

神经网络 Neural Networks

第九章

模糊神经网络

史忠植

中国科学院计算技术研究所
<http://www.intsci.ac.cn/>

内容提要



- 9.1 概述
- 9.2 算术模糊神经网络
- 9.3 模糊逻辑
- 9.4 模糊联想记忆
- 9.5 神经模糊推理系统
- 9.6 神经网络近似逻辑

模糊神经网络

ANN (Artificial Neural Network) 和 FLS (Fuzzy Logical Network) 的比较:

- 相同之处
- 1) 都是非数值型的非线性函数的逼近器、估计器、和动态系统;
 - 2) 不需要数学模型进行描述, 但都可用数学工具进行处理;
 - 3) 都适合于VLSI、光电器件等硬件实现。

不同之处: (一) 工作机制方面:

ANN——大量、高度连接, 按样板进行学习

FLS——按语言变量、通过隐含、推理和去模糊获得结果。

(二) 信息处理基本单元方面:

ANN——数值点样本, x_i, y_i

FLN——模糊集合 (A_i, B_i)

(三) 运行模式方面:

ANN——学习过程透明, 不对结构知识编码

FLN——不透明, 对结构知识进行编码, 推理过程外界

可知

(四) 应用上:

ANN——偏重于模式识别, 分类

FLN —— 偏重于控制

神经模糊网络——把ANN的学习机制和FLN的人类思维和推理结合起来。

结合方式有3种:

- 1) 神经模糊系统——用神经元网络来实现模糊隶属函数、模糊推理，基本上（本质上）还是FLN。
- 2) 模糊神经网络——神经网络模糊化，本质上还是ANN。
- 3) 模糊-神经混合系统——二者有机结合。

● 基于神经网络的模糊逻辑运算

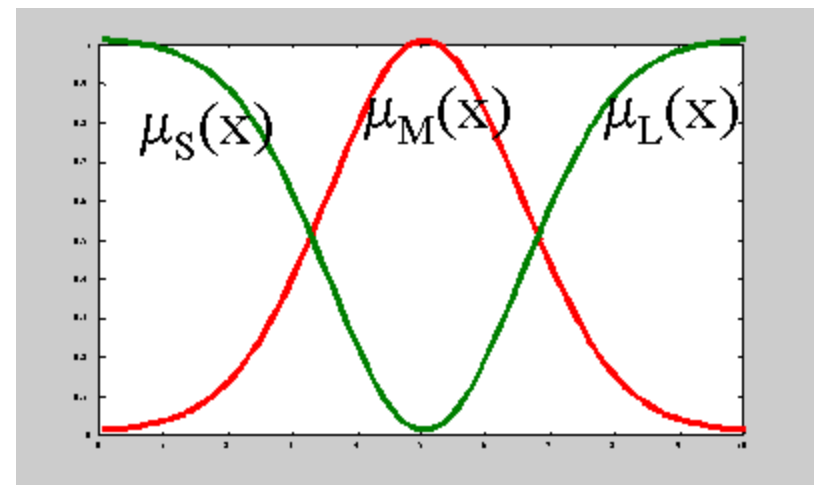
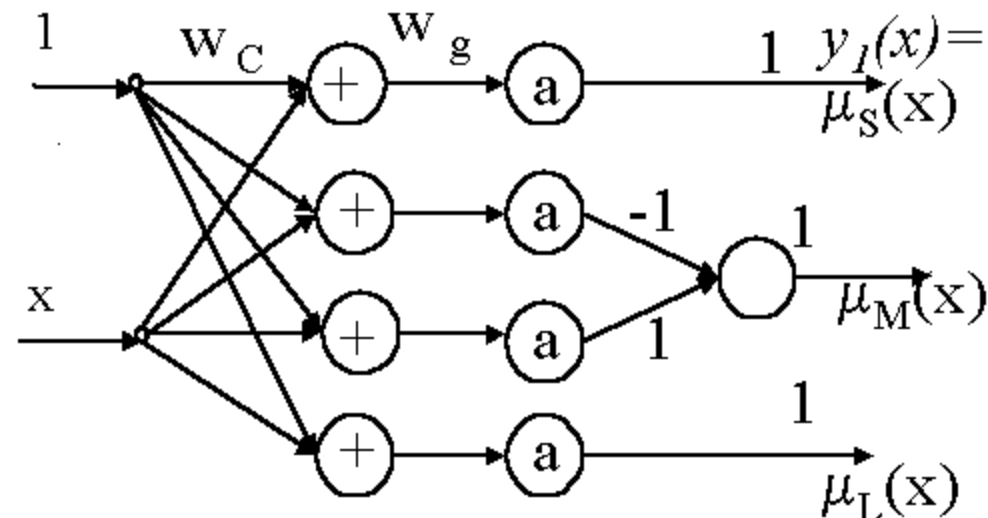
①用神经网络实现隶属函数

②神经网络驱动模糊推理

③神经网络的模糊建模

● 用神经网络实现隶属函数

$$y_1 = \mu_s(x) = \frac{1}{\{1 + \exp[-w_g(x + w_c)]\}}$$



a 是非线性函数，生成*sigmoid***函数。**

w_c 和 w_g 分别确定Sigmoid函数的中心和宽度
 $\mu_S(x), \mu_M(x), \mu_L(x)$

组成大、中、小三个论域的隶属函数。
 逻辑“与”可以用Softmin来实现：

$$(a \wedge b) = \text{soft min}(a, b) = \frac{ae^{-ka} + be^{-kb}}{e^{-ka} + e^{-kb}}$$

当 $k \rightarrow \infty$ 时， $\text{Softmin}(a, b) \rightarrow \text{Min}(a, b)$

●神经网络驱动模糊推理 (NDF)

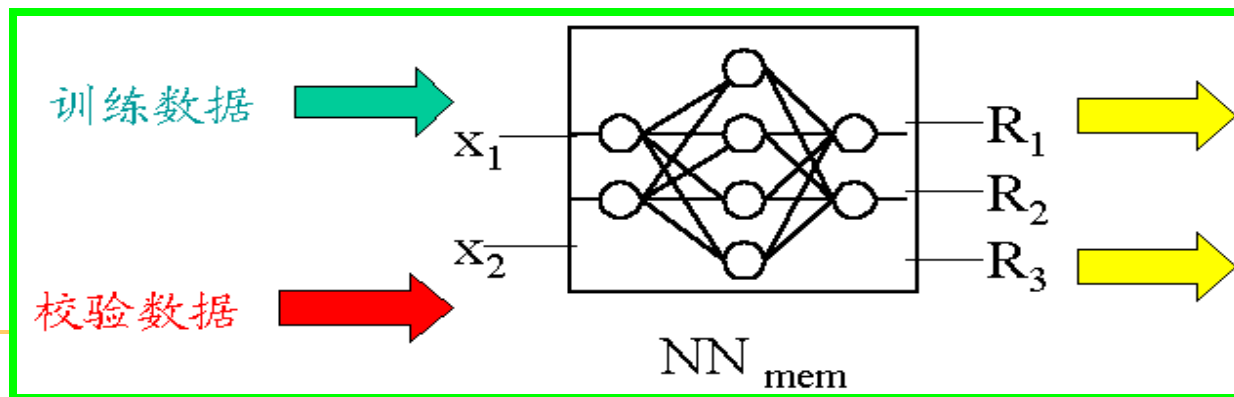
R^i : If X 是 A_i 和 Y 是 B_i , Then z 是 $f_i(x, y)$

解决模糊推理中二个问题：①缺乏确定的方法选择隶属函数；
②缺乏学习功能校正推理规则。

用神经网络实现T—S模型，称为神经网络驱动模糊推理(NDF).
网络由二部分组成：

R^s : If $X = (x_1, x_2, \dots, x_n)$ 是 A_s , Then $y_s = NN_s(x_1, x_2, \dots, x_n)$
 $s = 1, 2, \dots, r$

r 为规则数, A_s 是前提的模糊集合. NN_s 是模型的函数结构,由BP网络实现.



学习的网络和训练的步骤

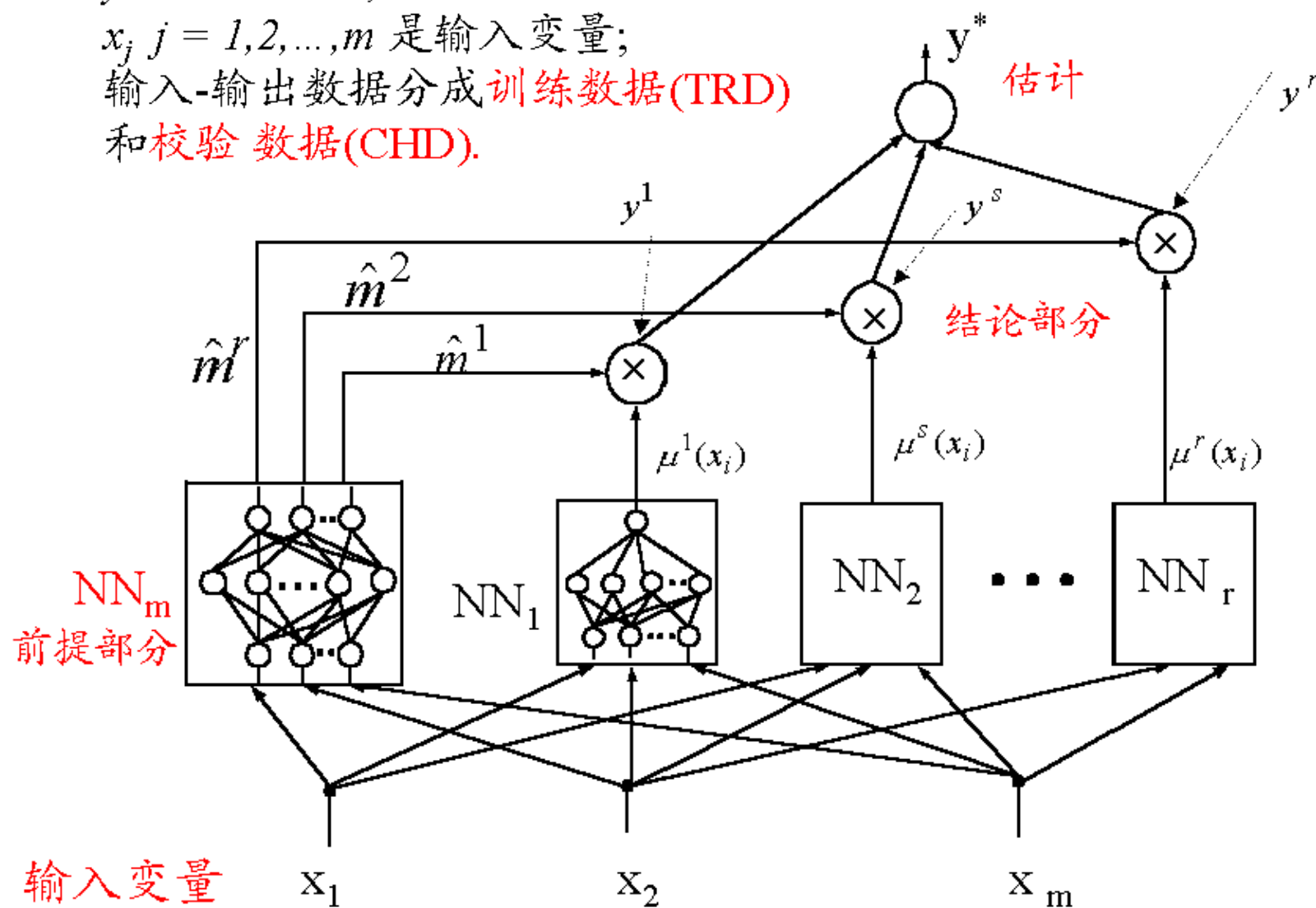
1) 选择输入-输出变量和训练数据.

y 为输出变量;

$x_j, j = 1, 2, \dots, m$ 是输入变量;

输入-输出数据分成训练数据(TRD)

和校验数据(CHD).



2) 将数据聚类. 分成 r 类. 即有 r 条规则. TRD的数据有 N_t 个.

3) 训练规则的前提部分网络 NN_m .

①对每个TRD输入向量 x_i , 定义一个向量:

$$m_i = (m_i^1, m_i^2, \dots, m_i^r)^T \text{ 使 } m_i^s = 1, m_i^k = 0, s \neq k.$$

②对数据对 (x_i, m_i) , $i = 1, 2, \dots, N_t$ 进行训练,

③ 推断校验数据 (CHD) x_i 对 R^s 的隶属度 $\hat{m}_i^s$

6

$$\text{即 } \mu_{A_s}(x_i) = \hat{m}_i^s, \quad i = 1, 2, \dots, N_c, \quad s = 1, 2, \dots, r.$$

4) 训练对应于规则 R^s 的后件部分 (Then部分) NN_s

NN_s 输入(TDR): $x_{i1}^s, x_{i2}^s, \dots, x_{im}^s \quad i = 1, 2, \dots, N_t$

NN_s 输出(TDR): y_i^s

6

训练以后, 将校验数据 (CHD) $x_{i1}, x_{i2}, \dots, x_{im} \quad i = 1, 2, \dots, N_c$

送到 NN_s , 计算误差:

$$E_m^s = \sum_{i=1}^{N_c} \{y_i - \mu_s(x_i)\mu_{A_s}(x_i)\}^2 \quad \text{或}$$

$$E_m^s = \sum_{i=1}^{N_c} \mu_{A_s}(x_i) \{y_i - \mu_s(x_i)\mu_{A_s}(x_i)\}^2 \quad (\text{加权})$$

6

5) 简化后件部分

在 NN_s 的输入端，任意消去 x_p ，比较误差：

$$E_m^s = \sum_{i=1}^{N_c} \{y_i - \mu_s(x_i)\mu_{A_s}(x_i)\}^2 \quad \text{消去前误差}$$

$$E_{m-1}^{sp} = \sum_{i=1}^{N_c} \{y_i - \mu_s(\hat{x}_i)\mu_{A_s}(\hat{x}_i)\}^2 \quad \text{消去后误差}$$

如果 $E_m^s > E_{m-1}^{sp}$ ，则 x_p 可以消去，否则另选变量

6) 最终输出

$$Y^* = \sum_{s=1}^r \mu_{A_s}(x_i)\mu_s(x_i) / \sum_{s=1}^r \mu_{A_s}(x_i)$$

●神经网络的模糊建模

有三种模型：

(1) 后件为恒值：

R^i : If x_1 是 A_{i_1} 和 x_2 是 A_{i_2} , Then $y = f_i$ ($i = 1, 2, \dots, n$)

$$y^* = \sum_{i=1}^n \mu_i f_i / \sum_{i=1}^n \mu_i = \sum_{i=1}^n \hat{\mu}_i f_i \quad i_1, i_2 = 1, 2, 3$$
$$\hat{\mu}_i(x_i) = \mu_i(x_i) / \sum_{i=1}^n \mu_i(x_i)$$



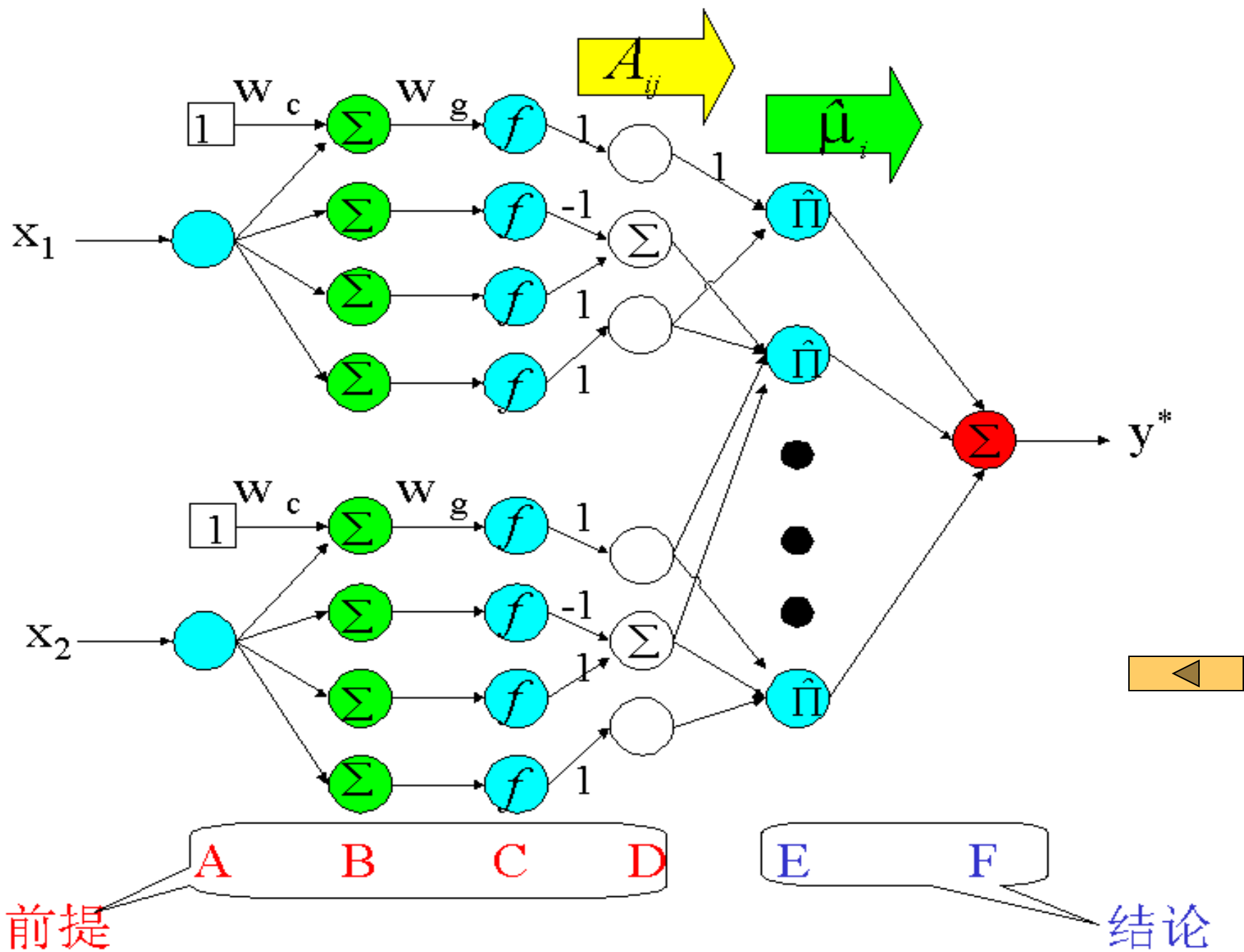
(2) 后件为一阶线性方程

R^i : If x_1 是 A_{i_1} 和 x_2 是 A_{i_2} , Then $y = f_i(x_1, x_2)$
 $i = 1, 2, \dots, n$

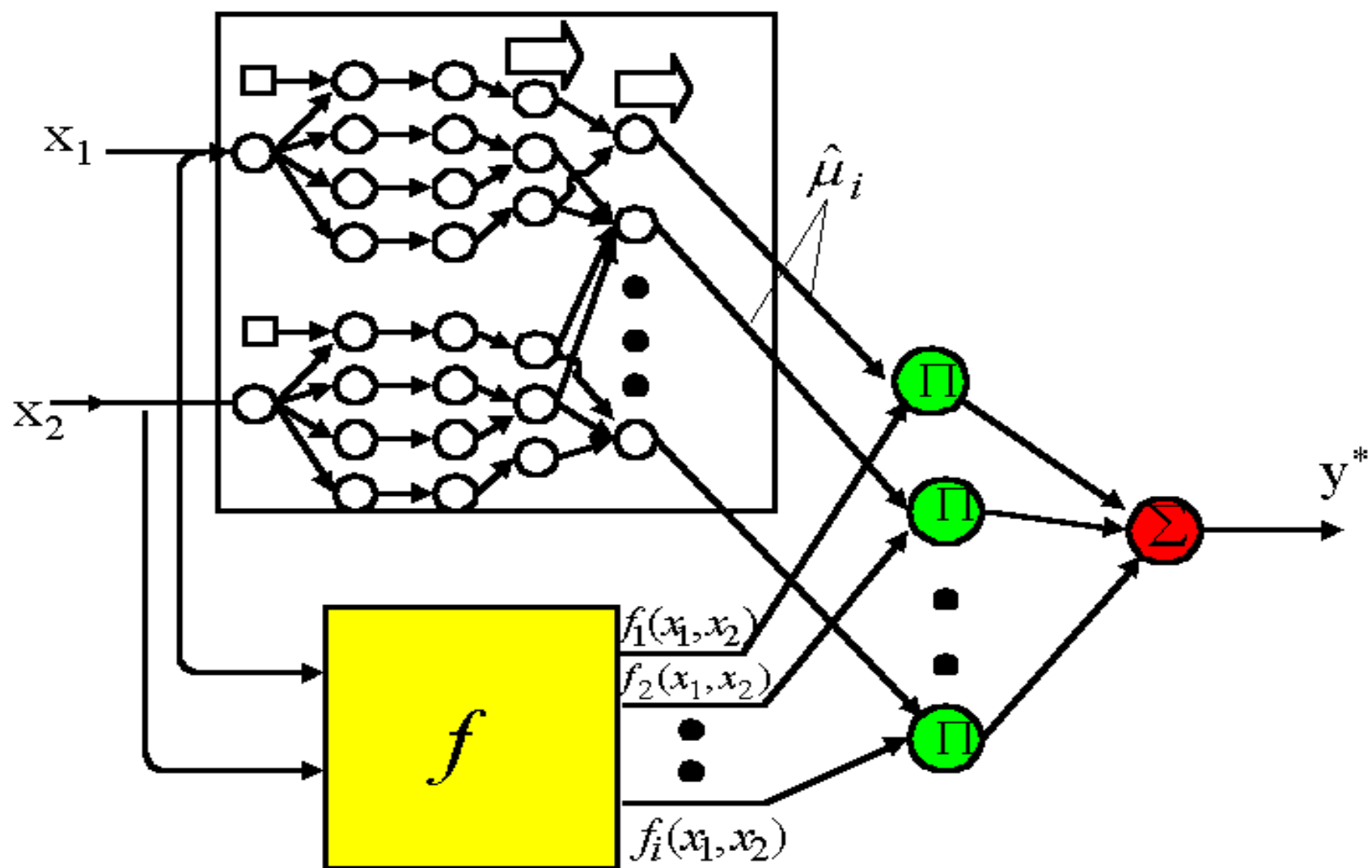
$$y^* = \sum_{i=1}^n \mu_i f_i(x_1, x_2) / \sum_{i=1}^n \mu_i = \sum_{i=1}^n \hat{\mu}_i f_i(x_1, x_2)$$



$f_i(x_1, x_2) = a_{i0} + a_{i1}x_1 + a_{i2}x_2$ a_{ij} ($j = 0, 1, 2$)是常数



前提部分



结论部分

(3)后件为模糊变量

$R_k^i : (If \ x_1 \text{是} A_{i_1} \text{和} x_2 \text{是} A_{i_2}, \text{Then } y \text{是} B_k) \text{是} \tau_{R_k^i}$

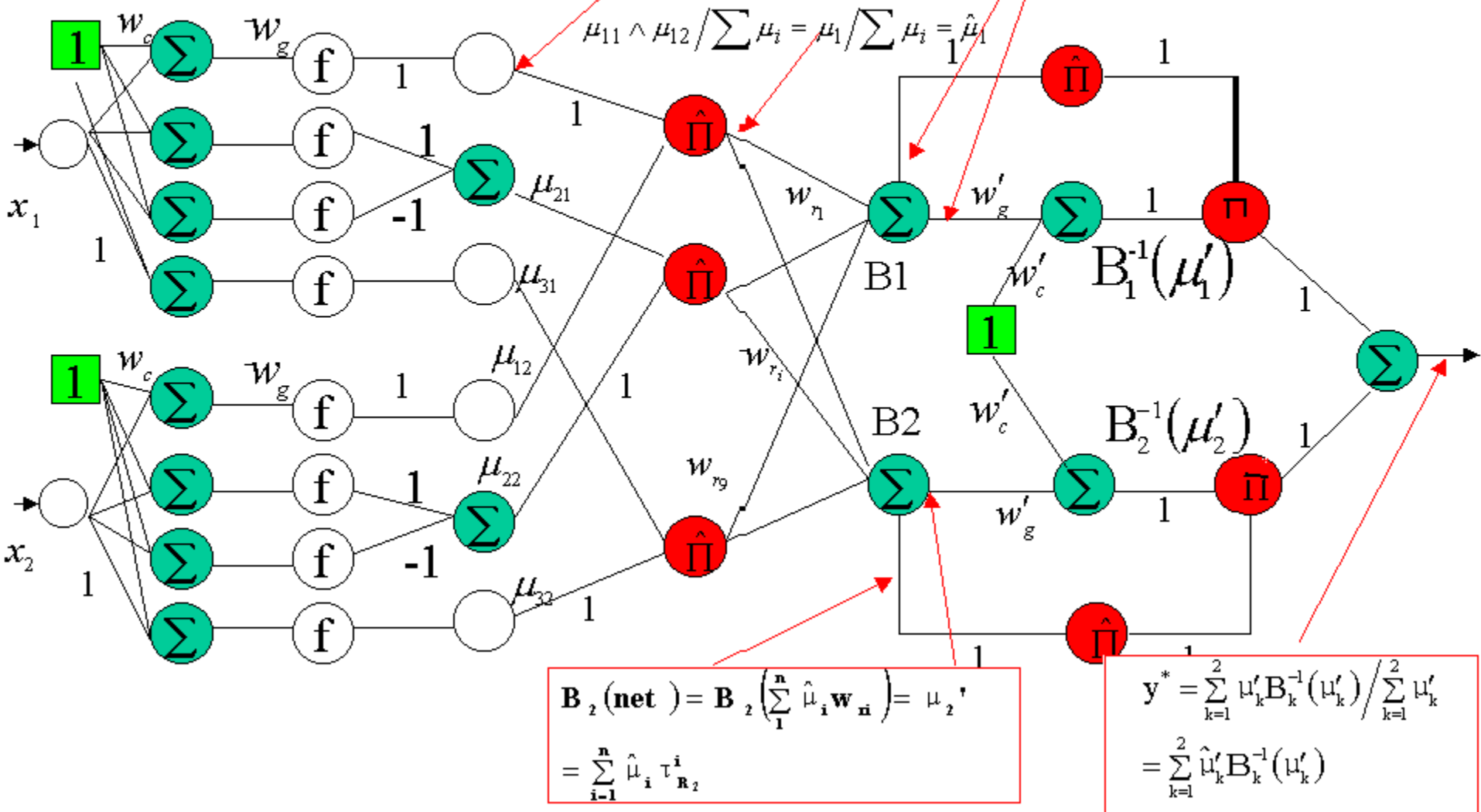
$$i = 1, 2, \dots, n. \quad k = 1, 2$$

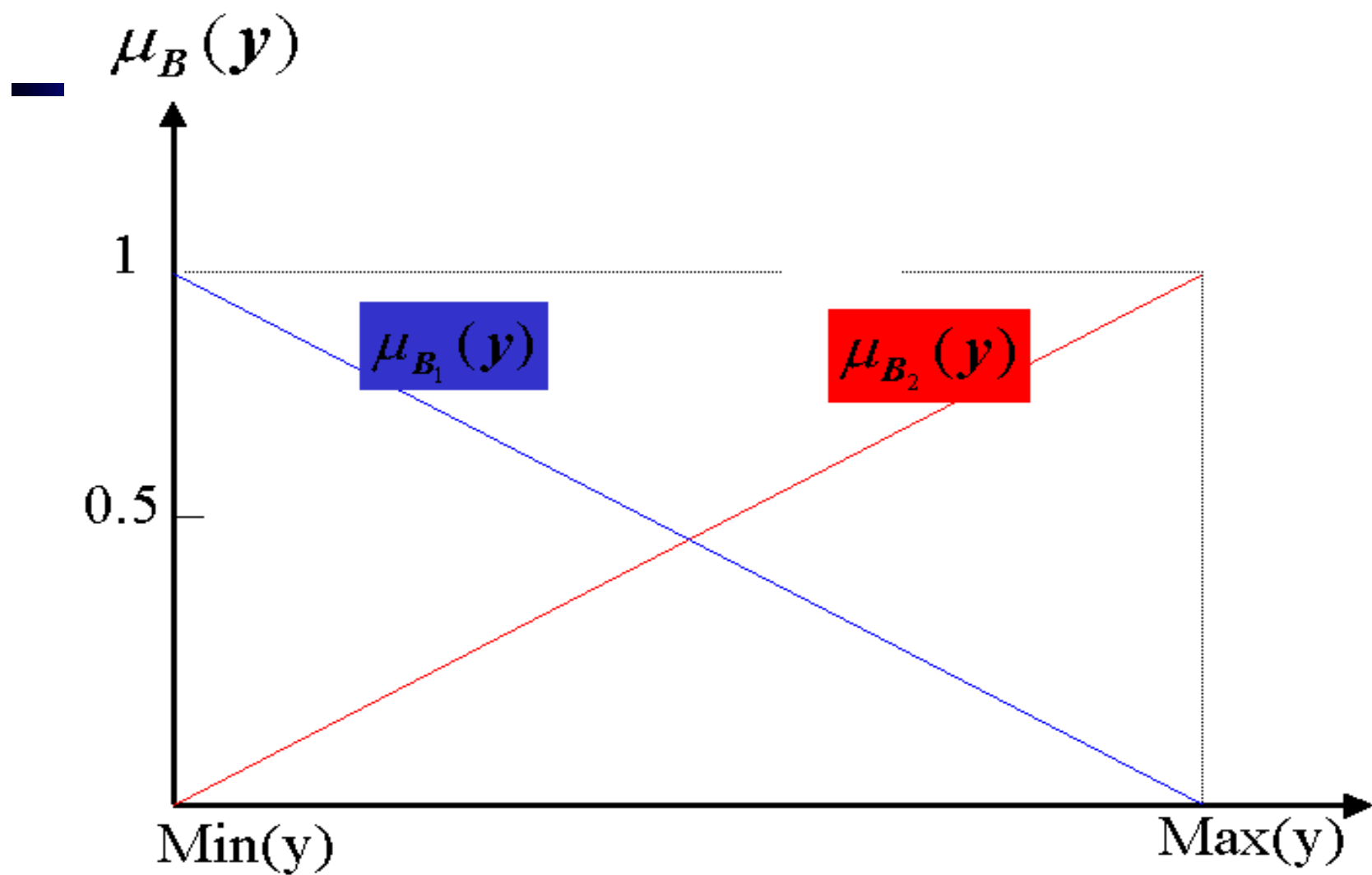
$$\mu_k' = \sum_{i=1}^n \hat{\mu}_i \tau_{R_k^i}$$

$$y^* = \sum_{k=1}^2 \mu_k' \mu_{B_k}^{-1}(\mu_k') / \sum_{k=1}^2 \mu_k' = \sum_{k=1}^2 \hat{\mu}_k' \mu_{B_k}^{-1}(\mu_k')$$

$$\mu_{11} = \frac{1}{\{1 + \exp[-w_g(x + w_c)]\}}$$

$$\begin{aligned} B_1(\text{net}) &= B_1\left(\sum_1^n \hat{\mu}_i w_i\right) = \mu_1' \\ &= \sum_{i=1}^n \hat{\mu}_i \tau_{R_1}^i \end{aligned}$$





应用

假定要辨识的系统为

$$g = (1 + x_1^{0.5} + x_2^{-1} + x_3^{1.5})^2$$

数据40 对，见表6.1

评判指标:

$$E_A = \sqrt{\sum_{i=1}^{n_A} (y_i^A - y_i^{AA})^2}$$

$$E_B = \sqrt{\sum_{i=1}^{n_B} (y_i^B - y_i^{BB})^2}$$

y_i^A : A 组数据所得的输出 (模型 A) ;

y_i^B : B 组数据所得的输出 (模型 B) ;

y_i^{AA}, y_i^{BA} : 用 A 组和 B 组数据, 输入到模型 A

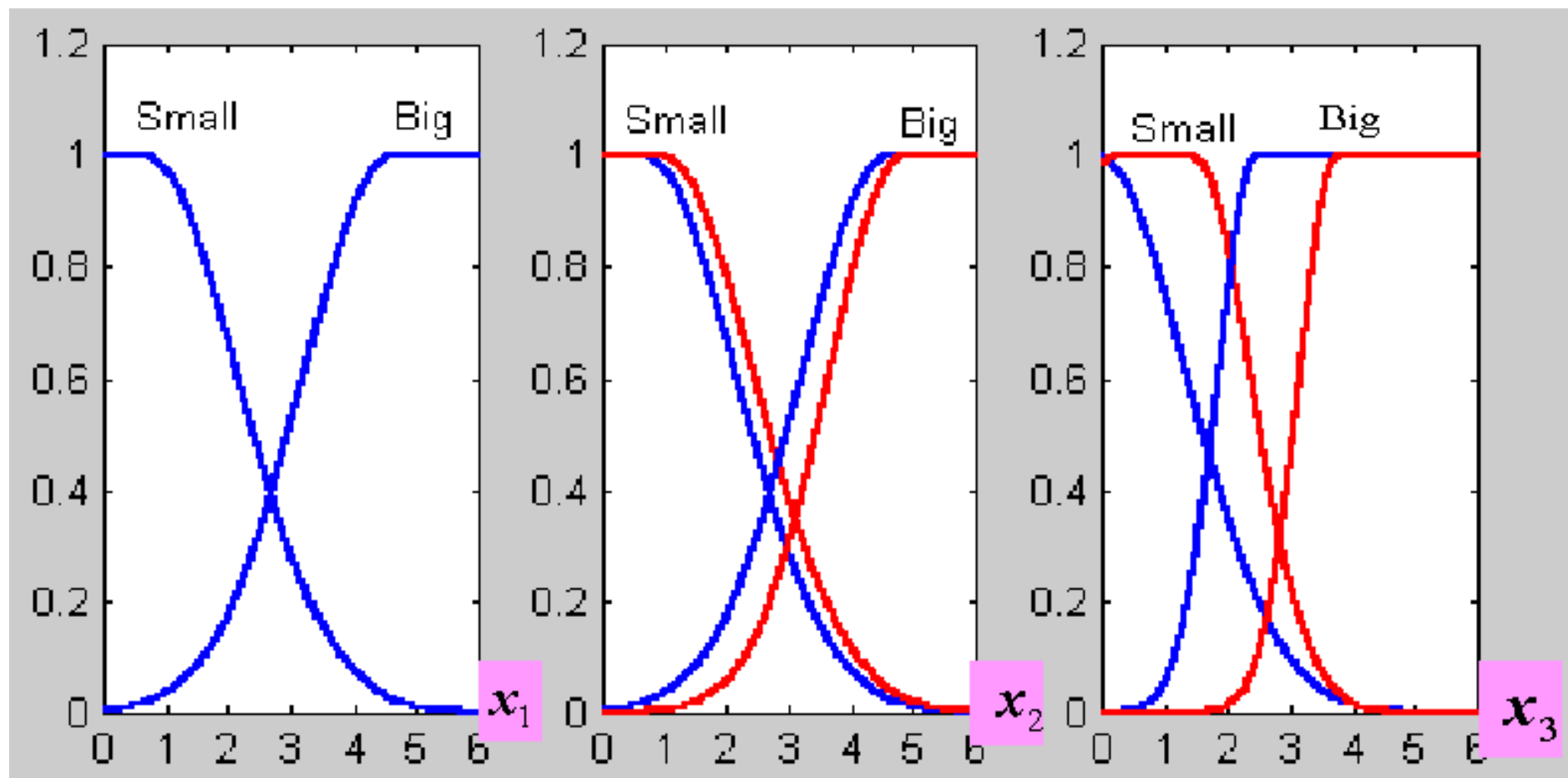
y_i^{BB}, y_i^{AB} : 用 A 组和 B 组数据, 输入到模型 B

$$UC = \sqrt{\sum_{i=1}^{n_A} (y_i^{AB} - y_i^{AA})^2 + \sum_{i=1}^{n_B} (y_i^{BA} - y_i^{BB})^2}$$

$$C = \sqrt{E_A^2 + E_B^2} + UC$$

前 件			后 件
x_1	x_2	x_3	f_i
<i>small</i>	<i>small</i>	<i>small</i>	15.9
<i>small</i>	<i>small</i>	<i>big</i>	9.7
<i>small</i>	<i>big</i>	<i>small</i>	10.8
<i>small</i>	<i>big</i>	<i>big</i>	5.2
<i>big</i>	<i>small</i>	<i>small</i>	28.3
<i>big</i>	<i>small</i>	<i>big</i>	18.8
<i>big</i>	<i>big</i>	<i>small</i>	20.3
<i>big</i>	<i>big</i>	<i>big</i>	12.5

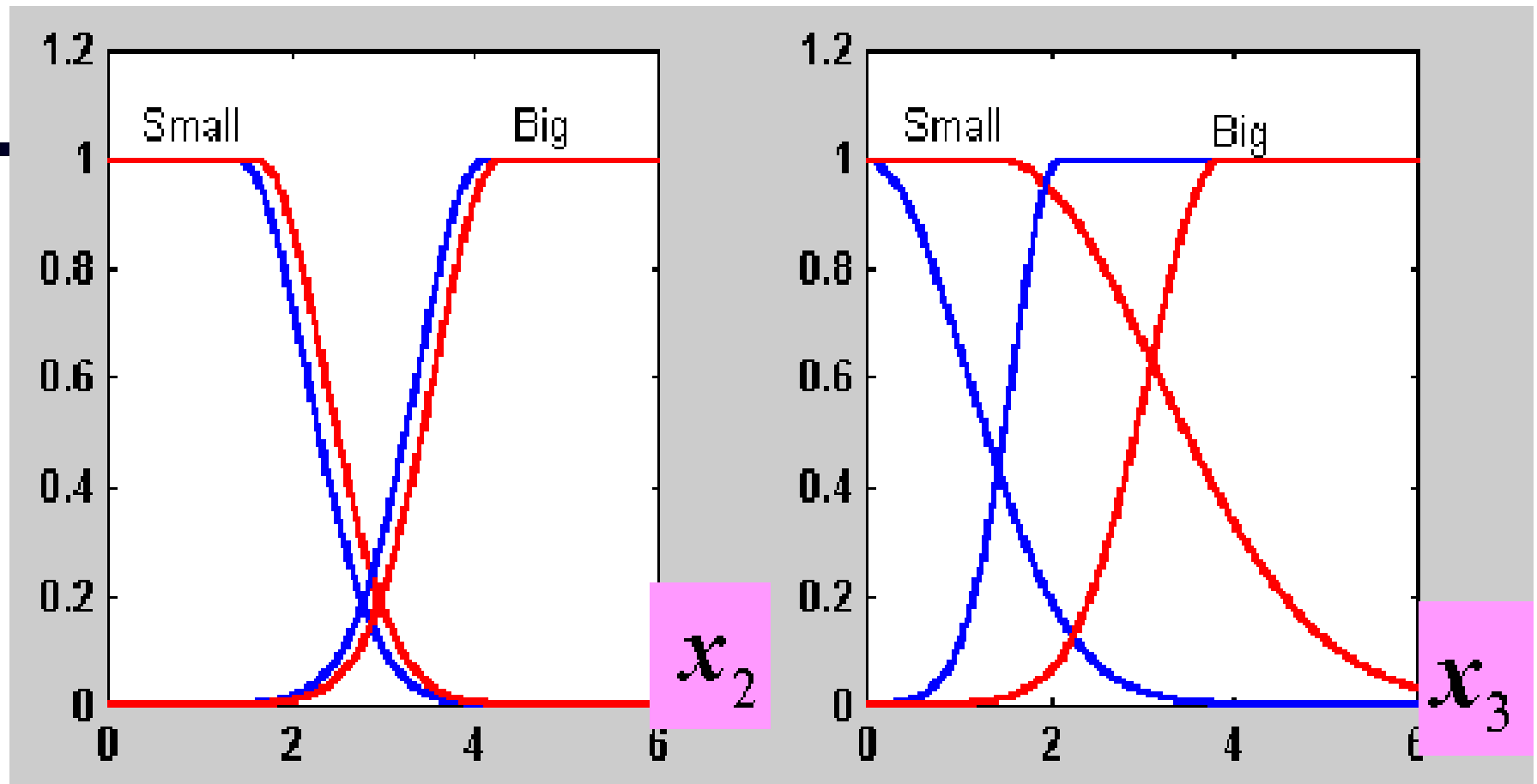
常 数 模 型



常 数 模 型 隶 属 函 数 的 变 化

前 件		后 件
x_2	x_3	$f_i(x_1, x_2, x_3)$
<i>small</i>	<i>small</i>	$15.6 + 2.59x_1 + 0.34x_2 - 0.15x_3$
<i>small</i>	<i>big</i>	$2.80 + 2.30x_1 + 0.01x_2 + 0.88x_3$
<i>big</i>	<i>small</i>	$6.30 + 2.31x_1 + 0.64x_2 - 0.37x_3$
<i>big</i>	<i>big</i>	$0.30 + 1.76x_1 + 0.14x_2 + 0.48x_3$

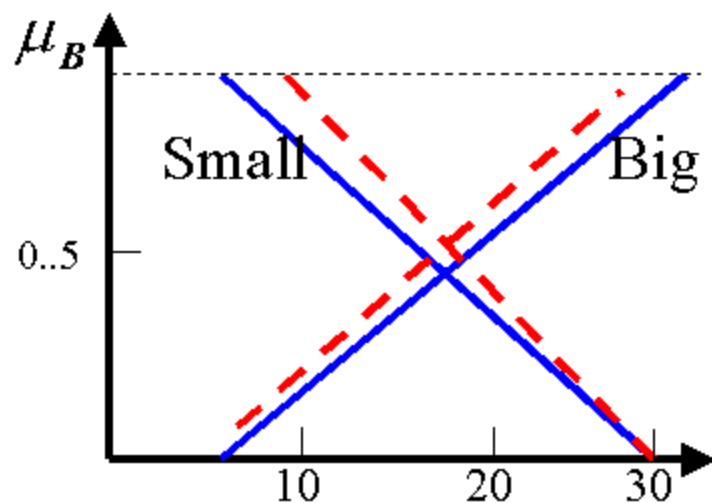
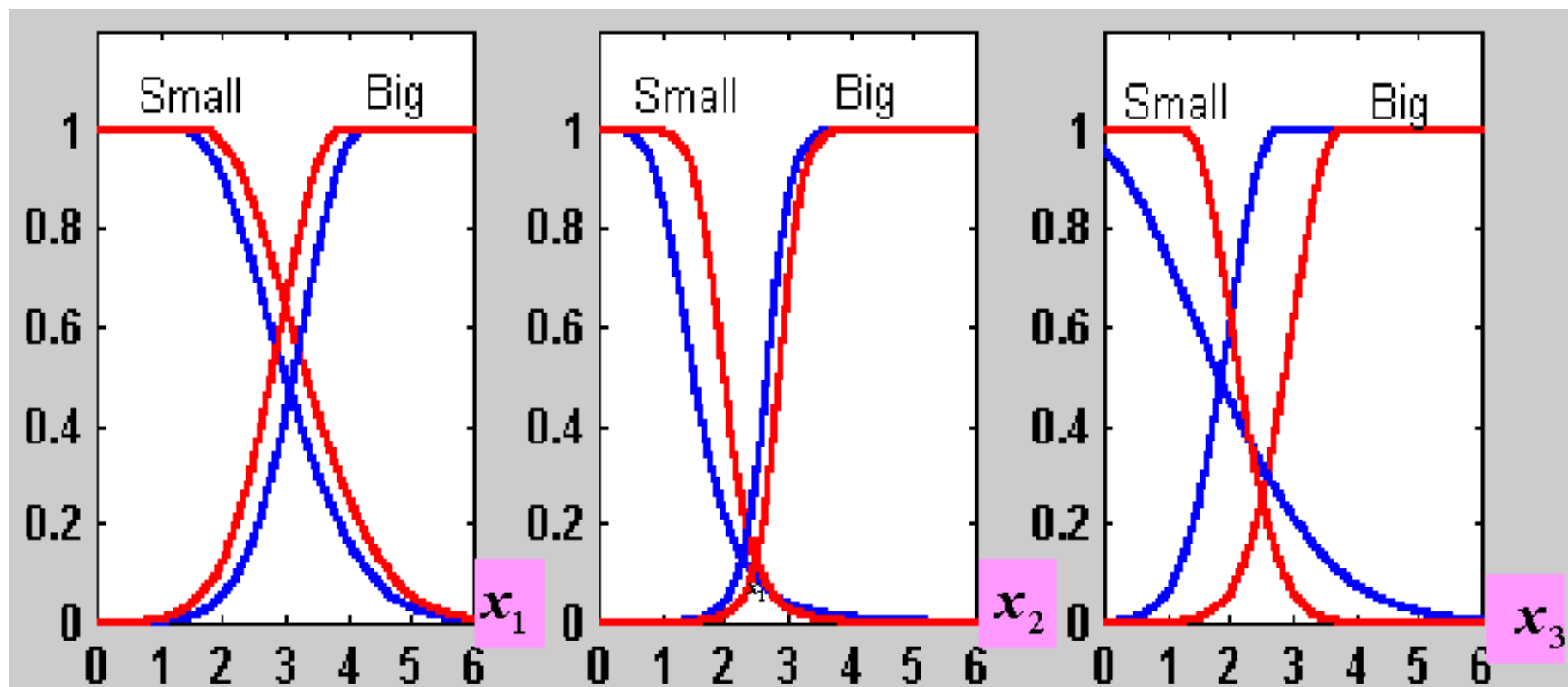
非 线 性 模 型



非线性模型隶属函数的变化

前 件			后 件	
x_1	x_2	x_3	$\tau_{R_S^i}$	$\tau_{R_B^i}$
<i>small</i>	<i>small</i>	<i>small</i>	0.53	0.47
<i>small</i>	<i>small</i>	<i>big</i>	0.76	0.28
<i>small</i>	<i>big</i>	<i>small</i>	0.73	0.31
<i>small</i>	<i>big</i>	<i>big</i>	0.94	0.15
<i>big</i>	<i>small</i>	<i>small</i>	0.09	1.00
<i>big</i>	<i>small</i>	<i>big</i>	0.38	0.63
<i>big</i>	<i>big</i>	<i>small</i>	0.36	0.68
<i>big</i>	<i>big</i>	<i>big</i>	0.64	0.38

语 言 输 出 模 型

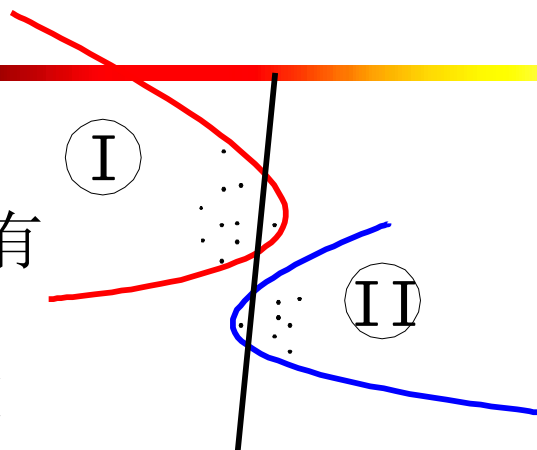


语言输出模型
隶属函数的变化

● 神经网络模糊化

① 模糊感知器

精确划分的问题：每个分量都有同样的“重要性”，用在分类时，当分类有重叠时（如图），得不到很好的结果。



模糊感知器的基本思想：给隶属函数以一定的修正量，对隶属度接近0.5的样本,在确定权值向量时,给予较小的影响：

$$W_k^{j+1} = W_k^j + \eta |\mu_1(x_k) - \mu_2(x_k)|^m [y_k^d - y_k] x_k^j, \\ 1 \leq j \leq n+1 (\text{迭代次数})$$

$$\sum_{i=1}^2 \mu_i(x_k) = 1, \quad i = 1, 2. \quad 0 < \sum_{k=1}^p \mu_i(x_k), \quad \mu_i(x_k) \in [0, 1]$$

p 是向量 x 的维数。 i 是分类的数目。

当 $\mu_1(x_k), \mu_2(x_k) \in \{0,1\}$ 时, $|\mu_1(x_k) - \mu_2(x_k)| = 1$,
模糊感知器变成精确感知器.

模糊感知器算法的问题:

- 1) 如何选择 m ?
- 2) 如何给向量赋与模糊隶属函数?
- 3) 算法的终止判据.

回答: 1) $m > 1$; 如隶属函数接近0.5.

$m \gg 1$; 如隶属函数大于0.5.

2) 给向量赋与模糊隶属函数的规则:

对类别I:
$$\mu_1(x_k) = 0.5 + \frac{e^{f(d_2-d_1)/d} - e^{-f}}{2(e^f - e^{-f})}$$

$$\mu_2(x_k) = 1 - \mu_1(x_k)$$

对类别II:
$$\mu_2(x_k) = 0.5 + \frac{e^{f(d_1-d_2)/d} - e^{-f}}{2(e^f - e^{-f})}$$

$$\mu_1(x_k) = 1 - \mu_2(x_k)$$

其中:

- d_1 是属于向量类别I均值(相当于离聚类中心)的距离;
- d_2 是属于向量类别II均值(相当于离聚类中心)的距离;
- d 是类别I和II均值之间的距离;
- f 是正常数,控制隶属函数下降到0.5的速率.

- 1) 输入向量等于大于所属类的平均值(d_1 或 d_2)隶属函数取1;
- 2) 输入向量等于其它类的平均值(d_1 或 d_2)隶属函数取0.5;
- 3) 输入向量等于二类的平均值 $[(d_1 + d_2)/2]$ 隶属函数取0.5;
- 4) 隶属函数不应小于0.5
- 5) 输入向量大于大于 $d_1(d_2)$ 和小于小于 $d_2(d_1)$ 隶属函数取1.0

3) 算法停止的判据: (当分类错误, 不确定向量不再产生另一迭代)

$$\mu_1(x_k) > 0.5 + Beta \text{ 或 } \mu_2(x_k) < 0.5 - Beta$$

$$Beta = \frac{1 - e^{-f}}{2(e^f - e^{-f})} + \varepsilon \quad \varepsilon \geq 0$$

$\varepsilon = 0.02$, $Beta$ 在 0.5 附近.

产生良好的结果.

② 模糊联想存储器 (FAM)

双向联想存储器的模糊化。把双向联想存储器的权矩阵变换成模糊集合的关系（关联）矩阵。

模糊关联矩阵M确定有二种方法：1) 相关最小编码

$$M = A^T \circ B$$
$$m_{ij} = \text{Min}(a_i, b_j)$$

假定A= (0.2 0.5 0.9 1.0), B= (0.9 0.5 0.6),则:

$$M = A^T \circ B = \begin{pmatrix} 0.2 \\ 0.5 \\ 0.9 \\ 1.0 \end{pmatrix} \begin{pmatrix} 0.9 & 0.5 & 0.6 \end{pmatrix} = \begin{pmatrix} 0.2 & 0.2 & 0.2 \\ 0.5 & 0.5 & 0.5 \\ 0.9 & 0.5 & 0.6 \\ 0.9 & 0.5 & 0.6 \end{pmatrix}$$

现在看，如果有A能否“回忆”起

$$AM = B; \quad B \circ M^T = [0.2 \ 0.5 \ 0.9 \ 0.9] = A' \neq A.$$

2) 相关乘积编码

$$M = A^T \bullet B$$
$$m_{ij} = a_i b_j$$

$$M = A^T B = \begin{pmatrix} 0.18 & 0.1 & 0.12 \\ 0.45 & 0.25 & 0.3 \\ 0.81 & 0.45 & 0.45 \\ 0.9 & 0.5 & 0.6 \end{pmatrix}$$

现在看，如果有A能否“回忆”起B？

- 如果 $A^T = (0\ 0\ 0\ 1)$ ，则 $A^T \cdot M = B$ ；如果 $A^T = (1\ 0\ 0\ 0)$ 则

由m个FAM组成的FAM系统只回忆起B的20%。

把m个关联 (A_k, B_k) 分别存到存贮库中，把m个记忆向量叠加起来。即：

$$B_k = A \circ M_k = A \circ (A_k^T \circ B_k) \quad k = 1, 2, \dots, m.$$

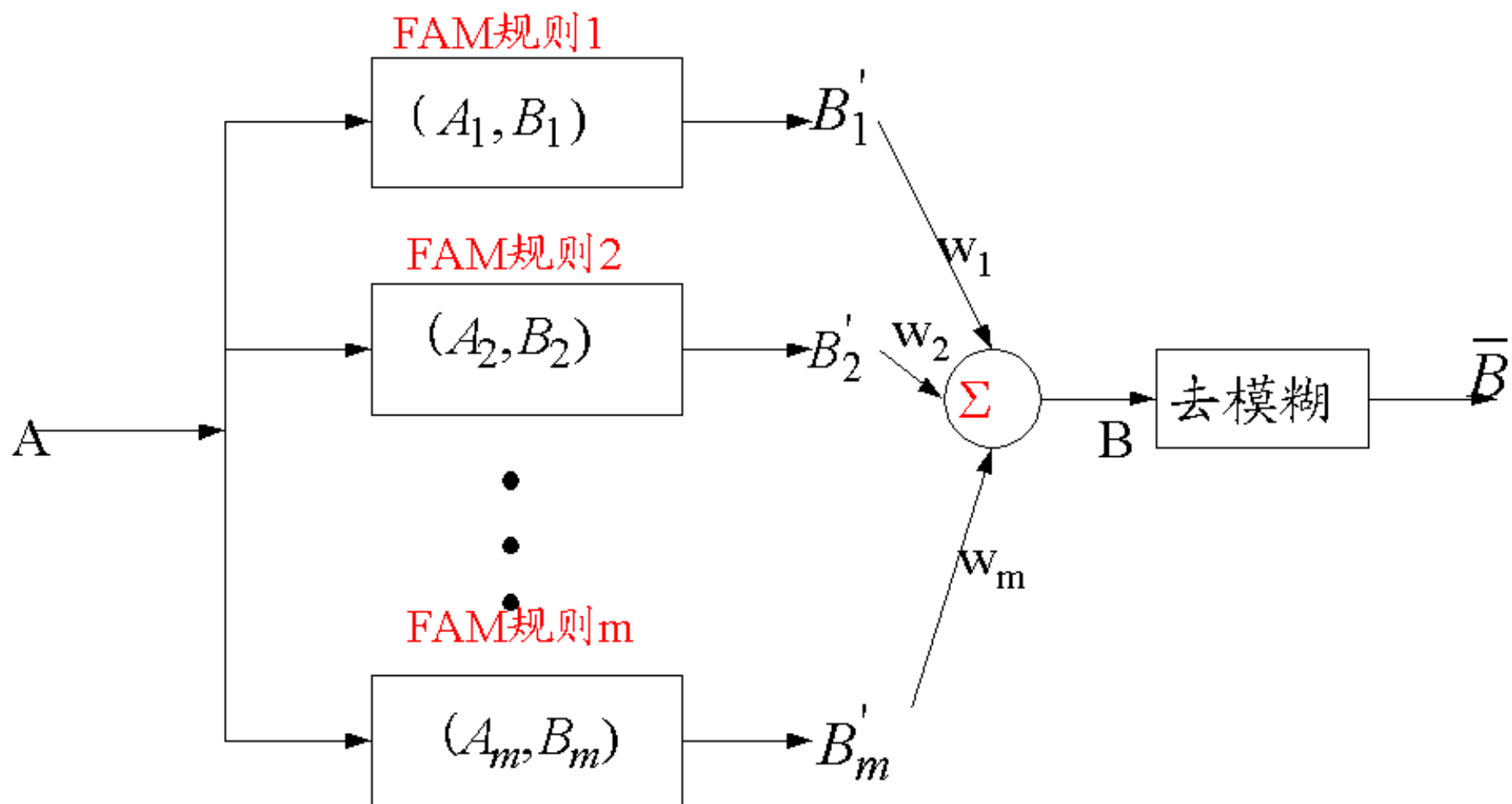
所记忆的隶属向量，等于各记忆向量的加权和：

$$B = \sum_{k=1}^m W_k B_k'$$

如在输出论域 $Y = (y_1, y_2, \dots, y_p)$ 需要一个单独的输出，则要去模糊：

$$\bar{B} = \sum_{i=1}^p y_i \mu_B(y_i) / \sum_{i=1}^p \mu_B(y_i)$$

A并行地加于各联想存贮器上。



FAM系统的结构

● 模糊自适应学习控制网络 (FALCON)

If x_1 是 A_1^1 和 x_2 是 A_2^1 x_n 是 A_n^1 then y 是 B_1

If x_1 是 A_1^2 和 x_2 是 A_2^2 x_n 是 A_n^2 then y 是 B_2

•

•

•

•

•

•

If x_1 是 A_1^n 和 x_2 是 A_2^n ... x_n 是 A_n^n then y 是 B_n

对任一节点*i* 输入与输出的关系:

输入: $net_i = f(u_1^k, u_2^k, \dots u_p^k, w_1^k, w_2^k \dots w_p^k)$

k 代表层次, f 代表组合

输出 $Output = o_i^k = a(net_i) = a(f)$

:

第1层: 输入 $w_i = 1$ ($u_i^1 = x_i$)
输出 = 输入, $f = u_i^{(1)}(x_i)$

第2层: 术语用钟型隶属函数

$$\text{输入: } f = M_{x_i}^j(m_{ij}, \sigma_{ij}) = -\frac{(\mu_i^{(2)} - m_{ij})^2}{\sigma_{ij}^2}, \text{ 输出 } a = e^f$$

第3层:

与运算, 取极小。

$$\text{输入: } f = \min(u_1^{(3)}, u_3^{(3)}, \dots, u_p^{(3)})$$

$$\text{输出: } a = f, \text{ 权值: } w_i^{(3)} = 1$$

第4层:

二种运算 : 1) 自下至上

$$\text{输入: } f = \sum_i u_i^{(4)}$$

$$\text{输出: } a = \min(1, f), w_i^{(4)} = 1$$

2) 自上至下与第2层一样, 产生隶属函数

第5层:

1) 由下至上: 决策、去模糊

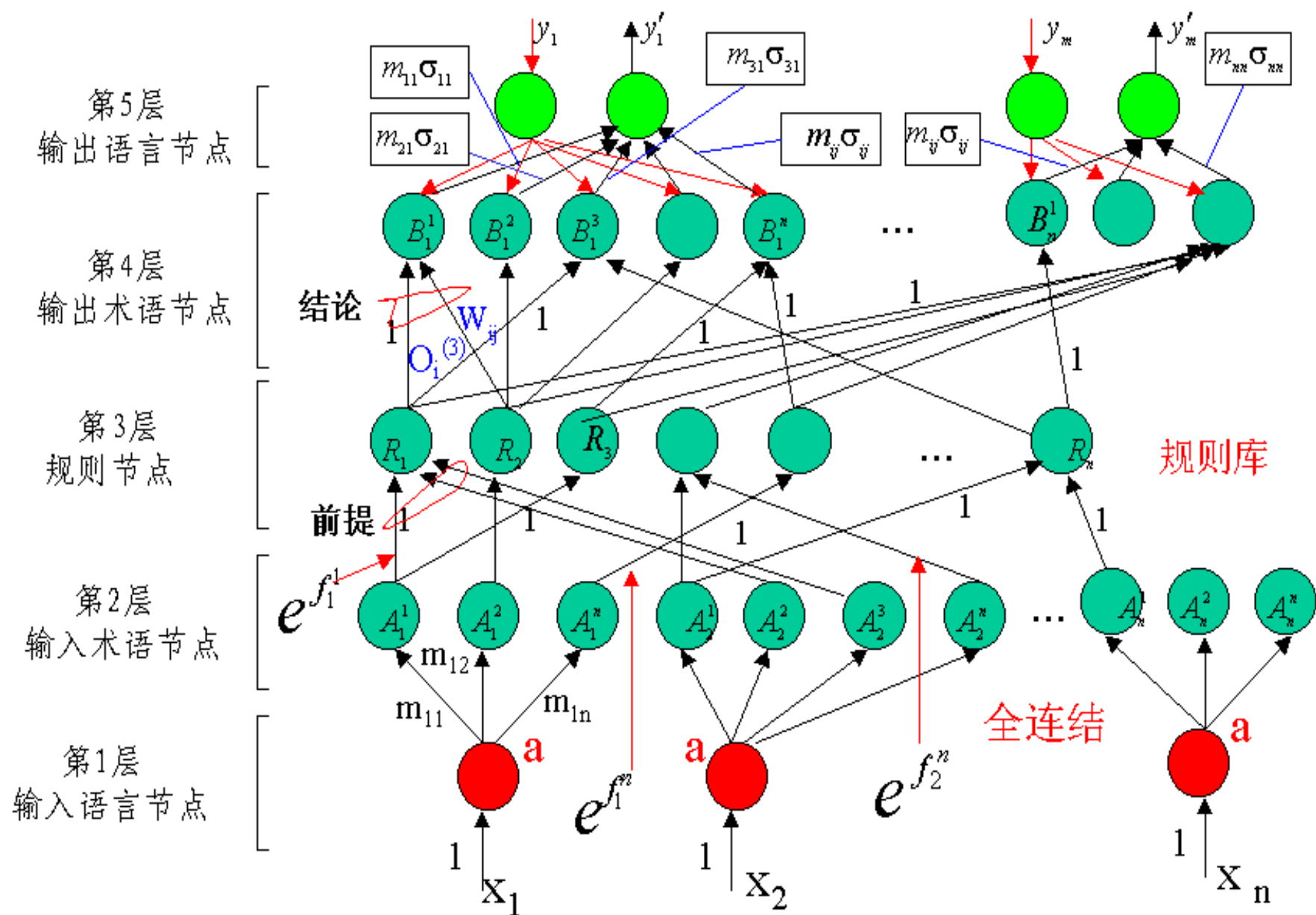
输入:
$$f = \sum_j w_{ij}^{(5)} u_{ij} = \sum_j (m_{ij} \sigma_{ij}) u_{ij}^{(5)}$$

隶属函数为钟形

输出:
$$a = f / \sum_j \sigma_{ij} u_{ij}^{(5)} = y_i$$

权值:
$$w_{ij}^{(5)} = m_{ij} \sigma_{ij}$$

2) 由上而下 与第1层一样.



学习（训练）

目的：1）决定第2层和第4层中的隶属函数中心 m_{ij} 和宽度 σ_{ij}

2）决定第3层和第4层中的规则

学习分二阶段

1）自组织学习。设置初始隶属函数，通过无监督学习检测规则的存在性。

2）有监督学习。最佳地调节隶属函数的参数，得到期望的输出

自组织学习

a) 输入变量 x_1 空间的划分

$$\begin{array}{c} |T(x_1)| \times |T(x_2)| \times \dots \times |T(x_n)| = |T(x)| \\ |T(y_i)| \end{array}$$

b) 第4层处在自上至下的模式

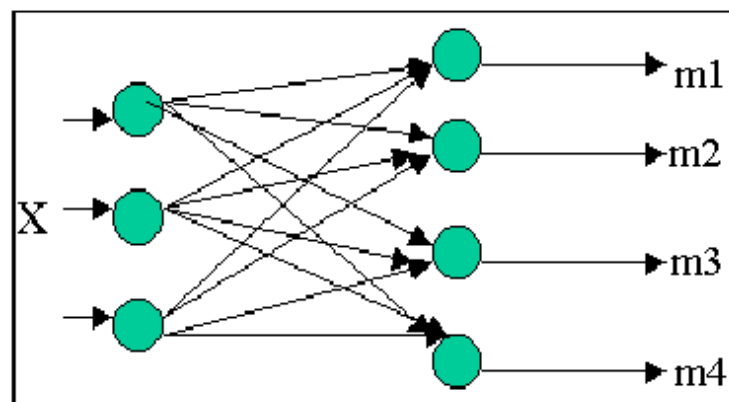
c) 先初步计算隶属函数的中心 m_i

随机选择 $x(t)$ 中第 i 个隶属函数的中心 m_i 利用 Kohonen 竞争学习规则, 确定最接近的一个:

$$\|x(t) - m_{closest}(t)\| = \min_{1 \leq i \leq k} \{ \|x(t) - m_i(t)\| \}$$
$$m_{closest}(t+1) = m_{closest}(t) + \alpha(t)[x(t) - m_{closest}(t)]$$
$$m_i(t+1) = m_i(t), \quad \text{对 } m_i \neq m_{closest}$$

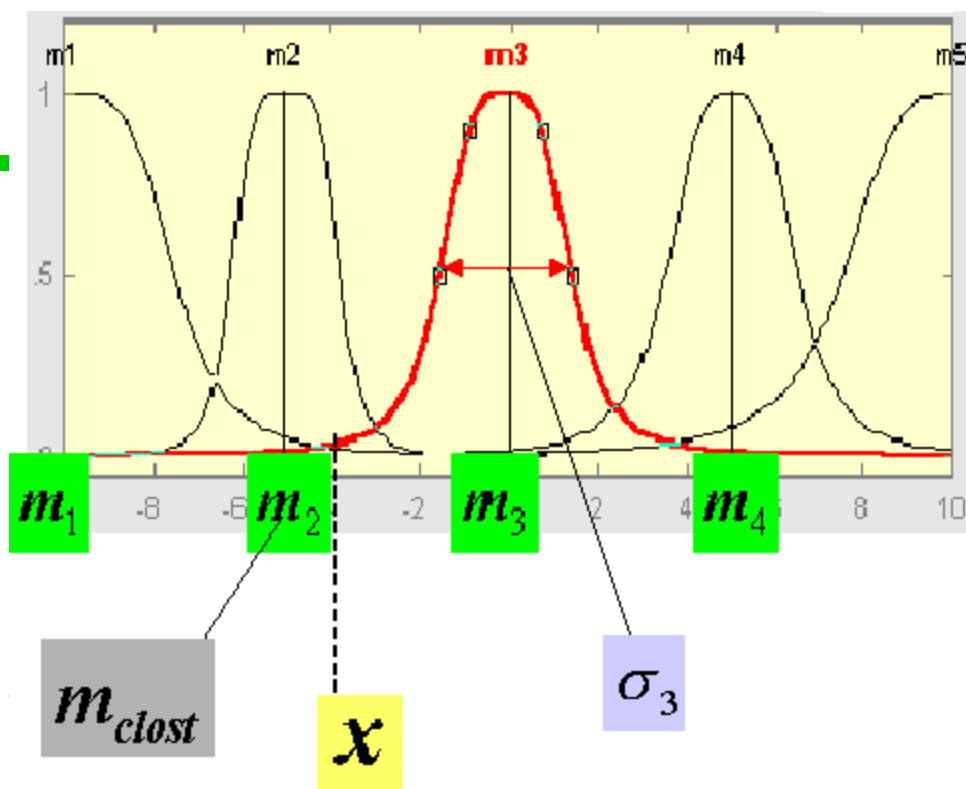
$x(t)$ 代表输入、输出语言变量

$x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m$ 的任意一个



这里: $k = |T(x)|$

m_i 确定后, 按下式计算 σ



$$E = \frac{1}{2} \sum_{i=1}^N \left[\sum_{j \in N_n} \left(\frac{m_i - m_j}{\sigma_i} \right)^2 - r \right]^2$$

$$\xi = \mathcal{J}$$

这里 N 是邻点的个数, N_n 是最远的邻点个数, r 是一个重叠系数

d) 确定连接和模糊规则

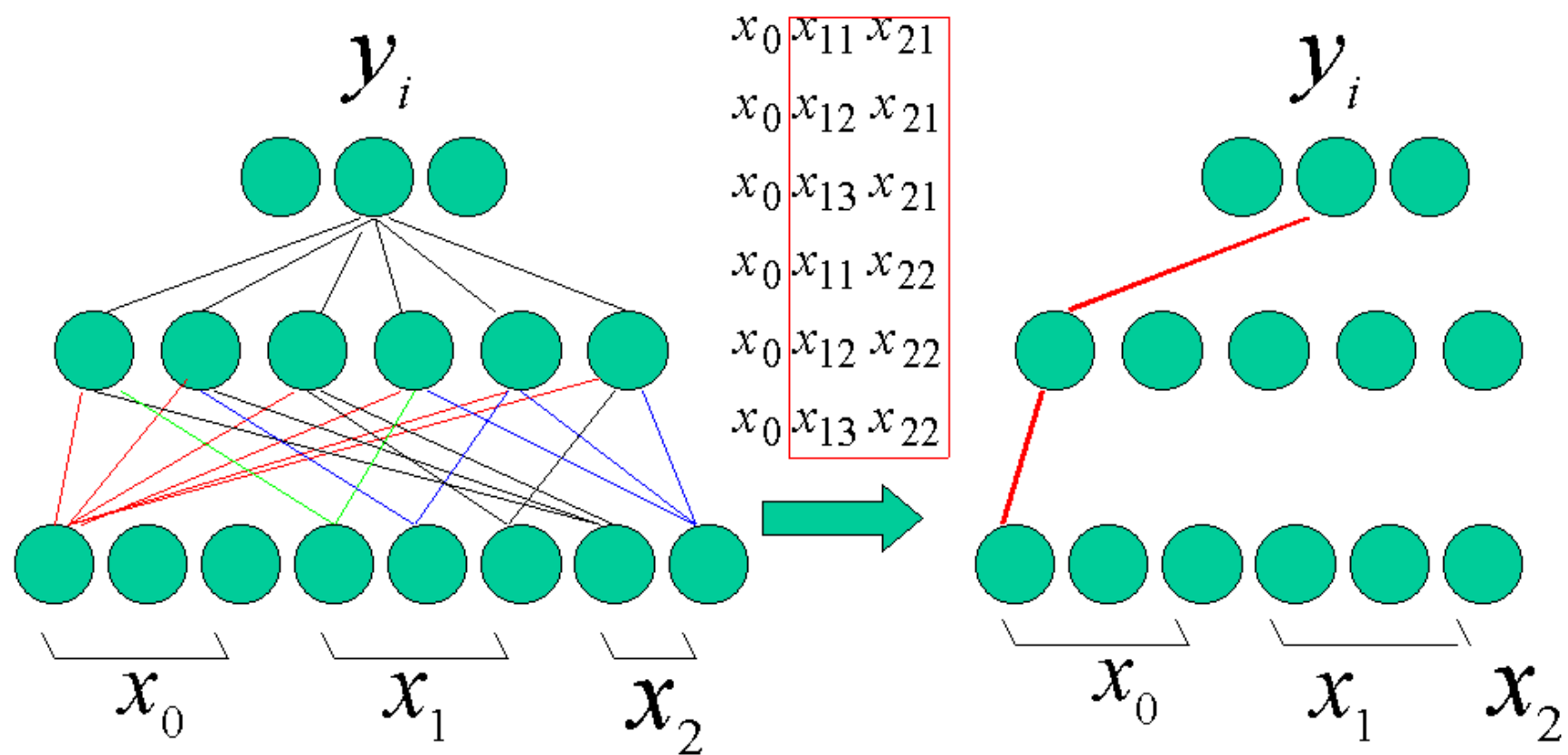
- 规则节点(第3层)产生激励强度 $O_i^{(3)}$, 输出术语节点(自上而下)产生结论 $O_i^{(4)}$, 二者的连接权值 W_{ij}

$$\dot{W}_{ji} = O_j^{(4)}(-W_{ji} + O_i^{(3)})$$

输赢指数

e) 规则合并, 减少规则

- 1) 有完全相同的结果
- 2) 前提一样的规则
- 3) 其它前提的并, 组成了某些输入变量的整个术语的集合



监督学习阶段

$$E = \frac{1}{2} (y^d(t) - y(t))^2$$

$$\frac{\partial E}{\partial m_i} = \frac{\partial E}{\partial a^{(5)}} \frac{\partial a^{(5)}}{\partial m_i} = -[y^d(t) - y(t)] \frac{\sigma u_i^{(5)}}{\sum_i \sigma_i u_i^{(5)}}$$

$$m_i(t+1) = m_i(t) + [y^d(t) - y(t)] \frac{\sigma_i u_i^{(5)}}{\sum_i \sigma_i u_i^{(5)}}$$

$$\frac{\partial E}{\partial \sigma_i} = \frac{\partial E}{\partial a^{(5)}} \frac{\partial a^{(5)}}{\partial \sigma_i} = -[y^d(t) - y(t)] \frac{m_i u_i^{(5)} (\sum_i \sigma_i u_i^{(5)}) - (\sum_i m_i \sigma_i u_i^{(5)}) u_i^{(5)}}{(\sum_i \sigma_i u_i^{(5)})^2}$$

$$\sigma_i(t+1) = \sigma_i(t) + [y^d(t) - y(t)] \frac{m_i u_i^{(5)} (\sum_i \sigma_i u_i^{(5)}) - (\sum_i m_i \sigma_i u_i^{(5)}) u_i^{(5)}}{(\sum_i \sigma_i u_i^{(5)})^2}$$

$$\delta^{(5)} = -\partial E / \partial a^{(5)} = \partial E / \partial y = y^d - y(t)$$

●神经-模糊网络（控制器）的参数学习（ANFIS）

- R^1 : If x_1 是 A_1^1 和 x_2 是 A_2^1 , Then $y = f_1 = a_0^1 + a_1^1 x_1 + a_2^1 x_2$
 R^2 : If x_1 是 A_1^2 和 x_2 是 A_2^2 , Then $y = f_2 = a_0^2 + a_1^2 x_1 + a_2^2 x_2$

$$y^* = \frac{\mu_1 f_1 + \mu_2 f_2}{\mu_1 + \mu_2} \quad \mu_j = \mu_{A_1^j}(x_1) \bullet \mu_{A_2^j}(x_2), j=1,2$$
$$= \bar{\mu}_1 f_1 + \bar{\mu}_2 f_2 = \bar{\mu}_1(a_0^1 + a_1^1 x_1 + a_2^1 x_2) + \bar{\mu}_2(a_0^2 + a_1^2 x_1 + a_2^2 x_2)$$

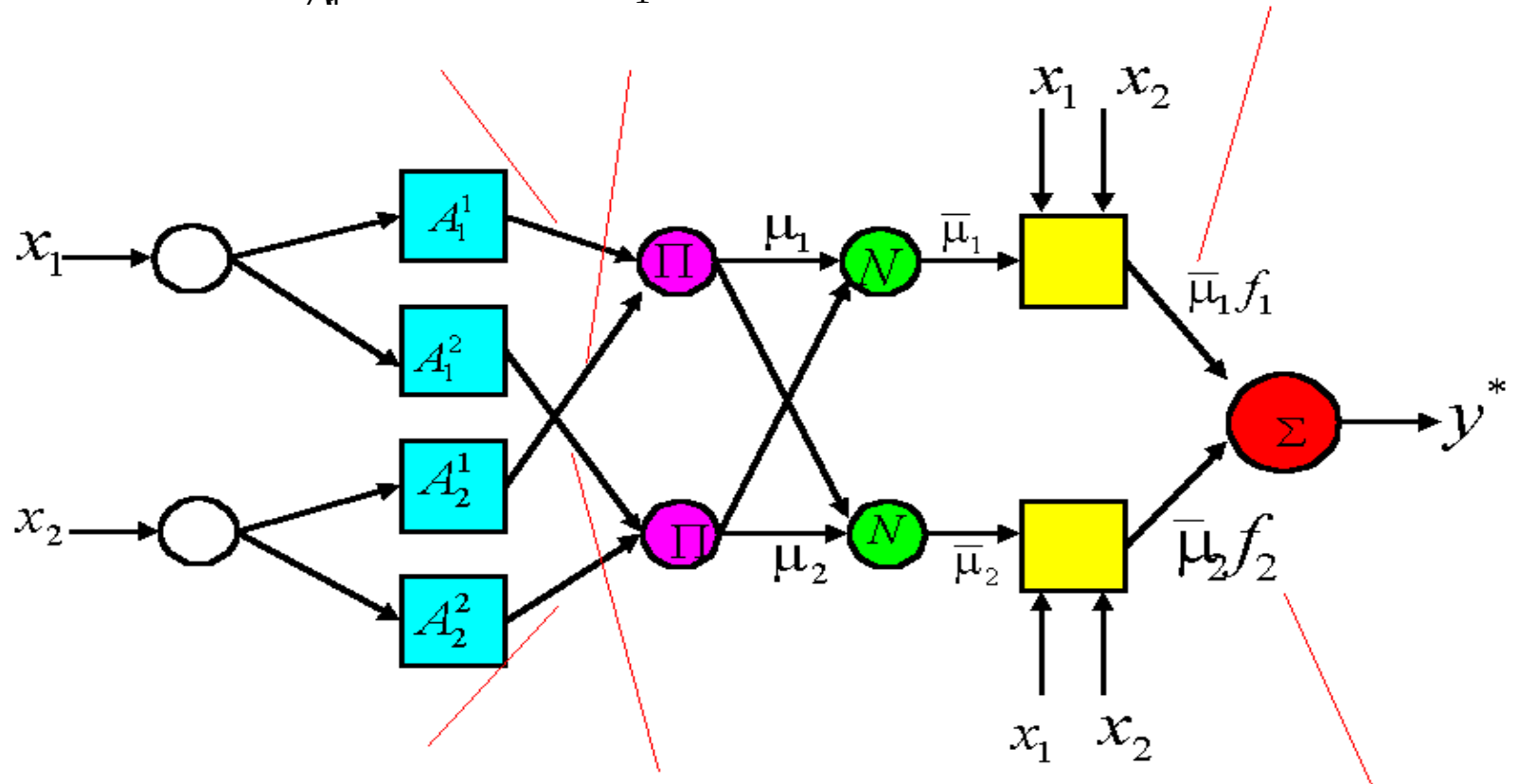
相应的ANFIS网络如图示。隶属函数为钟形：

$$\mu_{A_i^j}(x_i) = \frac{1}{(1 + \{[(x_i - m_i^j) / \sigma_i^j]^2\}^{b_i^j})}$$

或

$$\mu_{A_i^j}(x_i) = \exp \left\{ - \{[(x_i - m_i^j) / \sigma_i^j]^2\}^{b_i^j} \right\}$$

$$\bar{\mu}_j = \mu_j / \mu_{A_1^j}(x_1) + \mu_{A_2^j}(x_2)$$



$$\mu_{A_2^1}(x_2) \quad \mu_{A_2^2}(x_2)$$

$$f_j = (a_0^j + a_1^j x_1 + a_2^j x_2)$$

$$\mu_{A_i^j}(x_i) = \frac{1}{(1 + \{[(x_i - m_i^j) / \sigma_i^j]^2\}^{b_i^j})}$$

要调节的参数:

$$\{m_i^j, \sigma_i^j, b_i^j\} \text{ 和 } \{a_0^j, a_1^j, a_2^j\}$$

对后件参数, 可以用Kalman滤波方法进行计算, 此时, 把后件参数排列成向量:

$$(a_0^1, a_1^1, a_2^1, a_0^2, a_1^2, a_2^2)$$

一组线性方程求解

$$\begin{aligned} \bar{\mu}_1(a_0^1 + a_1^1 x_1^{(1)} + a_2^1 x_2^{(1)}) + \bar{\mu}_2(a_0^2 + a_1^2 x_1^{(1)} + a_2^2 x_2^{(1)}) &= d^{(1)} \\ \bar{\mu}_1(a_0^1 + a_1^1 x_1^{(2)} + a_2^1 x_2^{(2)}) + \bar{\mu}_2(a_0^2 + a_1^2 x_1^{(2)} + a_2^2 x_2^{(2)}) &= d^{(2)} \\ \bullet & \bullet \bullet \bullet \\ \bullet & \bullet \bullet \bullet \\ \bullet & \bullet \bullet \bullet \\ \bar{\mu}_1(a_0^1 + a_1^1 x_1^{(p)} + a_2^1 x_2^{(p)}) + \bar{\mu}_2(a_0^2 + a_1^2 x_1^{(p)} + a_2^2 x_2^{(p)}) &= d^{(p)} \end{aligned}$$

$$\begin{vmatrix}
 \bar{\mu}_1^{(1)} & \bar{\mu}_1^{(1)} x_1^{(1)} & \bar{\mu}_1^{(1)} x_2^{(1)} & \bar{\mu}_2^{(1)} & \bar{\mu}_2^{(1)} x_1^{(1)} & \bar{\mu}_2^{(1)} x_2^{(1)} \\
 \bar{\mu}_1^{(2)} & \bar{\mu}_1^{(2)} x_1^{(1)} & \bar{\mu}_1^{(2)} x_2^{(2)} & \bar{\mu}_2^{(2)} & \bar{\mu}_2^{(2)} x_1^{(1)} & \bar{\mu}_2^{(2)} x_2^{(2)} \\
 \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
 \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
 \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
 \bar{\mu}_1^{(p)} & \bar{\mu}_1^{(p)} x_1^{(p)} & \bar{\mu}_1^{(p)} x_2^{(p)} & \bar{\mu}_2^{(p)} & \bar{\mu}_2^{(p)} x_1^{(p)} & \bar{\mu}_2^{(p)} x_2^{(p)}
 \end{vmatrix}
 \begin{vmatrix}
 a_0^1 \\
 a_1^1 \\
 a_2^1 \\
 a_0^2 \\
 a_1^2 \\
 a_2^2
 \end{vmatrix}
 =
 \begin{vmatrix}
 d^{(1)} \\
 d^{(2)} \\
 \bullet \\
 \bullet \\
 \bullet \\
 d^{(p)}
 \end{vmatrix}$$

$(x_1^{(k)}, x_2^{(k)}, d^{(k)})$ 是样本, $k = 1, 2, \dots, p$ 是样板数目.

$$A = (X^T X)^{-1} X^T D$$

Thank You

Question!

Intelligence Science

<http://www.intsci.ac.cn/>

