模式分类若可用任一个线性函数来划分，则这些模式就称为线性可分的，否则就是非线性可分的。一旦线性函数的系数 wk 被确定，这些函数就可用作模式分类的基础。

对于 M 类模式的分类，多类情况 1 需要 M 个判别函数，而多类情况 2 需要 M*(M-1)/2 个判别函数，当 M 较大时，后者需要更多的判别式（这是多类情况 2 的一个缺点）。

采用多类情况 1 时，每一个判别函数都要把一种类别的模式与其余 M-1 种类别的模式分开，而不是将一种类别的模式仅与另一种类别的模式分开。

由于一种模式的分布要比 M-1 种模式的分布更为聚集，因此多类情况 2 对模式是线性可分的可能性比多类情况 1 更大一些（这是多类情况 2 的一个优点）。

答（2）广义线性判别函数出发点：

- 线性判别函数简单，容易实现；
- 非线性判别函数复杂，不容易实现；
- 若能将非线性判别函数转换为线性判别函数，则有利于模式分类的实现。

**采用广义线性判别函数的概念，可以通过增加维数来得到线性判别，但维数的大量增加会使在低维空间里在解析和计算上行得通的方法在高维空间遇到困难，增加计算的复杂性。所以某些情况下使用非线性判别函数或分段线性判别函数效果更好。**

解（3）假设该两类模式是线性可分的，则在三维空间中一个线性平面可以将这两类模式分开，所以判别函数可以写成：

$$d(x) = w_1 x + w_2 x + w_3 x + w_4$$

所以权向量需要 4 个系数。

对于 n 维 x 向量，采用 r 次多项式，d(x)的权系数 w 的项数为：

$$N_w = C_{n+r}^r = \frac{(n+r)!}{r!n!}$$

当 r=2，n=3 时，

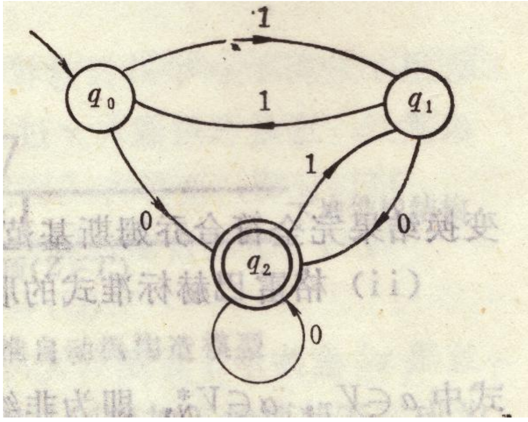
$$N_W = \frac{(n+2)!}{2!n!} = \frac{(n+2)(n+1)}{2} = 10$$

所以，此时权向量需要 10 个系数分量。

**5. 设一有限态自动机 $A = (\{0,1\}, \{q_0, q_1, q_2\}, \delta, q_0, q_2\}$，$\delta$ 定义如下：**

$$\delta(q_0,0) = q_2, \delta(q_1,0) = q_2, \delta(q_2,0) = q_2$$
$$\delta(q_0,1) = q_1, \delta(q_1,1) = q_0, \delta(q_2,1) = q_1$$

**试求等价的正则文法，使得 L(G)=T(A)。（10’）**




解：设由 A 得一正则文法 $G=(V_N,V_T,P，S)$，则 $V_N=\{S,x_1,x_2\}$，$V_T=\{0,1\}$，$S=q_0$

由 $\delta(q_0,1)=q_1$，得生成式 $S\longrightarrow 1x_1$

由 $\delta(q_0,0)=q_2$，得生成式 $S\longrightarrow 0,S\longrightarrow 0x_2$

由 $\delta(q_1,1)=q_0$，得生成式 $x_1\longrightarrow 1S$

由 $\delta(q_1,0)=q_2$，得生成式 $x_1\longrightarrow 0,x_1\longrightarrow 0x_2$

由 $\delta(q_2,1)=q_1$，得生成式 $x_2\longrightarrow 1x_1$

由 $\delta(q_2,0)=q_2$，得生成式 $x_2\longrightarrow 0,x_2\longrightarrow 0x_2$

对比实例：当扫描字符串 1110 时，A 按以下状态序列接受该字符串

$$q_0\xrightarrow{1}q_1\xrightarrow{1}q_0\xrightarrow{1}q_1\xrightarrow{0}q_2$$

用对应的正则文法 G 推导，得：

$$S\Rightarrow 1x_1\Rightarrow 11S\Rightarrow 111x_1\Rightarrow 1110$$

- 按有限态自动机确定正则文法

给定一个有限态自动机 $A=(\Sigma,Q,\delta,q_0,F)$，可确定一个正则文法 $G=(V_N,V_T,P,S)$，使得 L(G) = T(A)。

由 $Q=\{q_0,q_1,...,q_n,q_{n+1}\},q_{n+1}\in F$ ，可确定：$V_N=\{S,x_1,x_2,...,\mathrm{x}_n,x_{n+1}\}$，$S=q_0$，$x_i=q_i$，$V_T=\Sigma$。

从 $\delta$ 求 G 中的生成式 P 可按如下原则：

(1) 若 $\delta(q_i,a)=q_j$，则 $x_i\rightarrow ax_j$

(2) 若 $\delta(q_i,a)=q_{n+1}$，则 $x_i\rightarrow a,x_i\rightarrow ax_{n+1}$

**6. K-均值算法聚类：K=2，初始聚类中心为$x_1, x_2$，数据为：（10'）**

$$\{x_1=(0,0), x_2=(1,0), x_3=(0,1), x_4=(1,1), x_5=(8,7)$$
$$x_6=(9,7), x_7=(8,8), x_8=(9,8), x_9=(8,9), x_{10}=(9,9)\}$$

算法：

第一步：选$K$个初始聚类中心，$z_1(1), z_2(1), ..., z_k(1)$，其中括号内的序号为寻找聚类中心的迭代运算的次序号。可选开始的$K$ 个模式样本的向量值作为初始聚类中心。

第二步：逐个将需分类的模式样本$\{x\}$按最小距离准则分配给$K$个聚类中心中的某一个$z_j(1)$。即$D_j(k)=\min\{\|x-z_i(k)\|, i=1,2,\cdots K\}$，则$x\in S_j(k)$，其中$k$ 为迭代运算的次序号，第一次迭代$k=1$，$S_j$表示第$j$个聚类，其聚类中心为$z_j$。

第三步：计算各个聚类中心的新的向量值，$z_j(k+1), j=1,2,...,K$

求各聚类域中所包含样本的均值向量：

$$z_j(k+1)=\frac{1}{N_j}\sum_{x\in S_j(k)} x, \quad j=1,2,\cdots,K$$

其中$N_j$为第$j$个聚类域$S_j$中所包含的样本个数。**以均值向量作为新的聚类中心**，可使如下聚类准则函数最小：

$$J_j=\sum_{x\in S_j(k)}\|x-z_j(k+1)\|^2, \quad j=1,2,\cdots,K$$

**在这一步中要分别计算K个聚类中的样本均值向量，所以称之为K-均值算法。**

第四步：若$z_j(k+1)\neq z_j(k)$，则返回第二步，将模式样本逐个重新分类，重复迭代运算；

若$z_j(k+1)=z_j(k)$，则算法收敛，计算结束。

**7. 给出两类模式分布，每一列代表一个样本：**

$$\omega_1 \;:\; \boldsymbol{x_1}=\begin{pmatrix}-5 & -5 & -4 & -5 & -6\\ -5 & -4 & -5 & -6 & -5\end{pmatrix}$$

$$\omega_2 \;:\; \boldsymbol{x_2}=\begin{pmatrix}5 & 5 & 6 & 5 & 4\\ 5 & 6 & 5 & 4 & 5\end{pmatrix}$$

**试用K-L变换来做一维特征的提取（12'）。**

解：首先将所有样本看作一个整体，求出样本均值向量：

$$\boldsymbol{m} = \frac{1}{5}\sum_{j=1}^{5}\boldsymbol{x_{1j}} + \frac{1}{5}\sum_{j=1}^{5}\boldsymbol{x_{2j}} = \boldsymbol{0}$$

由于均值为 0，符合 K-L 变换的最佳条件。如果均值不为 0，则所有样本要减去均值向量。由于 $\omega_1$ 和 $\omega_2$ 的样本数相同，所以认为他们的先验概率相同，即：

$$P(\omega_1) = P(\omega_2) = 0.5$$

求出总体的自相关矩阵 $R$ 或协方差矩阵 $C$：

$$R = \sum_{i=1}^{2} P(\omega_i) E\{x_i x_i^T\} = \begin{pmatrix} 25.4 & 25 \\ 25 & 25.4 \end{pmatrix}$$

解特征方程 $|R - \lambda I| = 0$，求出 $R$ 的特征值：

$$\lambda_1 = 50.4, \lambda_2 = 0.4$$

求出对应于特征值的特征向量 $R\phi_i = \lambda_i \phi_i$：

$$\phi_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \phi_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

选取 $\lambda_1$ 对应的特征向量作为变换矩阵 $\Phi$，由 $\boldsymbol{y} = \Phi^T \boldsymbol{x}$ 得出变换后的一维模式：

$$\omega_1 \ : \ \boldsymbol{x_1'} = \begin{pmatrix} -\frac{10}{\sqrt{2}} & -\frac{9}{\sqrt{2}} & -\frac{9}{\sqrt{2}} & -\frac{11}{\sqrt{2}} & -\frac{11}{\sqrt{2}} \end{pmatrix}$$

$$\omega_2 \ : \ \boldsymbol{x_2'} = \begin{pmatrix} \frac{10}{\sqrt{2}} & \frac{11}{\sqrt{2}} & \frac{11}{\sqrt{2}} & \frac{9}{\sqrt{2}} & \frac{9}{\sqrt{2}} \end{pmatrix}$$

**8. 用第二类势函数的算法进行分类（10’）**

**选择指数型势函数，取α=1，在二维情况下势函数为：**

$$K(x, x_k) = e^{-\|x - x_k\|^2} = e^{-[(x_1 - x_{k_1})^2 + (x_2 - x_{k_2})^2]}$$

**这里：ω₁类为 x①=(0 0)$^T$, x②=(2 0)$^T$；ω₂类为 x③=(1 1)$^T$, x④=(1 -1)$^T$**

解：可以看出，这两类模式是线性不可分的。算法步骤如下：

第一步：取 $x_{(1)} = (0,0)^T \in \omega_1$，则

$$K_1(x) = K(x, x_{(1)}) = \exp\{-[(x_1 - 0)^2 + (x_2 - 0)^2]\} = \exp[-(x_1^2 + x_2^2)]$$

第二步：取 $x_{(2)} = (2,0)^T \in \omega_1$

因 $\exp[-(4+0)] = \exp(-4) > 0$，

故$K_2(x)=K_1(x)=\exp[-(x_1^2+x_2^2)]$

第三步：取$x_{(3)}=(1,1)^T\in\omega_2$

因$\exp[-(1+1)]=\exp(-2)>0$，

故

$$K_3(x)=K_2(x)-K(x,x_{(3)})=\exp[-(x_1^2+x_2^2)]-\exp\{-[(x_1-1)^2+(x_2-1)^2]\}$$

……

后面同理，就是不断将样本带入，如果分类正确，则势函数保持不变，即：

$$K_{k+1}(x)=K_k(x)$$

如果分类错误，则有两种情况：

- $x_{(k+1)}\in\omega_1,K_k(x_{(k+1)})\le 0$，则$K_{k+1}(x)=K_k(x)+K(x,x_{(k+1)})$
- $x_{(k+1)}\in\omega_2,K_k(x_{(k+1)})\ge 0$，则$K_{k+1}(x)=K_k(x)-K(x,x_{(k+1)})$

经过迭代，全部模式都已正确分类，因此算法收敛于判别函数。

得出：$d(x)=e^{-(x_1^2+x_2^2)}-e^{-[(x_1-1)^2+(x_2-1)^2]}-e^{-[(x_1-1)^2+(x_2+1)^2]}+e^{-[(x_1-2)^2+x_2^2]}$

**9. 有一种病，正常为$\omega_1$ ，不正常为$\omega_2$ ，已知：**

$$P(\omega_1)=0.9,P(\omega_2)=0.1$$

**现对某人进行检查，结果为$x$，由概率曲线查出：**

$$P(x|\omega_1)=0.2,P(x|\omega_2)=0.4$$

**风险代价矩阵为：**

$$L=\begin{bmatrix}L_{11} & L_{12}\\ L_{21} & L_{22}\end{bmatrix}=\begin{bmatrix}0 & 6\\ 1 & 0\end{bmatrix}$$

**对该检查者进行判决：**

**（1） 用贝叶斯最小错误概率判别，求出判决函数和决策分界面。**

**（2） 用贝叶斯最小风险判别，求出判别函数和决策分界面。**

解（1）：

$$P(\omega_1|x)\propto P(\omega_1)P(x|\omega_1)$$
$$P(\omega_2|x)\propto P(\omega_2)P(x|\omega_2)$$

由于

$$l=\frac{P(x|\omega_1)}{P(x|\omega_2)}=\frac{1}{2}>\frac{P(\omega_2)}{P(\omega_1)}=\frac{1}{9}$$

所以$x\in\omega_1$。

解（2）：

$$r_j(x)=\sum_{i=1}^{2}L_{ij}P(x\,|\,\omega_i)P(\omega_i),\ j=1,2$$

由于

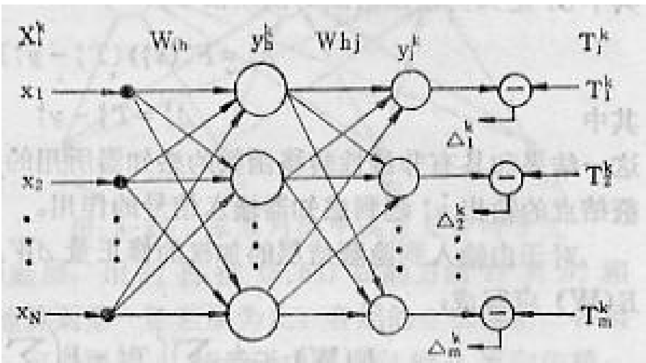
$$l'=\frac{P(x\,|\,\omega_1)}{P(x\,|\,\omega_2)}=\frac{1}{2}>\frac{P(\omega_2)}{P(\omega_1)}\frac{L_{21}-L_{22}}{L_{12}-L_{11}}=\frac{1}{54}$$

所以 $x\in\omega_1$。

## 10. 阐述误差反传算法（BP 算法）的原理，并写出其训练步骤。

答（1）：

- **BP 算法推算过程：**




当加入第 k 个输入时，隐蔽层 h 结点的输入加权和为：

$$s_h^k=\sum_i w_{ih}x_i^k$$

如果令第一层的加权矩阵为 $W_1$ ，则还可以表示为：

$$\boldsymbol{s}_h^k=\boldsymbol{W}_1^T\boldsymbol{x}^k$$

相应节点的输出为：

$$y_h^k=F(s_h^k)=F(\sum_i w_{ih}x_i^k)$$

写成矩阵形式为：

$$\boldsymbol{y}_h^k=F(\boldsymbol{s}_h^k)=F(\boldsymbol{W}_1^T\boldsymbol{x}^k)$$

同样，输出层 j 结点的输入加权和为：

$$s_j^k=\sum_h w_{hj}y_h^k=\sum_h w_{hj}F(\sum_i w_{ih}x_i^k)$$

令第二次的加权矩阵为 $W_2$，则可以写成：

$$\boldsymbol{s}_j^k = \boldsymbol{W}_2^T \boldsymbol{y}_h^k = \boldsymbol{W}_2^T F(\boldsymbol{W}_1^T \boldsymbol{x}^k)$$

相应点的输出：

$$y_j^k = F(s_j^k) = F(\sum_h w_{hj} y_h^k) = F[\sum_h w_{hj} F(\sum_i w_{ih} x_i^k)]$$

写成矩阵形式为：

$$\boldsymbol{y}_j^k = F(\boldsymbol{W}_2^T F(\boldsymbol{W}_1^T \boldsymbol{x}^k))$$

这里，各结点的阈值等效为一个连接的加权 $\theta = w_{0h}$ 或 $w_{0j}$，这些连接由各结点连到具有固定值-1 的偏置结点，其连接加权也是可调的，同其它加权一样参与调节过程。

误差函数为：

$$E(W) = \frac{1}{2}\sum_{k,j}(T_j^k - y_j^k)^2 = \frac{1}{2}\sum_{k,j}\{T_j^k - F[\sum_h w_{hj} F(\sum_i w_{ih} x_i^k)]\}^2$$

**为了使误差函数最小，用梯度下降法求得最优的加权，权值先从输出层开始修正，然后依次修正前层权值，因此含有反传的含义。根据梯度下降法，由隐蔽层到输出层的连接的加权调节量为：**

$$\Delta w_{hj} = -\eta \frac{\partial E}{\partial w_{hj}} = \eta \sum_k (T_j^k - y_j^k) F'(s_j^k) y_h^k = \eta \sum_k \delta_j^k y_h^k$$

其中 $\delta_{\mathrm{j}}^{\mathrm{k}}$ 为输出结点的误差信号：

$$\delta_j^k = F'(s_j^k)(T_j^k - y_j^k) = F'(s_j^k)\Delta_j^k$$

$$\Delta_j^k = T_j^k - y_j^k$$

在 BP 算法中常采用 Sigmoid 函数：$y = F(s) = \dfrac{1}{1+e^{-s}}$

其导数为：$F'(s) = F(s)(1-F(s)) = y(1-y)$

对应的误差为：$\delta_j^k = y_j^k(1-y_j^k)(T_j^k - y_j^k)$

对于输入层到隐蔽层结点连接的加权修正量 $\Delta w_{ih}$，必须考虑将 $E(W)$ 对 $w_{ih}$ 求导，因此利用分层链路法，有：

$$\begin{aligned}\Delta w_{ih} &= -\eta \frac{\partial E}{\partial w_{ih}} = -\eta \sum_k \frac{\partial E}{\partial y_h^k} \cdot \frac{\partial y_h^k}{\partial w_{ih}} = \eta \sum_{k,j}\{(T_j^k - y_j^k) F'(s_j^k) w_{hj} \cdot F'(s_h^k) x_i^k\} \\ &= \eta \sum_{k,j} \delta_j^k w_{hj} F'(s_h^k) x_i^k = \eta \sum_k \delta_h^k x_i^k\end{aligned}$$

其中：

$$\delta_h^k = F'(s_h^k)\sum_j w_{hj}\delta_j^k = F'(s_h^k)\Delta_h^k$$

$$\Delta_h^k = \sum_j w_{hj}\delta_j^k$$

这样就可以根据 $\Delta w_{hj}$ 和 $\Delta w_{ih}$ 分别调整输出层和隐层的权值了。

- **BP 训练算法实现步骤**

准备：设网络具有 m 层，$y_j^m$ 表示第 m 层中第 j 个结点的输出，$y_j^0$（零层输出）等于 $x_j$，即第 j 个输入。$w_{ij}^m$ 表示从 $y_i^{m-1}$ 到 $y_j^m$ 的连接加权。这里，m 代表层号，而不是向量的类号。

1.（**初始化加权矩阵**）将各加权随机置为小的随机数。可用均匀分布的随机数，以保证网络不被大的加权值所饱和。

2.（**输入数据**）从训练数据组中选一数据对 $(x^k, T^k)$，将输入向量加到输入层（m=0），使得对所有端点 i：$y_i^0 = x_i^k$，k 表示向量类号。

3.（**输出预测数据**）信号通过网络向前传播，即利用关系式：

$$y_j^m = F(s_j^m) = F(\sum_i w_{ij}^m y_i^{m-1})$$

计算从第一层开始的各层内每个结点 i 的输出 $y_j^m$，直到输出层的每个结点的输出计算完为止。

4.（**计算输出层误差**）计算输出层每个结点的误差值，对 Sigmod 函数：

$$\delta_j^m = F'(s_j^m)(T_j^k - y_j^m) = y_j^m(1 - y_j^m)(T_j^k - y_j^m)$$

它是由实际输出和要求目标值之差获得。

5.（**误差反传**）计算前面各层各结点的误差值

$$\delta_j^{m-1} = F'(s_j^{m-1})\sum_i w_{ji}\delta_i^m$$

这里逐层计算反传误差，直到将每层内每个结点的误差值算出为止。

6.（**修改权值**）利用加权修正公式：

$$\Delta w_{ij}^m = \eta\delta_j^m y_i^{m-1}$$

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}$$

修正所有连接权。一般 $\eta = 0.01 \sim 1$，称为训练速率系数。

7.（**运算至权值收敛**）返回第 2 步，为下一个输入向量重复上述步骤，直至网络收敛。