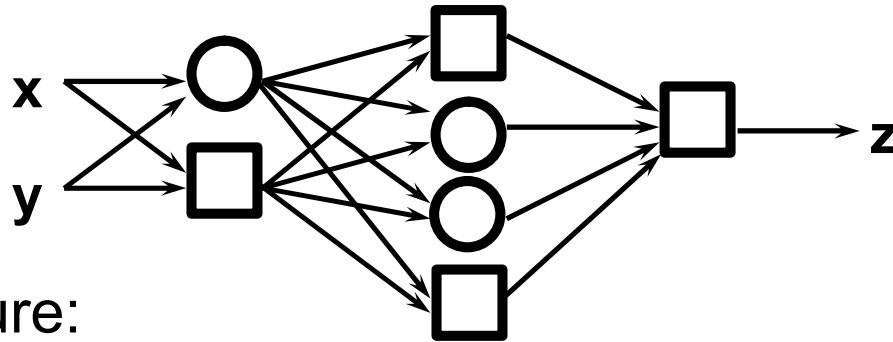


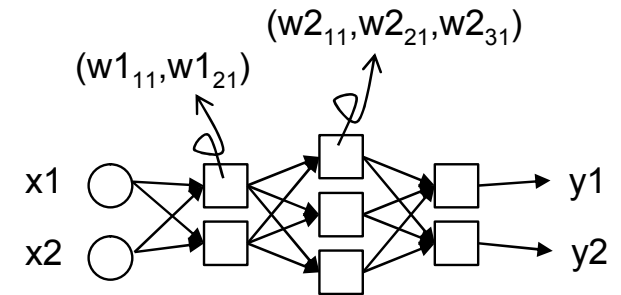
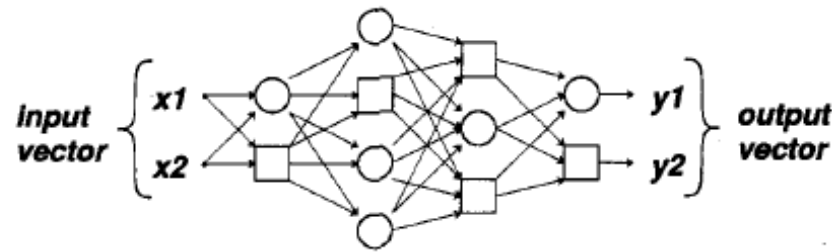
*ANFIS:
Adaptive Network-Based
Fuzzy Inference System*

Adaptive Networks



- Architecture:
 - Feedforward networks with differentiable node functions
 - No weights associated with the links
 - **Squares**: nodes with parameters
 - **Circles**: nodes without parameters
- Goal:
 - To achieve an I/O mapping specified by training data
 - By backpropagation and gradient descent

Training Adaptive Networks



$$O_i^k = O_i^k(O_1^{k-1}, \dots, O_{\#(k-1)}^{k-1}, a, b, c, \dots) \quad (1)$$

$$E_p = \sum_{m=1}^{\#(L)} (T_{m,p} - O_{m,p}^L)^2 \quad (2)$$

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L). \quad (3)$$

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \quad (4)$$

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha}, \quad (5)$$

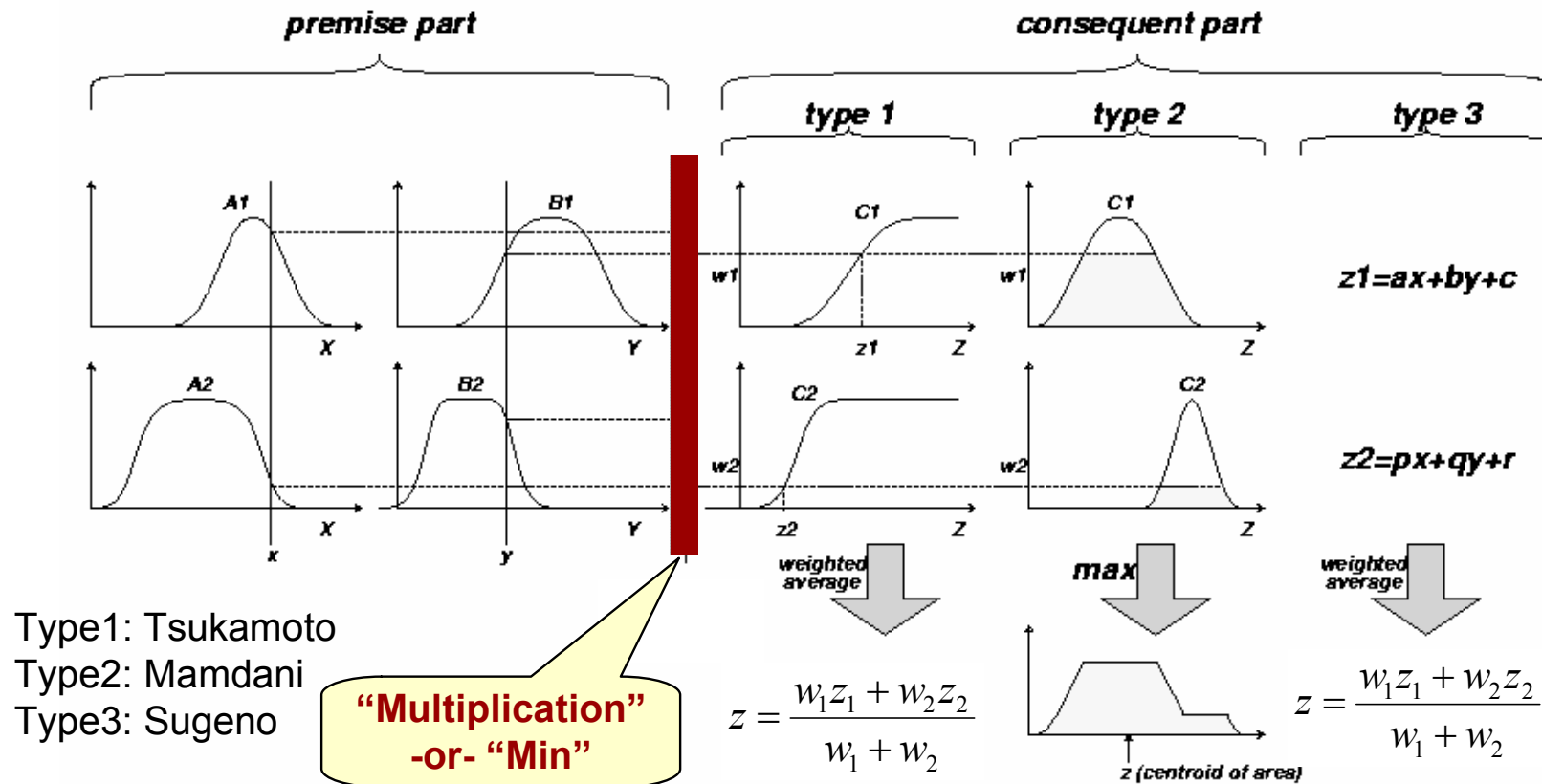
$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha}. \quad (6)$$

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \quad (7)$$

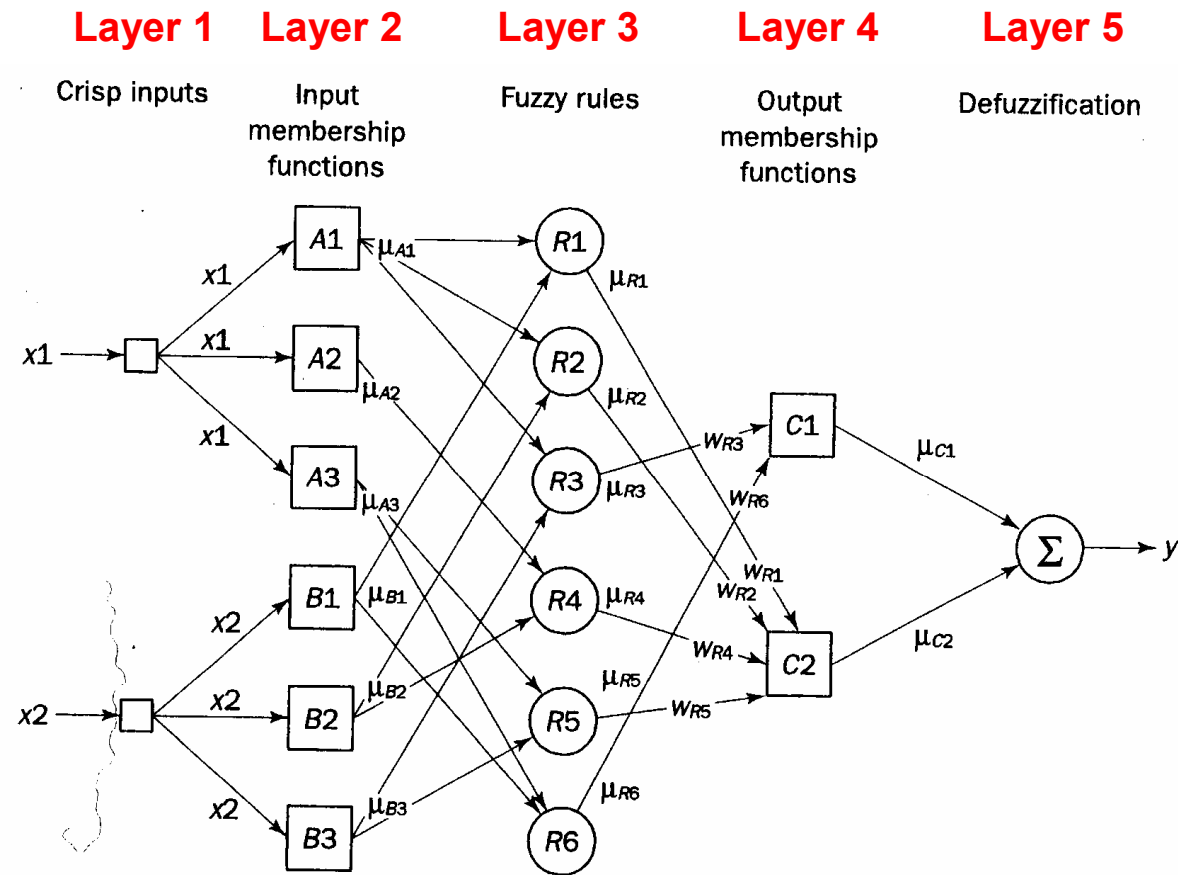
Output layer

Inner layer

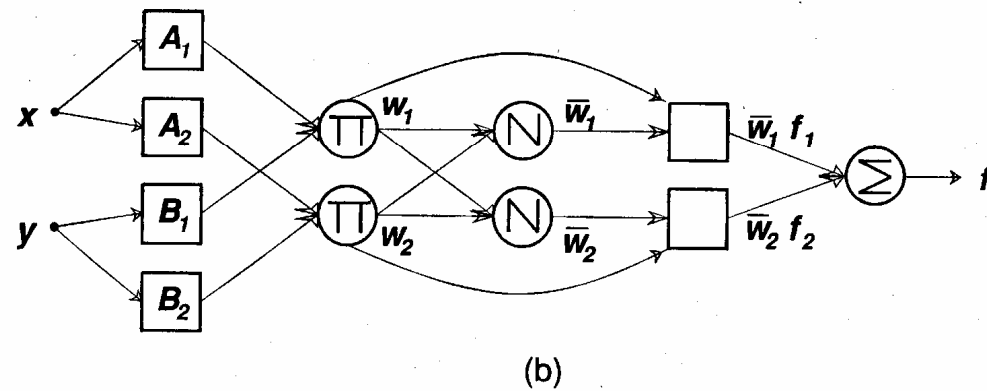
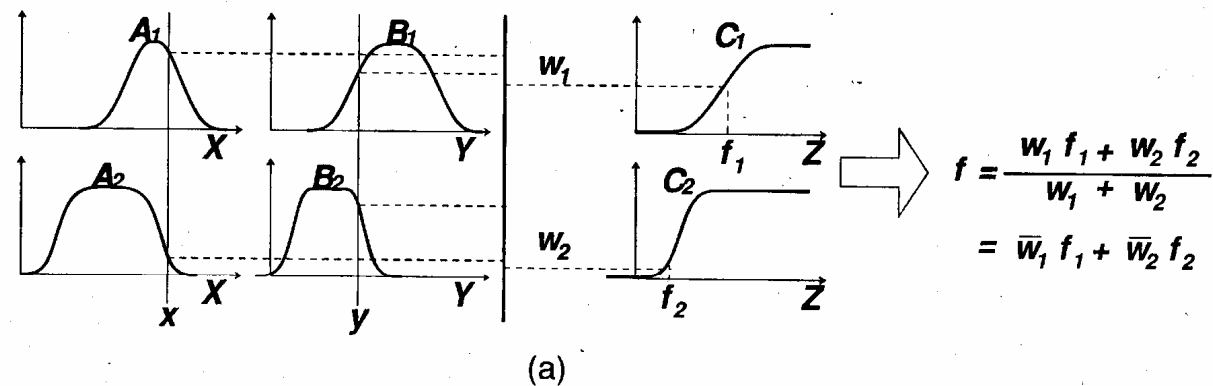
Commonly Used Fuzzy IF-THEN Rules and Fuzzy Reasoning Mechanisms



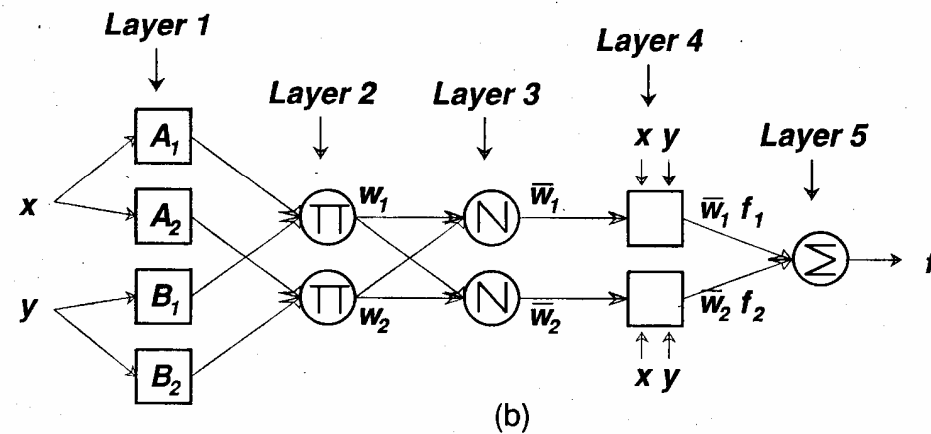
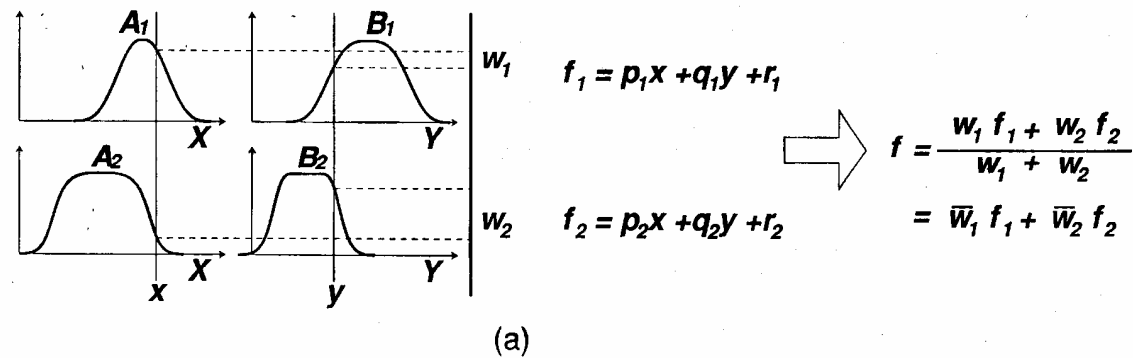
Mamdani's ANFIS (Type2)



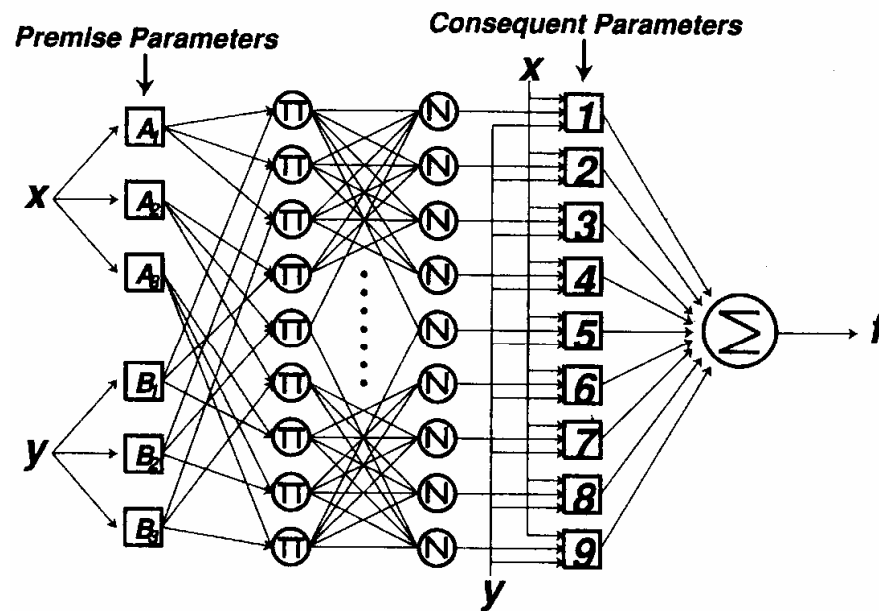
Tsukamoto's ANFIS (Type 1)



Sugeno's ANFIS (Type 3)

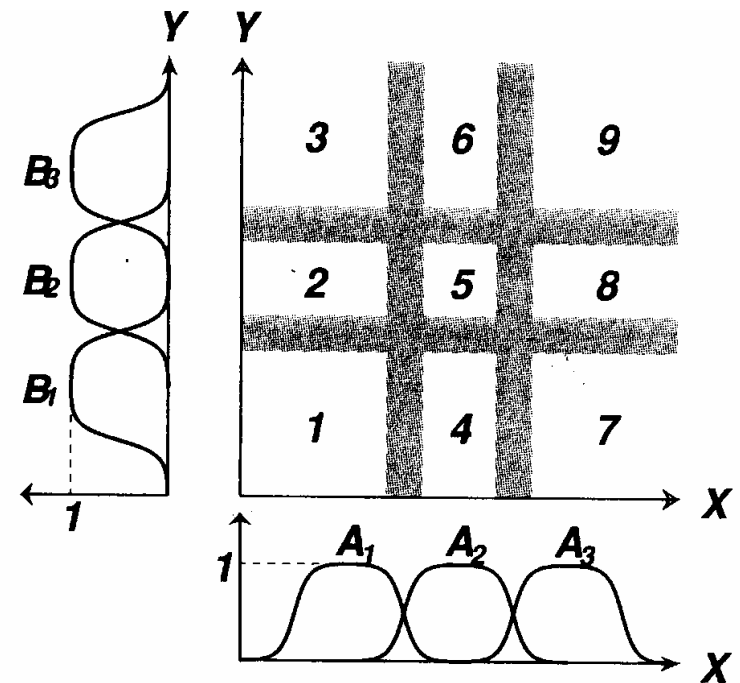


Example (Sugeno's ANFIS)



(a)

$$\mu_{A_i} = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}}$$



(b)

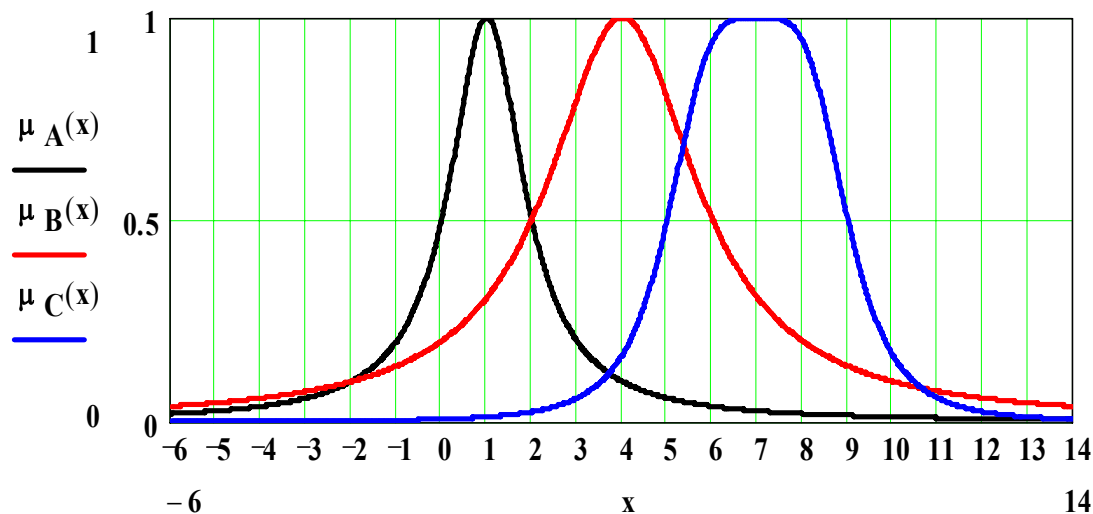
Example (Adaptive Formula)

$$\mu_{A1} = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

$$\Delta a = \alpha (y_d - y) f_i \bar{\mu}_i (1 - \bar{\mu}_i) \times \frac{1}{\mu_{A1}} \times \frac{\partial \mu_{A1}}{\partial a},$$

$$\frac{\partial \mu_{A1}}{\partial a} = - \frac{1}{\left[1 + \left(\frac{x1-a}{c} \right)^{2b} \right]^2} \times \frac{1}{c^{2b}} \times 2b \times (x1-a)^{2b-1} \times (-1)$$

$$= \mu_{A1}^2 \times \frac{2b}{c} \times \left(\frac{x1-a}{c} \right)^{2b-1}$$



$$\mu_A(x) := \frac{1}{1 + \left(\left| \frac{x-1}{1} \right| \right)^2}$$

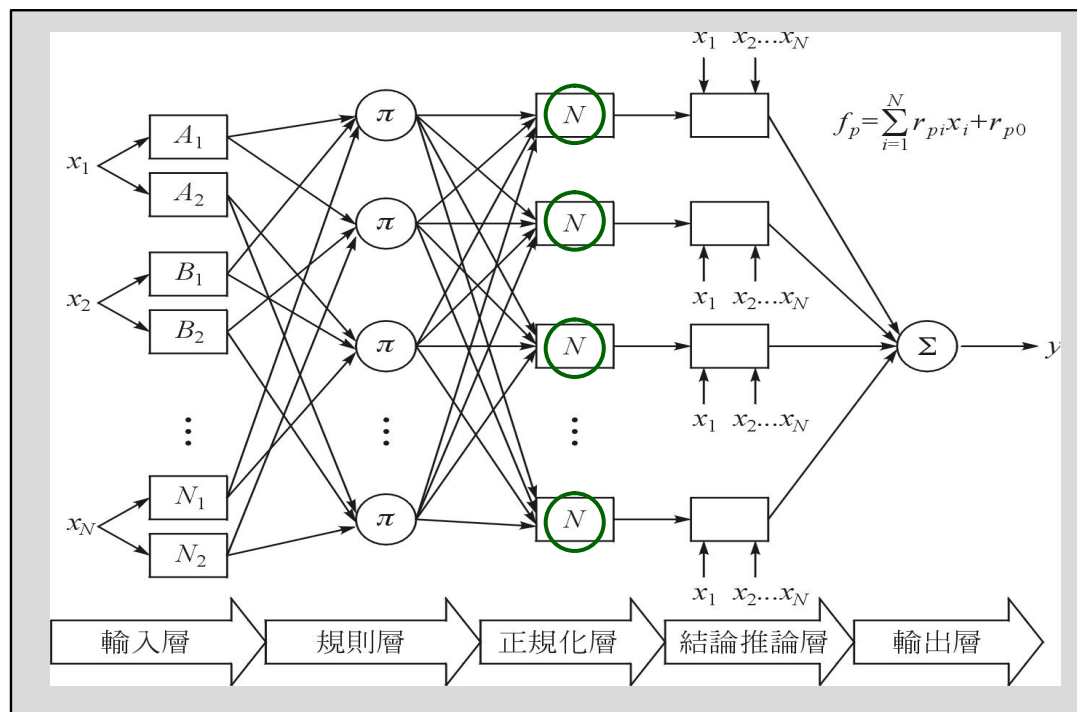
$$\mu_B(x) := \frac{1}{1 + \left(\left| \frac{x-4}{2} \right| \right)^2}$$

$$\mu_C(x) := \frac{1}{1 + \left(\left| \frac{x-7}{2} \right| \right)^4}$$



■ break

以下投影片 by 台大生工系 水資源資訊系統實驗室

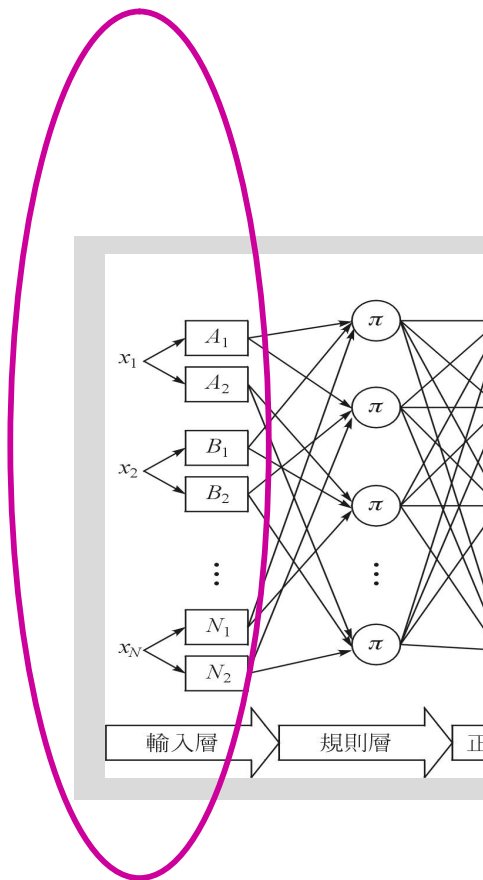


Example:
Sugeno's ANFIS
訂正 Layer 3
應是 circle layer

ANFIS 架構圖， N 個維度的輸入變數，1 個維度輸出變數

第一層：輸入層

將輸入變數映射到模糊集合中



$$O_{1,ji} = \mu_j(x_i) = \frac{1}{1 + \left| \frac{x_i - c_{ji}}{a_{ji}} \right|^{2b_{ji}}}$$

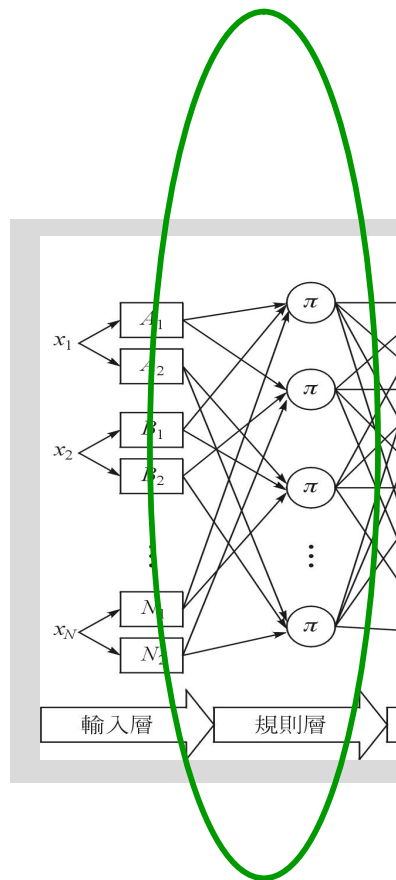
for $i = 1, 2, \dots, N; j = 1, 2, \dots, M_i$

式中 $\mu_j(x_i)$ 為第 i 個輸入變數的第 j 個集合之隸屬函數，在此設定為鐘型函數（具有三個參數 a_{ji}, b_{ji}, c_{ji} ）

在此層產生的參數即為前提部分的參數，以 NL 表示其參數集合，參數總個數為 $3S$ 個。

第二層：規則層

將各輸入變數間的模糊集合進行排列組合之配對後，進行模糊邏輯運算



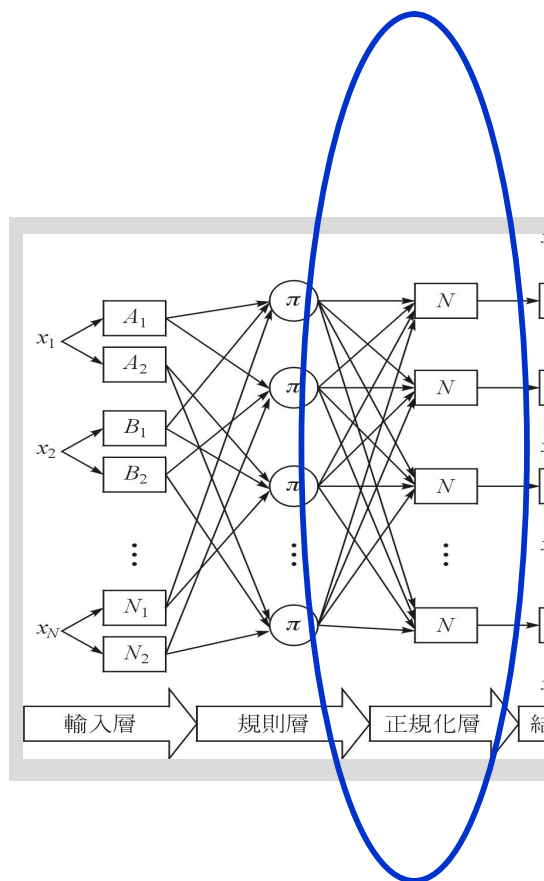
$$O_{2,p} = w_p = \prod_{i=1}^N \mu_{j_i}(x_i)$$

for $j_i = 1, \dots, M_i; p = 1, \dots, P$

本層選用 T-norm 進行模糊 AND 運算（本層節點以 Π 符號表示進行 AND 或乘積運算）

第三層：正規化層

將上一層各節點所得的結果正規化

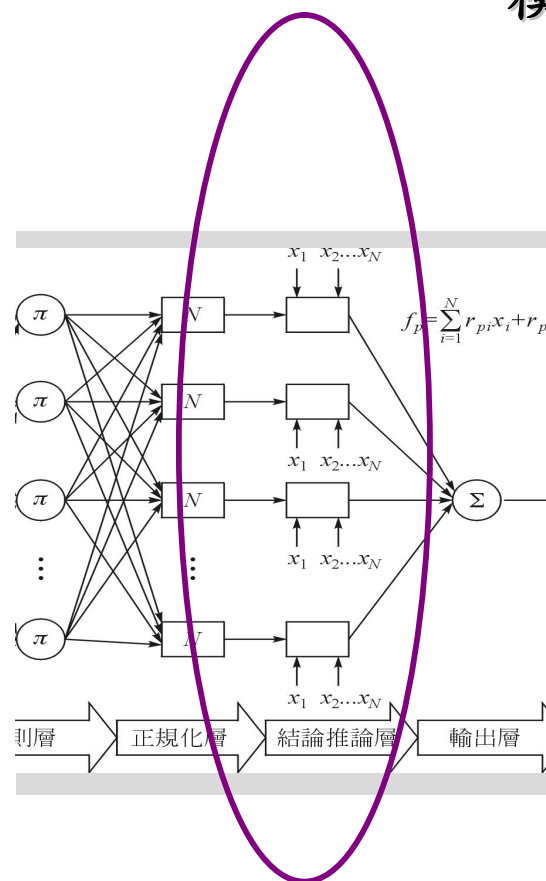


$$O_{3,p} = \bar{w}_p = \frac{w_p}{\sum_{p=1}^P w_p}$$

本層節點以N 符號表示，將第 i 條規則的輸出結果除以所有規則之輸出結果，使其輸出值介於 0 到 1 之間

第四層：結論推論層

將上層獲得之正規化結果與 Sugeno 模糊模式相乘

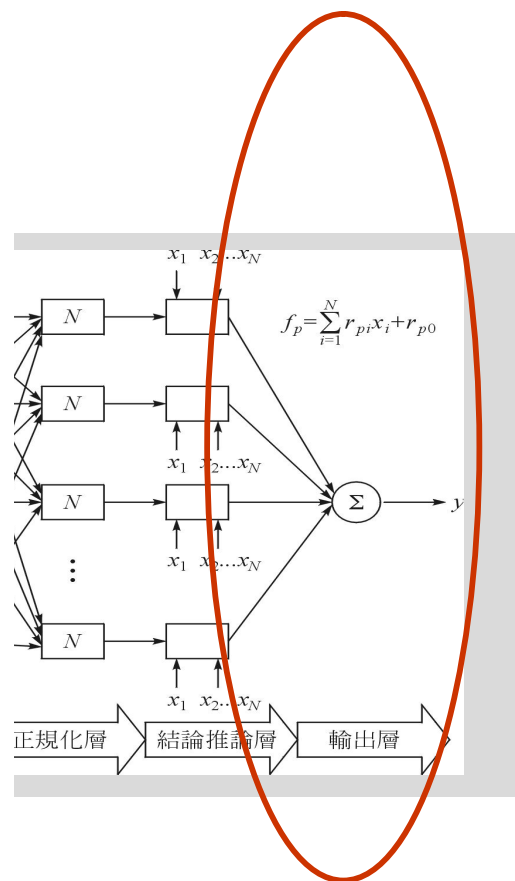


$$O_{4,p} = \bar{w}_p f_p = \bar{w}_p \left(\sum_{i=0}^N r_{pi} x_i \right), x_0 = 1$$

式中 r_{pi} 為一次 Sugeno 模糊模式的相關係數，亦為此層所產生的參數，稱為結論部分的參數，以 L 表示其參數集合，參數總個數為 $(N+1)P$ 個。

第五層：輸出層

總和上層各節點的輸出結果，以 Σ 符號表示



$$O_{5,1} = \sum_{p=1}^P \bar{w}_p f_p = \frac{\sum_{p=1}^P w_p f_p}{\sum_{p=1}^P w_p}$$

- ANFIS模式為結合前饋式類神經網路的監督式學習方法

- 代入訓練範例資料，比較真實輸出值與模式推估值之誤差，試圖求得誤差的最小平方和，使得模糊推論系統的所有參數可獲得適當的調整

前提部分參數與結論部分參數
(與其他模式為神經元間的連結權重向量不同)

在這兩大部分的參數中，可將其分為
非線性參數部分（即前提部分參數），以 **NL** 表示，
與
線性參數部分（即結論部分參數），以 **L** 表示；

第一層輸入層

輸出值 $O_{1,l} = \mu_{A_l}(x_1)$

$O_{1,m} = \mu_{B_m}(x_2)$ for $l, m, n = 1, 2$

$O_{1,n} = \mu_{C_n}(x_3)$

$$\text{式中 } \mu_{A_l} = \frac{1}{1 + \left| \frac{x_1 - c_{1l}}{a_{1l}} \right|^{2b_{1l}}}, \mu_{B_m} = \frac{1}{1 + \left| \frac{x_2 - c_{2m}}{a_{2m}} \right|^{2b_{2m}}}, \mu_{C_n} = \frac{1}{1 + \left| \frac{x_3 - c_{3n}}{a_{3n}} \right|^{2b_{3n}}}$$

相關參數 $\{a, b, c\}$ 為前提部分參數，總個數為 $3 \times 6 = 18$ 個。

第二層規則層

有 8 條規則，將第一層經過配對後，進行AND 模糊運算所得之輸出值如

$$w_{lmn} = \mu_{A_l}(x_1) \times \mu_{B_m}(x_2) \times \mu_{C_n}(x_3) \text{ for } l, m, n = 1, 2$$

第三層正規化層

將第二層各節點所得的結果正規化，因此，本層輸出值總個數與第二層相同（都是 8 個），其輸出結果介於 0 與 1 之間

$$O_{3,lmn} = \bar{w}_{lmn} = \frac{w_{lmn}}{\sum_{l,m,n=1}^2 w_{lmn}}$$

第四層結論推論層

輸出結果 $O_{4,lmn} = \bar{w}_{lmn} f_{lmn} = \bar{w}_{lmn} \left(\sum_{i=0}^3 r_{lmn,i} x_i \right)$

其相關參數 $r_{lmn,i}$ 為結論部分參數，共 $(3+1) \times 8 = 32$ 個。

第五層輸出層

將第四層的結果總合輸出，如

$$y = O_{5,1} = \sum_{l,m,n=1}^2 \bar{w}_{lmn} f_{lmn} = \sum_{l,m,n=1}^2 \bar{w}_{lmn} \left(\sum_{i=0}^3 r_{lmn,i} x_i \right)$$



訓練範例資料之目標輸出值與網路推估值的誤差平方，如

$$e_k = (d_k - y_k)^2$$

希望能夠對於整個訓練範例資料獲得最小誤差值，即E值最小 $E = \sum_{k=1}^K e_k$
在網路的學習過程中，可使用最陡坡降法來搜尋E值最小化的過程；
當輸入第k個訓練範例資料時，網路即小幅調整加權值的大小如

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} = -\eta \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \alpha} = 2\eta(d_k - y_k) \frac{\partial y_k}{\partial \alpha}$$

式中 α 為參數集合 (NL + L)， α_{NL} 表示 NL 集合的參數， α_L 則表示 L 集合的參數，

對 α_L 進行微分，可得 α_L 的修正值 $\Delta\alpha_L = 2\eta(d_k - y_k) \sum_{l,m,n} \bar{w}_{lmn}$

相似的方式將誤差值傳至第二層可以獲得 $\Delta\alpha_{NL}$

複合型學習演算法



- 參數集合有『線性』與『非線性』兩部分
複合型學習演算法 $\Rightarrow$ 利用 LSE 來調整線性參數集合，
以最陡坡降法調整非線性參數集合

非線性參數集合部分

以 NL 表示

$$\{ a_{ji}, b_{ji}, c_{ji} \}$$

線性參數集合部分

以 L 表示

$$\{ r_{pi} \}$$