第三章 前馈神经网络

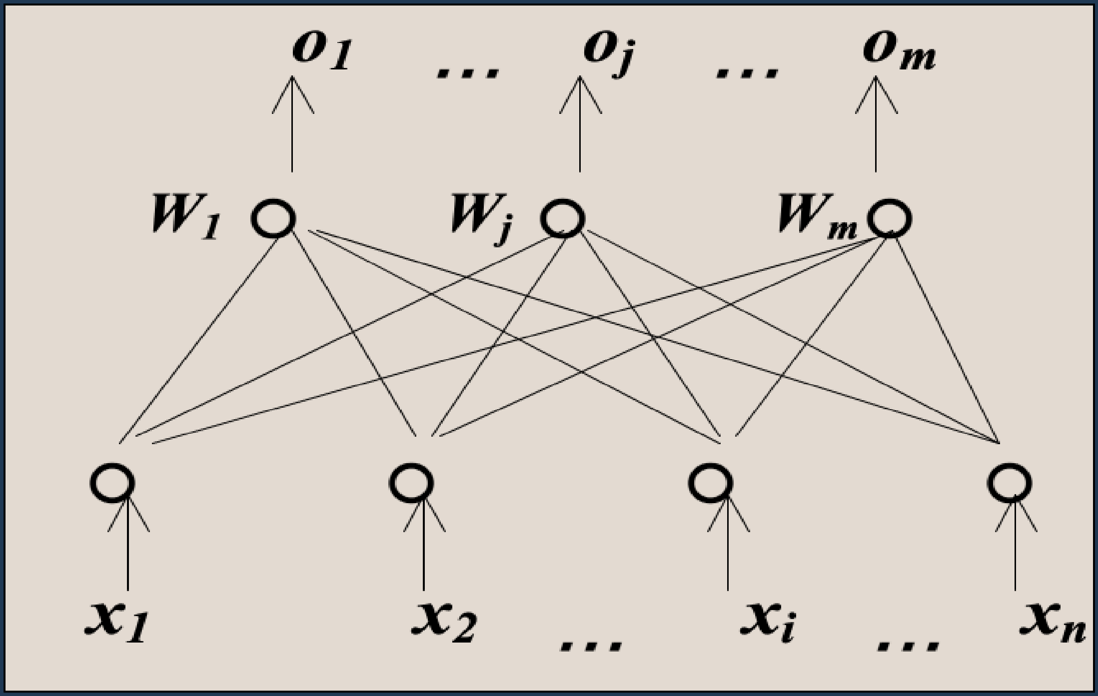
3.1 单层感知器

3.2 多层感知器

3 前馈神经网络

3.1 单层感知器

3.1.1 感知器模型



3.1.1 感知器模型

$$\boldsymbol{X} = (x_1, x_2, \ldots x_i, \ldots, x_n)^T$$

$$\boldsymbol{O} = (o_1, o_2, \ldots o_i, \ldots, o_m)^T$$

$$\boldsymbol{W}_j = (w_{1j}, w_{2j}, \ldots w_{ij}, \ldots, w_{nj})^T$$

j=1,2,…,m

前馈神经网络：单层感知器

3.1.1 感知器模型

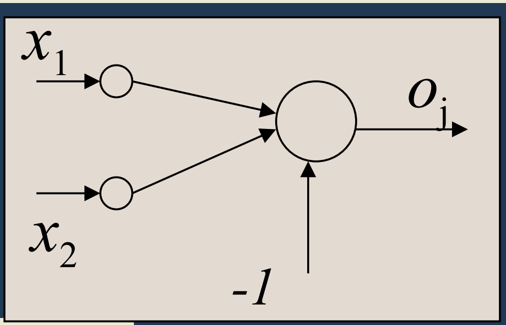
净输入：

$$net_j = \sum_{i=1}^{n} w_{ij} x_i \qquad (3.1)$$

输出：

$$o_j = sgn(net_j - T_j) = sgn\left(\sum_{i=0}^{n} w_{ij} x_i\right) = sgn(\boldsymbol{W}_j^T \boldsymbol{X}) \qquad (3.2)$$

3.1.2 感知器的功能



(1) 设输入向量 $X=(x_1, x_2)^T$

输出： $o_j = \begin{cases} 1 & w_{1j}x_1 + w_{2j}x_2 - T_j > 0 \\ -1 & w_{1j}x_1 + w_{2j}x_2 - T_j < 0 \end{cases}$

则由方程 $w_{1j}x_1+w_{2j}x_2-T_j=0$ **(3.3)**

确定了二维平面上的一条分界线。

（1）输入是二维

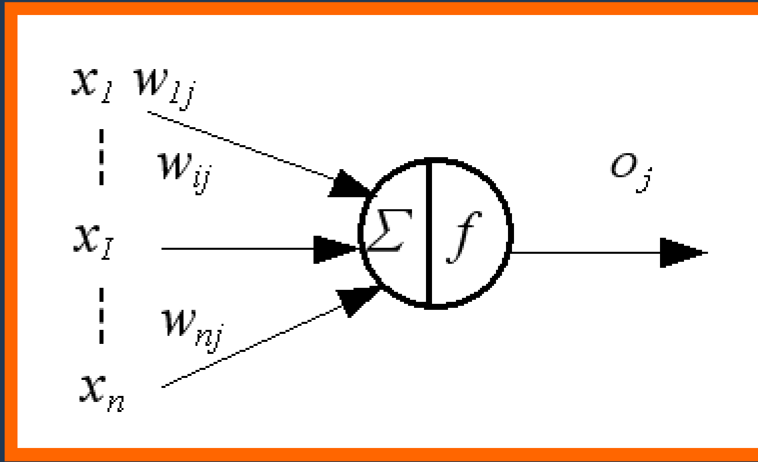
$$w_{1j}x_1+w_{2j}x_2-T_j=0$$

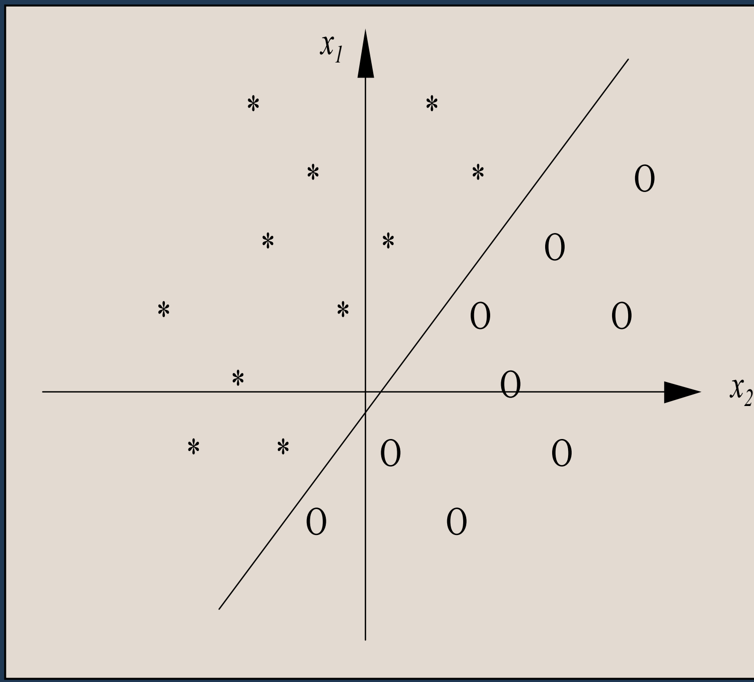
$$w_{1j}x_1=T_j-w_{2j}x_2$$

$$x_1=(T_j-w_{2j}x_2)/w_{1j}$$

$$=-(w_{2j}/w_{1j})x_2+T_j/w_{1j}$$

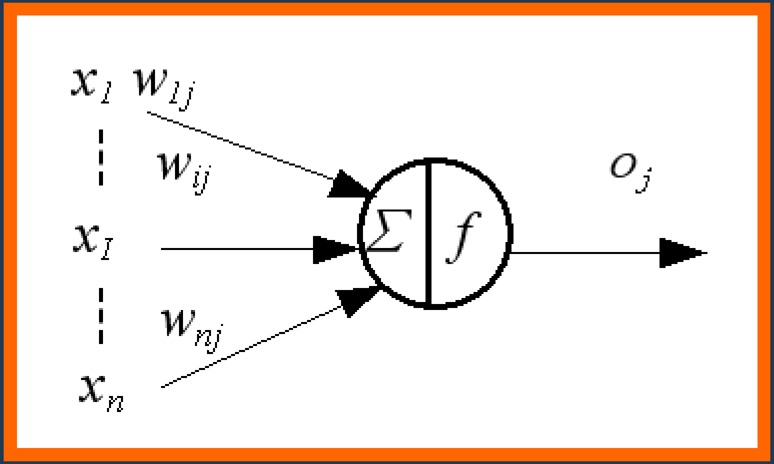
$$=a\,x_2+c$$

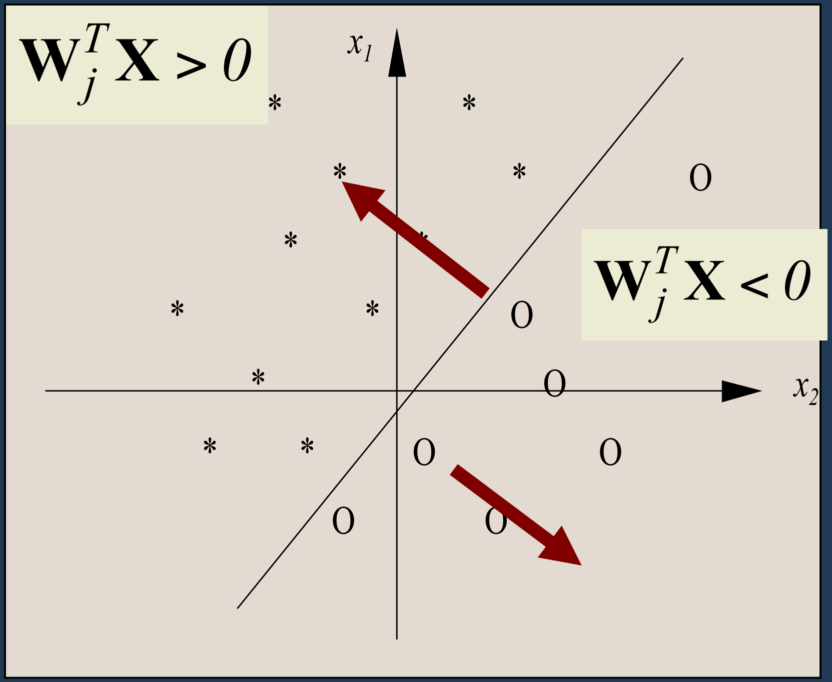




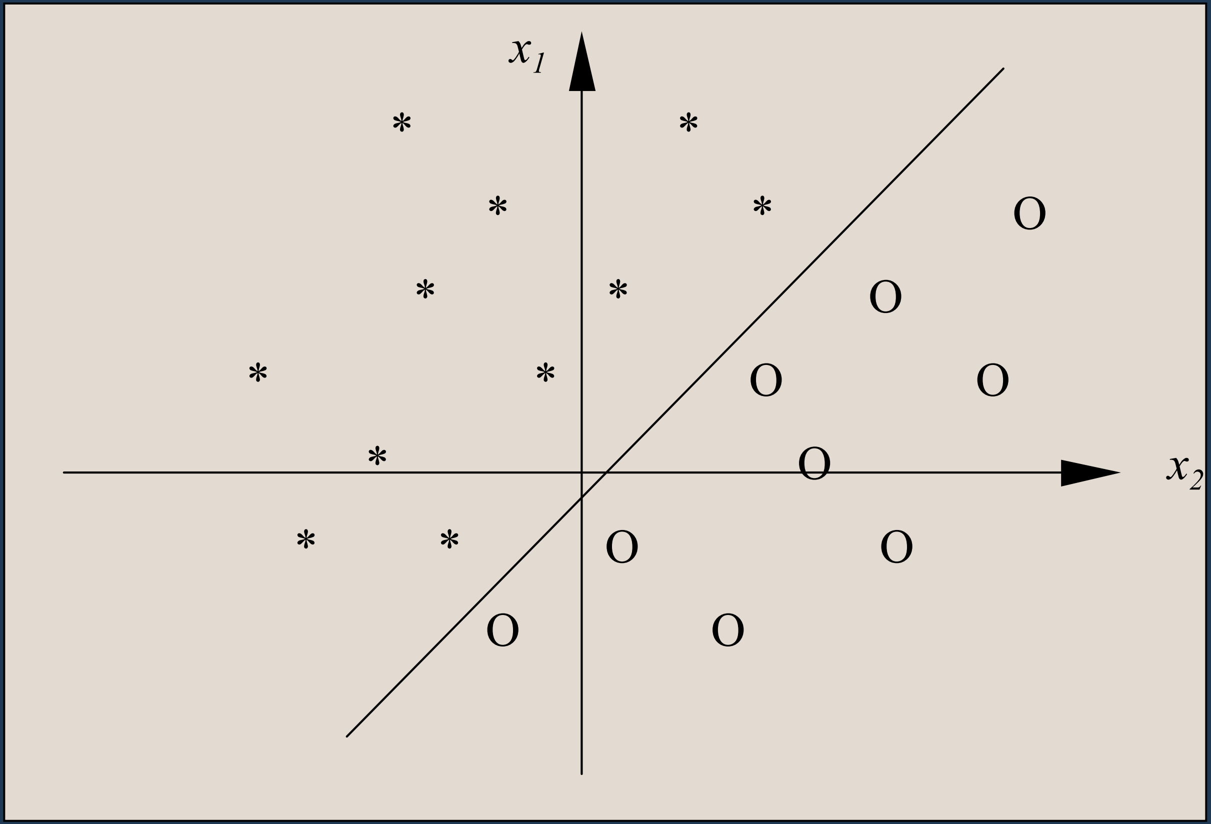
感知器的功能（二维）

$$\mathbf{W}_j^T\mathbf{X} = 0$$

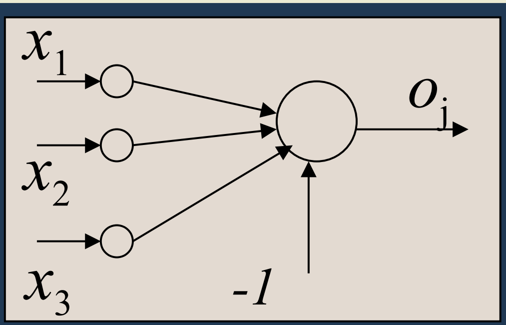




3.1.2 感知器的功能



3.1.2 感知器的功能



(2) 设输入向量 $X=(x1,x2,x_3)^T$

输出： $= \begin{cases} 1 & w_{1j}x_1 + w_{2j}x_2 + w_{3j}x_3 - T_j > 0 \\ -1 & w_{1j}x_1 + w_{2j}x_2 + w_{3j}x_3 - T_j < 0 \end{cases}$

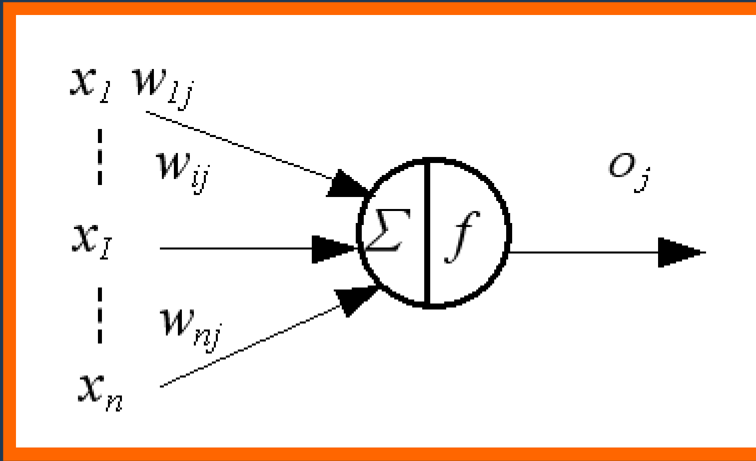
则由方程 $w_{ij}x_1+w_{2j}x_2+w_{3j}-T_j=0$ (3.4)

确定了三维空间上的一个分界平面。

（2）输入是三维

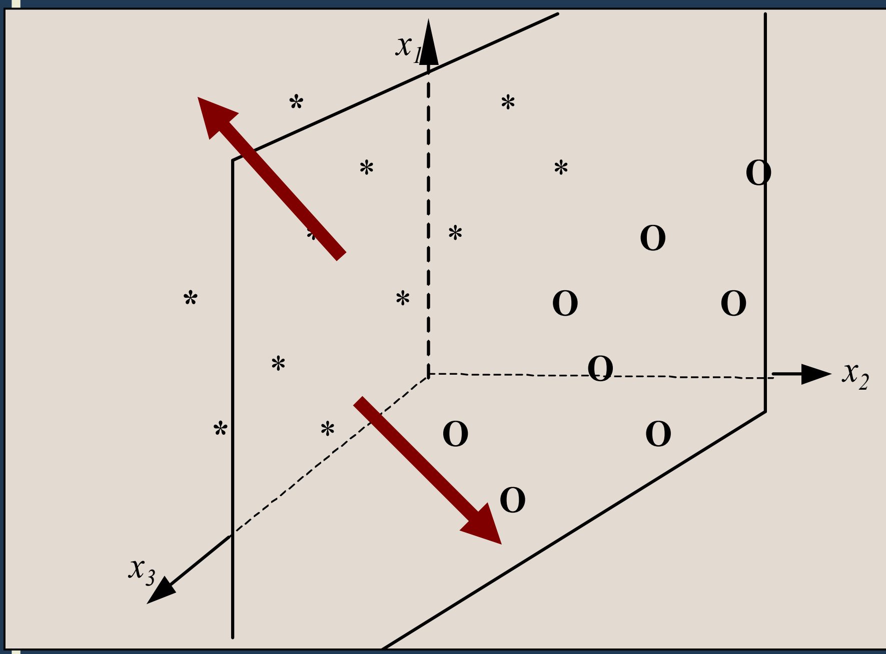
$$\mathbf{W}_j^T\mathbf{X} = 0$$

是什么？

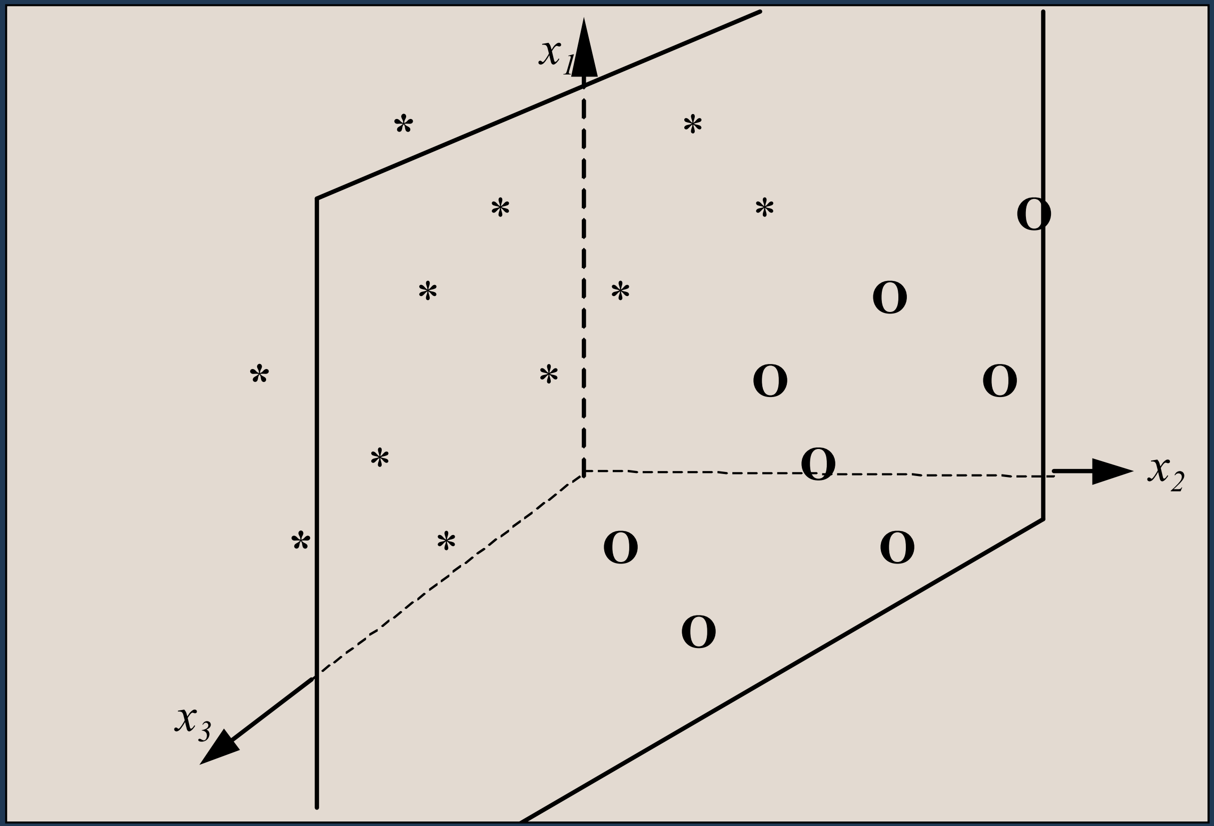


$$w_{ij}x_1 + w_{2j}x_2 + w_{3j}x_3 - T_j = 0$$

$$x_1 = a\,x_2 + b\,x_3 + c$$



3.1.2 感知器的功能



3.1.2 感知器的功能

前馈神经网络：单层感知器

(3) 设输入向量 $X=(x_1, x_2, \ldots, x_n)^T$

输出：$w_{ij}x_1+w_{2j}x_2+\ldots+w_{nj}-T_j=0$ (3.5)

则由方程 $w_{ij}x_1+w_{2j}x_2+\ldots+w_{nj}-T_j=0$ (3.6)

确定了 n 维空间上的一个分界平面。

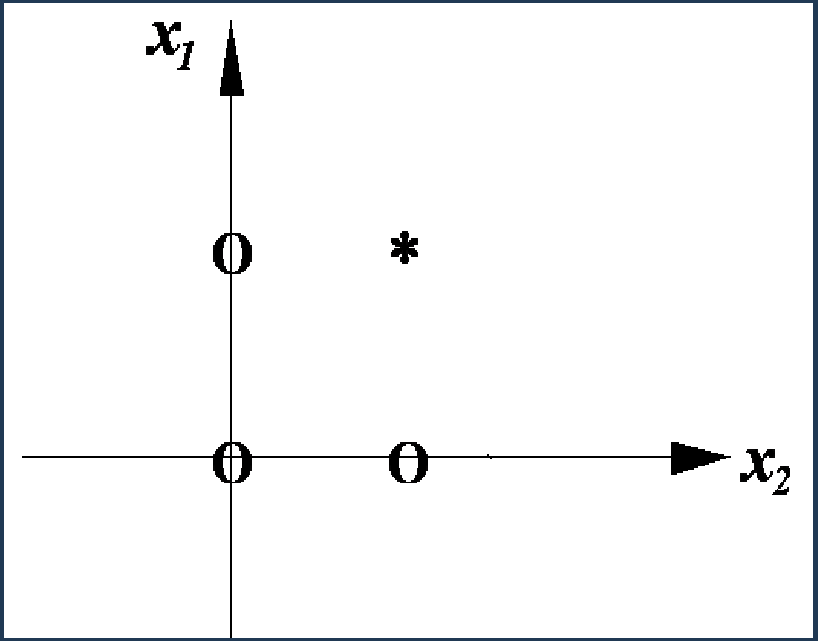
3.1.2 感知器的功能

一个最简单的单计算节点感知器具有分类功能。其分类原理是将分类知识存储于感知器的权向量（包含了阈值）中，由权向量确定的分类判决界面将输入模式分为两类。

例一 用感知器实现逻辑“与”功能

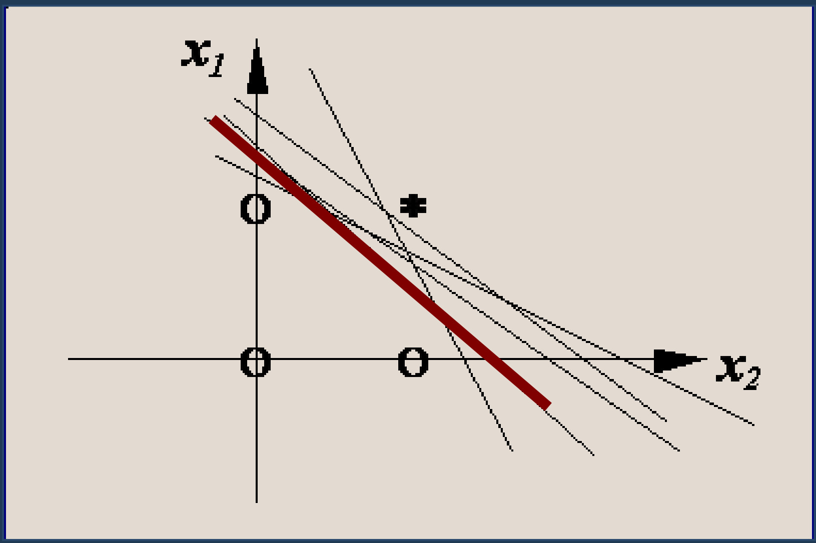
逻辑“与”真值表

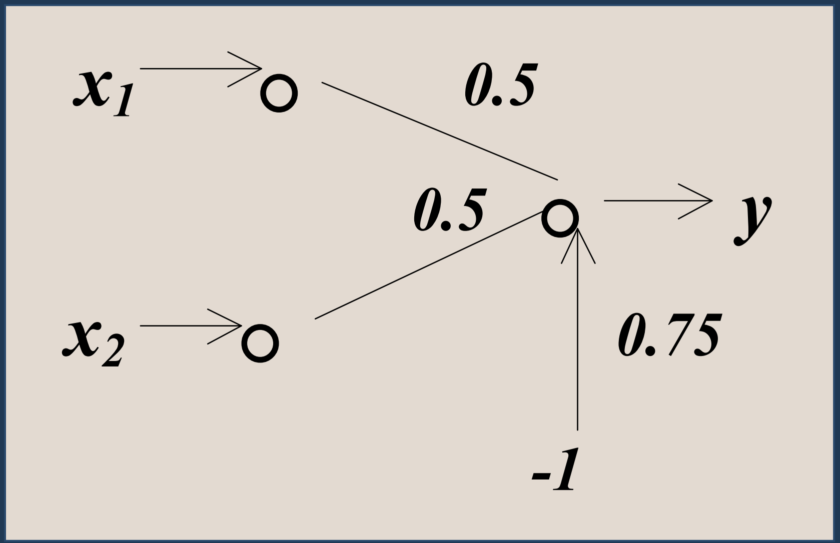
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



例一 用感知器实现逻辑“与”功能

感知器结构





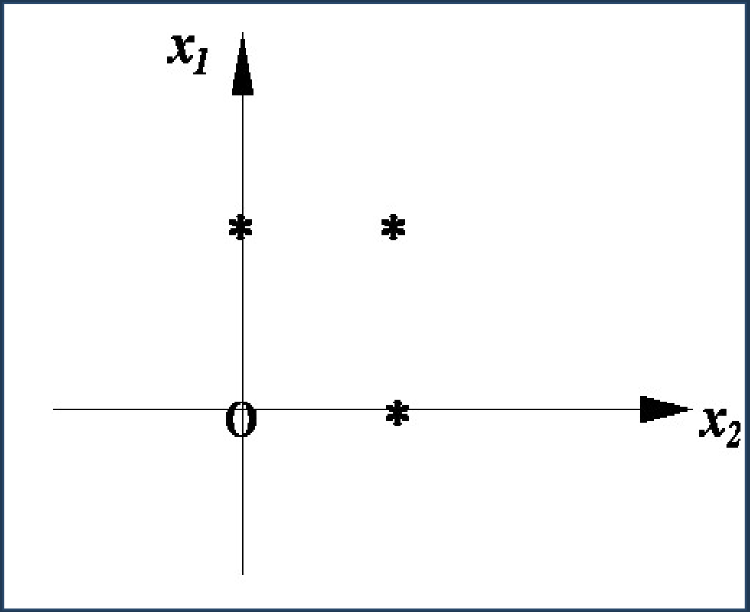
$$w_1x_1+w_2x_2-T=0$$

$$0.5x_1+0.5x_2-0.75=0$$

例二 用感知器实现逻辑“或”功能

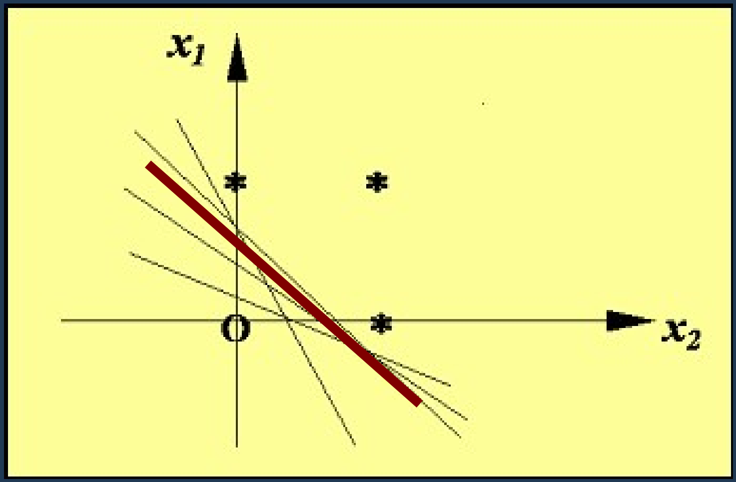
逻辑“或”真值表

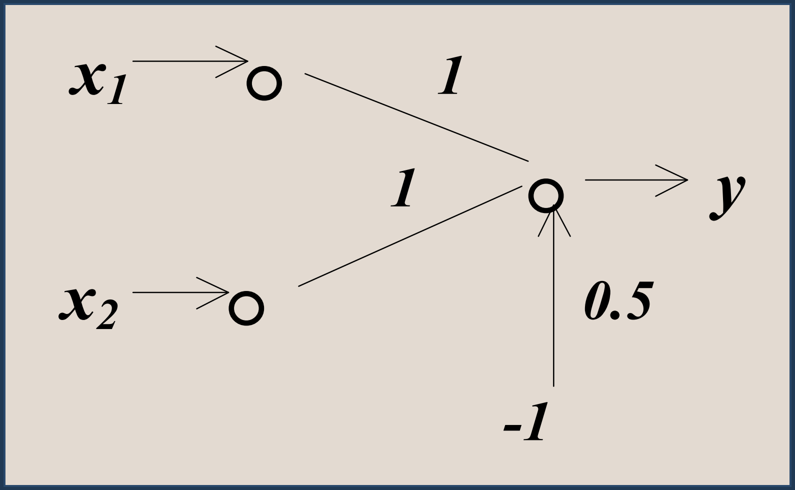
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



例二 用感知器实现逻辑“或”功能

感知器结构



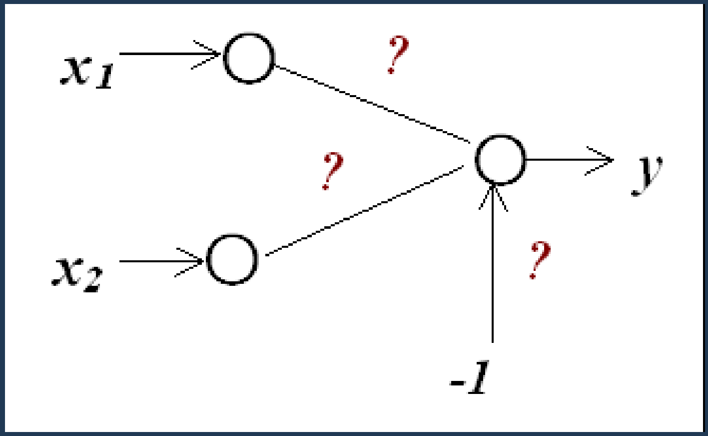


$$w_1x_1+w_2x_2-T=0$$

$$x_1+x_2-0.5=0$$

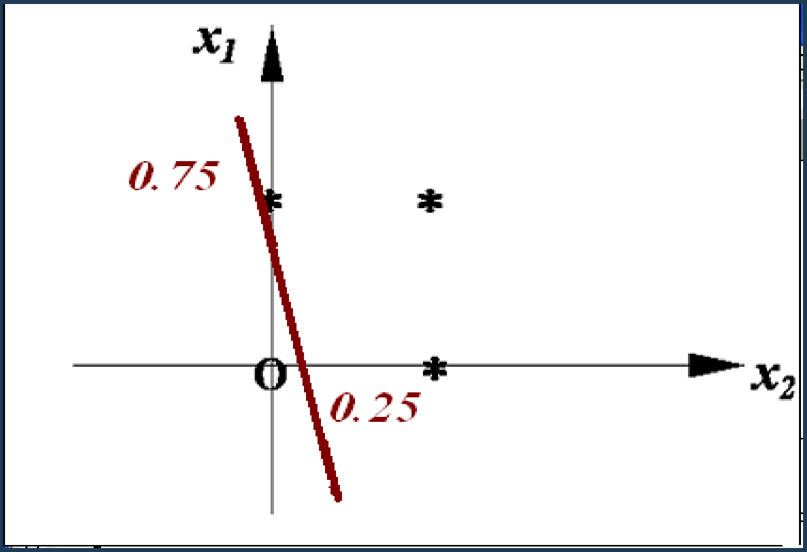
思考并回答

感知器结构



- 分界线的方程是什么？

$$\mathbf{W}_j^T \mathbf{X} = 0$$



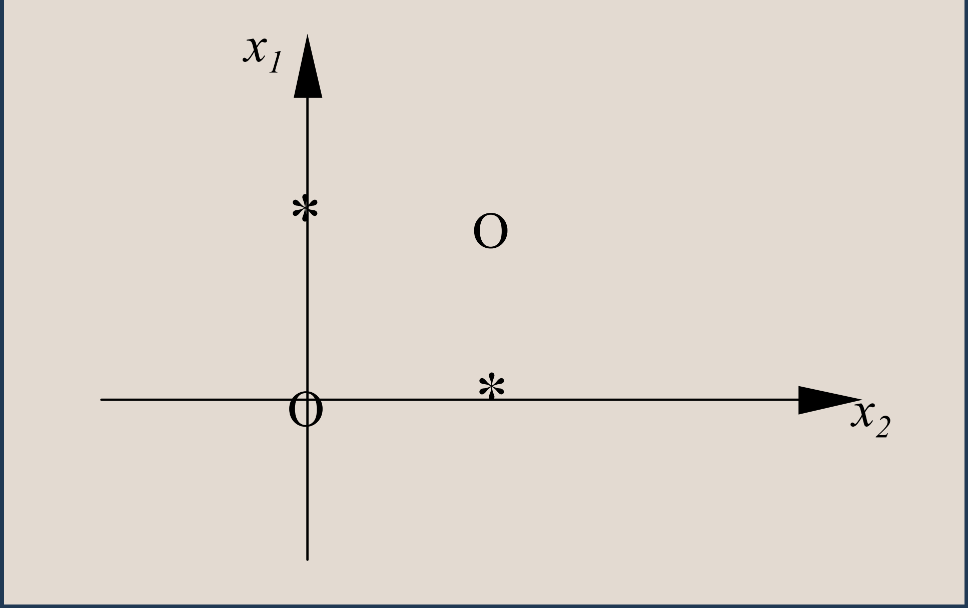
- 感知器的模型如何表示？
  - 图示？
  - 数学表达式？

3.1.3 感知器的局限性

问题：能否用感知器实现“异或”功能？

“异或”的真值表

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

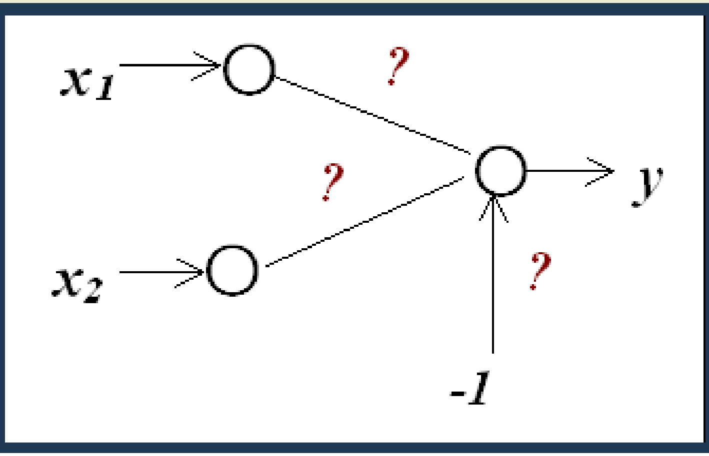


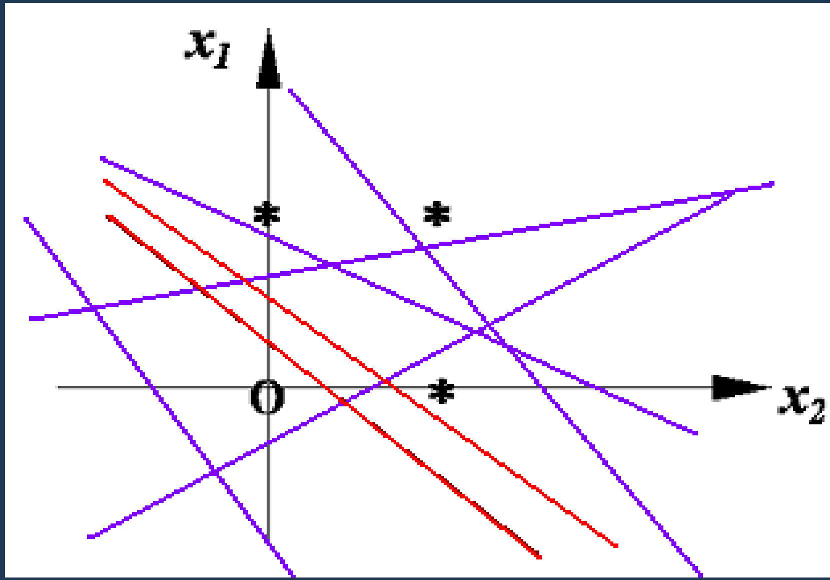
前馈神经网络：单层感知器

3.1.4 感知器的学习

- 关键问题就是求

$$\mathbf{W}_j^T\mathbf{X} = 0$$





3.1.4 感知器的学习算法

学习信号等于神经元期望输出(教师信号)与实际输出之差

$$r=d_j-o_j \tag{2.15}$$

因此，权值调整公式应为

$$\Delta \boldsymbol{W}_j=\eta\boxed{[d_j-sgn(\boldsymbol{W}_j^T\boldsymbol{X})]}\boldsymbol{X} \tag{2.17a}$$

$$\Delta w_{ij}=\eta[d_j-sgn(\boldsymbol{W}_j^T\boldsymbol{X})]x_i \qquad i=0,1,\cdots,n \tag{2.17b}$$

式中，当实际输出与期望值相同时，权值不需要调整。感知器学习规则代表一种有导师学习。

3.1.4 感知器的学习算法

感知器学习规则的训练步骤：

(1) 对各权值 $w_{0j}(0), w_{1j}(0), \cdots, w_{nj}(0)$ ，$j=1, 2, \cdots, m$

（m 为计算层的节点数）赋予较小的非零随机数；

(2) 输入样本对 $\{X^p, d^p\}$ ，其中 $X^p=(-1, x_1^p, x_2^p, \cdots, x_n^p)$ ，

d^p 为期望的输出向量（教师信号），上标 p 代表样本对的模式序号，设样本集中的样本总数为 P ，则 $p=1,2,\cdots,P$ ；

3.1.4 感知器的学习算法

感知器学习规则的训练步骤：

(3) 计算各节点的实际输出 $o_j^p(t)=sgn[W_j^T(t)X^p], j=1,2,\ldots,m$ ；

(4) 调整各节点对应的权值，$W_j(t+1)= W_j(t)+\eta[d_j^p-o_j^p(t)]X^p$, $j=1, 2,\cdots,m$, 其中 η 为学习率，用于控制调整速度，太大会影响训练的稳定性，太小则使训练的收敛速度变慢，一般取 $0 < \eta\leq 1$ ；

(5) 返回到步骤 (2) 输入下一对样本，周而复始直到对所有样本，感知器的实际输出与期望输出相等。

感知器学习规则的训练步骤：

- （1）权值初始化
- （2）输入样本对
- （3）计算输出
- （4）根据感知器学习规则调整权值
- （5）返回到步骤(2)输入下一对样本，周而复始直到对所有样本，感知器的实际输出与期望输出相等。

3.1.4 感知器的学习算法

例三　单计算节点感知器，3 个输入。给定 3 对训练样本对如下：

$X^1 = (-1,\ 1,\ -2,\ 0)^T \qquad d^1 = -1$

$X^2 = (-1,\ 0,\ 1.5,\ -0.5)^T \qquad d^2 = -1$

$X^3 = (-1,\ -1,\ 1,\ 0.5)^T \qquad d^3 = 1$

设初始权向量 $W(0)=(0.5,1,-1,0)^T$，$\eta=0.1$。注意，输入向量中第一个分量 x_0 恒等于 -1，权向量中第一个分量为阈值，试根据以上学习规则训练该感知器。

3.1.4 感知器的学习算法

解：第一步 输入 X^1 ，得

$$W^T(0)X^1=(0.5,1,-1,0)(-1,1,-2,0)^T=2.5$$

$$o^1(0)=sgn(2.5)=1$$

$$W(1)= W(0)+\eta[d^1- o^1(0)] X^1$$

$$=(0.5,1,-1,0)^T+0.1(-1-1)(-1,1,-2,0)^T$$

$$=(0.7,0.8,-0.6,0)^T$$

3.1.4 感知器的学习算法

第二步 输入 X^2 ，得

$$W^T(1)X^2=(0.7,0.8,-0.6,0)(-1,0,1.5,-0.5)^T=-1.6$$

$$o^2(1)=sgn(-1.6)=-1$$

$$W(2)= W(1)+\eta[d^2- o^2(1)]\, X^2$$

$$=(0.7,0.8,-0.6,0)^T+0.1[-1-(-1)](-1,0,1.5,-0.5)^T$$

$$=(0.7,0.8,-0.6,0)^T$$

由于 $d^2= o^2(1)$ ，所以 $W(2)= W(1)$ 。

3.1.4 感知器的学习算法

前馈神经网络：单层感知器

第三步 输入 X^3 ，得

$$W^T(2)X^3=(0.7,0.8,-0.6,0)(-1,-1,1,0.5)^T=-2.1$$

$$O^3(2)=sgn(-2.1=-1$$

$$W(3)= W(2)+\eta[d^3- o^3(2)] X^3$$

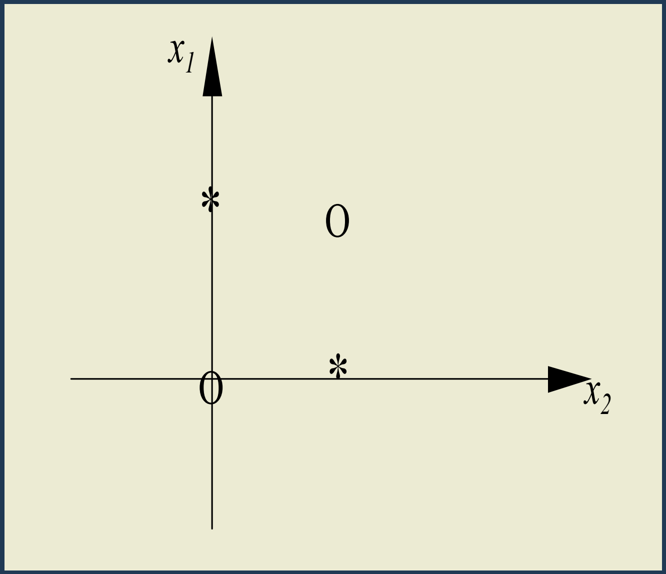
$$=(0.7,0.8,-0.6,0)^T+0.1[1-(-1)](-1,-1,1,0.5)^T$$

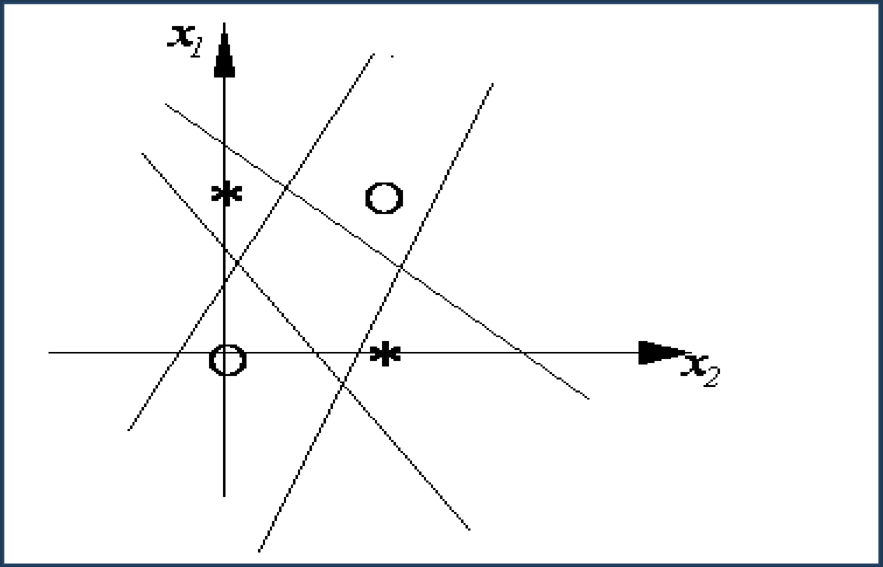
$$=(0.5,0.6,-0.4,0.1)^T$$

第四步 返回到第一步，继续训练直到 $d^p- o^p=0$ ， $p=1,2,3$ 。

3.1.3 单层感知器的局限性

- 问题：能否用感知器解决如下问题？



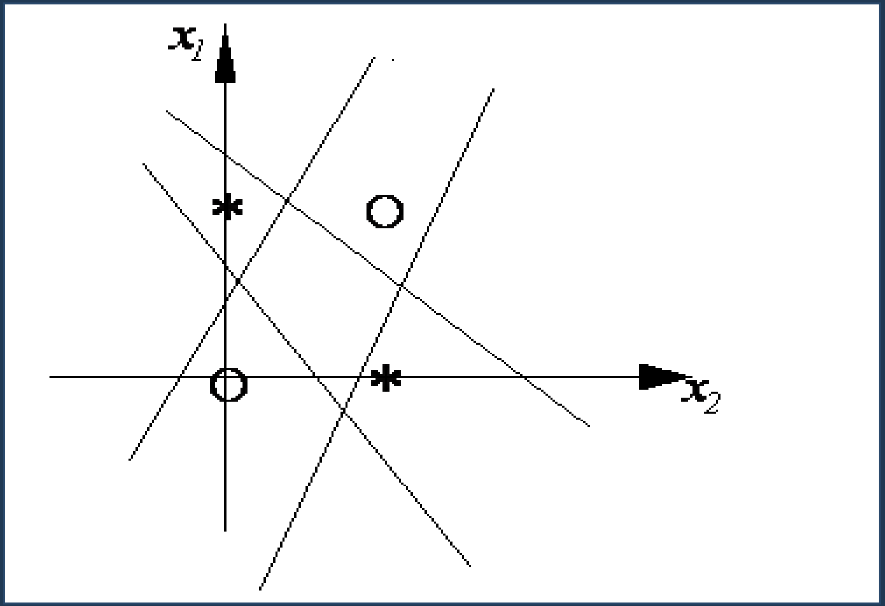


3.1.3 单层感知器的局限性

- 无法解决“异或”问题

“异或”的真值表

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

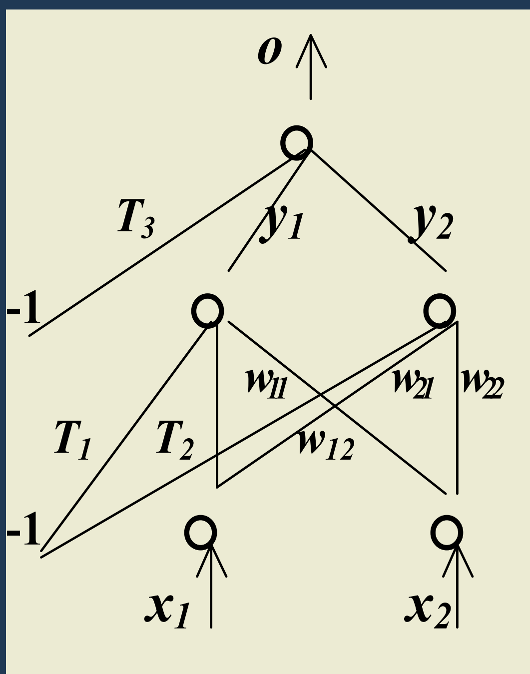


- 只能解决线性可分问题

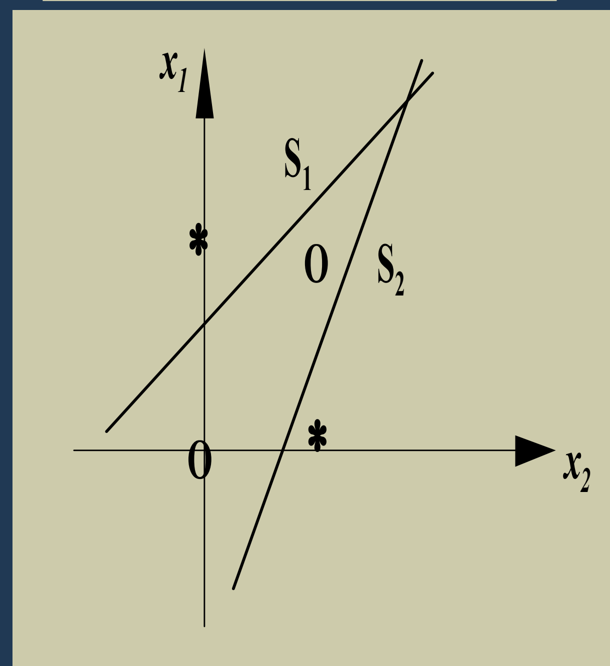
3.2 多层感知器

例四 用两计算层感知器解决“异或”问题。

双层感知器



“异或”问题分类



“异或”的真值表

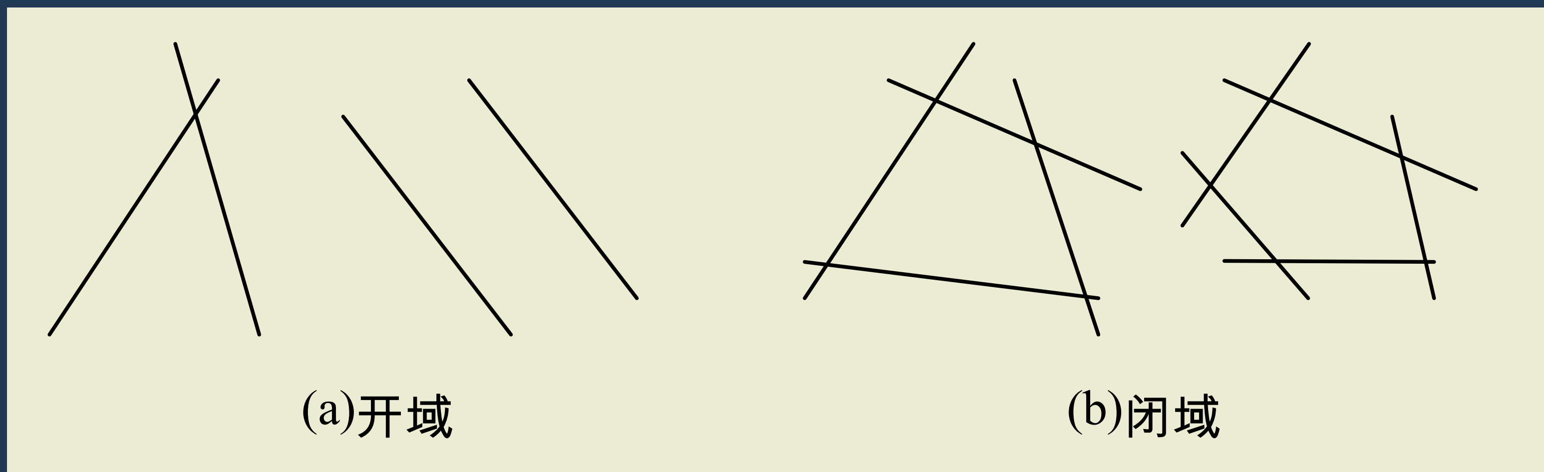
x_1	x_2	y_1	y_2	o
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	1	1	0

3.2.1 多层感知器的提出

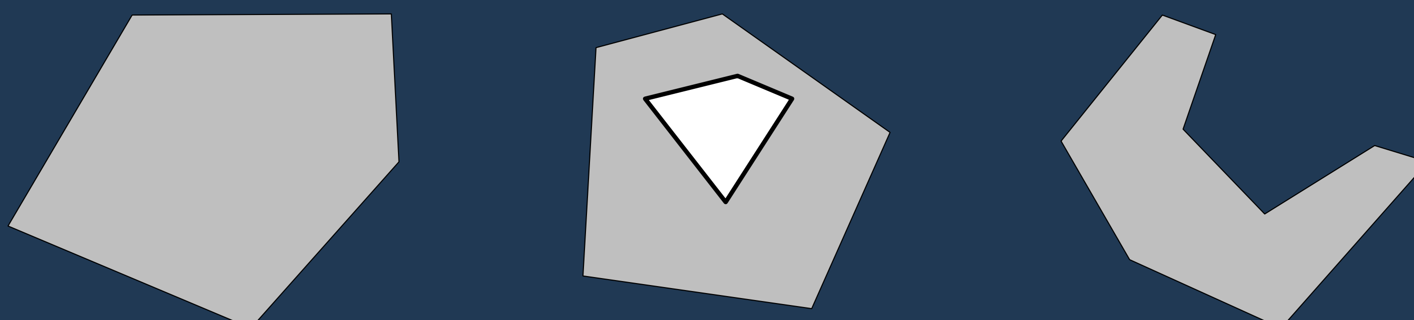
提出的动因

- 单计算层感知器的局限性：
  - 只能解决线性可分问题，而大量的分类问题是线性不可分的。
- 解决的有效办法
  - 在输入层与输出层之间引入隐层作为输入模式的“内部表示”，将单计算层感知器变成*多（计算）层感知器*。
  - 采用*非线性连续函数作为转移函数*，使区域边界线的基本线素由直线变成曲线，从而使整个边界线变成连续光滑的曲线。

3.2 多层感知器



(a)开域　　(b)闭域

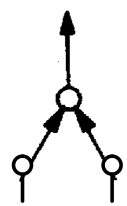
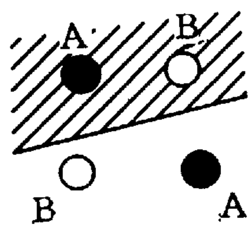
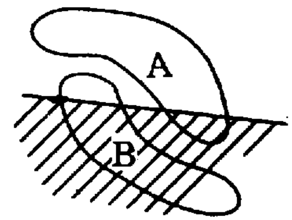
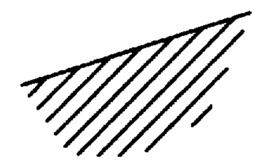
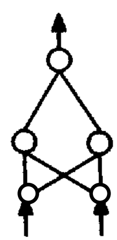
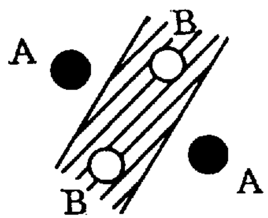
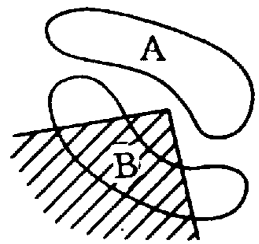
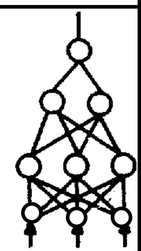
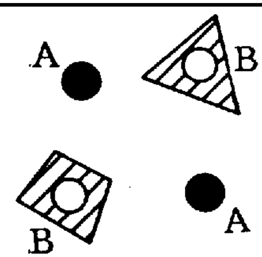
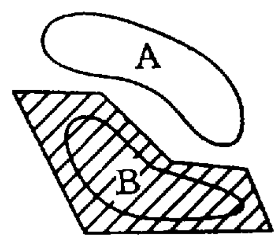
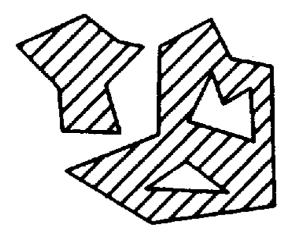


前馈神经网络—多层感知器

3.2 多层感知器

$\Delta W_j(t)=\eta[d_j-o_j(t)]X$

具有不同隐层数的感知器的分类能力对比

感知器结构	异或问题	复杂问题	判决域形状	判决域
无隐层 				半平面
单隐层 				凸 域
双隐层 				任意复杂形状域