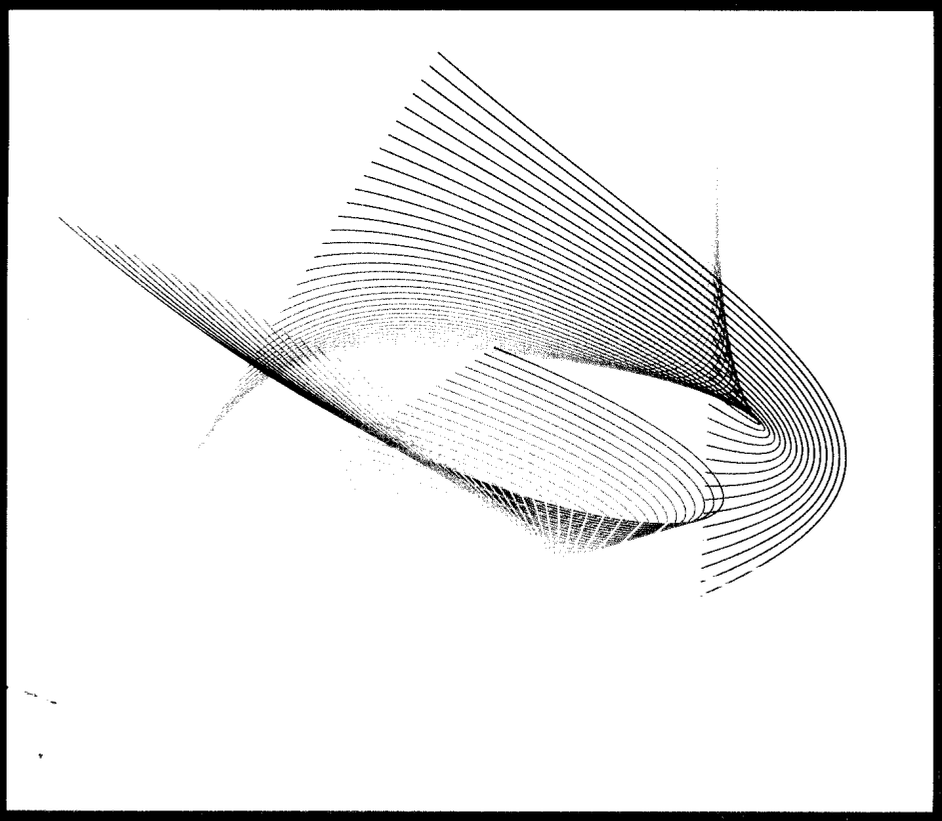
3

基于MATLAB的系统分析与设计

——神经网络



西安电子科技大学出版社

内 容 简 介

MATLAB 的推出得到了各个领域专家学者的广泛关注，其强大的扩展功能为用户提供了强有力的支持。

本书针对具有应用前景且被广泛关注的神经网络领域，简要介绍了神经网络的基本概念和学习算法，详细介绍了由 MATLAB 提供的神经网络工具箱函数的用法指南，最后以大量的应用示例，说明了基于 MATLAB 进行神经网络设计与应用的方法。

本书可作为神经网络原理、神经网络应用等课程的参考书，对课程学习可起到事半功倍的效果。对神经网络领域的教师、研究生、高年级本科生和广大科研人员都有重要的参考价值，对其它领域的科研人员也有一定的借鉴作用。

基于 MATLAB 的系统分析与设计

——神经网络

楼顺天　施　阳　编著

责任编辑　毛红兵　马继红

出版发行　西安电子科技大学出版社出版

（西安市太白南路 2 号）

邮　　编　710071

电　　话　(029)8227828

经　　销　新华书店

印　　刷　陕西省富平印刷有限责任公司

版　　次　1998 年 9 月第 1 版

1998 年 9 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印张 13.375

字　　数　312 千字

印　　数　1 ～ 4 000 册

定　　价　17.50 元

ISBN 7-5606-0646-6/TP・0327

前言

MATLAB[R]是 MathWorks 公司于 1982 年推出的一套高性能的数值计算和可视化软件，它集数值分析、矩阵运算、信号处理和图形显示于一体，构成了一个方便的、界面友好的用户环境。MATLAB 的推出得到了各个领域专家学者的广泛关注，其强大的扩展功能为各个领域的应用提供了基础。由各个领域的专家学者相继推出了 MATLAB 工具箱，其中主要有信号处理(signal processing)，控制系统(control system)，神经网络(neural network)，图像处理(image processing)，鲁棒控制(robust control)，非线性系统控制设计(nonlinear control system design)，系统辨识(system identification)，最优化(optimisation)，μ 分析与综合(μ analysis and synthesis)，模糊逻辑(fuzzy logic)，小波(wavelet)，样条(spline)等工具箱，而且工具箱还在不断增加，这些工具箱给各个领域的研究和工程应用提供了有力的工具。借画于这些“巨人肩上的工具”，各个层次的研究人员可直观、方便地进行分析、计算及设计工作，从而大大地节省了时间。

由我们编写的《MATLAB 程序设计语言》(西安电子科技大学出版社出版)受到了读者的一致好评，它从程序设计的角度出发，深入浅出地叙述了 MATLAB 的方方面面，是初学者的入门教材。在此基础上，我们又编写了更深层次的《基于 MATLAB 的系统分析与设计》系列图书，希望能给读者以帮助。

针对应用广泛的信号处理、控制系统和具有应用前景且被广为关注的神经网络这三个工具箱函数，我们编写了系列图书中的三本，除了详细介绍各个工具箱函数之外，简要介绍各领域的基本理论，着重介绍利用 MATLAB 对系统进行分析与设计的实例，因此，对这三个领域的教师、研究生、高年级本科生和广大科研人员都有重要的参考价值。

本书为神经网络部分，全书分三章。第 1 章简要介绍神经网络的基本理论，它对神经网络领域中的基本网络结构、学习算法及应用技术等内容做了简要介绍。第 2 章详细

叙述了MATLAB的神经网络工具箱函数。第3章以大量的应用示例，说明如何利用MATLAB进行神经网络的分析与设计。为了查阅方便，本书最后给出了两个具有重要参考价值的附录。附录A分类列出了MATLAB的基本命令及函数；附录B给出了部分重要工具箱中所包含的实用函数及其功能。另外，在第2章的前面，列出了本工具箱函数的索引，以方便读者查阅。

本书的出版得到了MathWorks公司的认可，有关购买MATLAB和SIMULINK软件及其它业务可直接与MathWorks公司联系：

The MathWorks, Inc.
24 Prime Park Way
Natick, MA 01760 - 1500
Phone: (500)647 - 7000
Fax: (508)647 - 7001
E-mail: info@mathworks.com
WWW:http://www.mathworks.com

本书在构思过程中得到了戴树荪教授的指点，在此深表谢意。卢君明和谭伏会研究生为本书第2章提供了初稿，在此致谢。本书的出版得到了西安电子科技大学出版社的大力支持，特别是毛红兵和马继红同志对本书进行了细致的编辑，做了大量的工作，在此深表谢意！

为获得本书的源程序，可通过下列Email地址与作者联系：

shtlou@xidian.edu.cn

楼顺天

1998年6月10日

符 号 说 明

由于本书涉及到大量的计算机程序，而程序中无法输入斜体和希文字母，因此为统一起见，本书中使用的符号均为正体；程序中采用国际上惯用的像形符号，例如在叙述中使用的符号 ω，在程序中用 w(或 W)代替；叙述中使用的带上下标符号如 a_1，ω_s，ω_p，F_s，T_s 等，在程序中用 a1，Ws，Wp，Fs，Ts 等代替。

目 录

第 1 章

神经网络基本理论

1.1 神经网络概述

1.1.1 神经网络的发展概况

1943 年心理学家 W. McCulloch 和数理逻辑学家 W. Pitts 首先提出了一个简单的神经网络模型，其神经元的输入输出关系为

$$y_j = \mathrm{sign}\left(\sum_i w_{ji} x_i - \theta_j\right)$$

其中，输入、输出均为二值量，w_{ji}为固定的权值。利用该简单网络可以实现一些逻辑关系。虽然该模型过于简单，但它为进一步的研究打下了基础。

1949 年 D. O. Hebb 首先提出了一种调整神经网络连接权值的规则，通常称为 Hebb 学习规则，其基本思想是，当两个神经元同时兴奋或同时抑制时，它们之间的连接强度便增加。这可表示为

$$w_{ij} = \begin{cases} \sum_{k=1}^{n} x_i^{(k)} x_j^{(k)} & i \neq j \\ 0 & i = j \end{cases}$$

这种学习规则的意义在于，连接权值的调整正比于两个神经元活动状态的乘积，连接权值是对称的，神经元到自身的连接权值为零。现在仍有不少神经网络采用这种学习规则。

1958 年 F. Rosenblatt 等人研究了一种特殊类型的神经网络，称为感知器(Perceptron)。他们认为这是生物系统感知外界传感信息的简化模型。这种模型主要用于模式分类。

1969 年 M. Miskey 和 S. Papert 发表了名为“感知器”的专著。他们在专著中指出，简单的线性感知器的功能是有限的，它无法解决线性不可分的两类样本的分类问题。典型的例子如“异或”运算，即简单的线性感知器不可能实现“异或”的逻辑关系。要解决这个问题，必须加入隐层节点。但是对于多层网络，如何找到有效的学习算法尚是难于解决的问题。

因此它使整个 70 年代神经网络的研究处于低潮。

美国物理学家 J. J. Hopfield 在 1982 年和 1984 年发表了两篇神经网络的文章，引起了很大的反响。他提出一种反馈互连网络，并定义了一个能量函数，它是神经元的状态和连接权值的函数，利用该网络可以求解联想记忆和优化计算的问题。这种网络后来称为 Hopfield 网络，最典型的应用例子是，网络成功地解决了旅行商最优路径问题。

1986 年 D. E. Rumelhart 和 J. L. Mcclelland 等人提出了多层前馈网络的反向传播算法(Back Propagation)，简称 BP 网络或 BP 算法。这种算法解决了感知器所不能解决的问题。

1987 年 6 月，在美国圣地亚哥举行的第一届神经网络国际学术讨论会期间，会议的组织者与主持人曾经发出一个不同凡响的宣言，声称："人工智能已亡，神经网络万岁"。虽然上述宣言对于人工智能和神经网络的极端态度不无哗众取宠之嫌，但是神经网络近年来的发展的确引人注目！

自 80 年代初重又兴起的第二次神经网络热潮是有目共睹的。神经网络理论是在现代神经科学研究成果的基础上提出来的，它反映了人脑功能的若干特性，但并非神经系统的逼真描述，而只是其简化、抽象和模拟。换言之，人工神经网络是一种抽象的数学模型。出自不同的研究目的和角度，它可用作大脑结构模型、认识模型、计算机信息处理方式或算法结构。迄今为止的神经网络研究，大体上可分为三个大的方向：

(1) 探求人脑神经系统的生物结构和机制，这实际上是神经网络理论的初衷；

(2) 用微电子学或光学器件形成特殊功能网络，这主要是新一代计算机制造领域所关注的问题；

(3) 将神经网络理论作为一种解决某些问题的手段和方法，这类问题在利用传统方法时或者无法解决，或者在具体处理技术上尚存困难。

1.1.2 神经网络的结构及类型

神经网络由许多并行运算的功能简单的单元组成，这些单元类似于生物神经系统的单元。神经网络是一个非线性动力学系统，其特色在于信息的分布式存储和并行协同处理。虽然单个神经元的结构极其简单，功能有限，但大量神经元构成的网络系统所能实现的行为却是极其丰富多彩的。和数字计算机相比，神经网络系统具有集体运算的能力和自适应的学习能力。此外，它还具有很强的容错性和鲁棒性，善于联想、综合和推广。

一般而言，神经网络是一个并行和分布式的信息处理网络结构，它一般由许多个神经元组成，每个神经元只有一个输出，它可以连接到很多其它的神经元，每个神经元输入有多个连接通路，每个连接通路对应于一个连接权系数。

严格地说，神经网络是一个具有下列性质的有向图：

(1) 每个节点有一个状态变量 x_j；

(2) 节点 i 到节点 j 有一个连接权系数 w_{ji}；

(3) 每个节点有一个阈值 θ_j；

(4) 每个节点定义一个变换函数 $f_j[x_i, w_{ji}, \theta_j(i \neq j)]$，最常见的情形为

$$f\left(\sum_i w_{ji}x_i - \theta_j\right)$$

神经网络模型各种各样，它们是从不同的角度对生物神经系统不同层次的描述和模

拟。有代表性的网络模型有感知器、多层映射 BP 网络、RBF 网络、双向联想记忆(BAM)、Hopfield 模型等。利用这些网络模型可实现函数逼近、数据聚类、模式分类、优化计算等功能。因此，神经网络广泛应用于人工智能、自动控制、机器人、统计学等领域的信息处理中。

1.2 感 知 器

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

1.2.1 重要的感知器神经网络函数

感知器(Perceptron)的初始化、训练、仿真将分别用到 MATLAB 神经网络工具箱中的三个函数：initp、trainp 和 simup。

1.2.2 感知器神经元模型

图 1.1 描述了一个由符号函数阈值元件(hardlim)组成的感知器神经元。

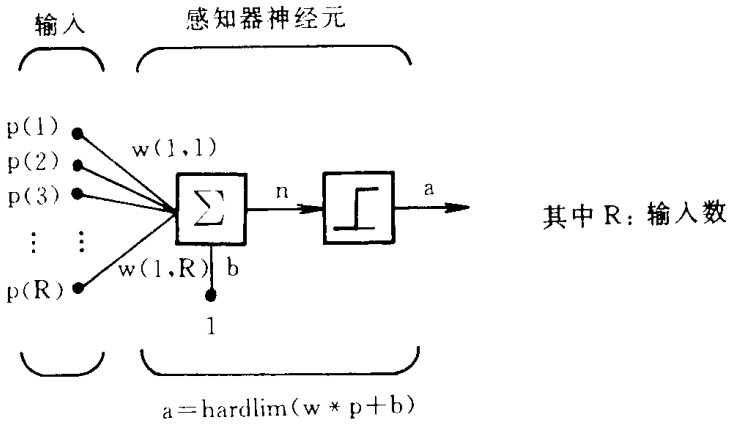


图 1.1 感知器神经元模型

图中，R 是网络输入的个数。在 MATLAB 中，感知器神经元的输出可通过两种方法得到

```
a=simup(p,w,b)
a=hardlim(w * p+b)
```

感知器神经元的每一个输入都对应于一个合适的权值，所有的输入与其对应权值的乘积之和输入给一个符号函数阈值单元，阈值单元还有一个输入为：常数 1 乘以阈值 b。

在感知器中使用了符号函数阈值单元，从而使得感知器能够将输入向量分为两个区域。最简单的一种情形是：当输入大于或等于 0 时输出为 1，当输入小于 0 时则输出为 0。下面以二输入感知器神经元为例，权值 w(1，1)＝－1 、w(1，2)＝＋1 ，阈值 b＝＋1，其分类情况如图 1.2 所示。

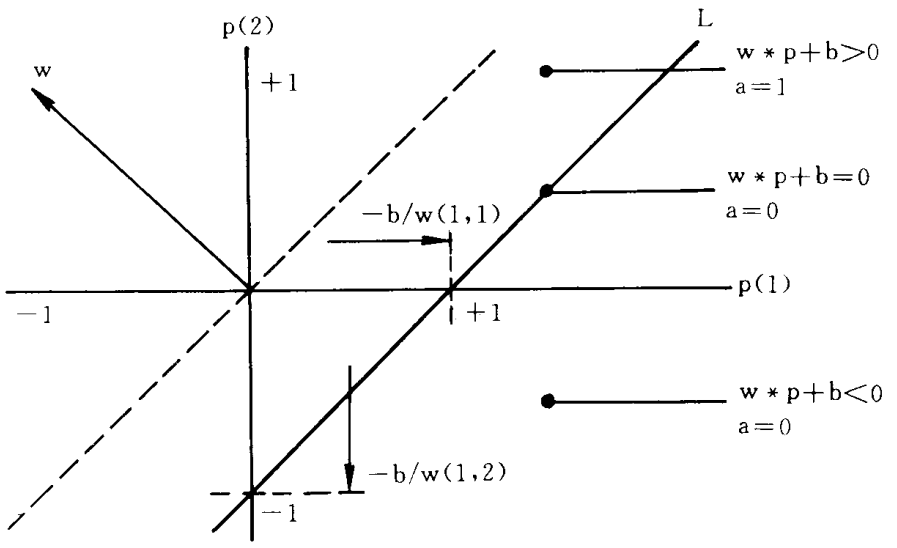


图 1.2　分类情况

1.2.3　感知器神经网络的网络结构

单层感知器神经网络可通过两种不同的方式加以描述，如图 1.3 所示，网络有 R 个输入，通过权值 w(i,j)与 S 个感知器神经元连接。

从图 1.3 可以看出，感知器神经网络只有一层神经元，这是由感知器学习规则造成的，感知器学习规则只能训练单层网络。感知器神经网络这种结构上的局限性也在一定程度上限制了其应用范围。

1.2.4　感知器神经网络的初始化

MATLAB 神经网络工具箱中的 initp 函数可自动产生[－1，＋1]区间中的随机初始权值和阈值，例如，对二输入、八神经元的感知器神经网络，初始化语句为

```
[w,b]=initp(2,8)
```

也可以利用输入向量 p 和目标向量 t 来初始化

```
[w,b]=initp(p, t)
```

1.2.5　感知器神经网络的学习规则

当给定输入向量 p 和感知器的学习误差 e 时，函数 learnp 可计算出权值和阈值的修正量 dw 和 db，而学习误差即为目标向量和感知器神经元输出之间的差值。感知器学习语句为

```
e=t−a;
```

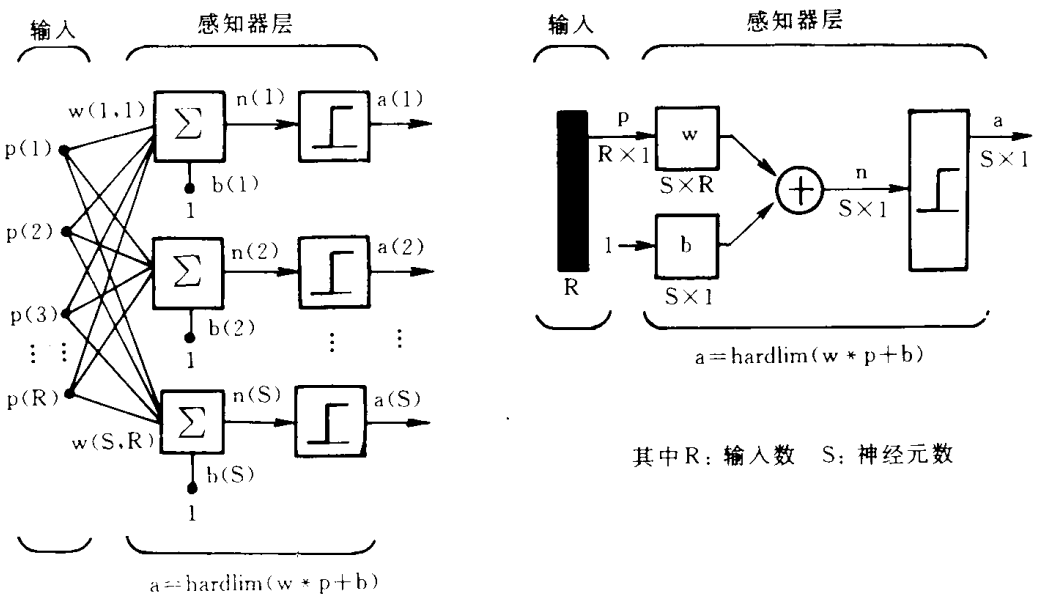


图 1.3 感知器神经网络结构

```
[dw,db]=learnp(p,e);
```

其中 p，e 为单个向量，也可以是相应的输入和误差的矩阵，这时 MATLAB 语句为

```
a=simup(p,w,b);
e=t-a;
[dw,db]=learnp(p,e);
w=w+dw;
b=b+db;
```

1.2.6 感知器神经网络的训练

利用函数 simup 和 learnp 反复地对感知器神经网络进行仿真和学习，最终可以得到最优的网络权值和阈值。这个过程可由函数 trainp 完成

```
tp=[disp_freq max_epoch];    [w,b,te]=trainp(w,b,p,t,tp);
```

其中 w 和 b 是网络的初始权值和阈值，p 是输入向量，t 是目标向量，tp 是训练的控制参数集合，包括显示频率和训练的最大步数。函数 trainp 完成每一步训练后，返回新的网络权值和阈值，并显示已经完成的训练步数 ep 及误差 te。

值得注意的是，函数 trainp 并不能保证感知器网络在取训练所得到的网络权值和阈值时，就可以顺利达到要求。因此，在训练完成后，最好要验证一下！验证的语句为

```
a=simup(p,w,b);
if all(a==t) , disp('It Works !') , end
```

如果发现网络不能成功地运行，那么可以继续使用函数 trainp 对网络进行训练。经过足够的训练后，如果网络还是达不到要求，那么应该认真地分析一下，感知器神经网络是否适合于这个场合。

1.2.7 感知器神经网络应用范围的局限性

由于感知器神经网络在结构和学习规则上的限制，其应用也有一定的局限性。首先，感知器神经网络的输出只能取 0 或 1；其次，单层感知器神经网络只能对线性可分的向量集合进行分类，这也是感知器的致命弱点。

1.2.8 双层感知器神经网络

单层感知器神经网络不能解决线性不可分的输入向量的分类问题，为解决这一问题可对输入的线性不可分向量进行预处理，使得其能够线性可分。换句话说，适当地设计多层感知器神经网络可以实现任意形状的划分。图 1.4 为一种常用的双层感知器神经网络。

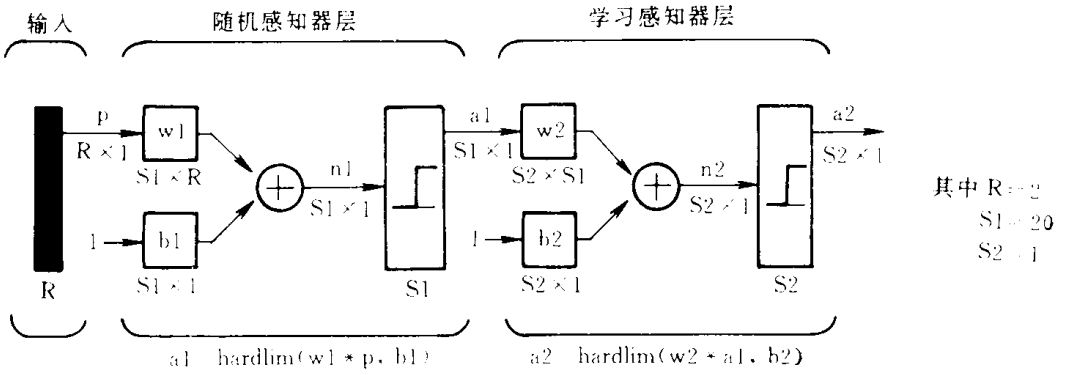


图 1.4 双层感知器网络结构图

1.3 线性神经网络

线性神经网络是最简单的一种神经元网络，由一个或多个线性神经元构成。50 年代末 Widrow 提出的 Adaline 是线性神经网络最早的典型代表。线性神经网络不同于感知器神经网络，其中每个神经元的传递函数为线性函数，因此线性神经网络的输出可以取任意值，而感知器神经网络的输出只能是 0 或 1。线性神经网络可采用 Widrow－Hoff 学习规则或者 LMS(Least Mean Square)算法来调整网络的权值和阈值。

1.3.1 重要的线性神经网络函数

函数 initlin、solvelin、simulin 分别用于线性神经网络的初始化、设计和仿真，函数 trainwh 可以对线性神经网络进行离线训练，函数 adaptwh 可以对线性神经网络进行在线自适应训练。

1.3.2 线性神经元模型

图 1.5 描述了一个由纯线性函数(purelin)组成的线性神经元。

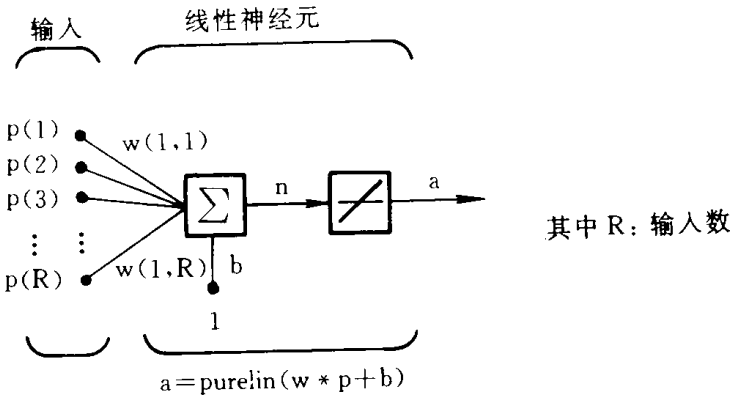


图 1.5　线性神经元

由于线性神经网络中神经元的传递函数为线性函数，其输入输出之间是简单的比例关系。因此对于单个神经元，可通过下列语句计算输出 a

a=purelin(w * p+b)

a=w * p+b

当网络的输入是矩阵形式、神经元个数不止一个时，网络的输出用下式求得

a=purelin(w * p,b)

网络输出也可以用仿真函数 simulin( )来求

a=simulin(p, w, b)

1.3.3　线性神经网络的网络模型

图 1.6 以两种形式给出了具有 R 个输入的单层(有 S 个神经元)线性神经元网络，其权值矩阵为 w。这种网络也称为 Madaline 网络。

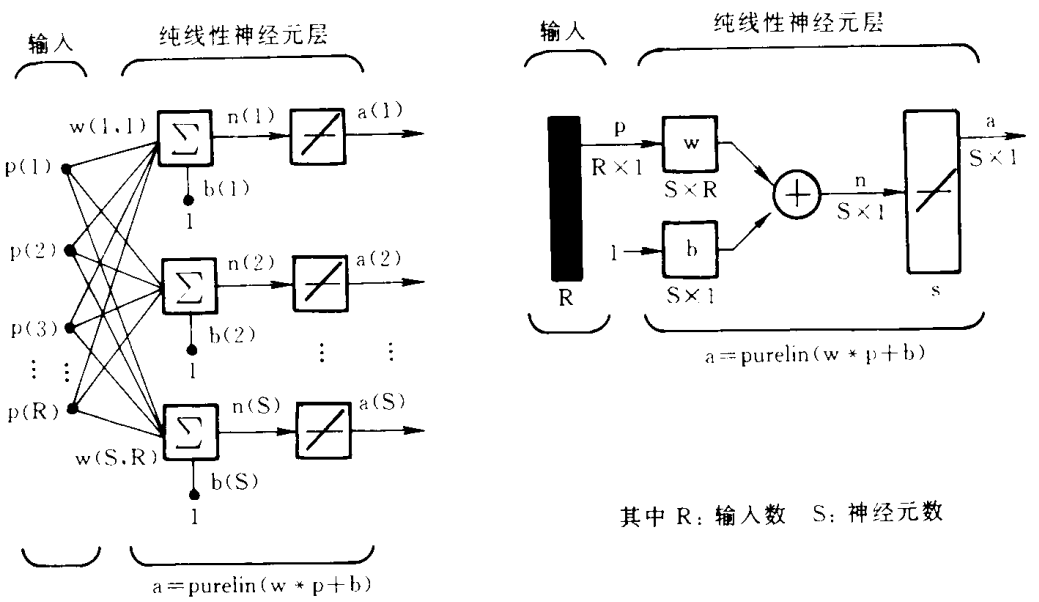


图 1.6　线性神经元网络

Widrow－Hoff 学习规则只能训练单层的线性神经元网络，但这并不影响单层线性神经网络的应用，因为对多层线性神经网络而言，可以设计出一个性能完全相当的单层线性神经网络。

1.3.4 线性神经网络的初始化

函数 initlin( )对线性神经网络初始化时，将权值和阈值取为很小的正数或负数。以一个有五个输入、七个神经元的线性神经网络为例，其初始化语句为

```
[w, b]=initlin(7, 5);
```

如果已知输入向量为 p，目标向量为 t，那么初始化语句为

```
[w, b]=initlin(p, t);
```

1.3.5 线性神经网络的设计

与大多数其它神经网络不同，只要已知其输入向量和目标向量，就可以直接设计出线性神经网络。函数 solvelin 无须经过训练，可直接求出网络的权值和阈值，使得网络误差的平方和最小。设计的语句为

```
[w, b]=solvelin(p, t);
```

函数 solvelin 在运行时，遵循下列原则：

(1) 如果网络能够达到零误差，那么函数 solvelin 就可以直接找到这时的网络权值和阈值；

(2) 如果不存在如此完美的零误差解，那么函数 solvelin 将找到一组权值和阈值，使得网络的误差平方和最小

$$\text{sse} = \sum_{k=1}^{Q} e(k)^2 = \sum_{k=1}^{Q} (t(k) - a(k))^2$$

(3) 如果网络有多个零误差解，那么函数 solvelin 将取最小的一组权值和阈值。

当然，函数 solvelin 并不是万能的，当计算的数值稳定性不能保证时，函数 solvelin 会给出警告信息

```
Warning : Matrix is singular to working precision .
```

这时，应该利用函数 learnwh 和 trainwh 来训练网络以得到网络的权值和阈值。

1.3.6 线性神经网络的学习规则

线性神经网络采用 Widrow－Hoff 学习规则，可利用函数 learnwh 求得权值和阈值的修正值。学习的语句

```
e=t－a;
[dw , db]=learnwh(p, e, lr);
```

其中，lr 是学习率。当 lr 较大时，学习过程加快；但是当 lr 过大时，学习过程将变得不稳定，且误差会加大，因此学习率 lr 的取值是至关重要的。函数 maxlinlr 可以求出合适的学习率 lr

```
lr=maxlinlr(p)
lr=maxlinlr(p,'bias')
```

它们分别对应于没有阈值和有阈值时的最大学习速率。一旦学习率确定，线性神经网络可以利用 learnwh 函数进行训练

```
a=simulin(p, w, b);
e=t-a;
[dw, db]=learnwh(p, e, lr);
w=w + dw;
b=b + db;
```

1.3.7 线性神经网络的训练

函数 trainwh 可以完成线性神经网络的训练，它包含：

(1) 调用函数 simulin 进行仿真计算；

(2) 求得网络误差；

(3) 调用函数 learnwh 求得网络新的权值和阈值；

(4) 直到网络误差达到要求为止。

1.3.8 自适应网络

图 1.7 表示一个自适应网络。自适应网络可通过 adaptwh 函数来训练

```
[a, e, w, b]=adaptwh(w, b, p, t, lr);
```

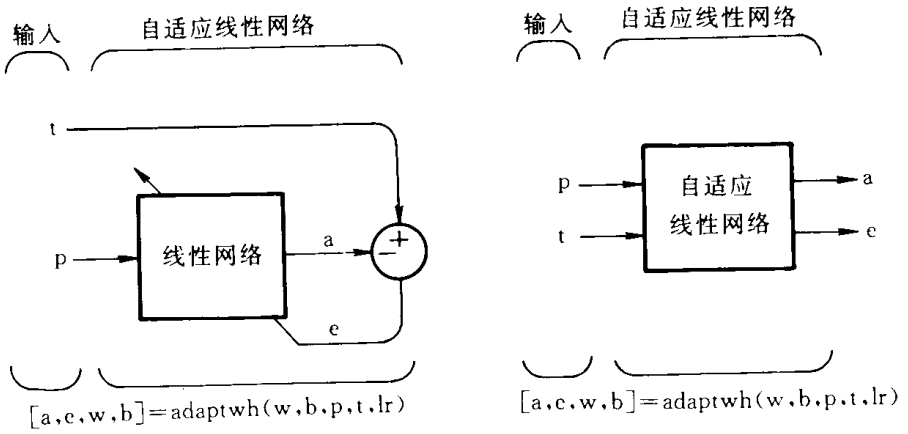


图 1.7 自适应网络

1.4 BP 网 络

Minsky 和 Papert 的论点曾使许多人对神经网络的研究失去了信心，但仍有许多学者坚持这一方面的研究。Rumelhart、McClelland 和他们的同事洞察到神经网络信息处理的重要性，于 1982 年成立了一个 PDP 小组，研究并行分布信息处理方法，探索人类认知的微

结构。1985 年发展了 BP 网络学习算法，实现了 Minsky 的多层网络设想。

由于感知器神经网络中神经元的变换函数采用符号函数，其输出为二值量，因此它主要用于模式分类。BP 网络也是一种多层前馈神经网络，其神经元的变换函数是 S 型函数，因此输出量为 0 到 1 之间的连续量，它可以实现从输入到输出的任意的非线性映射。由于权值的调整采用反向传播(Back Propagation)的学习算法，因此也常称其为 BP 网络。在确定了 BP 网络的结构后，利用输入输出样本集对其进行训练，也即对网络的权值和阈值进行学习和调整，以使网络实现给定的输入输出映射关系。经过训练的 BP 网络，对于不是样本集中的输入也能给出合适的输出，这种性质称为泛化(generalization)功能。从函数拟合的角度看，这说明 BP 网络具有插值功能。

1.4.1 重要的 BP 网络函数

函数 initff 和 simuff 可以用来初始化和仿真不超过三层的前向网络。函数 trainbp、trainbpx、trainlm 可用来训练 BP。其中 trainlm 的训练速度最快，但它需要更大的存储空间；trainbpx 的训练速度次之；trainbp 最慢。

1.4.2 BP 神经元模型

图 1.8 给出一个基本的 BP 神经元，它具有 R 个输入，每个输入都通过一个适当的权值 w 与下一层相连，网络输出可表示成

$$a=f(w*p,b)$$

BP 网络中隐层神经元的变换函数通常是 log-sigmoid 型函数 logsig，也可以采用 tan-sigmoid 型函数 tansig，在某些特定情况下，还可能采用纯线性函数 purelin。

如果 BP 网络的最后一层是 sigmoid 型神经元，那么整个网络的输出就限制在一个较小的范围内；如果 BP 网络的最后一层是 purelin 型线性神经元，那么整个网络的输出可以取任意值。

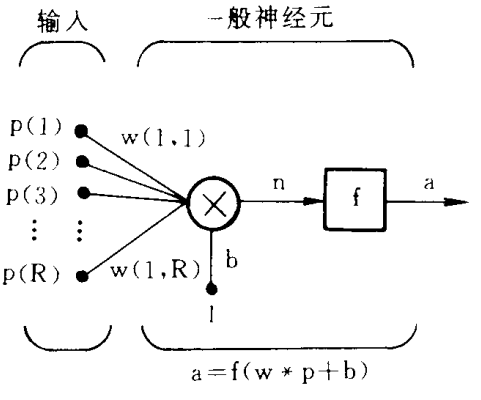


图 1.8 BP 神经元

BP 网络所采用的传递函数均是可微的单调递增函数。在 BP 网络的训练过程中，计算函数 logsig、tansig、purelin 的导数是非常重要的，神经网络工具箱提供了这些求导函数：deltalog、deltatan、deltalin。

这些函数都是在设计 BP 网络时要经常用到的。如果用户在实际的设计和应用中需要用到其它的函数，也可以进行自定义，MATLAB 系统提供了丰富的扩展功能。

1.4.3 BP 网络的网络结构

典型的 BP 网络的结构如图 1.9 所示。

BP 网络通常有一个或多个隐层，隐层中的神经元均采用 sigmoid 型变换函数，输出层的神经元采用纯线性变换函数。图 1.10 描述了一个具有一个隐层的 BP 网络。

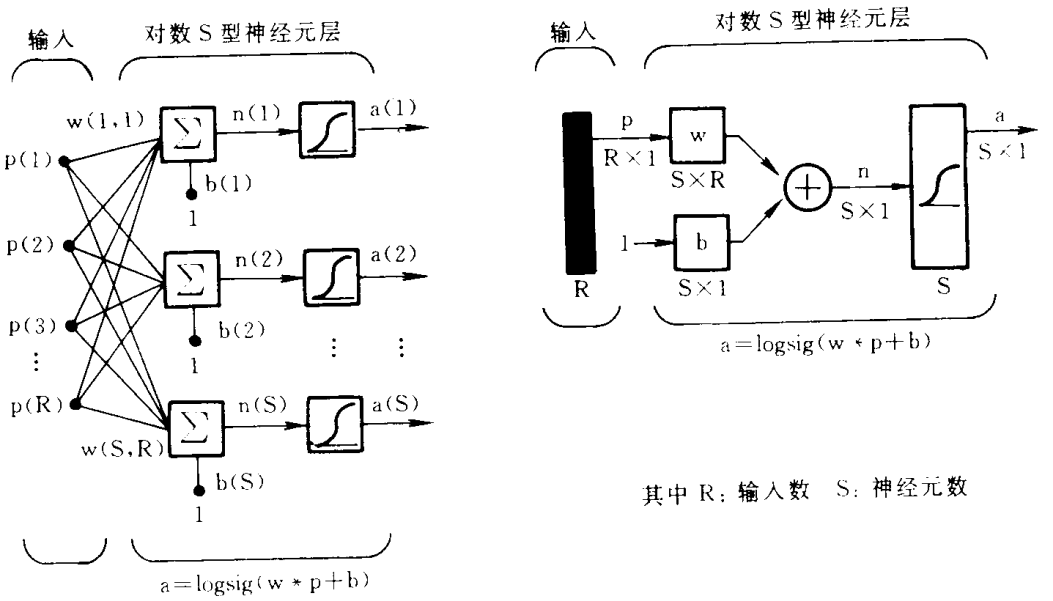


图 1.9 BP 网络结构

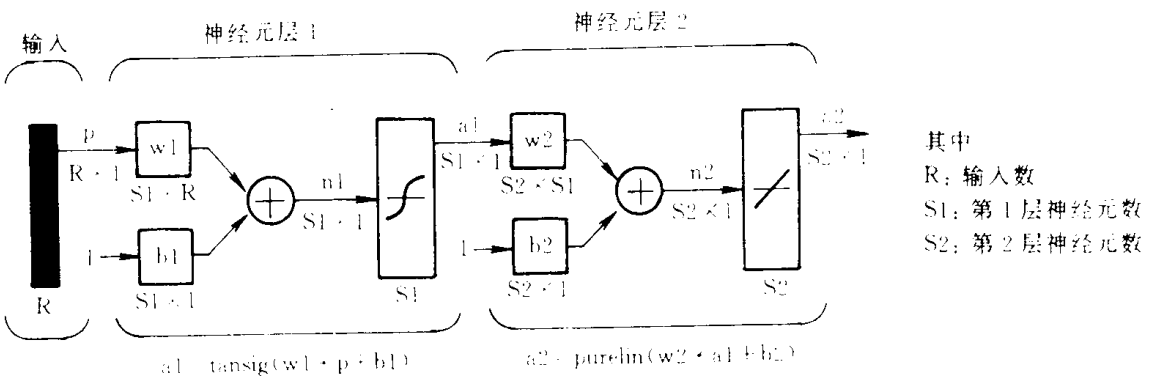


图 1.10 具有一个隐层的 BP 网络

图 1.10 的网络可以用来逼近非线性函数。在设计中，可以利用函数 simuff 对 BP 网络进行仿真

```
[a1 , a2]=simuff(p, w1,b1, 'tansig', w2, b2, 'purelin');
```

函数 simuff 可以对单层、两层、三层的 BP 网络进行仿真

```
a=simuff(p, w, b, 'tansig');
[a1, a2]=simuff(p, w1, b1, 'logsig', w2, b2, 'purelin');
[a1, a2, a3]=simuff(p, w1, b1, 'tansig', w2, b2, 'logsig', w3, b3,'purelin');
```

函数 simuff 也可以只计算输出层的输出值

```
a2=simuff(p, w1, b1, 'logsig', w2, b2, 'purelin');
```