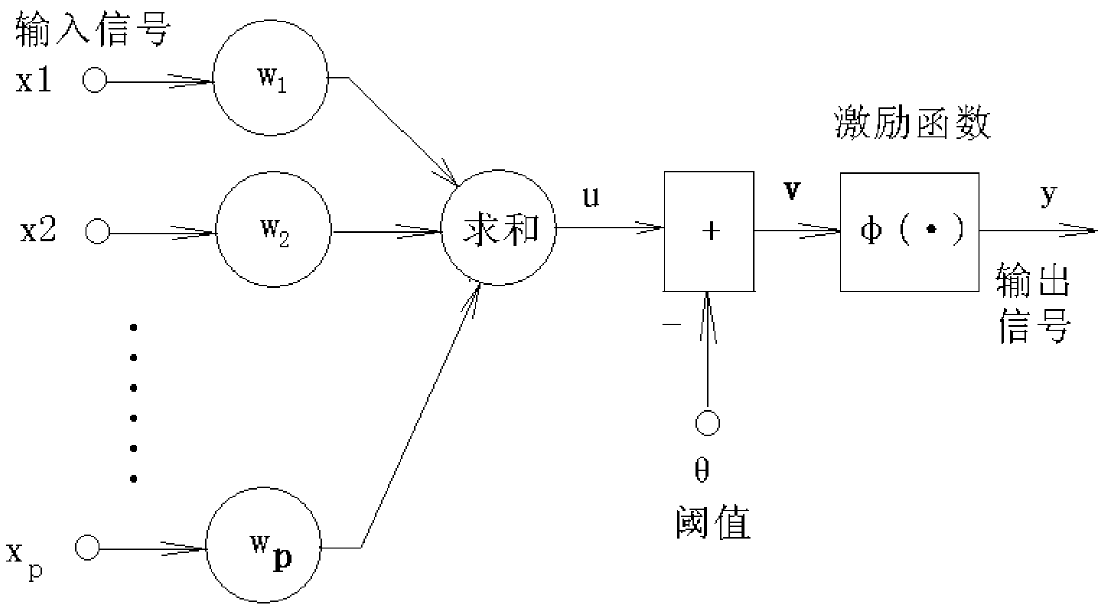
复习：人工神经元模型

1. 单个神经元的结构和数学模型



$$y=\varphi(v)=\varphi(u-\theta)=\varphi(\sum_{j=1}^{p}w_jx_j-\theta)$$

$$=\varphi(\sum_{j=0}^{p}w_jx_j)$$

最后一种写法为“扩维”表示法，主要是为了书写简便。$x_0=-1$，$w_0=\theta$ 。

2. 多个神经元组成神经网络时，各个神经元的结构和数学模型

● 问题的提出：组成网络时，神经元的结构需要简化。

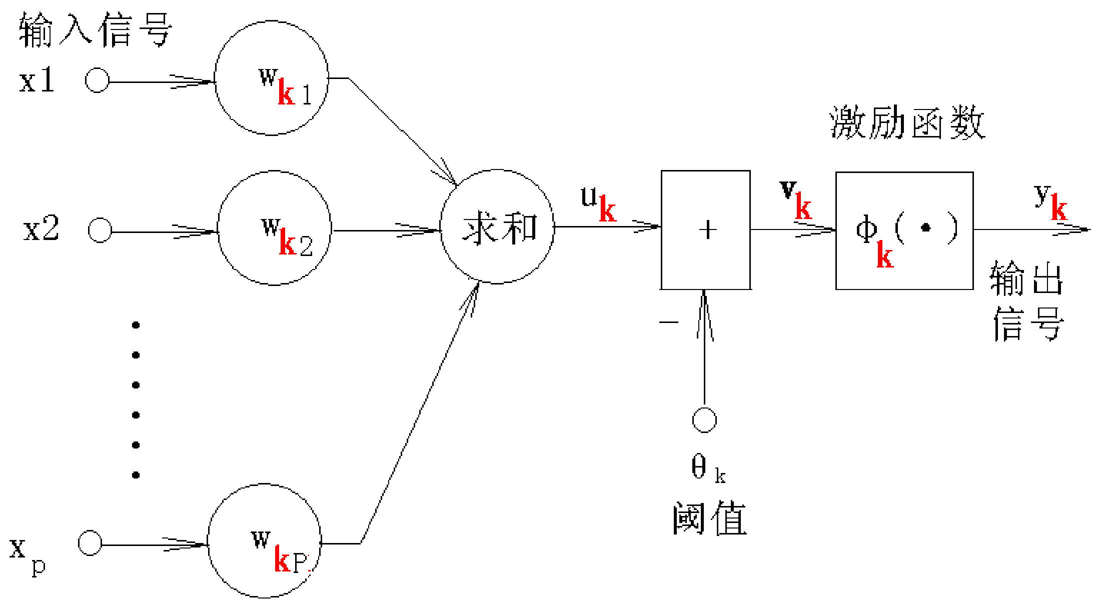


图 神经网络中单个神经元的信号表示

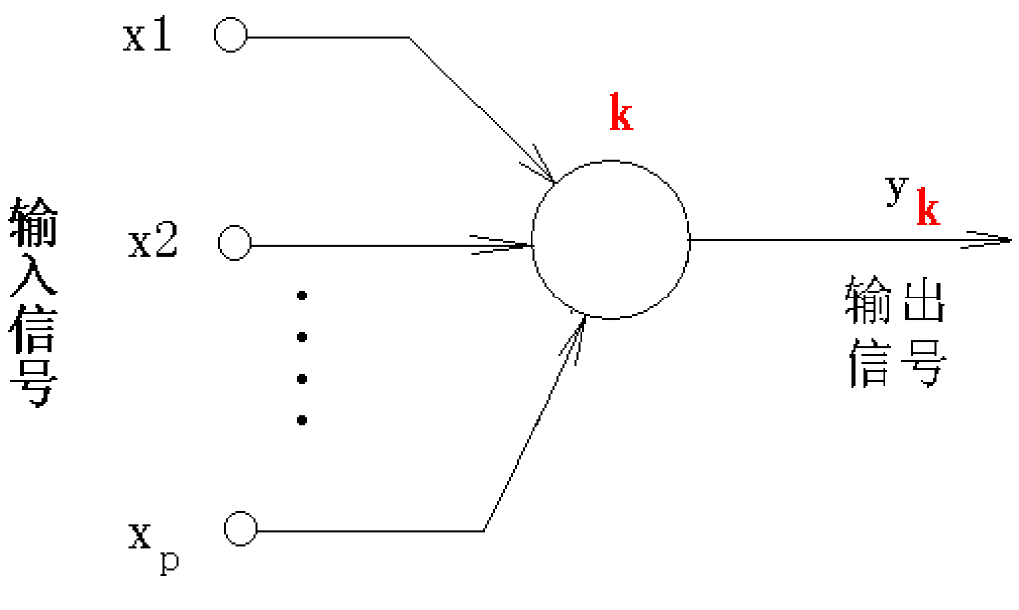


图 神经网络中单个神经元的简化表示

神经网络中单个神经元的数学模型：

$$v_k = \sum_{j=1}^{p} w_{kj} x_{kj} - \theta_k$$

$$y_k = \varphi(v_k)$$

- 利用神经元的简化表示法，构成神经网络：

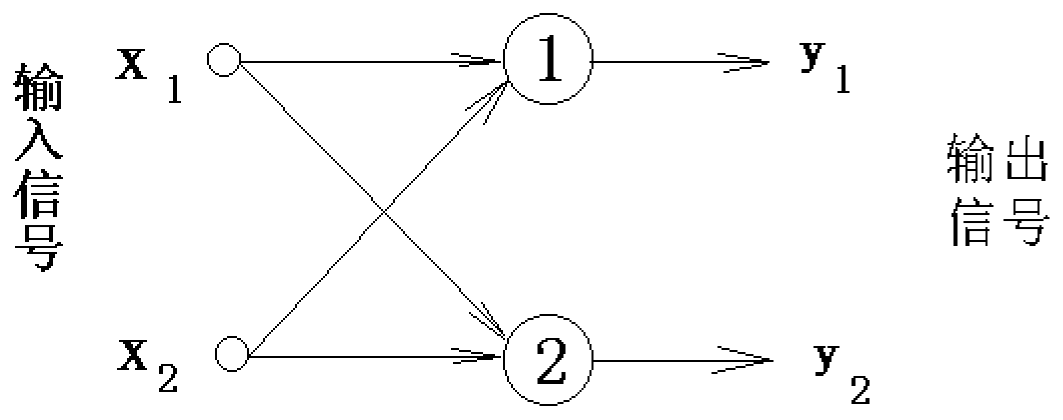


图　多输入、多输出的神经网络

神经经网络中单个神经元的数学模型：

$$v_k = \sum_{j=1}^{p} w_{kj} x_{kj} - \theta_k$$

$$y_k = \varphi(v_k)$$

第二章　前馈网络

§2.1　感知器（线性阈值单元）

- perceptron 是模拟人的视觉，接受环境信息，并由神经冲突进行信息传递的神经网络。
- 分类：单层、多层。
- 特点：具有学习能力。

一．单层感知器

由美国学者 F. Rosenblatt 建立于 1957 年。

1.　结构

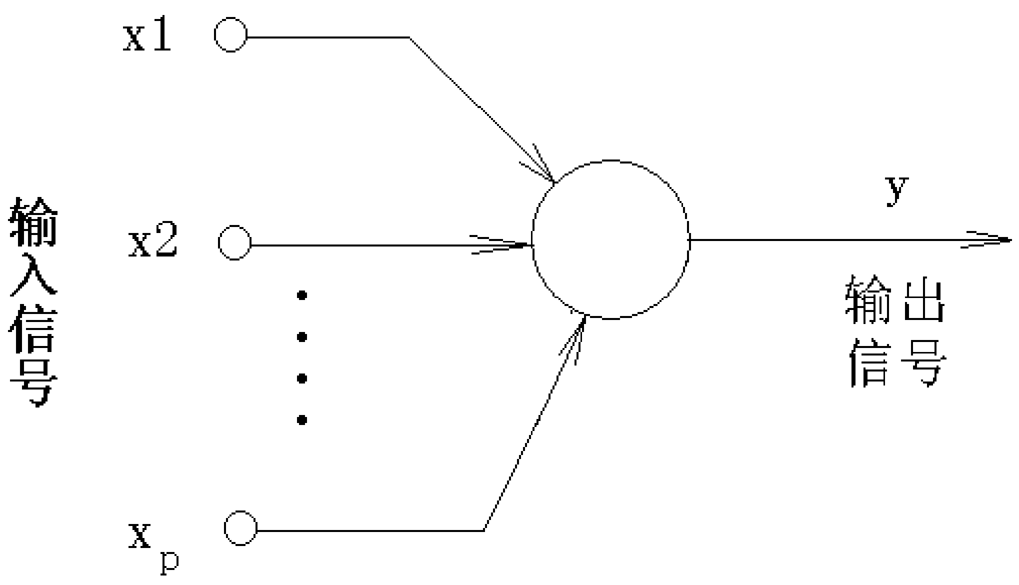


图 2.1 感知器的结构

2. 特点：

- 一个神经元，用 M-P 模型（阈值函数）

$$v=\sum_{j=1}^{p}w_jx_j-\theta=\mathbf{w}^T\mathbf{x}-\theta$$

$$y=\varphi(v)=\mathrm{sgn}(v)\begin{cases}+1, & v\ge 0\\ -1, & v<0\end{cases}$$

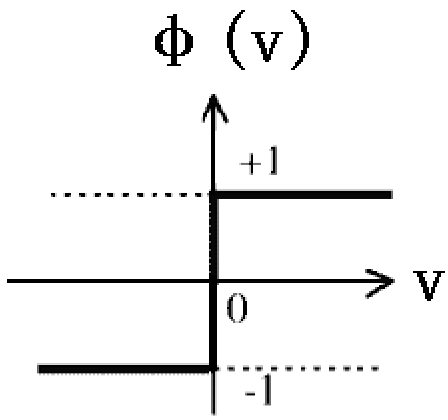


图 2.2　阈值作用函数

- 一组可调权值：

$$w_i\in R,\quad \mathbf{w}^T=(w_1,w_2,\cdots w_p)$$

- 多输入（p个）、单输出：

$$x_i\in R,\quad x^T=(x_1,x_2,\cdots x_p)$$

$$y\in\{-1,\quad +1\}$$

二. 线性可分集合的概念：用 NN 实现“与非门”

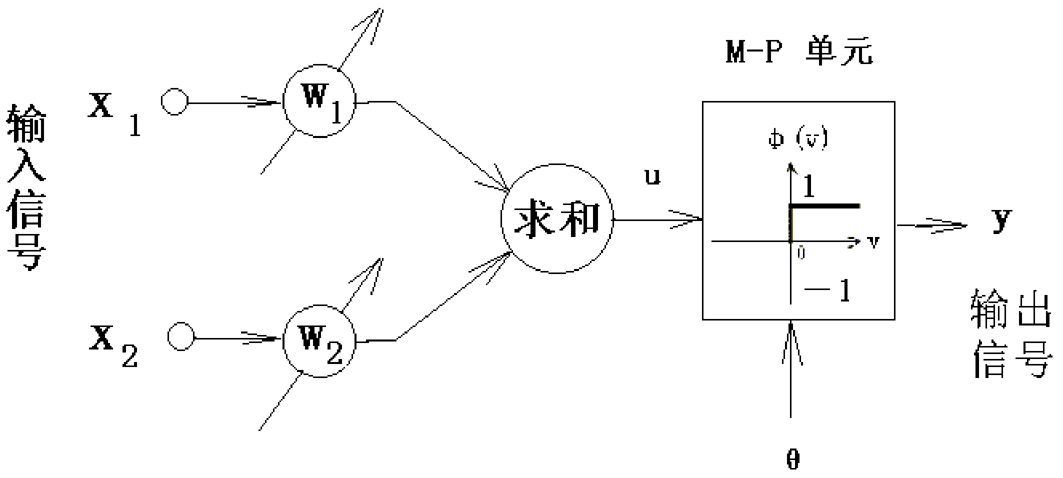


图 2.3 单层感知器神经网络

- 数学模型：

$$y=\varphi(w_1x_1+w_2x_2-\theta)$$
$$=\begin{cases}1, & w_1x_1+w_2x_2-\theta\geq 0\\ 0, & w_1x_1+w_2x_2-\theta<0\end{cases}$$

- y 的不同取值的分界线：

$$w_1x_1+w_2x_2-\theta=0\quad\Rightarrow x_2=\frac{\theta}{w_2}-\frac{w_1}{w_2}x_1$$

● 特例：$w_1=-1, w_2=-1, \quad \theta=-\frac{3}{2}$

$$y=\varphi(-x_1-x_2+\frac{3}{2})$$

把 X_1，X_2 带入上式，可得表中 y 的数值：

表 2－1 “与非门”逻辑表

X_1	X_2	y
0	0	1
1	0	1
0	1	1
1	1	0

思考题：对上述单层感知器，其权值、阈值不取上述数值时，也能实现“与非门”吗？

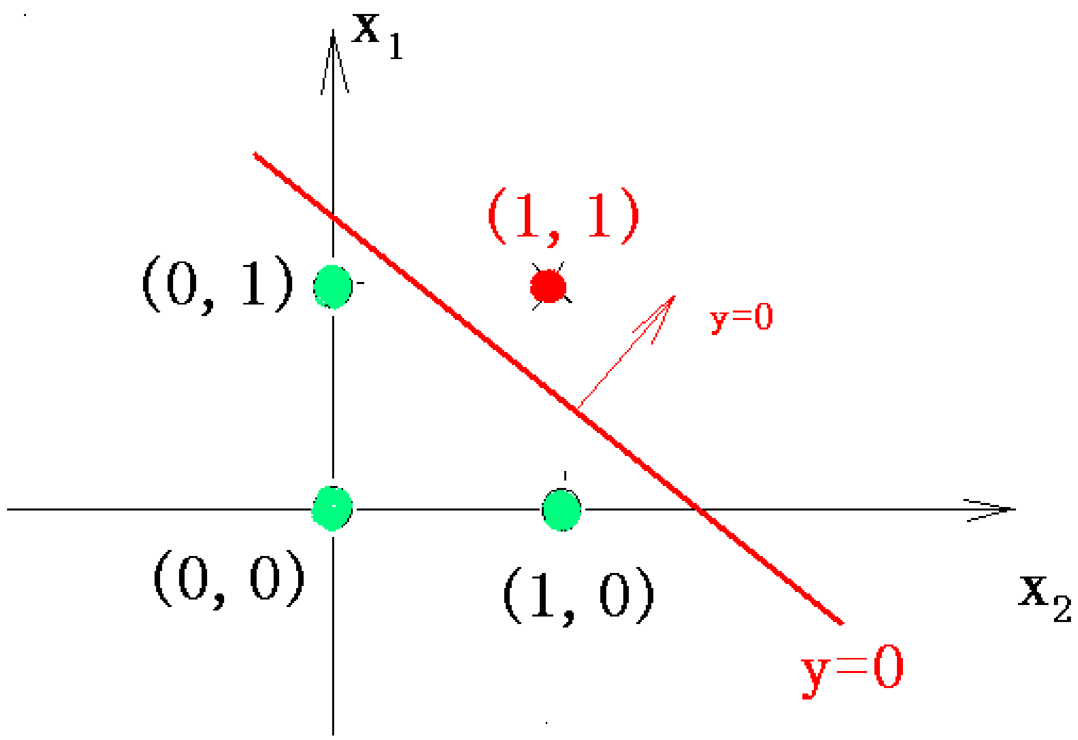


图 2.4　单层感知器的数据分类（与非门）

● 讨论：

（1）线性可分集合的概念：

对于输入信号（样本）是两维的情况：能用一条直线将两类不同的模式分开的集合，称为～

对于输入信号（样本）是高维的情况：能用一个超平面将两类不同的模式分开的集合，称为～。

（2）线性可分性：对于线性可分集合，一定存在一组权值和一个阈值，使得单层感知器能将两类不同的样本区分开来。即能解决“与非门”问

题。

（3）单层感知器的局限性：不能解决线性不可分集合的分类问题。不能实现“异或（XOR）”。

表 2—2　“异或（XOR）”逻辑表

X_1	X_2	y
0	0	0
0	1	1
1	0	1
1	1	0

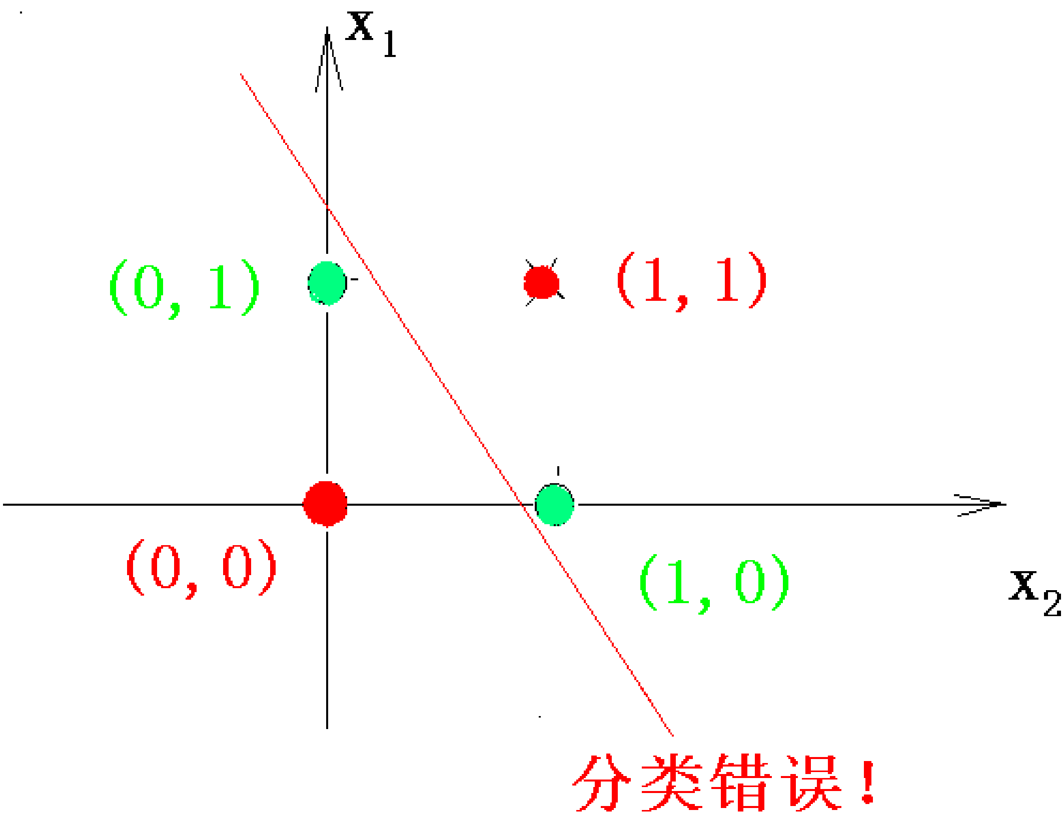


图 2.5　线性不可分集合的例子

（4）问题：线性不可分集合及其分类。

“多层感知器”的分类能力。

思考题：

1．对于线性可分集合，“分类直线”（权值和阈值）是唯一的吗？

§2.1 单层感知器的学习算法——机器自动调整权值、阈值的过程

（1）设置权值和阈值的初值$w_j(0)$

$(j=0,1,2,\cdots\cdot p)$ 为较小的随机非零值。

（2）给定输入/输出样本对，即：导师信号

$x_i(t)/d_i(t)$，$(i=0,1,2,\cdots\cdot L)$：

例：由 表2—2 “异或门”

$$x_i(t)/d_i(t)=x_i(1)/d_i(1)$$

$$=\begin{bmatrix} x_{1,1}(1) \\ x_{2,1}(1) \end{bmatrix} \Big/ d_1(1)=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \Big/ 0$$

$$x_2(2)/d_2(2)=\begin{bmatrix} 0 \\ 1 \end{bmatrix} \Big/ 1 \quad , \quad \cdot \ \cdot \ \cdot \ \cdot \ \cdot$$

$$x_L(L)/d_L(L)=\begin{bmatrix} 1 \\ 1 \end{bmatrix} \Big/ 0$$

注意：p， L 的区别：

$j,\ p$——输入信号的第j维，共p 维：

$$x_{j\,i}=\begin{bmatrix} x_{1\,i} \\ x_{p\,i} \end{bmatrix};$$

i, L　　——第 i 个样本，共 L 个样本$(L=4)$

(t) ——第 t 次迭代运算（修改）;

$x_i(t)$　取单下标时表示输入向量，未区分输入的第几个分量;

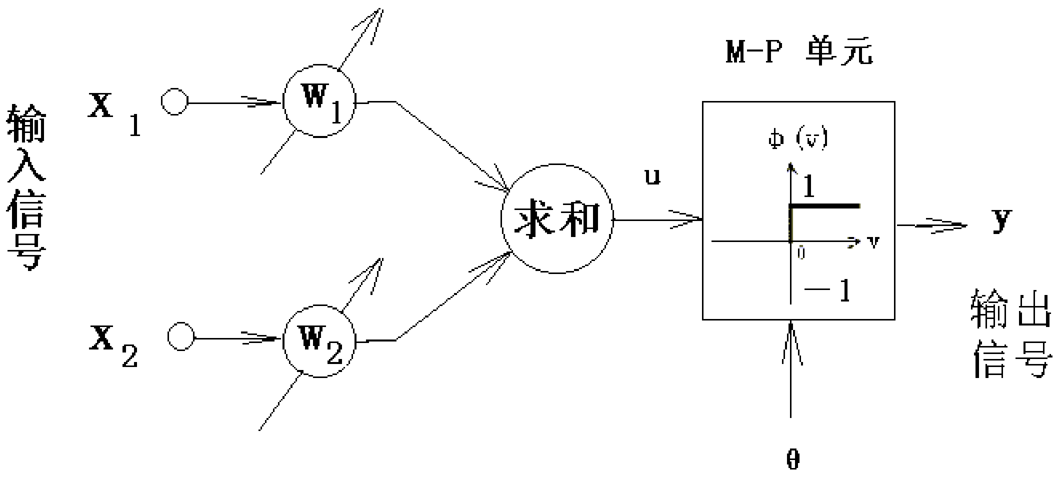
$d_i(t)$　第 i个样本对应的期望输出。

（3）求感知器的输出（第i 个样本输入时）：

$$y_i(t)=\varphi(\sum_{j=0}^{p} w_j(t)x_{j\,i})$$

特例　$y \ =\varphi(w_1x_1+w_2x_2-\theta)$

注：只有一个样本时，无需区分第几个样本 i。



$$y_i(t)=\varphi\left(\sum_{j=0}^{p} w_j(t)x_{ji}\right)$$

j， i， t分别是：第 j个输入分量，第i 个样本对（与非门有 4 个），第 t 次运算（迭代）。

样本序号i	X_1	X_2	y
1	0	0	1
2	1	0	1
3	0	1	1
4	1	1	0

（4）权值调整：与第 j 个输入分量对应的权值：

$$w_j(t+1)=w_j(t)+\eta(d_i(t)-y_i(t))x_{ji}$$

$w_j(t+1)$　修正后（新的）的权值；

$w_j(t)$　修正前（旧的）的权值；

$\eta\,(0<\eta<1)$ 学习率，控制权值修正速度；

（5）若 $|y_i(t)-d_i(t)|<\varepsilon$（精度要求），则学习结束，否则返回步骤（3），输入另一对样本，重新学习。

讨论：

（1）信息的分布式存储：学习结束后的网络，将样本模式信息，以权值（阈值）的形式分布记忆（存储）于网络中。

（2）NN 的容错能力：考虑下述 4 入 4 出网络：

$$w=\begin{bmatrix}-0.25 & 0.25 & 0.25 & -0.25\\ -0.25 & 0.25 & 0.25 & -0.25\\ 0.25 & -0.25 & -0.25 & 0.25\\ 0.25 & -0.25 & -0.25 & 0.25\end{bmatrix}$$

$$\theta = 0, \quad v = \sum_{j=1}^{4} w_j x_j$$

$$y = \varphi(v) = \varphi(wx) = \begin{cases} +1 & v \geq 0 \\ -1 & v < 0 \end{cases}$$

设下式的x为传感器的正确输入，则得

$$x = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \Rightarrow y = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

若x的第三个分量为 0（故障），仍得 y 值不变：

$$x = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \quad \Rightarrow y = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

说明 NN 有较强的容错能力。

（2007-9-24 第一次课 #）

§2.2　多层感知器

1．多层感知器的网络结构

在输入、输出层间加一层（或多层）隐含层（Hidden Unit），可构成～，也称多层前馈神经网络。

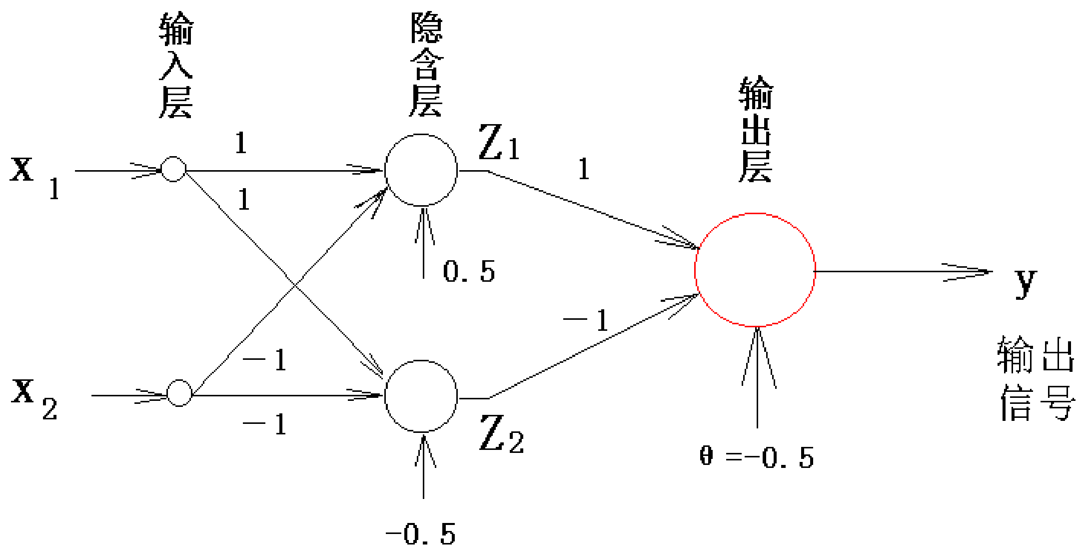


图 2.5　多层（三层）感知器神经网络

2．多层感知器的特点和实例

- 作用函数：隐含层、输出层相同：

$$\varphi(v)=\begin{cases}1 & v\geq 0\\ 0 & v<0\end{cases}$$

带入图 2.5 的数据可得：

$$\begin{cases} z_1 = \varphi(1 \times x_1 + (-1) \times x_2 - 0.5) \\ z_2 = \varphi(1 \times x_1 + (-1) \times x_2 + 0.5) \end{cases}$$

$$y = \varphi(z_1 - z_2 + 0.5)$$

- 带入表 2—3 的输入数据，得 Z_1，Z_2，y：

表 2—3　“异或（XOR）”逻辑表

	X_1	X_2		Z_1	Z_2	y
A	0	0		0	1	0
B	0	1		0	0	1
C	1	0		1	1	1
D	1	1		0	1	0

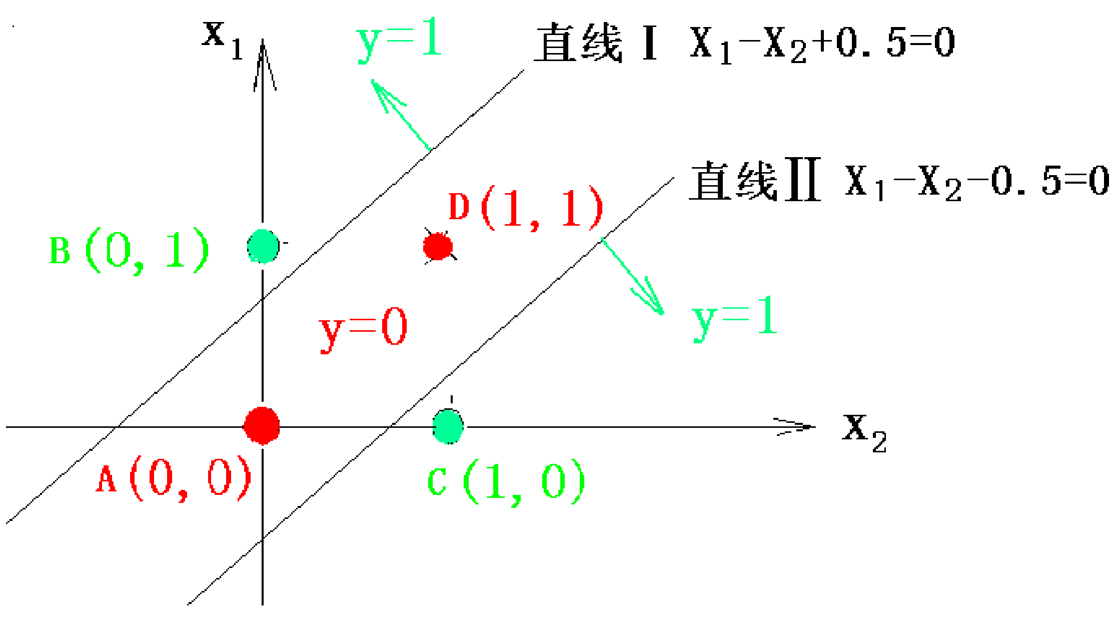


图 2.6 用三层感知器将样本空间分成两类

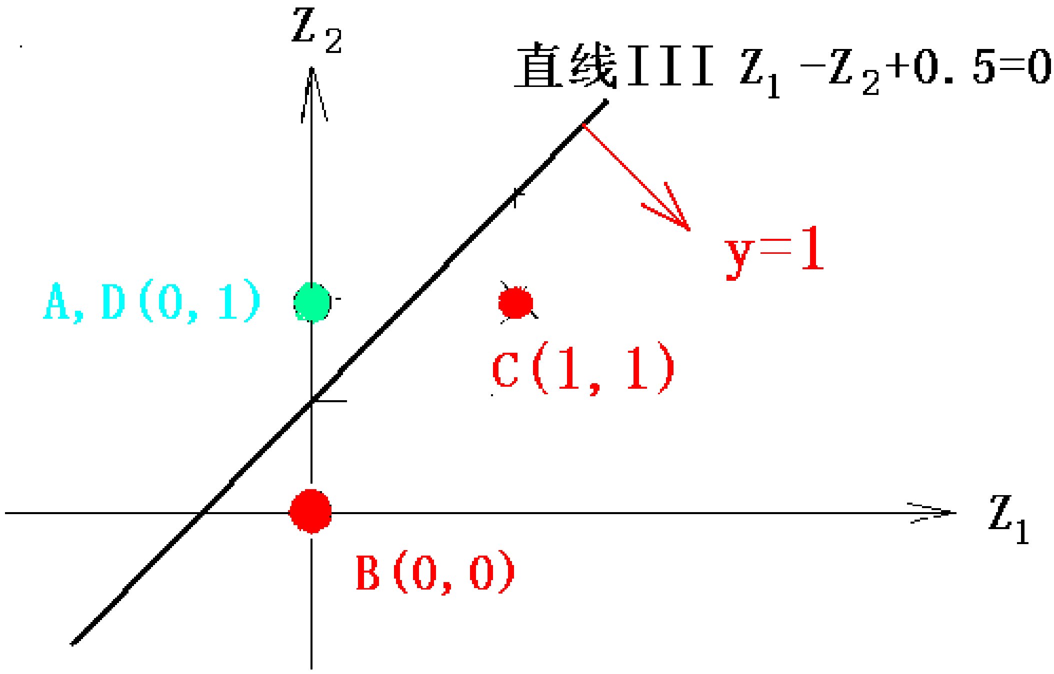


图 2.6 隐含层到输出层的映射

（A，D 两个样本映射到隐含层时变成一个点，将线性不可分集合变成线性可分集合）

结论

【推论】用三层感知器网络可识别任一凸多边形或无界的凸区域。更多层的感知器网络可识别更复杂的图形。

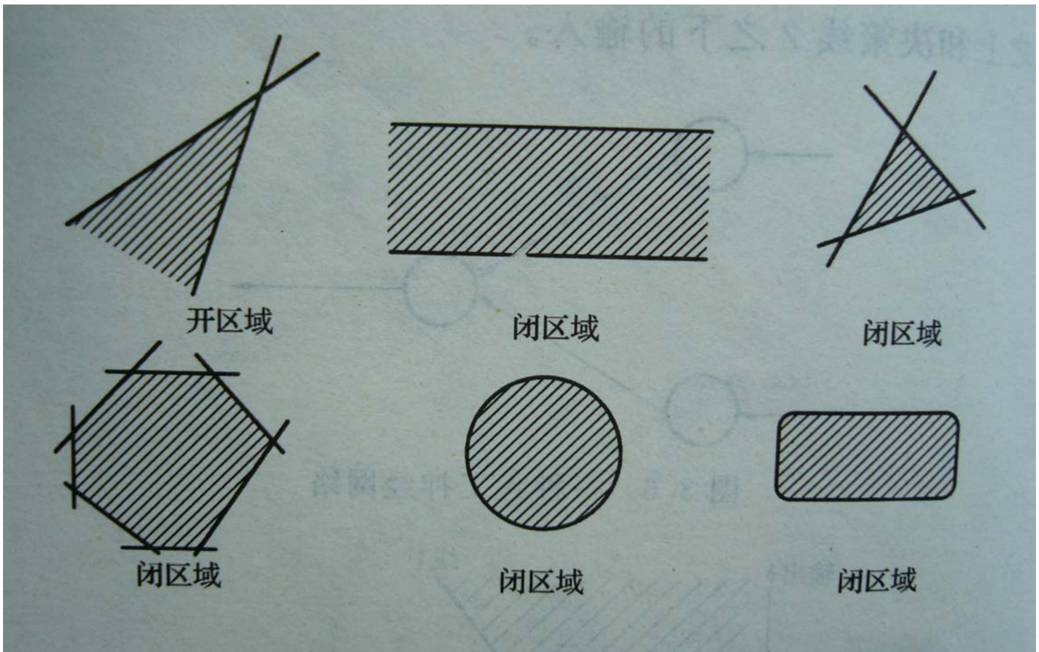


图 3.8 各种凸区域示意图

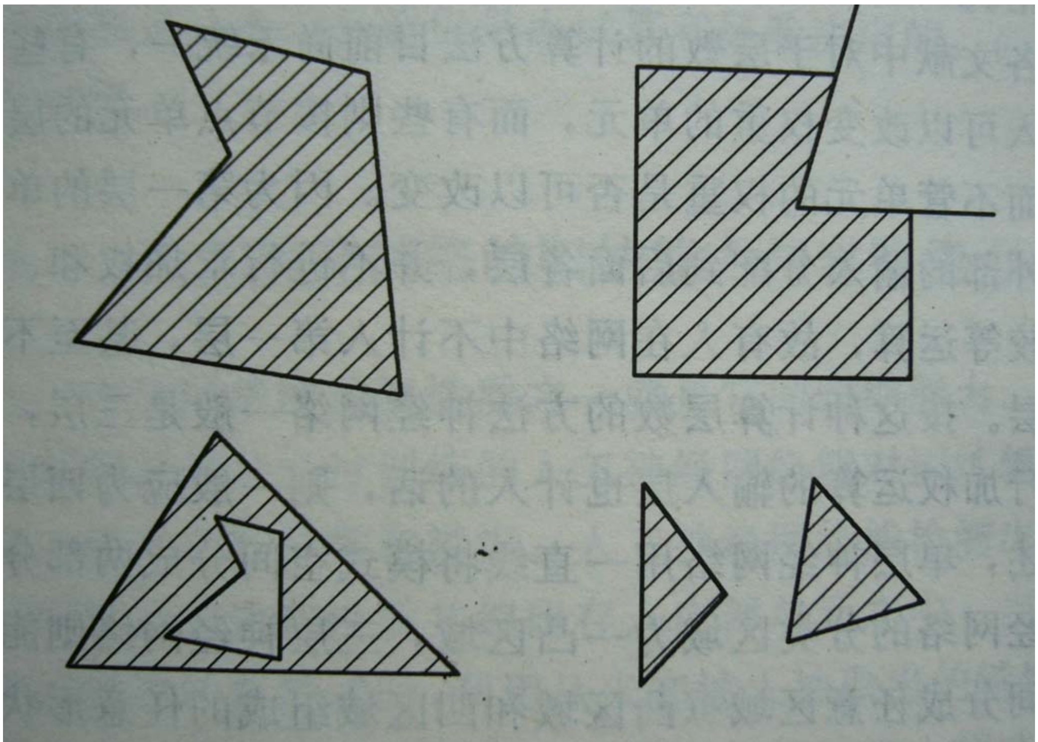


图 3.9 由凸区域组合得到任意区域

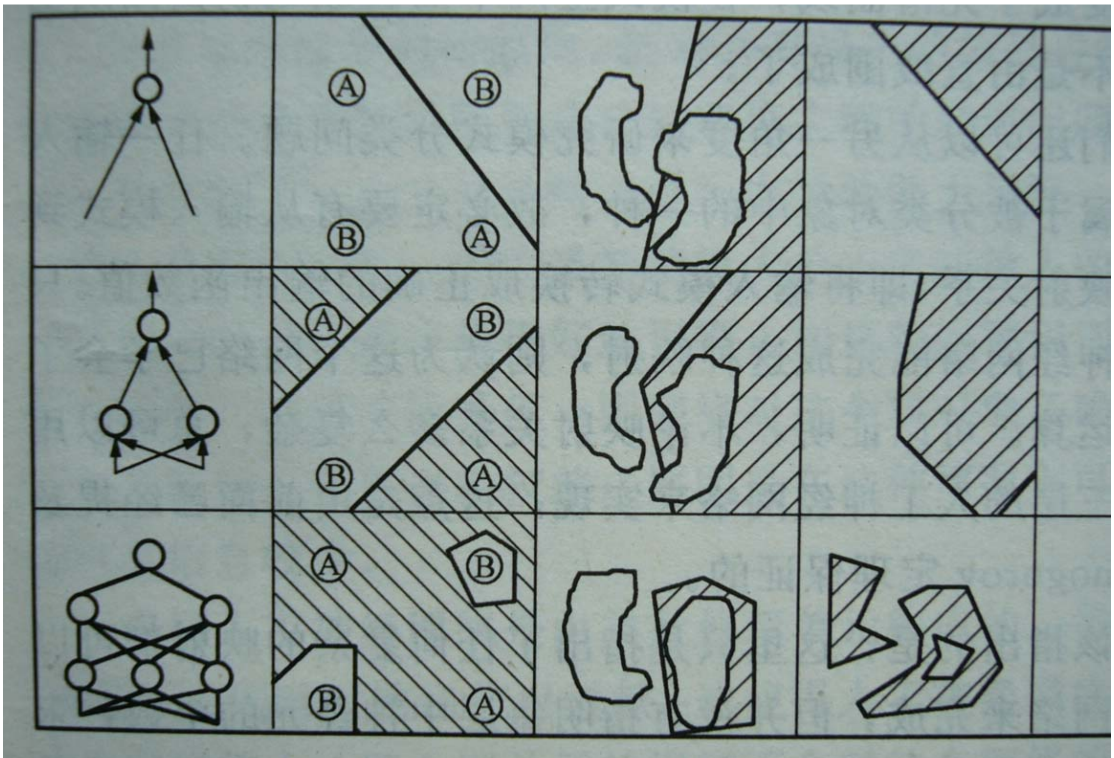
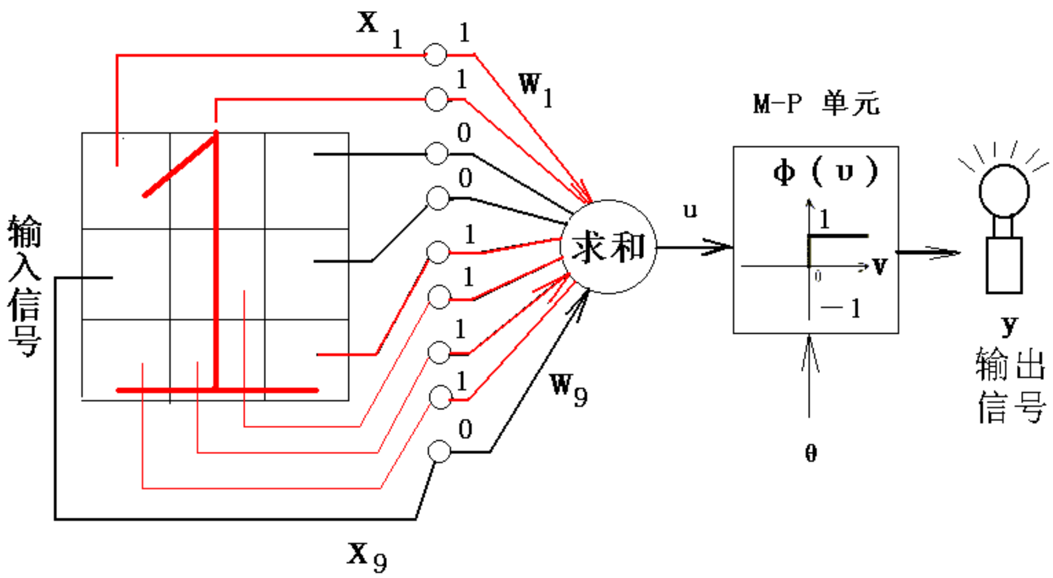


图 3.10　三种神经网络及其可能的决策区域

【定理 1】　若隐层节点可任意设置，则用三层阈值节点的网络，可以实现任意的二值逻辑。

【定理 2】　若隐层节点可任意设置，则用三层 S 型非线性作用函数节点的网络，可以一致逼近紧集上的连续函数或按 L_2 范数逼近紧集上的平方可积函数。

3. 感知器网络应用实例



● 说明：

输入：9 个，笔划通过小块时，输出为“1”，否则为“0”；

输出：1 个，当 $\sum x_j w_j$ 超过阈值时灯亮，

否则灯灭；

目的：奇偶数识别。数字卡为奇数时，灯亮（输出 1）。将 0～9 都输入一遍，结果正确，什么都不变，否则，修正权值，减小误差。

思考题：

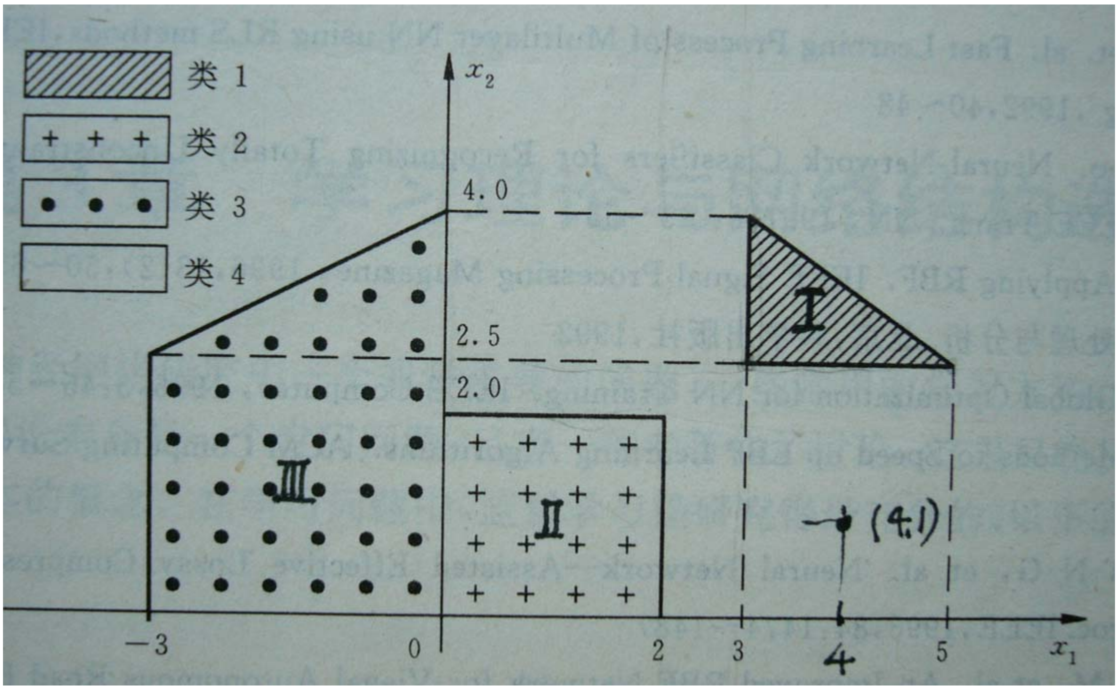
1. 对于线性可分集合，“分类直线”（权值和阈值）是唯一的吗？
2. 验证 表 2—3 的“异或（XOR）”逻辑，即：由输入 X_1，X_2，验证 Z_1，Z_2，y 的取值。
3. 试述神经网络为什么具有“容错能力”？。

【作 业】

（1）奇偶数字卡片识别利用单层感知器学习算法，求解上述“奇偶辨认”例题（即：让机器自动求解权值和阈值）。

（2）用两种不同的网络结构，即单个神经元、三层神经网络来进行上述“数字卡片识别”，并比较其识别效果的异同点。

（3）P28 习题 2.4



（4）算法的理论推导（略）。