据的统计分布规律。

本论文深入地研究了神经网络的理论基础、各种学习算法和拓扑结构，着重分析了有监督学习规则的 BP 模型和无监督学习规则的 Kohonen 自组织模型，并以此为基础在模式识别领域的图像认识和理解方面做了许多应用研究工作，取得了如下的一些成果：

- (1) 提出一种新的层次神经网络模型，以解决遥感图像中“同谱异构”的混合像元分解问题，并对葡萄牙里斯本地区 TM 图像的地面实况数据取得了极高的分类精度。
- (2) 提出一种用 Kohonen 网络自动检测文字倾斜角的新方法，即用两级 Kohonen 网络模型提取二值图像的文字中心，结合曲线拟合方法，以实现文字倾斜角的自动提取，实验结果较为满意。
- (3) 提出用分形理论中的分数维、多重分形中的广义维数谱和空隙参数等特征定义纹理图像的特征空间，结合神经网络分类器，实现对纹理图像的分割，并取得了很好的实验结果。

另外，还在神经网络的理论研究方面，作了如下一些工作：

- (1) 理论分析了 BP 学习规则的改进算法：MBP 算法和前向网络的自构形学习算法；
- (2) 理论分析了 Kohonen 网络的自组织原理，以及 Kohonen 网络的推广模型；
- (3) 研究了分形理论在几何图形维数定义上的独到之处，分形理论的局限性以及多重分形理论和广义维数谱的必要性；

另外，我还初步学习了混沌理论，研究了混沌神经网络与普通神经网络的关系。混沌与神经网络具有一些共同的本质特征，即系统的非线性和状态的模拟性。脑神经系统是由非线性神经元组成的复杂系统，人们通过对健康人脑电图的分析，发现其中存在混沌现象，从而证明混沌也是神经系统的正常特性。我们可以从两个方面将静态神经网络改造成具有混沌动力学行为的动态混沌神经网络，即从改造神经元模型的角度，使节点具有混沌响应，而导致整个网络具有混沌行为；或者从改造网络连接方式的角度，使节点具有动态连接方式，使整个网络同时具有混沌和收敛的特性。由于在这方面我还处在理论研究阶段，未做深入地实验研究，所以不在本论文中做详细说明。

由于分形理论和混沌理论本身就是一个完整的理论体系，本文在这方面的认识还很肤浅，仍然需要继续深入地研究和探索。

神经网络与其它理论的结合也是神经网络理论今后的发展趋势。期待着它在模式识别等信息科学领域中发挥越来越大的作用。

参考文献

- [1] Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Springer-Verlag, 1989.
- [2] Kohonen, *Self-Organization Maps*. Proceedings of the IEEE, Special Issue on Neural Networks.
- [3] Jacek M.ZURADA, *Introduction to Artificial Neural Systems*, West Publishing Company, 1992.
- [4] B.B.Mandelbrot, *The Fractal Geometry of Nature*, Freeman, San Francisco, 1983.
- [5] J.M.Keller, S.Chen, et al. *Texture Description and Segmentation through Fractal Geometry*. CVGIP 45, 1989:150-166.
- [6] N.Sarkar, B.B.Chaudhuri. *An Efficient Approach to Estimate Fractal Dimension of Textural Images*. Pattern Recognition. Vol. 25, No. 9, 1992: 1035-1041.
- [7] 张济忠, 分形, 清华大学出版社, 1995年。
- [8] 谢和平, 薛秀谦, 分形应用中的数学基础与方法, 科学出版社, 1997年。
- [9] 王东生, 曹磊, 混沌、分形及其应用, 中国科学技术大学出版社, 1995年。
- [10] 钟义信, 潘新安, 杨义先, 智能理论与技术——人工智能与神经网络, 人民邮电出版社, 1992年。
- [11] 刘政凯, 李葆馨, Kohonen 自组织特征映射模型的推广, 自动化学报, Vol.20, No.3, 1994年5月。
- [12] 焦李成, 神经网络的应用与实现, 西安电子科技大学出版社, 1995年。
- [13] 焦李成, 神经网络计算, 西安电子科技大学出版社, 1995年。
- [14] 李志勇, 模糊控制、神经控制和智能控制, 哈尔滨工业大学出版社, 1996年。
- [15] 史忠值, 神经计算, 电子工业出版社, 1993年。
- [16] 黄昌宁, 夏莹, 语言信息处理专论, 清华大学出版社, 广西科学技术出版社, 1996年。
- [17] H.A.Rowley, S.Baluja and T.Kanade, *Neural Network-Based Face Detection*, IEEE Trans. PAMI, Vol.20, No.1, 1998年。
- [18] C.M.Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, 1995.
- [19] R.Schalkoff, *Pattern Recognition: Statistical, Structural, and Neural Approaches* (New York: Wiley), 1992.
- [20] J.A.Benediktsson, P.H.Swain, and O.K.Ersoy, *Neural network approaches versus statistical methods in classification of multisource remote sensing data*. I.E.E.E. Transactions on Geoscience and Remote Sensing, 28,540-552, 1990.
- [21] D.L.Civco, *Artificial neural networks for land cover classification and mapping*. International Journal of Geographical Information Systems, 7, 173-186, 1993.
- [22] I.L.Thomas, V.M.Benning, and N.P.Ching, *Classification of Remotely Sensed Images* (Bristol: Adam Hilger), 1987.
- [23] Wilkinson, G.G., Fierens, F. and Kanellopoulos, I., 1995, *Integration of neural and statistical approaches in spatial data classification*. Geographical Systems, 2, 1-20.
- [24] P.M.Atkinson and A.R.L.Tatnall, *Introduction: Neural networks in remote sensing*, International Journal of Remote Sensing, Vol.18. No.4, 1997.
- [25] I. Kanellopoulos and G.G.Wilkinson, *Strategies and best practice for neural network*

- image classification, *International Journal of Remote Sensing*, Vol. 18, No.4, 1997.
- [26] Z.K.Liu and J.Y.Xiao, Classification of remotely-sensed image data using artificial neural networks, *International Journal of Remote Sensing*, Vol.12, No.11, 1991.
- [27] G.G.Wilkinson, I.Kanellopoulos, Z.K.Liu and S.Folving, Integrated land cover mapping from satellite imagery using artificial neural networks, *SPIE*, Vol.1941, *Ground Sensing*, 1993.
- [28] 李厚强, 王宜主, 刘政凯, 一种适用于多类别遥感图像分类的新方法, *遥感学报*, Vol.1, No.4, 1997年。
- [29] 中国科学技术大学生物医学工程跨系委员会, *神经网络及其应用*, 中国科学技术大学出版社, 1992年。
- [30] 徐希孺等, 混合象元的因子分解方法极其在大面积冬小麦种植面积估算中的应用, *科学通报*, 946~949, 1989年。
- [31] 刘永怀, 混合象元分解的理论和方法, *遥感技术与应用*, 7(1), 1992年。
- [32] 赵振宇等, *模糊理论和神经网络的基础与应用*, 清华大学出版社, 1995年。
- [33] 赵喜鹏等, 应用神经网络模型分解 AVHRR 混合像元, *遥感学报*, 2(1): 51~56, 1998年。
- [34] Josef, Cihlar, HungLy, and QinghanXiao, Land cover classification with AVHRR multichannel composites in northern environments, *Remote Sensing of Environment* 58: 36~51, 1996.
- [35] Charles Ichoku and Arnon Karieli, A review of mixture modeling techniques for sub-pixel land cover estimation, *Remote Sensing Review* 13: 161~186, 1996.
- [36] Wang, F, Fuzzy supervised classification of remote sensing images, *IEEE Transaction Geoscience and Remote Sensing* 28(2): 194~201, 1990.
- [37] Foody, GandCox, D, Sub-pixel land cover composition estimation using a linear mixing model and fuzzy membership functions, *International Journal of Remote Sensing* 15(3): 619~631, 1994.
- [38] Aaron Moody, etc, Artificial neural network responded to mixed pixels in coarse-resolution satellite data, *Remote Sensing of Environment* 58: 329~343, 1996.
- [39] Strahler, A, etc, MODIS land cover production algorithm technical basis document, *NASA document for MODIS Product No.11*, 1994.
- [40] Heermann, P, etc, Classification of multispectral remote sensing data using a back-propagation neural network, *IEEE Transactions on Geoscience and Remote Sensing* 30: 81~88, 1992.
- [41] J.D.Paola, etc, A review and analysis of back-propagation neural networks for classification of remotely-sensed multispectral imagery, *International Journal of Remote Sensing* 16(16): 3033~3058, 1995.
- [42] A. L. Yuille, D.S. Cohen and P. W. Hallinan, Feature extraction from faces using deformable templates, *Proc. CVPR'89*, 104~109, 1989.
- [43] Xiaobo Li and Nicholas Roeder, Face contour extraction from front-view images, *Pattern Recognition*, 28(8): 1167~1179, 1995.
- [44] Nicholas Roeder and Xiaobo Li, Accuracy analysis for facial feature detection, *Pattern Recognition*, 29(1): 143~157, 1996.
- [45] Chung-Lin Huang and Ching-Wen Chen, Human facial feature extraction for face interpretation and recognition, *Pattern Recognition*, 25(12): 1435~1444, 1992.

- [46] G. Chow and X. Li, Towards a system for automatic facial feature detection, *Pattern Recognition*, 26(12), 1739~1755, 1993.
- [47] X. Xie, R. Sudhakar and H. Zhuang, On improving eye feature extraction using deformable templates, *Pattern Recognition*, 27, 791~799, 1994.
- [48] Kin-Man Lam and Hong Yan, Locating and extracting the eye in human face images, *Pattern Recognition*, 29 (5): 771~779, 1996.
- [49] 罗杰波, 先进遥感模式识别方法的研究, 中国科学技术大学研究生毕业论文, 1992年。
- [50] 李厚强, 遥感图像模式识别神经网络方法的研究, 中国科学技术大学研究生毕业论文, 1997年。

攻读硕士学位期间发表或待发表的学术论文

1. 林峰, 刘政凯, 俞能海, “基于图像分割的人眼边缘提取方法的研究”,
《计算机应用与研究》(已录用);
2. 刘政凯, 林峰, 顾军捷, “遥感图像中波谱混迭像元分离方法的研究”,
已投《遥感学报》;
3. 李厚强, 刘政凯, 林峰, “基于分形理论的航空图像分类方法”, 已投
《遥感学报》;
4. 李厚强, 刘政凯, 林峰, “基于分形理论和 Kohonen 神经网络的纹理图
像分割方法”, 《计算机工程与应用》已录用。

附录 部分实验结果



图 3.12 第一级 Kohonen 网络的设置及输出结果

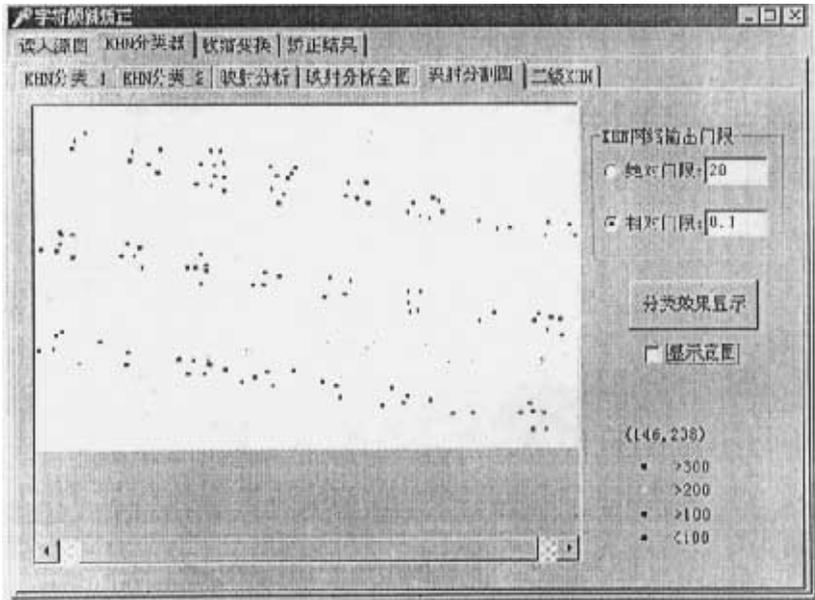


图 3.13 第一级 Kohonen 网络分类结果

75

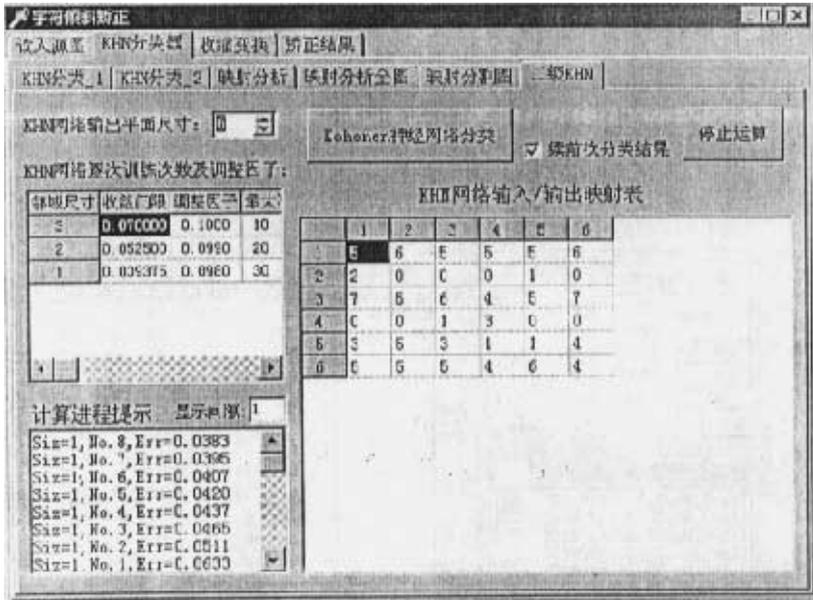


图 3.14 第二级 Kohonen 网络的设置及输出结果

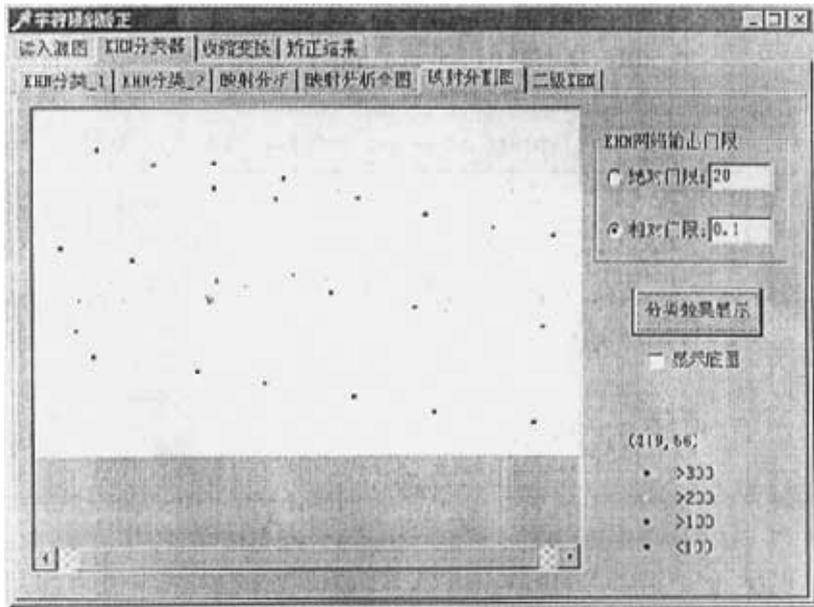


图 3.15 第二级 Kohonen 网络的分类结果

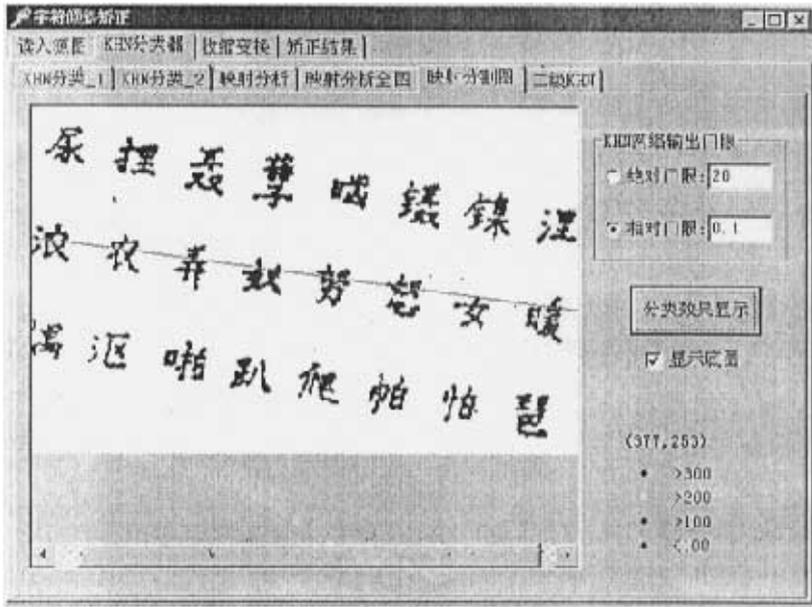


图 3.16 最终字符倾斜角度的获得

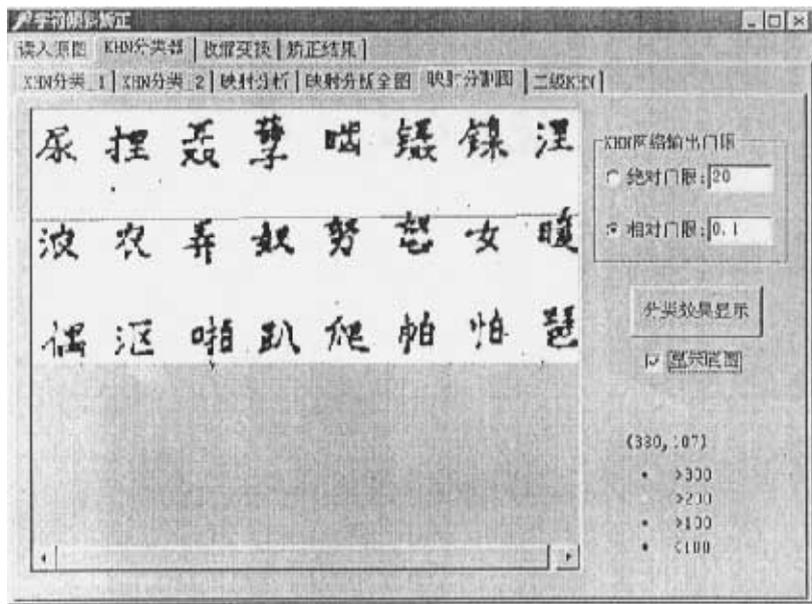
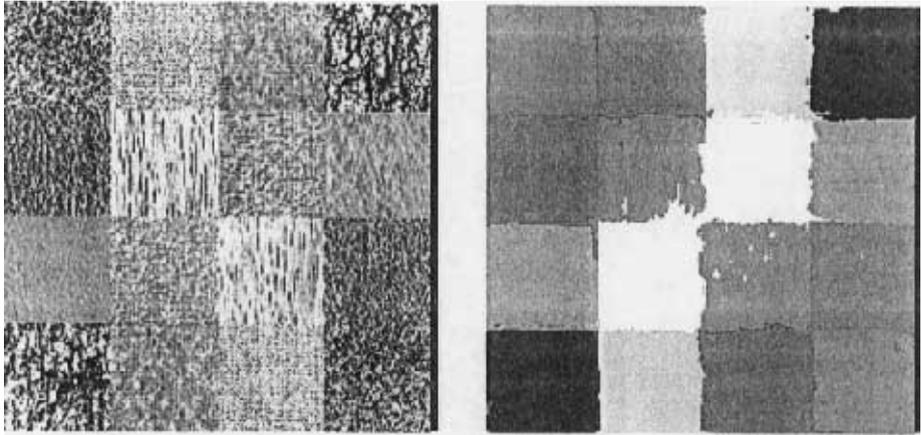


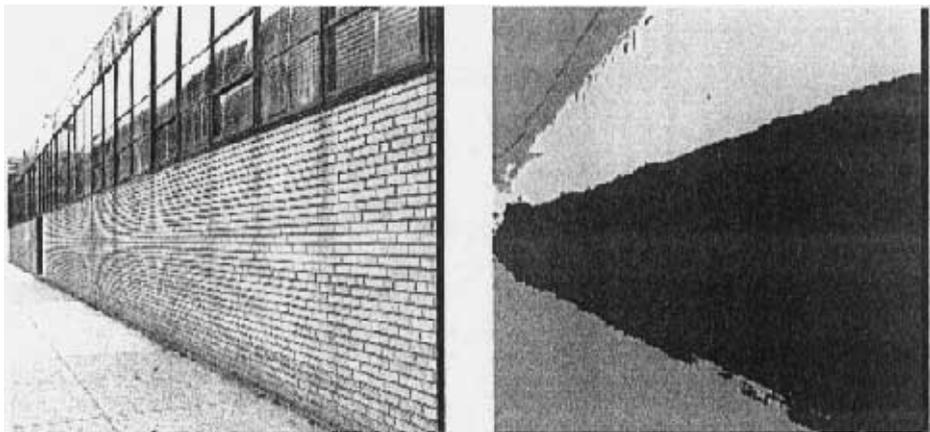
图 3.17 另一幅图像的实验结果



(a) 8幅Brodatz图象构成的Mosaic

(b) BP网络分类器生成的分割图

图4.4 8幅Brodatz图象构成的Mosaic及其分割图

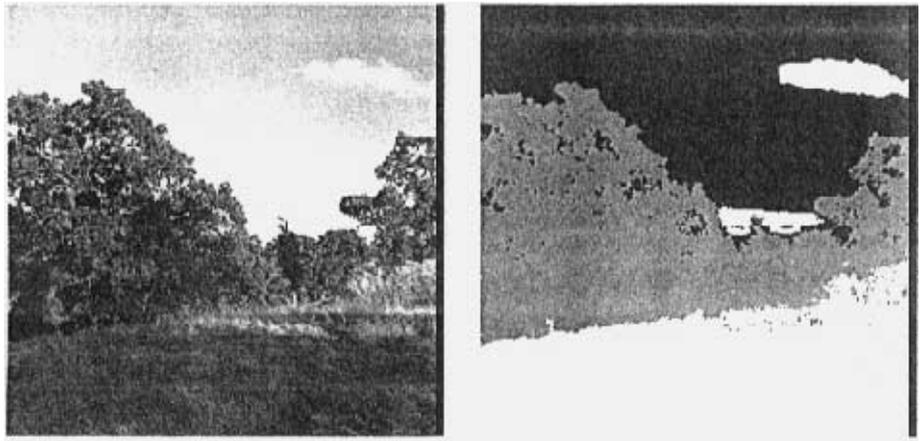


(a) 房屋图象

(b) BP网络分类器生成的分割图

图4.5 房屋图象及其分割图

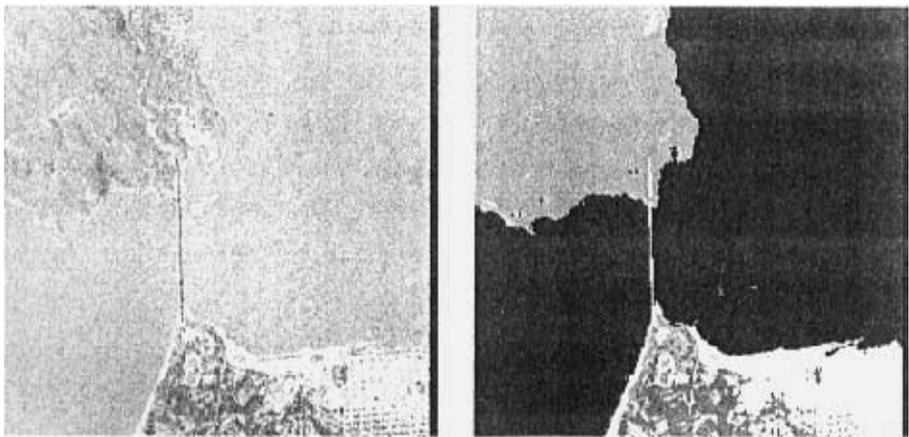
79



(a) 彩色自然景物图象

(b) BP网络分类器生成的分割图

图4.6 彩色自然景物图象及其分割图



(a) 彩色航空照片

(b) BP网络分类器生成的分割图

图4.7 彩色航空照片及其分割图