第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配

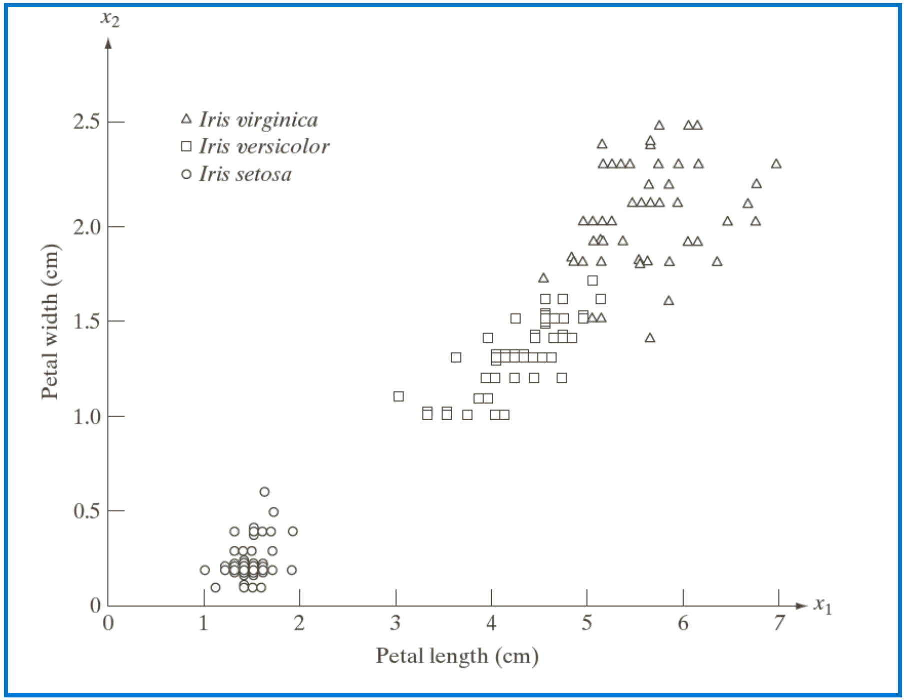
第12章　目标识别

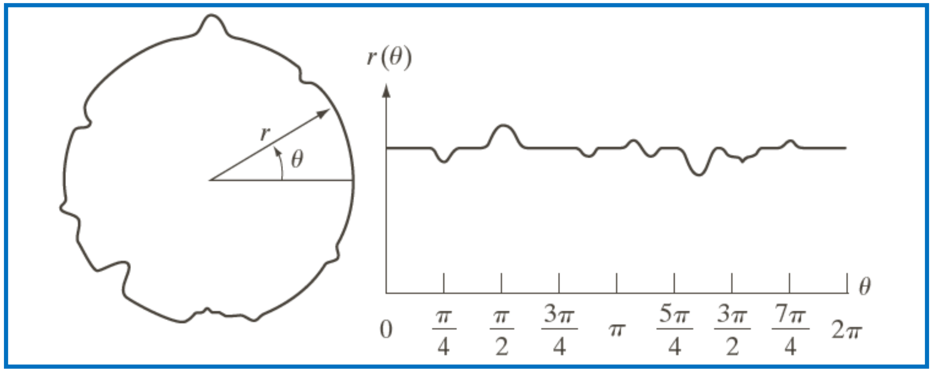
- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配

12.1 模式和模式类

模式向量

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$$

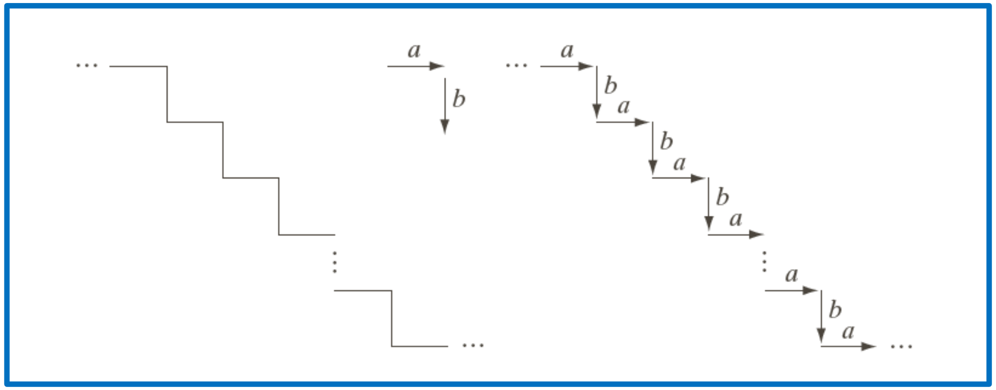




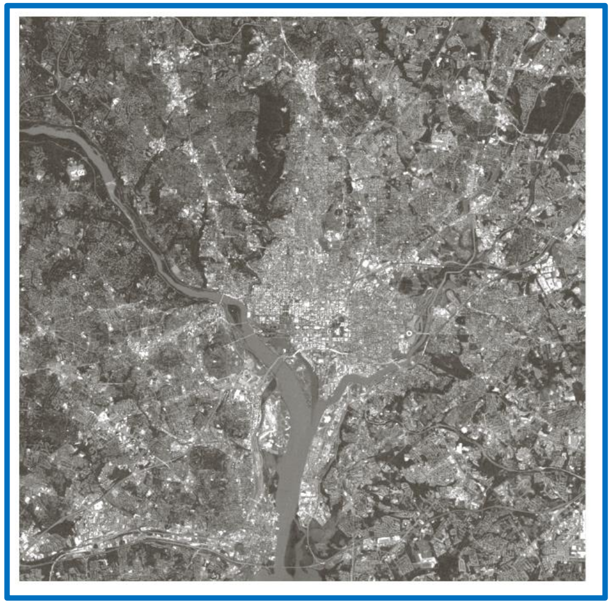
一个带有噪声的目标及其相应的信号

由两个测度描述的三种鸢尾花

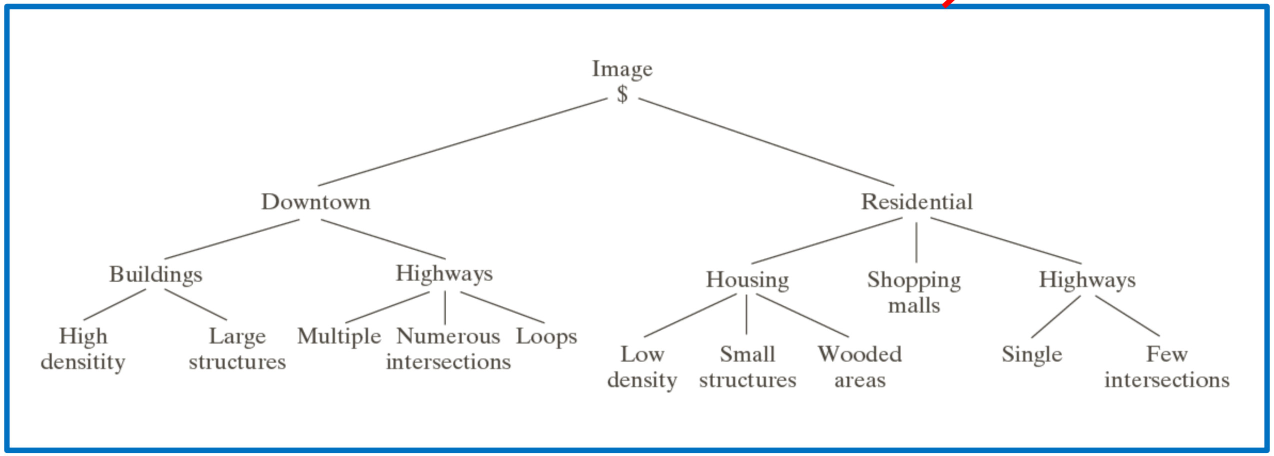
12.1 模式和模式类



(a) 阶梯结构 (b) 使用基元a和b对结构编码，生成串描述…$ababab$ …



建筑物密集的城市中心区（华盛顿特区）和周围居民区的卫星图像（原图像由NASA提供）



第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配

12.2 基于决策理论方法的识别

- W个模式类$\omega_1, \omega_2, \cdots, \omega_W$，决策函数$d_1(x), d_2(x), \cdots, d_W(x)$

  如果模式$\boldsymbol{x}$属于类ω_i，则

$$d_i(x) > d_j(x)\,, \qquad \forall\, j = 1,2,\cdots,W;\ j \neq i$$

- 决策边界

$$d_i(x) = d_j(x)$$

12.2.1 匹配

最小距离分类器

模式类的原型

$$\boldsymbol{m}_j = \frac{1}{N_j}\sum_{x\in\omega_j} \boldsymbol{x}_j \quad j = 1,2,\dots,W$$

距离测度

$$D_j(\boldsymbol{x}) = \| \boldsymbol{x} - \boldsymbol{m}_j \| \quad j = 1,2,\dots,W$$

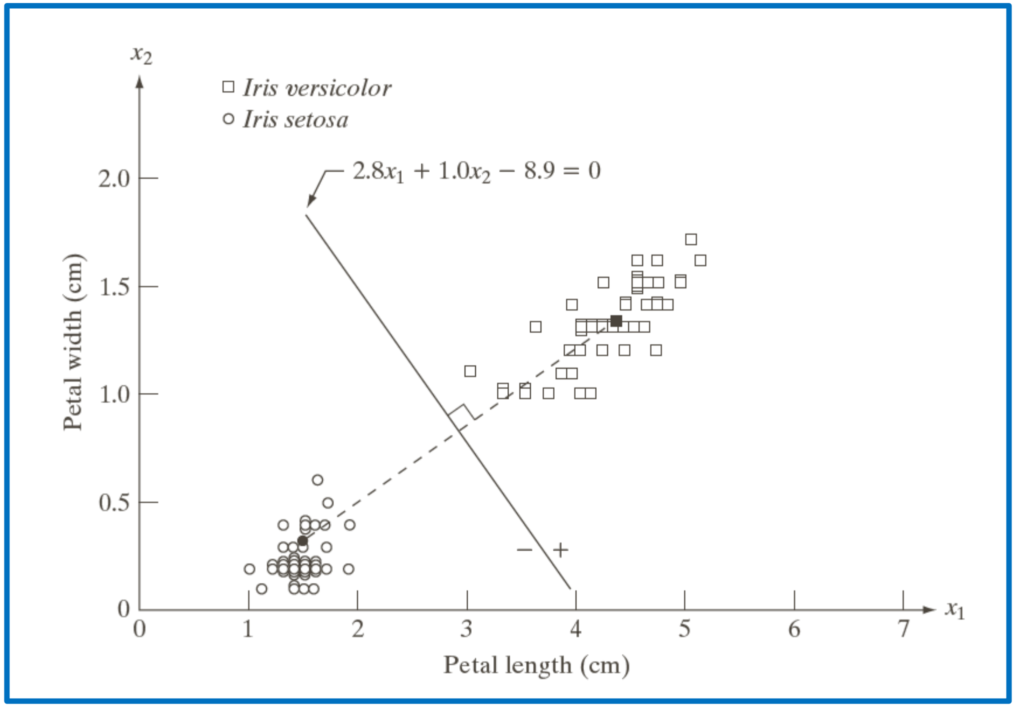
选择最小距离等价于

$$d_j(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{m}_j - \frac{1}{2}\boldsymbol{m}_j^{\mathrm{T}}\boldsymbol{m}_j \quad j = 1,2,\dots,W$$

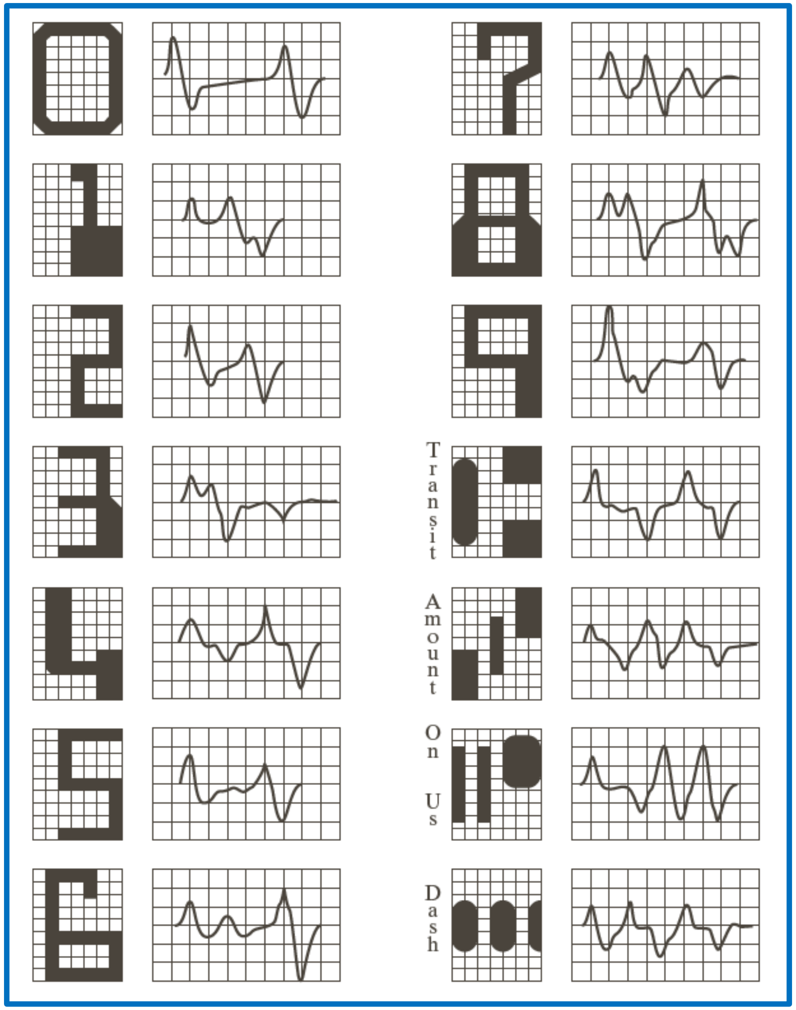
决策边界

$$d_{ij}(\boldsymbol{x}) = d_i(\boldsymbol{x}) - d_j(\boldsymbol{x})$$
$$= \boldsymbol{x}^{\mathrm{T}}(\boldsymbol{m}_i - \boldsymbol{m}_j) - \frac{1}{2}(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}}(\boldsymbol{m}_i + \boldsymbol{m}_j) = 0$$

12.2.1 匹配



类*Iris versicolor*和类*Iris setosa*的最小距离分类器的决策边界



美国银行家协会E-138字符集和对应波形

12.2.1 匹配

相关匹配

相关表示

$$c(x,y)=\sum_s \sum_s w(s,t)f(x+s,y+t)$$

模板 图像

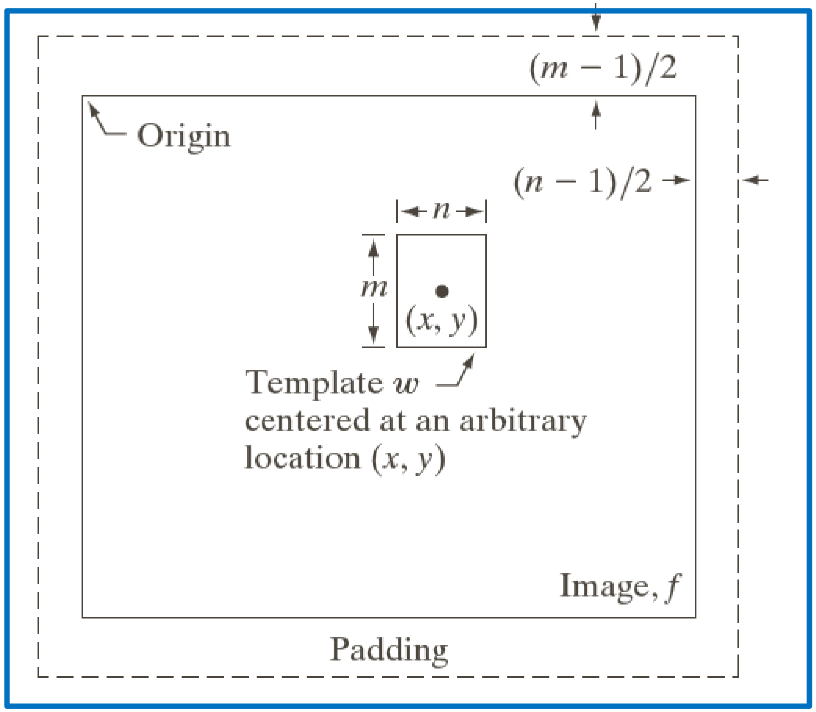
空间相关

$$f(x,y) \star w(x,y) \Leftrightarrow F^*(u,v)W(u,v)$$

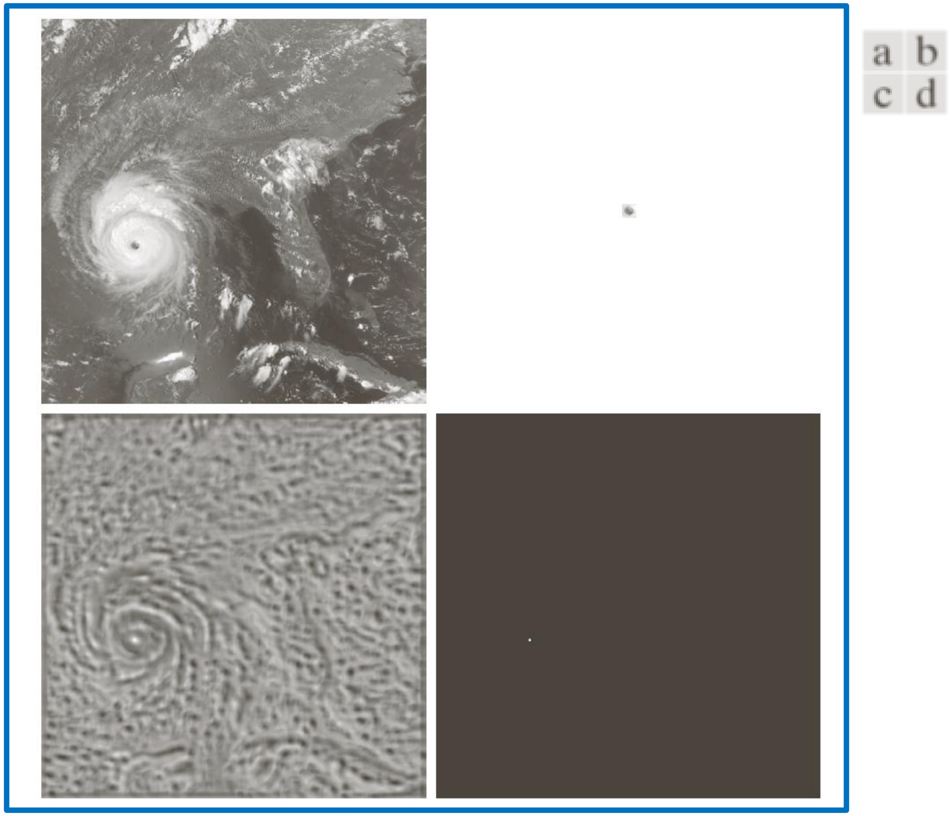
$$\gamma(x,y)=\frac{\sum_s \sum_t [w(s,t)-\overline{w}][f(x+s,y+t)-\overline{f_{xy}}]}{\{\sum_s \sum_t [w(s,t)-\overline{w}]^2 \sum_s \sum_t [f(x+s,y+t)-\overline{f_{xy}}]\}^{\frac{1}{2}}}$$

归一化相关系数

12.2.1 匹配



模板匹配的机理



（a）飓风“安德鲁”的卫星图像，摄于1992年8月24日；（b）飓风眼的模板；（c）显示为图像的相关系数（注意最亮点）；（d）匹配最好的位置。该点是单个像素，但其已被放大，以便于查看（原图像由NOAA提供）

12.2.2 最佳统计分类器

- 基础知识
  - 模式$\boldsymbol{x}$ 被分为第k 类的概率 $p(\omega_k|\boldsymbol{x})$，样本属于第$k$ 类但被判为第j 类的风险 L_{kj} ，则$\boldsymbol{x}$ 被分为第j 类的平均损失：

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{W} L_{kj}p(\omega_k|\boldsymbol{x}) = \frac{1}{p(\boldsymbol{x})}\sum_{k=1}^{W} L_{kj}p(\boldsymbol{x}|\omega_k)P(\omega_k)$$

去掉$\frac{1}{p(\boldsymbol{x})}$不影响$r_j(\boldsymbol{x})$的对不同$j$ 的相对大小排序

  - 定义风险函数

$$L_{ij} = 1 - \delta_{ij}$$

  - 更新后的平均损失函数为：

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{W}(1 - \delta_{ij})p(\boldsymbol{x}|\omega_k)P(\omega_k) = P(\boldsymbol{x}) - p\left(\boldsymbol{x}\middle|\omega_j\right)P(\omega_j)$$

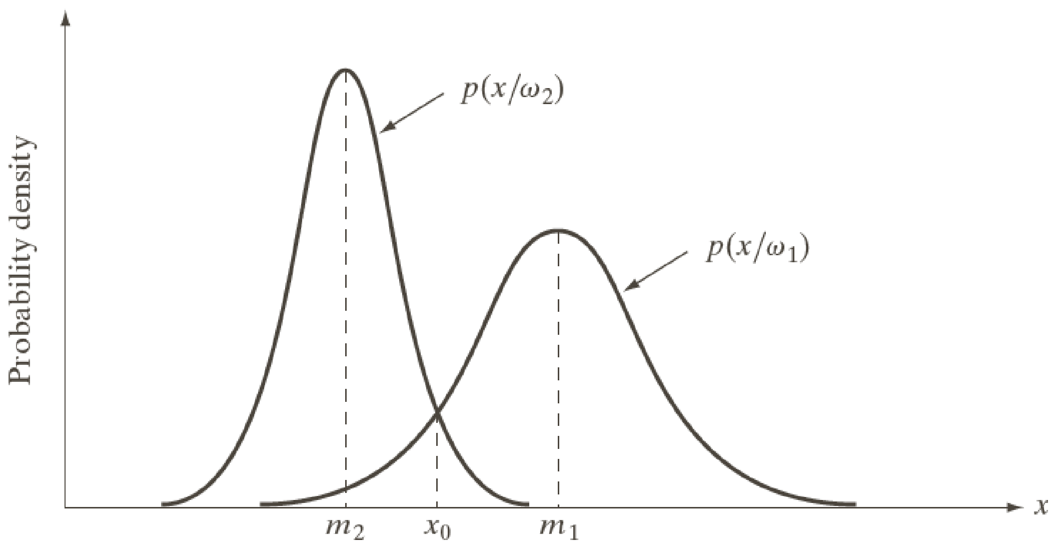
12.2.2 最佳统计分类器

- 高斯模式类的贝叶斯分类器

贝叶斯决策函数

$$d_j(x) = p(x|\omega_j)P(\omega_j)$$

$$= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \quad j = 1,2$$



两个一维模式类的概率密度函数。如果两个类出现概率相等，那么所示的点x_0就是决策边界

12.2.2 最佳统计分类器

- 高斯模式类的贝叶斯分类器

n维情形下，j个模式类中的向量高斯密度

$$p(\boldsymbol{x}|\omega_j) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{C}_j|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_j)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{m}_j)}$$

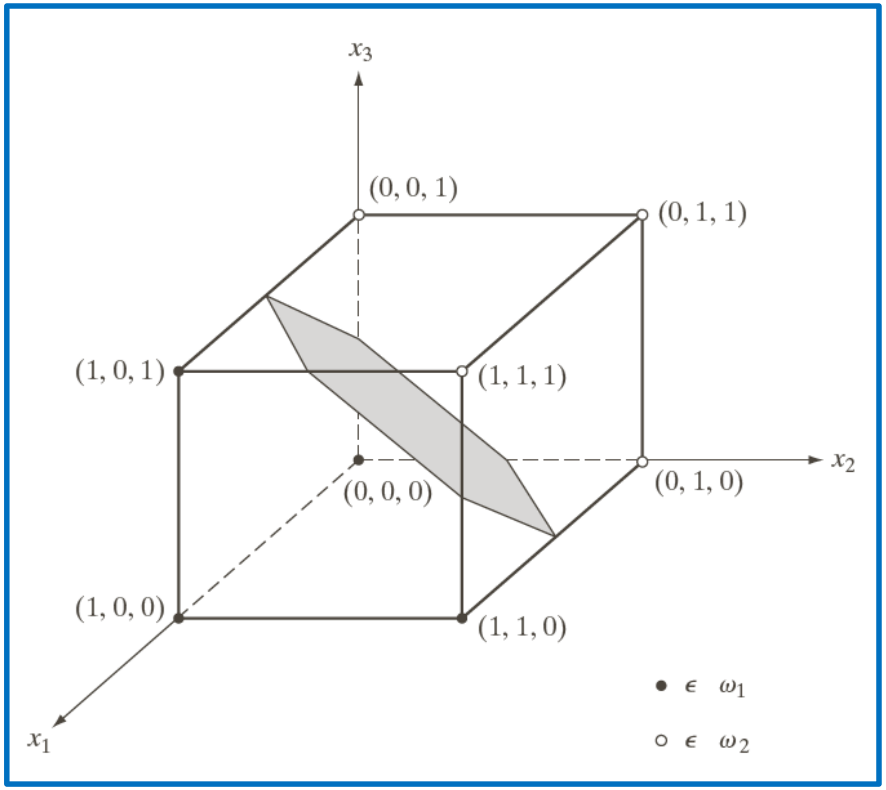
$$\boldsymbol{m}_j = \frac{1}{N_j}\sum_{\boldsymbol{x}\in\omega_j}\boldsymbol{x} \qquad \boldsymbol{C}_j = \frac{1}{N_j}\sum_{\boldsymbol{x}\in\omega_j}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} - \boldsymbol{m}_j\boldsymbol{m}_j^{\mathrm{T}}$$

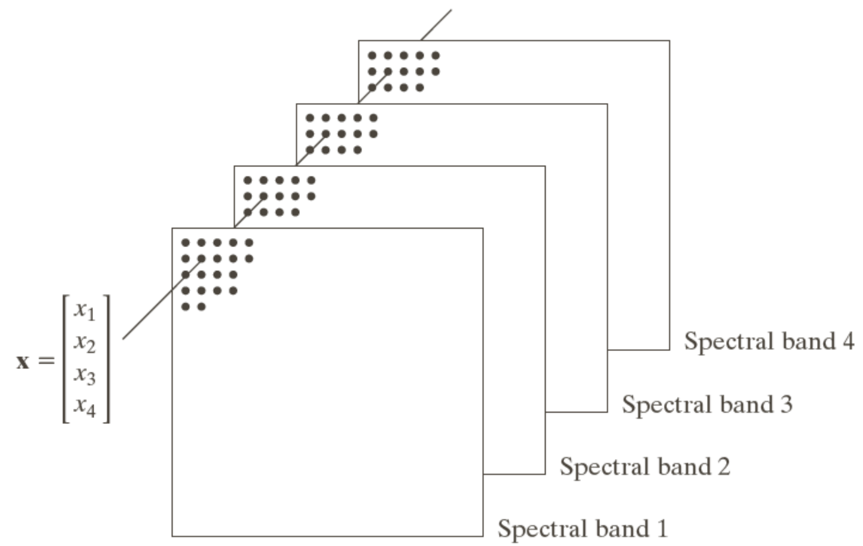
线性决策函数（超平面）

$$d_j(\boldsymbol{x}) = \ln P(\omega_j) + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{m}_j - \frac{1}{2}\boldsymbol{m}_j^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{m}_j$$

12.2.2 最佳统计分类器

- 高斯模式类的贝叶斯分类器



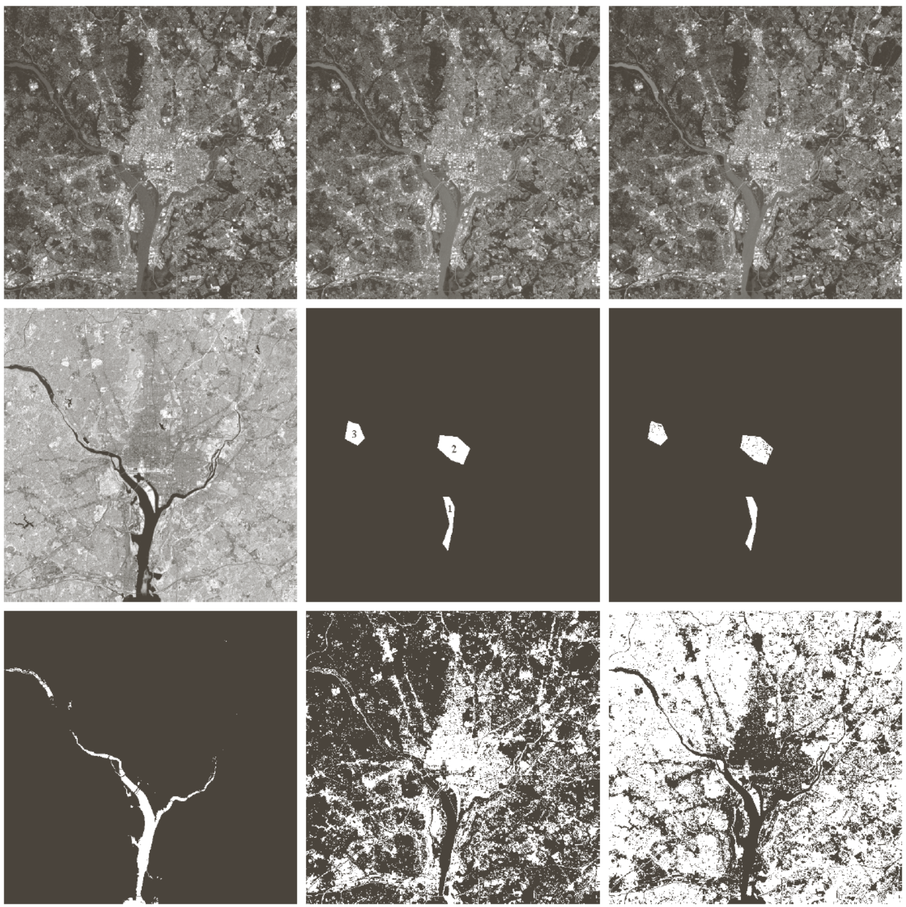


两个简单的模式类及其贝叶斯决策边界（阴影所示）

由多光谱扫描器生成的4幅数字图像经像素配准后，所形成的的模式向量

12.2.2 最佳统计分类器

- 高斯模式类的贝叶斯分类器

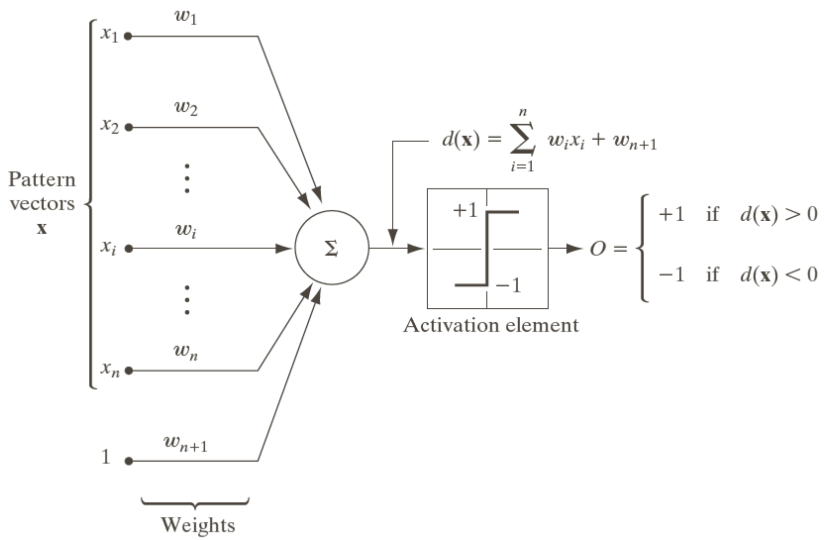


a b c
d e f
g h i

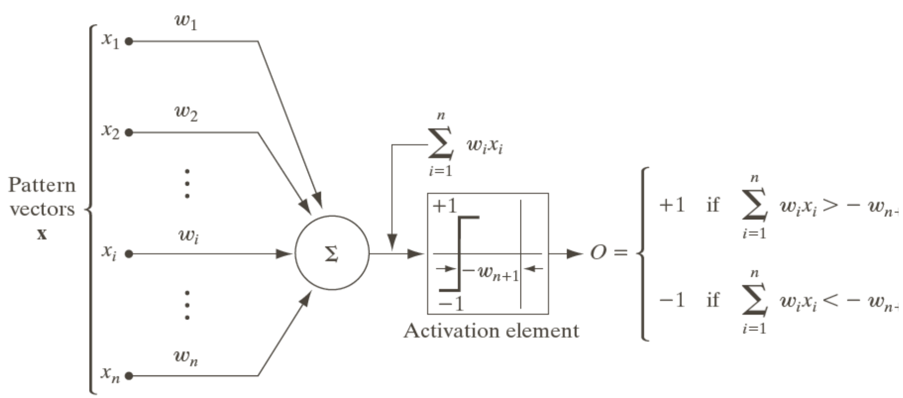
多光谱数据的贝叶斯分类：
(a)~(d) 可见蓝光、可见绿光、可见红光和古今红外波长图像；
(e) 显示(1)水体、(2)市区和(3)植被的样本区域的模板；
(f) 分类结果。黑点表示为正确分类的点，其他（白）点是正确分类的点；
(g) 分类为水体的所有图像像素（白色）；
(h) 分类为市区的所有图像像素（白色）；
(i) 分类为植被的所有图像像素（白色）

12.2.3 神经网络

- 两个模式类的感知机



$$d(x) = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$



$$O = \begin{cases} +1, \sum_{i=1}^{n} w_i x_i > -w_{n+1} \\ -1, \sum_{i=1}^{n} w_i x_i < -w_{n+1} \end{cases}$$

12.2.3 神经网络

向量化表达

- 记$\boldsymbol{y}=\{x_1,x_2,\cdots,x_n,1\}^T$，$\boldsymbol{w}=\{w_1,w_2,\cdots,w_n,w_{n+1}\}^T$，则判决函数:

$$d(\boldsymbol{y})=\sum_{i=1}^{n+1} w_i y_i=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}$$

训练算法

- 线性可分的类:

  如果$\boldsymbol{y}(k)\in\omega_1$且$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}(k)\le 0$，则

  $$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+c\boldsymbol{y}(k)$$

  相反，如果$\boldsymbol{y}(k)\in\omega_2$且$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}(k)\ge 0$，则

  $$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-c\boldsymbol{y}(k)$$

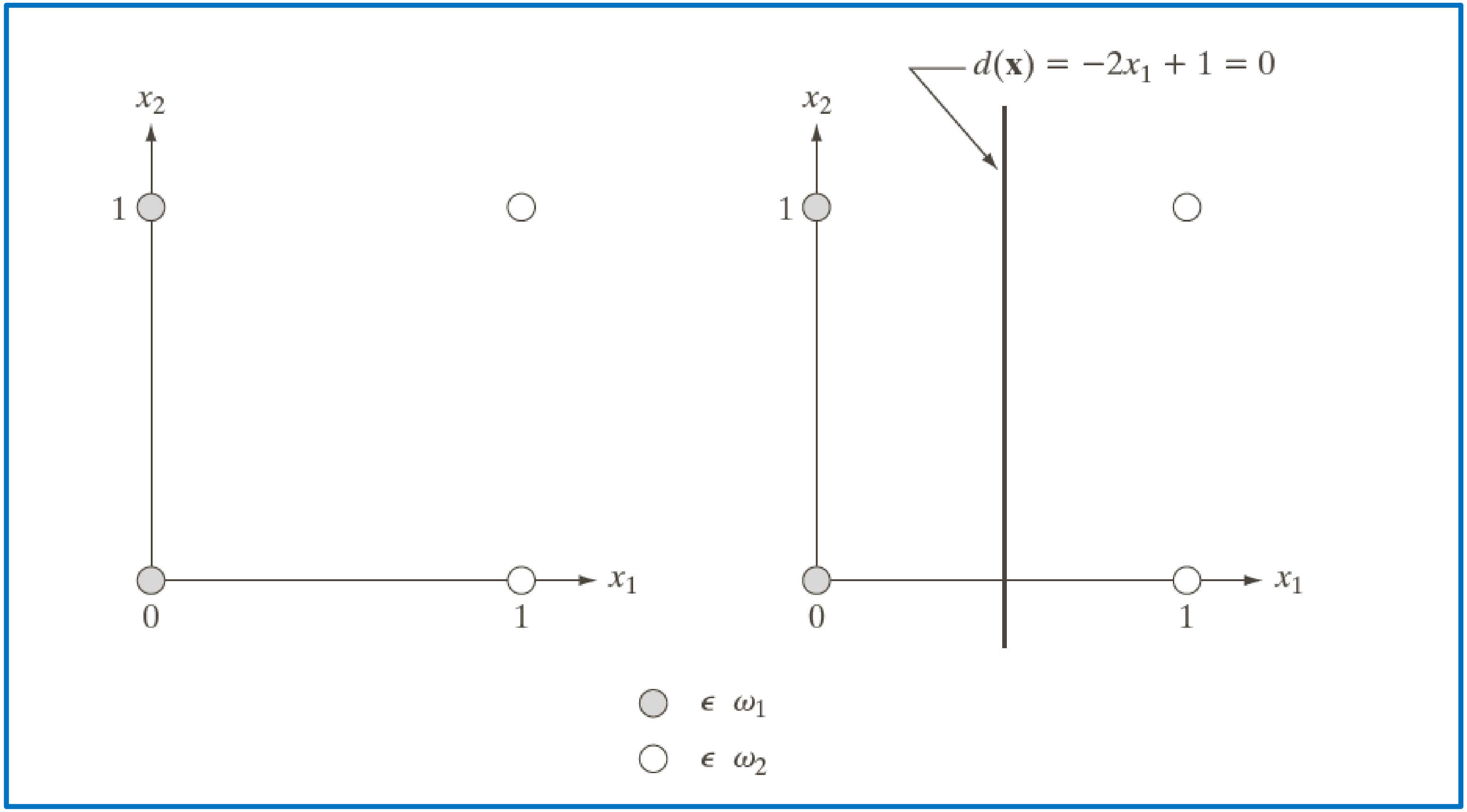
  否则，$\boldsymbol{w}(k)$保持不变

- 线性不可分的类:

  准则函数 $J(\boldsymbol{w})=\frac{1}{2}\left(r-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}\right)^2$

  梯度下降算法$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-\alpha\left[\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\right]_{\boldsymbol{w}=\boldsymbol{w}(k)}$

12.2.3 神经网络



(a) 属于两个类的模式；
(b) 由训练确定的决策边界

12.2.3 神经网络

多层前馈神经网络

多层前馈神经网络模型。放大部分显示了整个网络中每个神经元的基本结构。

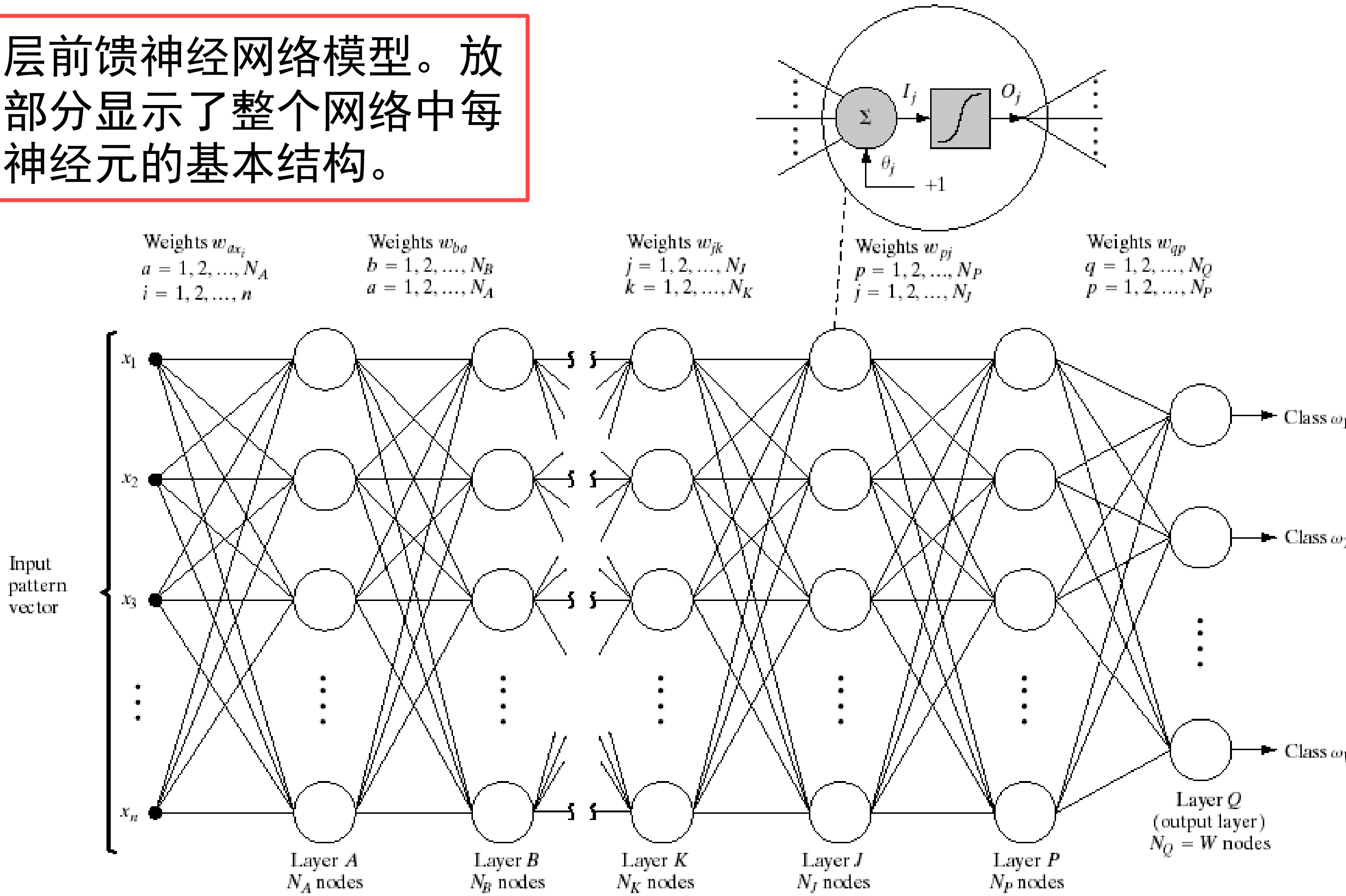


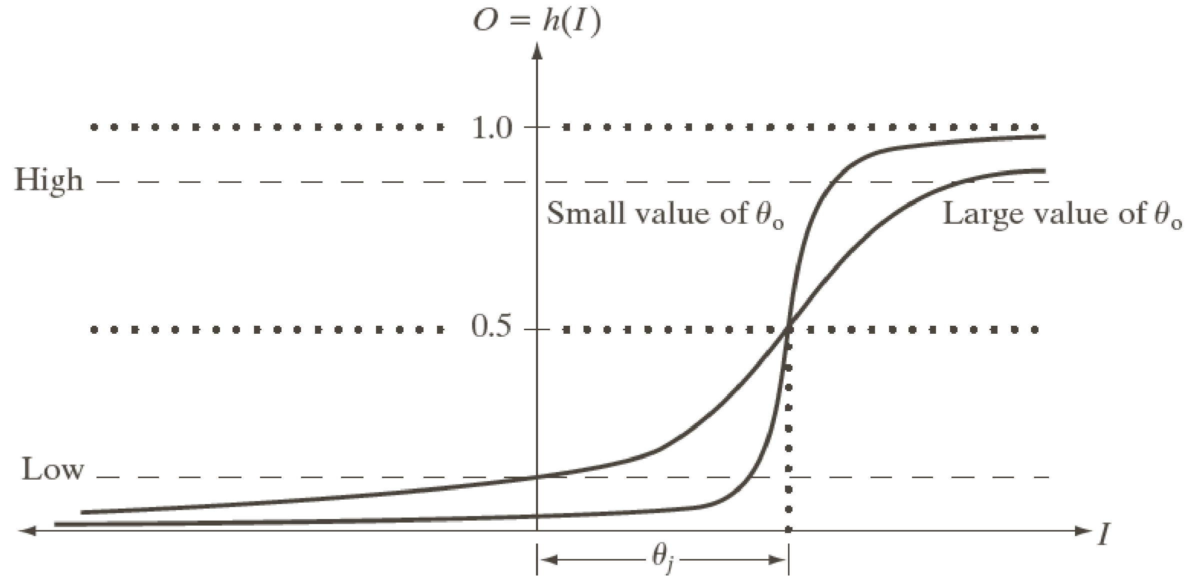
FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset, θ_j, is treated as just another weight.

12.2.3 神经网络

- 多层前馈神经网络

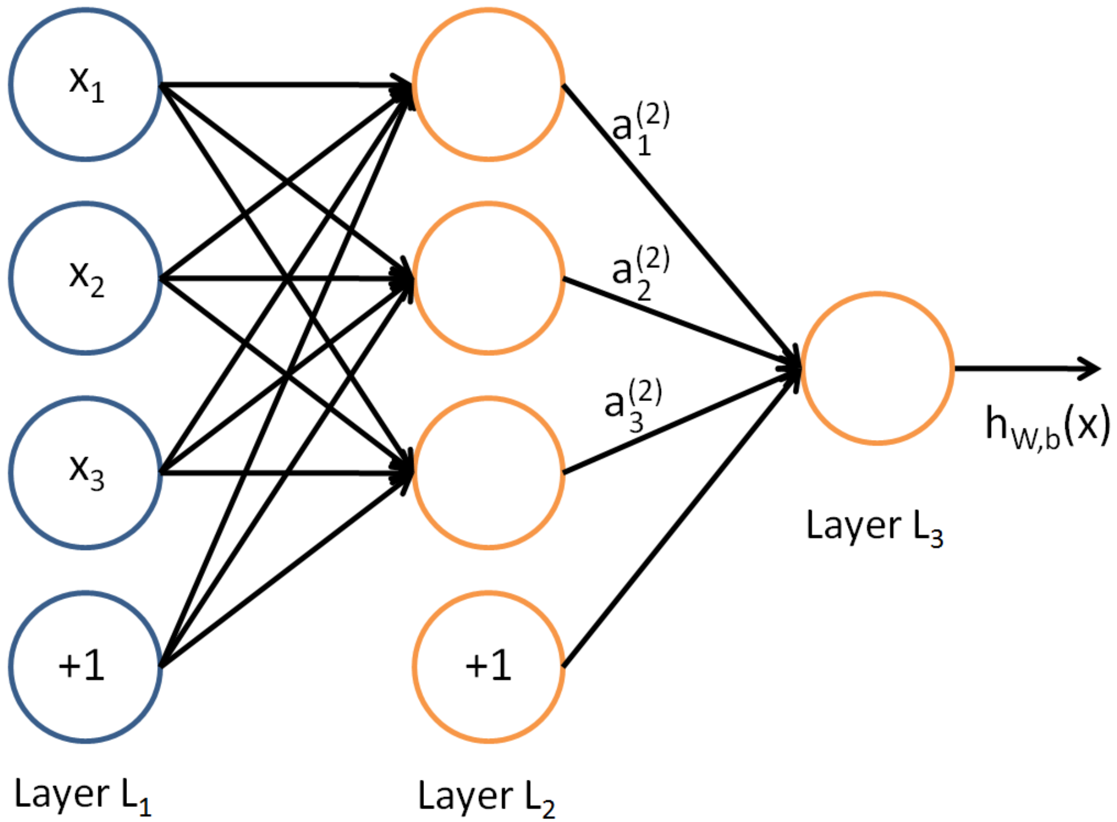
激活函数

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j+\theta_j)/\theta_o}}$$



Forward Network（前馈网络）

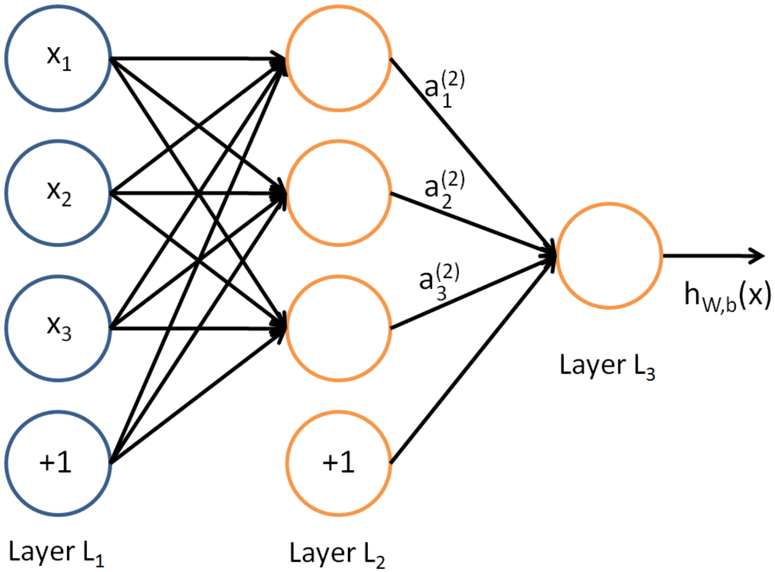
- 神经网络模型



输入层　　隐藏层　　输出层

Forward Network（前馈网络）

□ 神经网络模型



$a_i^{(l)}$：第l层第i单元的输出值

$z_i^{(l)}$：第l层第i单元的输入加权和

$$a_1^{(2)} = f\left(z_1^{(2)}\right)$$
$$= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$

$$a_2^{(2)} = f\left(z_2^{(2)}\right)$$
$$= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$

$$a_3^{(2)} = f\left(z_3^{(2)}\right)$$
$$= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$

$$\mathrm{h}_{\mathbf{W},b}(\mathbf{x}) = a_1^{(3)} = f(z_1^{(3)})$$
$$= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_2^{(2)} + b_1^{(2)})$$

训练前馈网络

- 误差后向传播算法（BP）
  - 训练样本集$\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)})\}$
  - 代价函数

对于单个样例$(\mathbf{x}, y)$的代价函数 $J(\mathbf{W}, b; \mathbf{x}, y) = \frac{1}{2}\left\|\mathrm{h}_{\mathbf{W},b}(\mathbf{x}) - y\right\|^2$

对于训练集的整体代价函数

$$J(\mathbf{W}, b) = \left[\frac{1}{m}\sum_{i=1}^{m} J(\mathbf{W}, b; x^{(i)}, y^{(i)})\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2$$

$$= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|\mathrm{h}_{\mathbf{W},b}\left(\mathbf{x}^{(l)}\right) - y\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2$$

以上公式中第一项是一个**均方差项**，第二项是一个**正则化项**（也叫权重衰减weight decay项），其目的是减小权重的幅度，防止过度拟合。

训练前馈网络

- 后向传播算法
  - 参数随机初始化
  - 进行前馈传导计算，利用前向传导公式，得到L_2, $L_3, \ldots, L_{n_l}$的输出值；
  - 对于第n_l层（输出层）的每个输出单元i，我们根据以下公式计算残差：

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} J(\mathbf{W}, b; \mathbf{x}, y) = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \left\| y - \mathrm{h}_{\mathbf{W},b}(\mathbf{x}) \right\|^2$$

$$= -(y - a_i^{(n_l)}) \cdot f'\left(z_i^{(n_l)}\right)$$

训练前馈网络

后向传播算法

- 对$l = n_l - 1, n_l - 2, n_l - 3, \ldots, 2$的各个层，第$l$层的第$i$个节点的残差计算方法如下：

$$\delta_i^{(n_l-1)} = \frac{\partial}{\partial z_i^{(n_l-1)}} J(\mathbf{W}, b; \mathbf{x}, y) = \frac{\partial}{\partial z_i^{(n_l)}} J(\mathbf{W}, b; \mathbf{x}, y) \cdot \frac{\partial z_i^{(n_l)}}{\partial z_i^{(n_l-1)}}$$

$$= \delta_i^{(n_l)} \cdot \frac{\partial z_i^{(n_l)}}{\partial z_i^{(n_l-1)}} = \delta_i^{(n_l)} \cdot \frac{\partial}{\partial z_i^{(n_l-1)}} \sum_{j=1}^{s_{n_l}} W_{ji}^{(n_l-1)} f\left(z_i^{(n_l-1)}\right)$$

$$= \left(\sum_{j=1}^{s_{n_l}} W_{ji}^{(n_l-1)} \delta_i^{(n_l)}\right) f'\left(z_i^{(n_l-1)}\right)$$

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_i^{(l+1)}\right) f'\left(z_i^{(l)}\right)$$

训练前馈网络

- 后向传播算法（BP）
  - 计算我们需要的偏导数，计算方法如下：

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(\mathbf{W}, b; \mathbf{x}, y) = a_j^{(l)} \delta_i^{(l+1)}$$

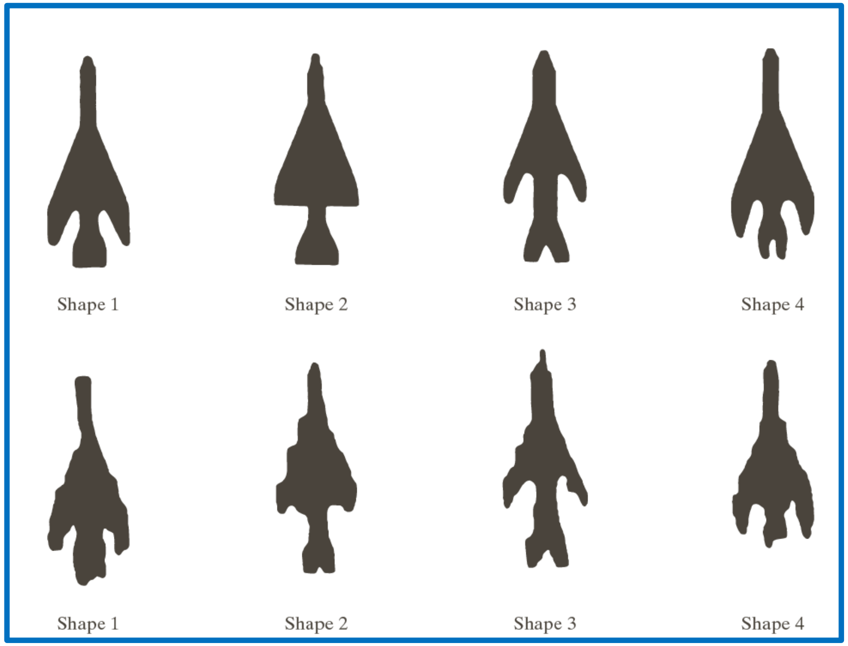
$$\frac{\partial}{\partial b_i^{(l)}} J(\mathbf{W}, b; \mathbf{x}, y) = \delta_i^{(l+1)}$$

  - 更新参数：

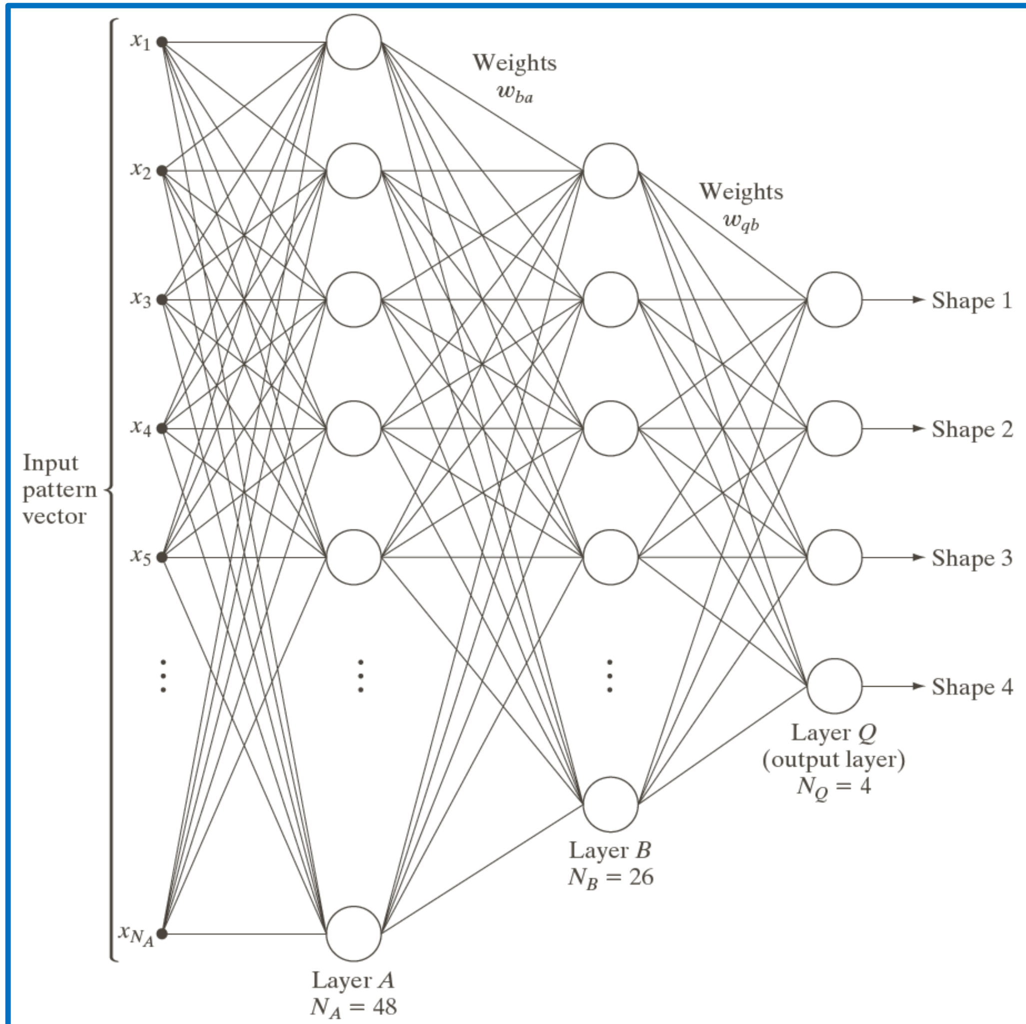
$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \left[\frac{1}{m} a_j^{(l)} \delta_i^{(l+1)} + \lambda W_{ij}^{(l)} \right]$$

$$b_i^{(l)} = b_i^{(l)} - \frac{\alpha}{m} \delta_i^{(l+1)}$$

12.2.3 神经网络

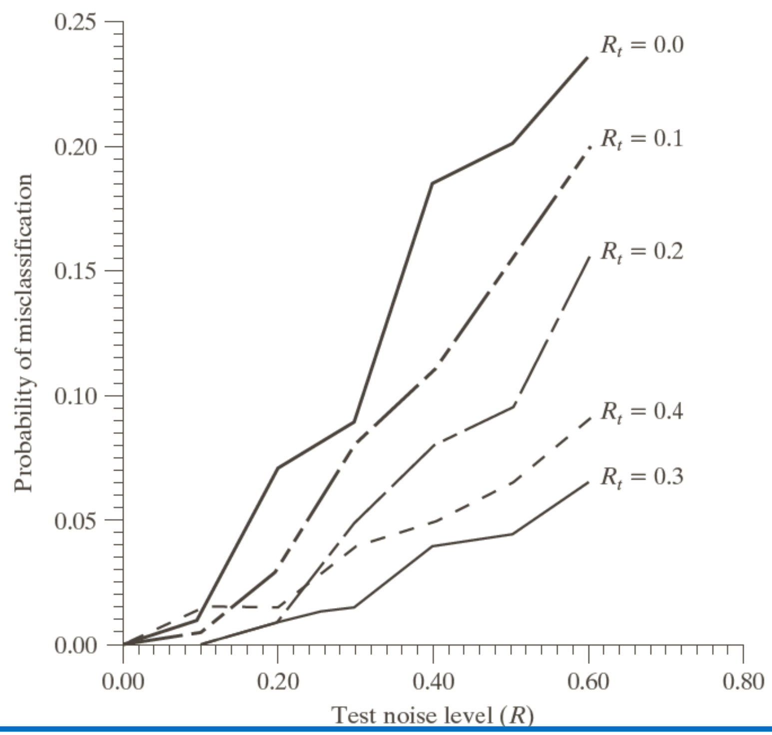


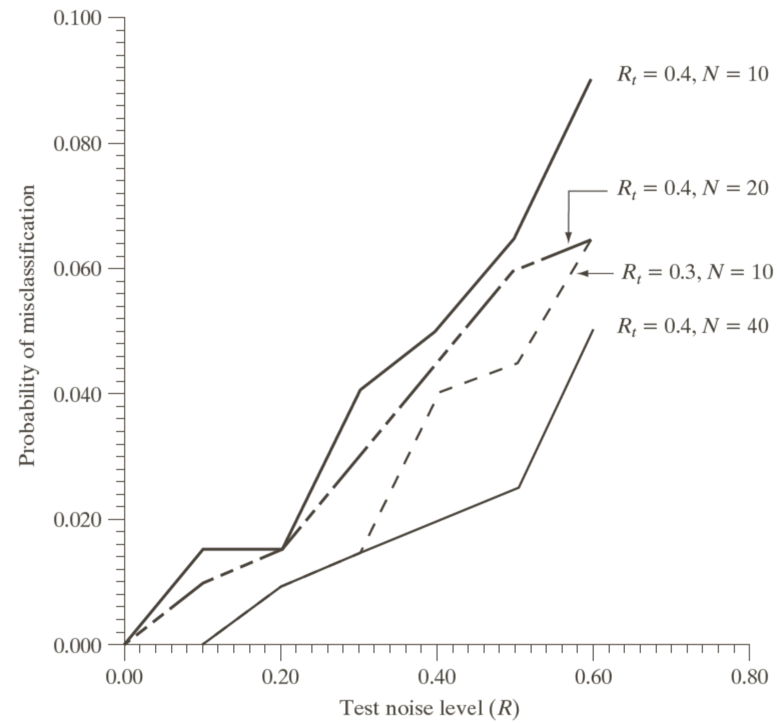
训练右图中的神经网络时所使用的（上）参考图形和（下）典型的带噪声图形



用于识别左图所示形状的三层神经网络

12.2.3 神经网络



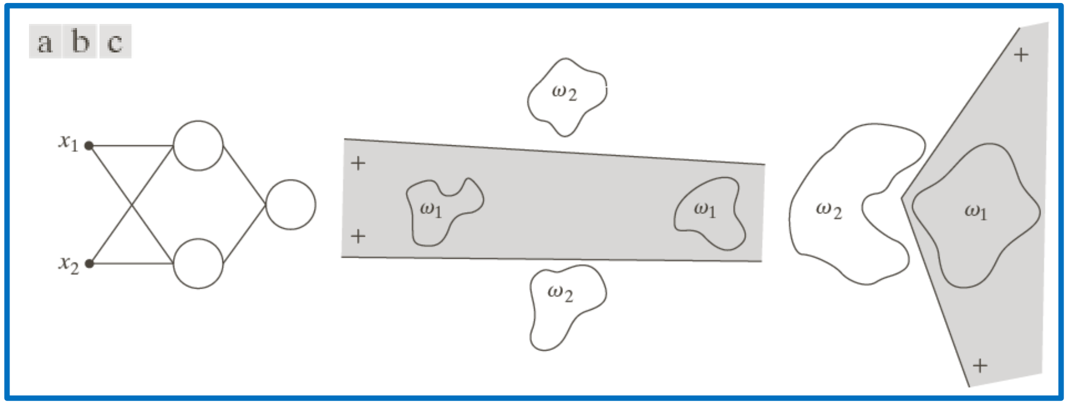


神经网络的性能是噪声水平的函数

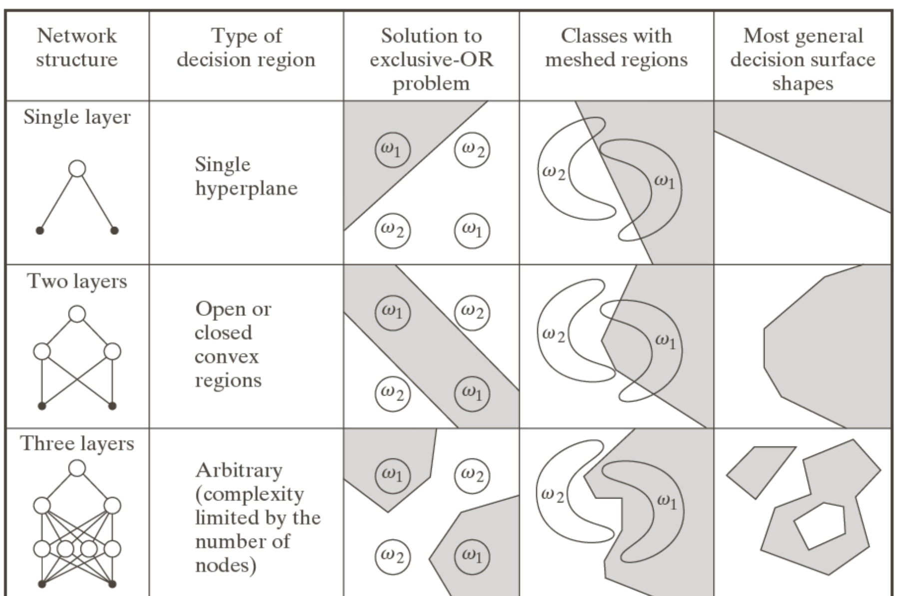
增加训练模式的数量时，$R_t = 0.4$时的性能改进（$R_t = 0.3$时的曲线仅作为参考）

12.2.3 神经网络

决策面的复杂性



(a)一个双输入双层前馈神经网络；
(b)和(c)可使用该网络实现的决策边界示例



可以由单层和多层前馈网络带有一层或两层隐藏单元与两个输入形成的决策区域的类型

第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配

12.3 结构方法

- 匹配形状数

$$s_j(a) = s_j(b), \qquad i = 4,6,8,\dots,k$$
$$s_j(a) \neq s_j(b), \qquad j = k+2, k+4, \dots$$

距离

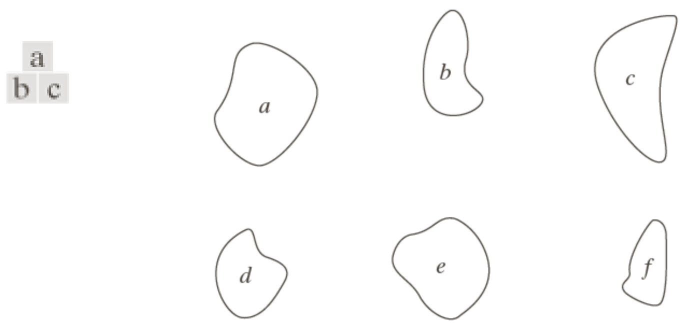
$$D(a,b) = \frac{1}{k}$$

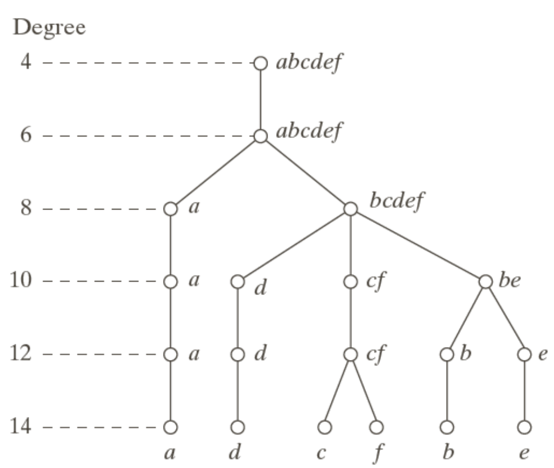
性质

$$D(a,b) \geq 0$$
$$D(a,b) = 0, \qquad if\ a = b$$
$$D(a,c) \leq \max[(D(a,b), D(b,c))]$$

12.3 结构方法

匹配形状数





	a	b	c	d	e	f
a	∞	6	6	6	6	6
b		∞	8	8	10	8
c			∞	8	8	12
d				∞	8	8
e					∞	8
f						∞

(a) 各种形状；
(b) 假想的相似树；
(c) 相似性矩阵