# KON 426E
# INTELLIGENT CONTROL SYSTEMS

## LECTURE 12
## 23/05/2022

# ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

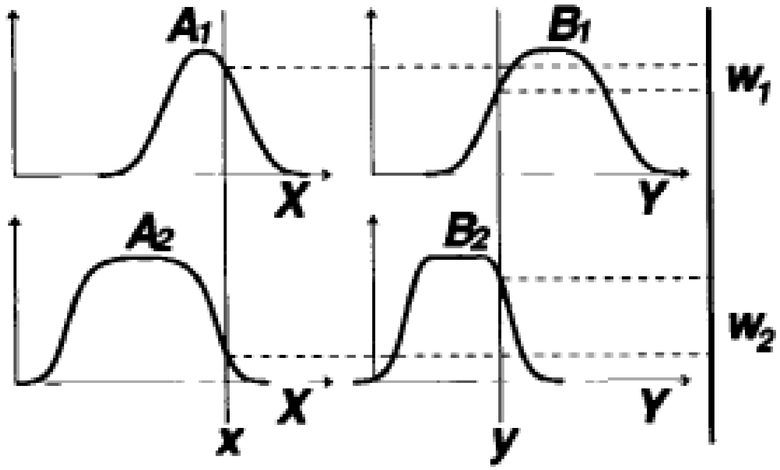
1. Learning ability
2. Parallel computation
3. Structural information representation
4. Better integration with other methods

NN's are better

FIS's are better

- Remember a TSK (Takagi-Sugeno-Kang) type of FIS (Shortly: Sugeno)
- A two input, first-order Sugeno fuzzy model with two rules:

$$\begin{aligned}
&\text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1\text{, then } f_1 = p_1x + q_1y + r_1\\
&\text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2\text{, then } f_2 = p_2x + q_2y + r_2
\end{aligned}$$

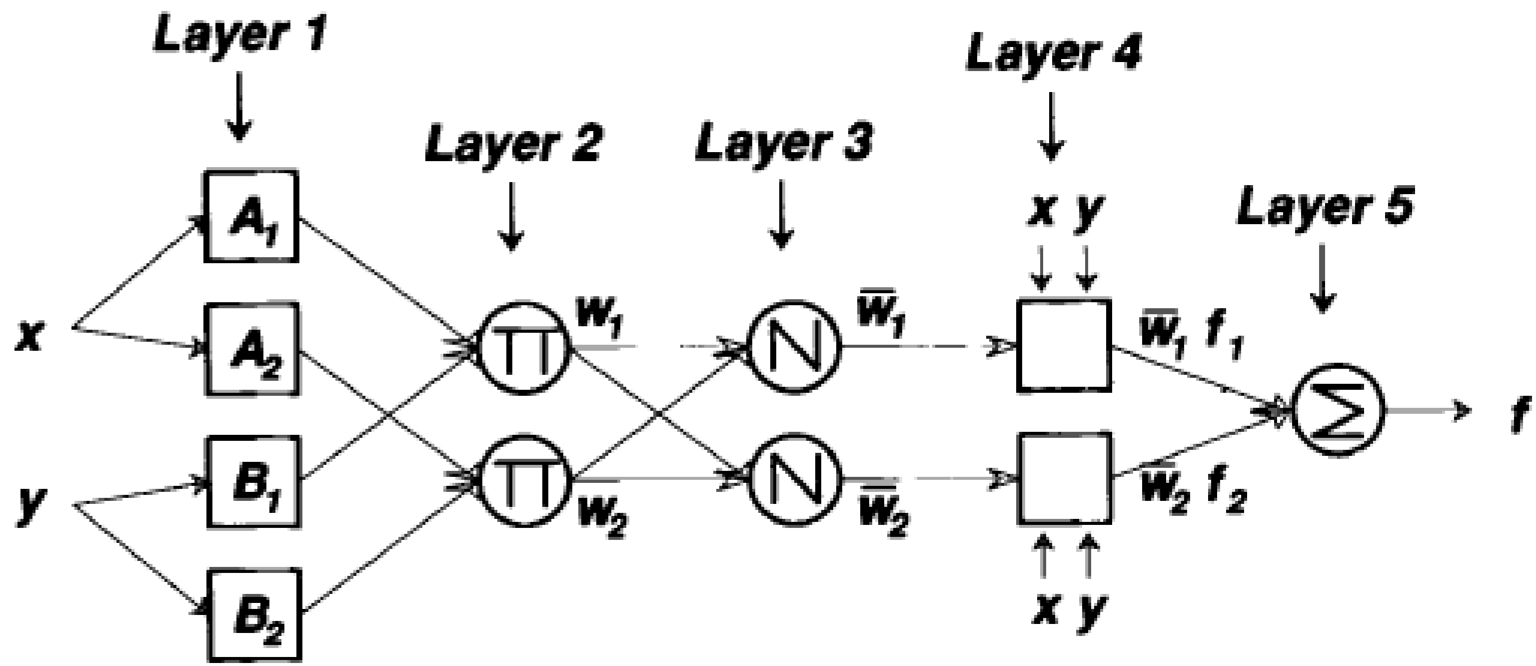



$$f_1 = p_1x + q_1y + r_1$$

$$f_2 = p_2x + q_2y + r_2$$

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 f_1 + \bar{w}_2 f_2$$

- An equivalent ANFIS arcitecture:



## Layer 1:

Every node $i$ in this layer is an **adaptive node** with a node function:

$$O_{1,i} = \mu_{A_i}(x), \qquad \text{for } i = 1, 2, \text{ or}$$
$$O_{1,i} = \mu_{B_{i-2}}(y), \quad \text{for } i = 3, 4,$$

$x$ and $y$ are inputs to node $i$

$A_i$ (or $B_{i-2}$) is a linguistic label (like “small” or “large”)

$O_{1,i}$ is the membership grade of a fuzzy set $A_1, A_2, B_1$ or $B_2$

And it specifies the degree to which the given input (x or y) satisfies the quantifier A.

MF can be any appropriate parameterized and piecewise differentiable MF.

Parameters in this layer are called **premise parameters**.

For example:

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

$\{a_i, b_i, c_i\}$ is a parameter set.

$$\mu_B(x) = \exp\left[-\left(\frac{x-m_i}{\sigma_i}\right)^2\right]$$

$\{m_i, \sigma_i\}$ is a parameter set.

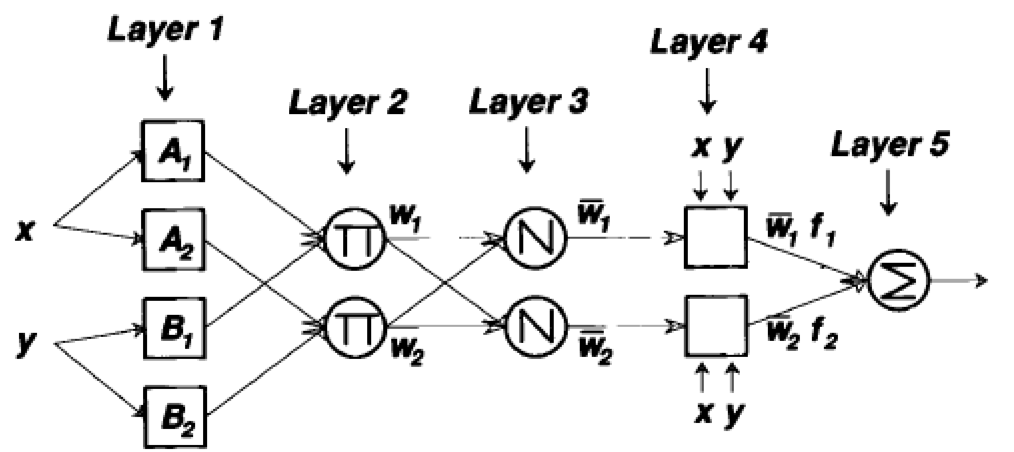
## Layer 2:

Every node in this layer is a **fixed node** labeled $\Pi$. Its output is the **product** of all incoming signals:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y),\ i = 1, 2$$

Each node output represents the **firing strength** of a rule.

In general, any other T-norm operator that performs fuzzy AND can be used as the node function in this layer.



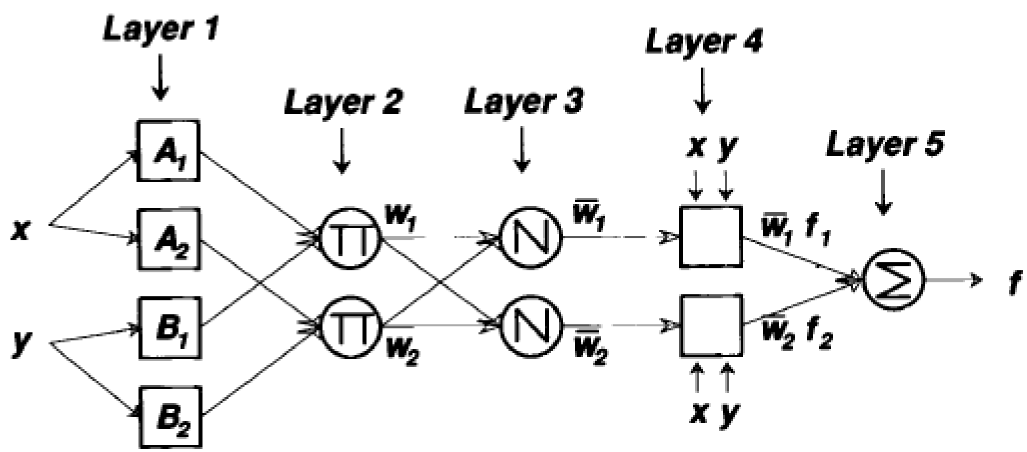
## Layer 3

Every node in this layer is a **fixed node** labeled $\mathrm{N}$ .

The $i$th node calculates the ratio of the $i$th rule's firing strength to the sum of all rules' firing strengths:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \; i = 1, 2$$

The output of this layer is called **"normalized firing strength"**



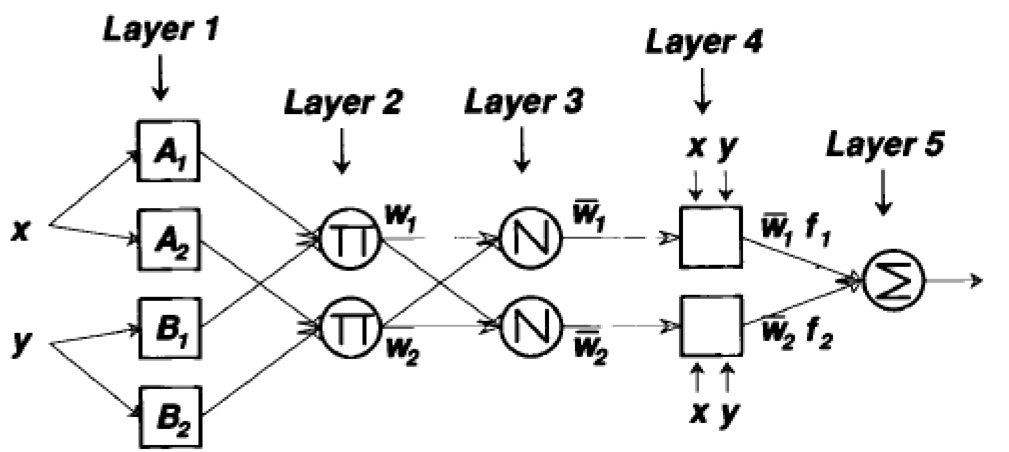
**Layer 4**

Every node $i$ in this layer is an **adaptive node** with a node function:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i)$$

$\overline{w}_i$ : normalized firing strength from layer 3.

$\{p_i,\ q_i,\ r_i\}$ is the parameter set of this node. They are called as **"consequent parameters"**

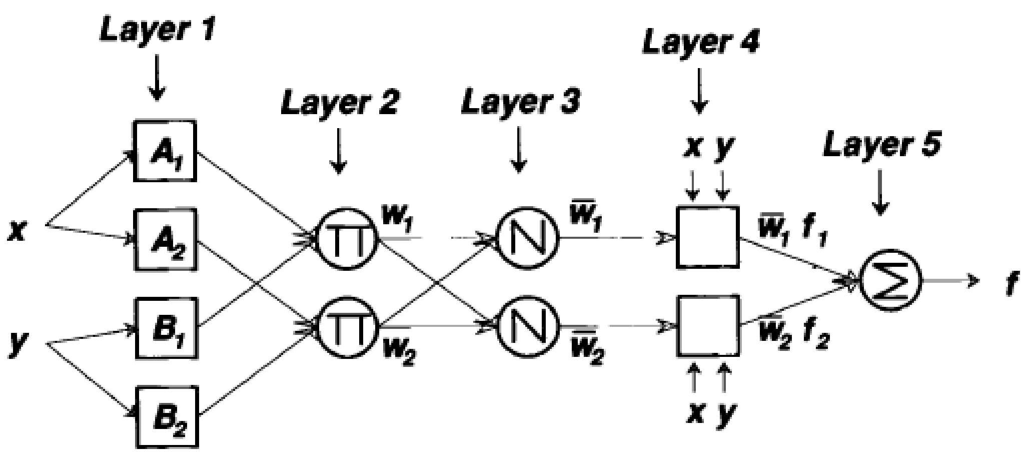


## Layer 5

There is a single fixed node labeled $\Sigma$ which computes the overall output as the summation of all incoming signals:

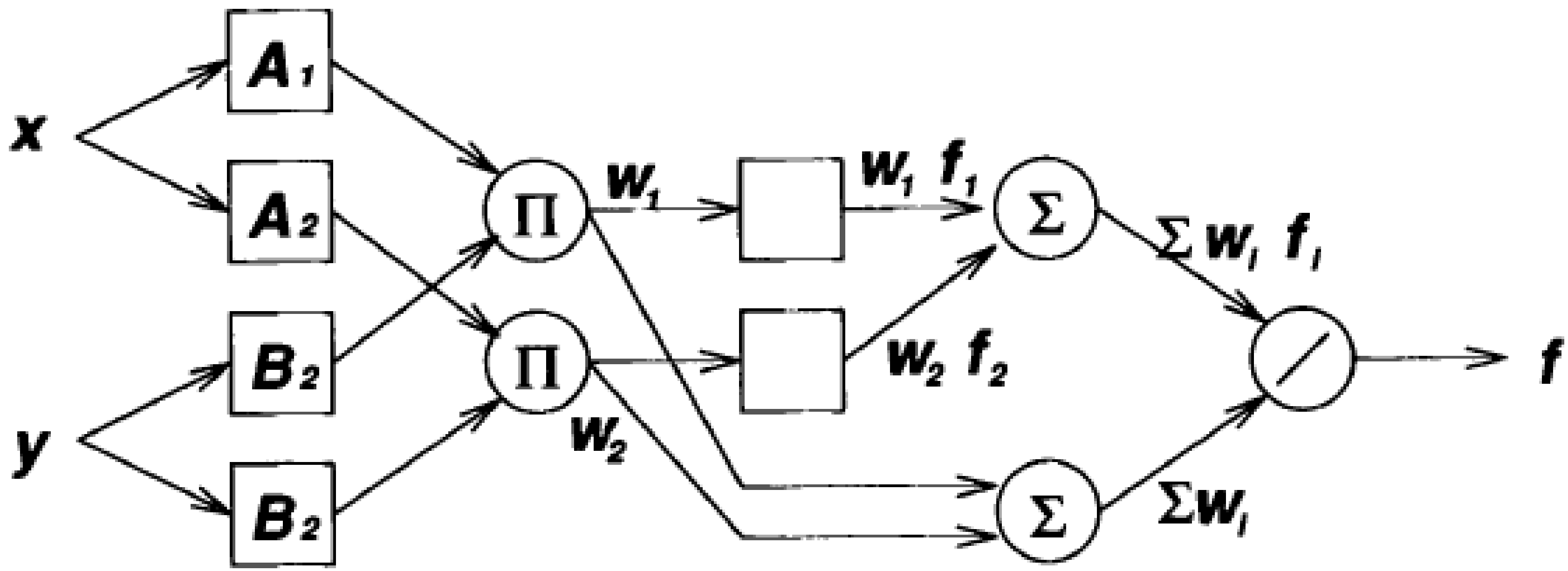
$$\text{overall output} = O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

$$f = [\overline{w}_1 \quad \overline{w}_1 x \quad \overline{w}_1 y \quad \overline{w}_2 \quad \overline{w}_2 x \quad \overline{w}_2 y] \begin{bmatrix} r_1 \\ p_1 \\ q_1 \\ r_2 \\ p_2 \\ q_2 \end{bmatrix}$$



- This is an adaptive network that is functionally equivalent to a Sugeno fuzzy model.
- The structure of this adaptive network is not unique.
- We can combine layers 3 and 4 to obtain an equivalent network with only four layers.
- Also we can perform weight normalization at the last layer.

An alternative architecture for ANFIS for the Sugeno fuzzy model (weight normalization is performed at the last layer)



- Extension from Sugeno ANFIS to Tsukamoto ANFIS is straightforward.
- For the Mamdani FIS, an ANFIS can be obtained if max-min composition and discrete approximation to replace the integrals in the centroid defuzzification scheme are used.
- However, resulting ANFIS is much more complicated than Sugeno-ANFIS or Tsukamoto-ANFIS.
- The extra complexity in structure and computation of Mamdani-ANFIS does not imply better learning capability or approximation power.

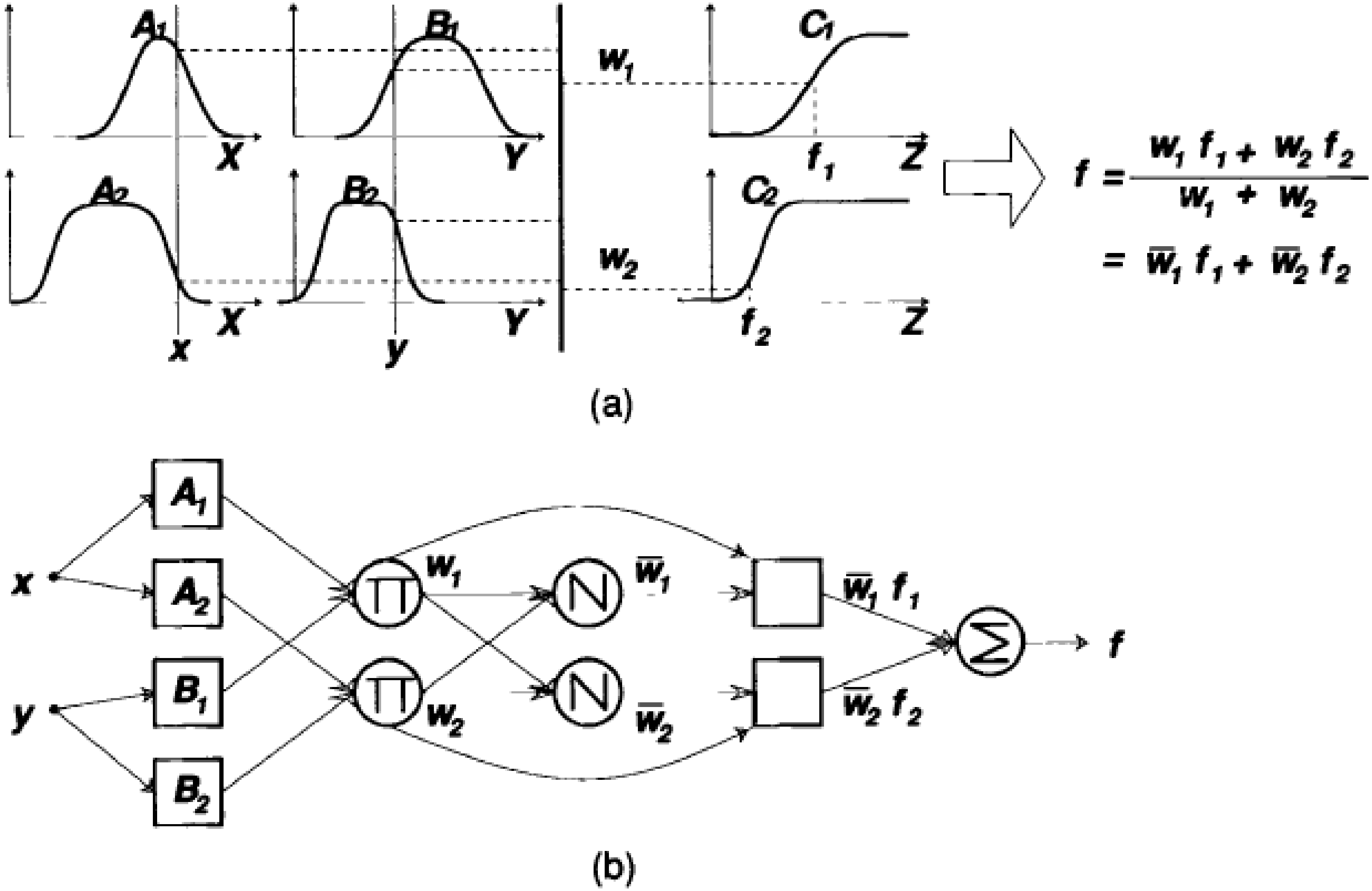



**Figure 12.3.** *(a) A two-input two-rule Tsukamoto fuzzy model; (b) equivalent ANFIS architecture.*

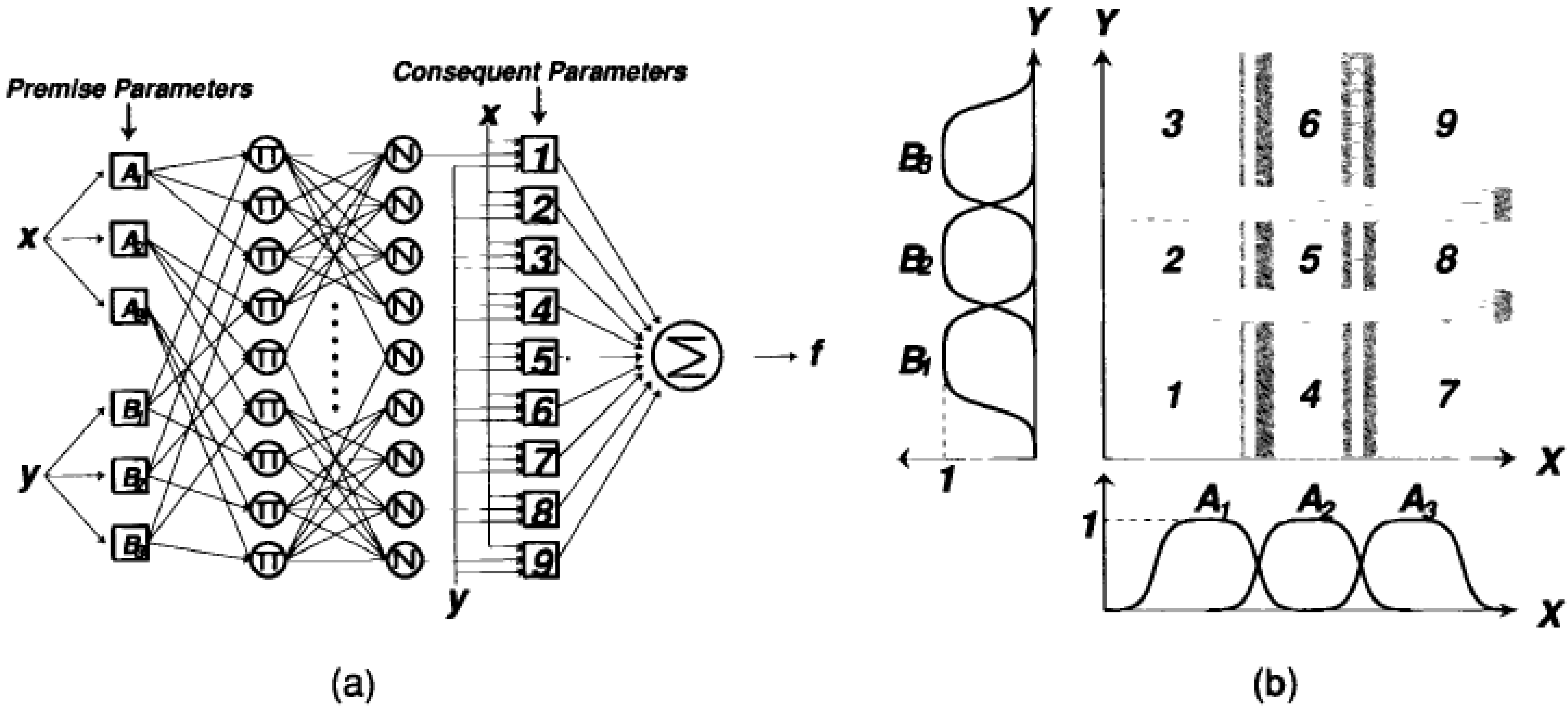



**Figure 12.4.** *(a) ANFIS architecture for a two-input Sugeno fuzzy model with nine rules; (b) the input space that are partitioned into nine fuzzy regions.*

# Backpropagation Algorithm for ANFIS

Parameters to be updated:

- **Premise parameters**

If you use Gaussian MF's with $\mu_{A_i}(x) = \exp\left[-\left(\frac{x - m_i}{\sigma_i}\right)^2\right]$

premise parameters will be $m_i$ and $\sigma_i$

- **Consequent parameters**

For a Sugeno ANFIS with first order polynomials as outputs

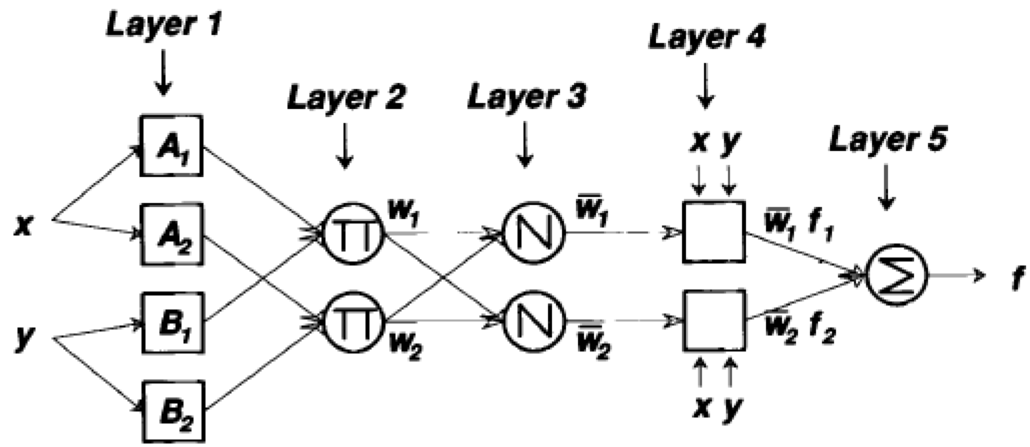
they will be $p_i, q_i, r_i$

$\boldsymbol{d}$: desired output

$\boldsymbol{f}$: actual output

$e = d - f$

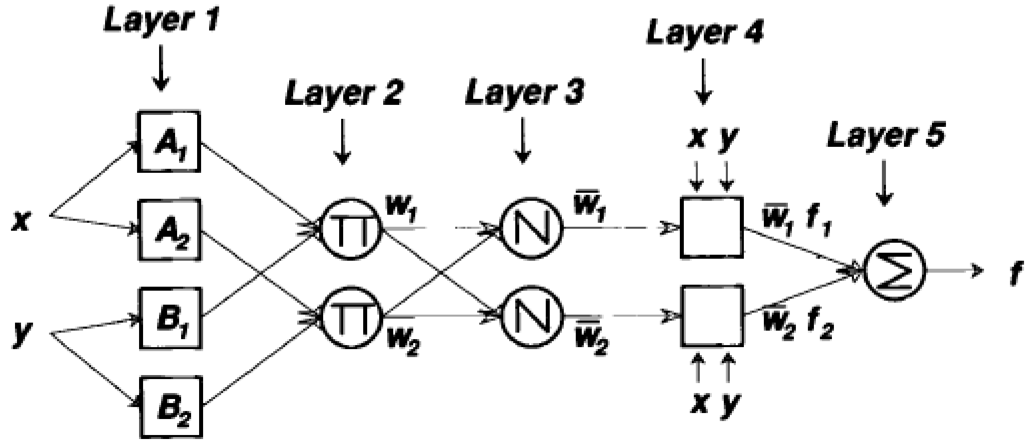
Cost function: $E = \frac{1}{2} e^2$




**Layer 5:** There is no update in this layer.

$$\delta_i^5 = -\frac{\partial E}{\partial f} = -\frac{\partial E}{\partial e}\frac{\partial e}{\partial f} = -\frac{1}{2}\,2\,e\,(-1) = e$$

$$E = \frac{1}{2}e^2 \qquad e = d - f$$




$(i = 1, 2, \dots)$

**Layer 4**: The consequent parameters $p_i, q_i, r_i$ are updated.

$$p_i(k+1) = p_i(k) + \eta\, \Delta p_i(k)$$

$$q_i(k+1) = q_i(k) + \eta\, \Delta q_i(k)$$

$$r_i(k+1) = r_i(k) + \eta\, \Delta r_i(k)$$

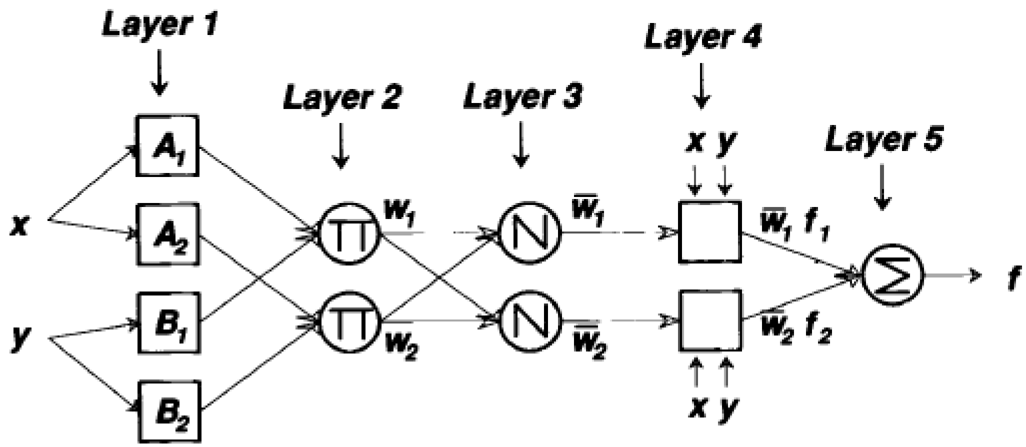
$$\Delta p_i = -\frac{\partial E}{\partial p_i} = -\frac{\partial E}{\partial f}\frac{\partial f}{\partial p_i} = e(\bar{w}_i\, x)$$

$$\Delta q_i = -\frac{\partial E}{\partial q_i} = -\frac{\partial E}{\partial f}\frac{\partial f}{\partial q_i} = e(\bar{w}_i\, y)$$

$$f = \sum_i \bar{w}_i f_i = \sum_i \bar{w}_i (p_i x + q_i y + r_i)$$

$$\Delta r_i = -\frac{\partial E}{\partial r_i} = -\frac{\partial E}{\partial f}\frac{\partial f}{\partial r_i} = e\, \bar{w}_i$$

$$\frac{\partial f}{\partial p_i} = \bar{w}_i\, x$$

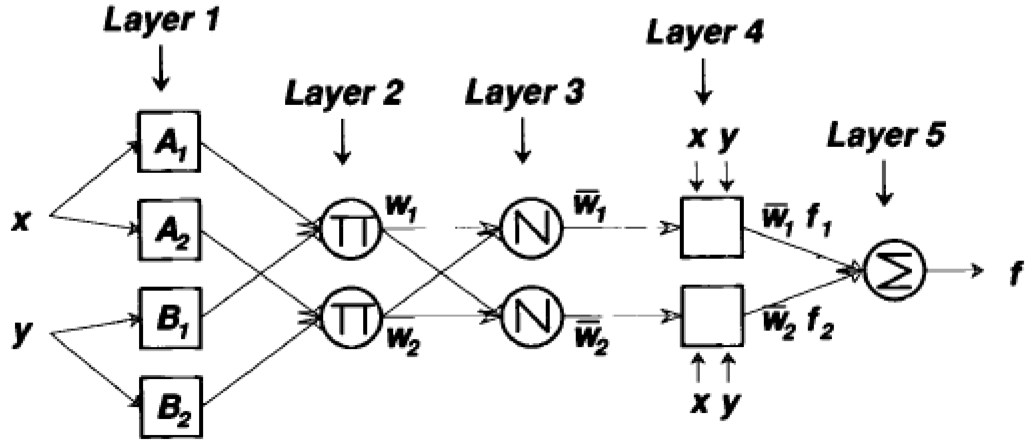



**Layer 3**: There is no update in this layer.

$$\delta_i^3 = -\frac{\partial E}{\partial \bar{w}_i} = -\frac{\partial E}{\partial f}\frac{\partial f}{\partial \bar{w}_i} \qquad (i = 1, 2, \dots.)$$

$$f = \bar{w}_1 f_1 + \bar{w}_2 f_2 \qquad \frac{\partial f}{\partial \bar{w}_i} = f_i$$

$$\delta_i^3 = e\, f_i$$




**Layer 2**: There is no update in this layer.

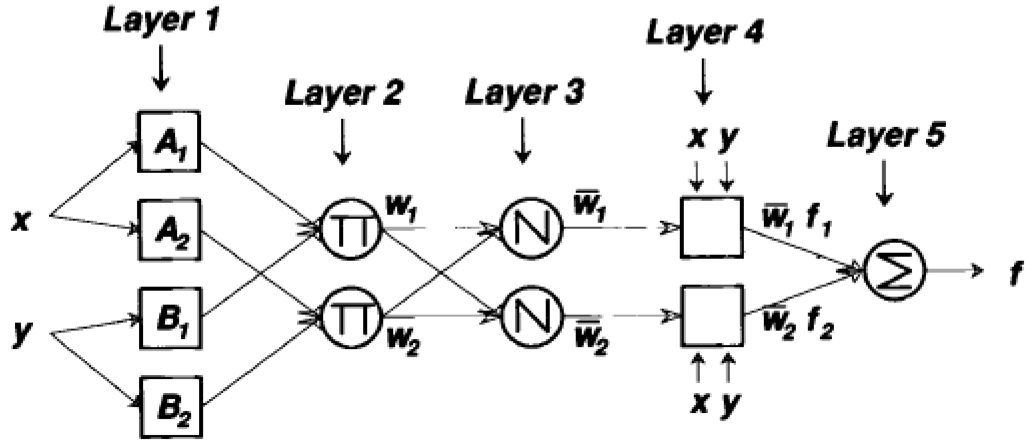
$$\delta_i^2 = -\frac{\partial E}{\partial w_i} = -\frac{\partial E}{\partial f}\frac{\partial f}{\partial \bar{w}_i}\frac{\partial \bar{w}_i}{\partial w_i} \qquad (i = 1, 2, \ldots .)$$

$$\bar{w}_i = \frac{w_i}{\sum_i w_i} \qquad a = \sum_i w_i \qquad \frac{\partial \bar{w}_i}{\partial w_i} = \frac{a - w_i}{a^2}$$

For example: $\bar{w}_1 = \dfrac{w_1}{w_1 + w_2 + w_3}$

$$\frac{\partial \bar{w}_1}{\partial w_1} = \frac{(w_1 + w_2 + w_3) - w_1}{(w_1 + w_2 + w_3)^2} = \frac{w_2 + w_3}{(w_1 + w_2 + w_3)^2} = \frac{a - w_1}{a^2}$$

$$\delta_i^2 = e\, f_i \left[\frac{a - w_1}{a^2}\right] = \delta_i^3 \left[\frac{a - w_1}{a^2}\right]$$




**Layer 1**: In this layer "premise parameters" related to the MF's are updated.

Using Gaussian MF's:

$$\mu_{A_i}(x) = \exp\left[-\left(\frac{x - m_i}{\sigma_i}\right)^2\right]$$

$$\mu_{A_i}(x) = exp[v_i(x)]$$

$$v_i(x) = -\left(\frac{x - m_i}{\sigma_i}\right)^2$$

$m_i$ and $\sigma_i$ should be updated.

$$m_i(k+1) = m_i(k) + \eta\,\Delta m_i(k)$$

$$\sigma_i(k+1) = \sigma_i(k) + \eta\,\Delta\sigma_i(k)$$

$$\delta_i^1 = -\frac{\partial E}{\partial \mu_{A_i}} = \left[\underbrace{-\frac{\partial E}{\partial f}\frac{\partial f}{\partial \bar{w}_i}\frac{\partial \bar{w}_i}{\partial w_i}}_{\delta_i^2}\frac{\partial w_i}{\partial \mu_{A_i}}\right]$$

$$w_i = \mu_{A_i}(x)\,\mu_{B_i}(y) \qquad\qquad i = 1,2$$

$$\frac{\partial w_i}{\partial \mu_{A_i}} = \mu_{B_i}(y)$$

$$\delta_i^1 = -\frac{\partial E}{\partial \mu_{A_i}} = \delta_i^2\,\mu_{B_i}(y)$$

Similarly, $\delta_{i+2}^1 = -\dfrac{\partial E}{\partial\, \mu_{B_i}} = \left[-\dfrac{\partial E}{\partial f}\dfrac{\partial f}{\partial \bar{w}_i}\dfrac{\partial \bar{w}_i}{\partial w_i}\dfrac{\partial w_i}{\partial\, \mu_{B_i}}\right]$

$$\frac{\partial w_i}{\partial \mu_{B_i}} = \mu_{A_i}(x) \qquad\qquad \delta_{i+2}^1 = \delta_i^2\,\mu_{A_i}$$

Now update $m_i$ and $\sigma_i$

For $\mu_{A_i}$ :

$$\Delta m_i = -\frac{\partial E}{\partial m_i} = \left[ \underbrace{-\frac{\partial E}{\partial f}\frac{\partial f}{\partial \bar{w}_i}\frac{\partial \bar{w}_i}{\partial w_i}}_{\delta_i^2} \underbrace{\frac{\partial w_i}{\partial \mu_{A_i}}}_{\mu_{B_i}} \underbrace{\frac{\partial \mu_{A_i}}{\partial m_i}}_{\frac{\partial \mu_{A_i}}{\partial m_i} = \frac{\partial \mu_{A_i}}{\partial v_i}\frac{\partial v_i}{\partial m_i}} \right]$$

$$\frac{\partial \mu_{A_i}}{\partial v_i} = \mu_{A_i}$$

$$\frac{\partial v_i}{\partial m_i} = -2\left(\frac{x - m_i}{\sigma_i}\right)\left(\frac{-1}{\sigma_i}\right) = \frac{2(x - m_i)}{\sigma_i^2}$$

$$\Delta m_i = \delta_i^2\, \mu_{B_i}\mu_{A_i}\frac{2(x - m_i)}{\sigma_i^2}$$

For $\mu_{B_i}$ $\longrightarrow$ $\Delta m_{i+2} = \delta_i^2\, \mu_{B_i}\mu_{A_i}\frac{2(y - m_{i+2})}{\sigma_{i+2}^2}$

For $\mu_{A_i}$

$$\Delta\sigma_i = -\frac{\partial E}{\partial \sigma_i} = \left[ \underbrace{-\frac{\partial E}{\partial f}\frac{\partial f}{\partial \bar{w}_i}\frac{\partial \bar{w}_i}{\partial w_i}}_{\delta_i^2} \underbrace{\frac{\partial w_i}{\partial \mu_{A_i}}}_{\mu_{B_i}} \underbrace{\frac{\partial \mu_{A_i}}{\partial v_i}}_{\mu_{A_i}} \frac{\partial v_i}{\partial \sigma_i} \right] \qquad i = 1,2$$

$$v_i(x) = -\left(\frac{x - m_i}{\sigma_i}\right)^2 = -(x - m_i)^2 \sigma_i^{-2}$$

$$\frac{\partial v_i}{\partial \sigma_i} = -(x - m_i)^2(-2)\sigma_i^{-3} = \frac{2(x - m_i)^2}{\sigma_i^3}$$

$$\Delta\sigma_i = \delta_i^2\, \mu_{B_i}\mu_{A_i} \frac{2(x - m_i)^2}{\sigma_i^3}$$

$$\Delta\sigma_{i+2} = \delta_i^2\, \mu_{B_i}\mu_{A_i} \frac{2(y - m_{i+2})^2}{\sigma_{i+2}^3}$$

$$m_i(k+1) = m_i(k) + \eta\, \Delta m_i(k)$$

$$\sigma_i(k+1) = \sigma_i(k) + \eta\, \Delta\sigma_i(k)$$

# Hybrid Learning Algorithm

When the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters.

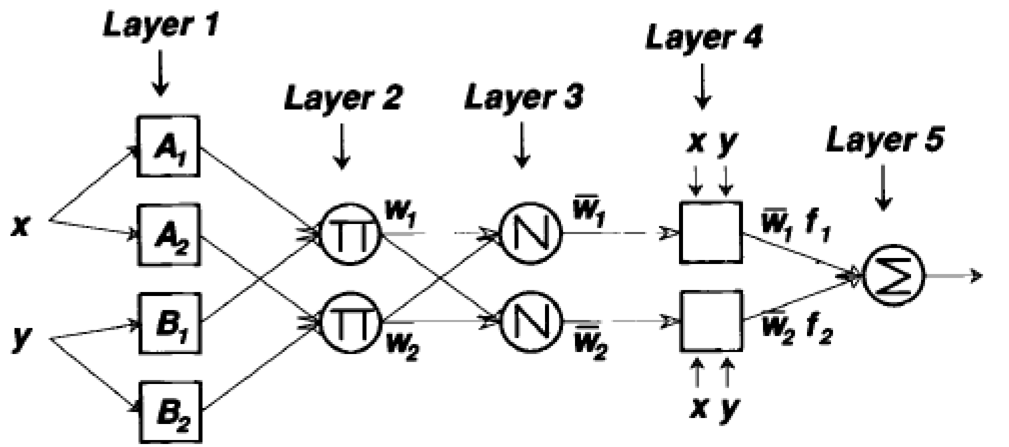
$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2$$

$$f = \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2)$$

$$f = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2$$

So, this is linear in the consequent parameters:

$$p_1, q_1, r_1, p_2, q_2, r_2$$



***S***= set of total parameters

***S1***= set of premise (nonlinear) parameters

***S2***= set of consequent (linear) parameters

***S1***= $\{m_1, \sigma_1, m_2, \sigma_2, m_3, \sigma_3, m_4, \sigma_4\}$ $\longrightarrow$ Updated by gradient descent

***S2***= $\{p_1, q_1, r_1, p_2, q_2, r_2\}$ $\longrightarrow$ Updated by least squares

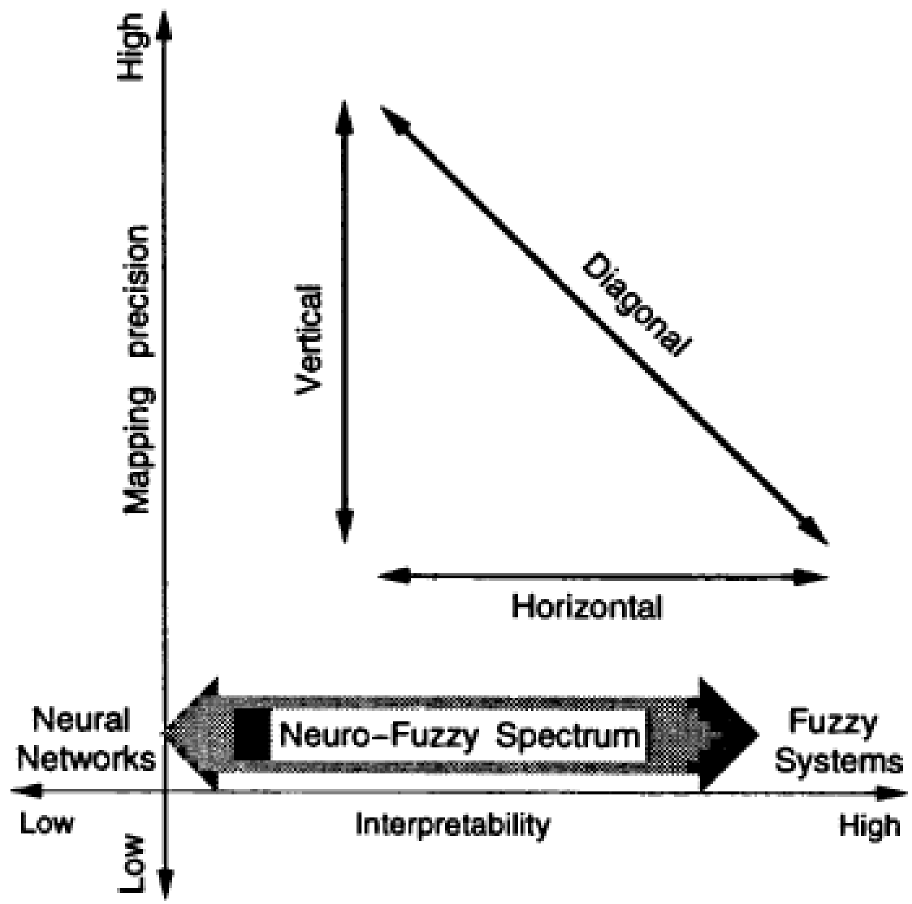
Two passes in the hybrid learning procedure for ANFIS

| | Forward pass | Backward pass |
|---|---|---|
| Premise parameters | Fixed | Gradient descent |
| Consequent parameters | Least-squares estimator | Fixed |
| Signals | Node outputs | Error signals |

- Consequent parameters are optimal under the condition that premise parameters are fixed.

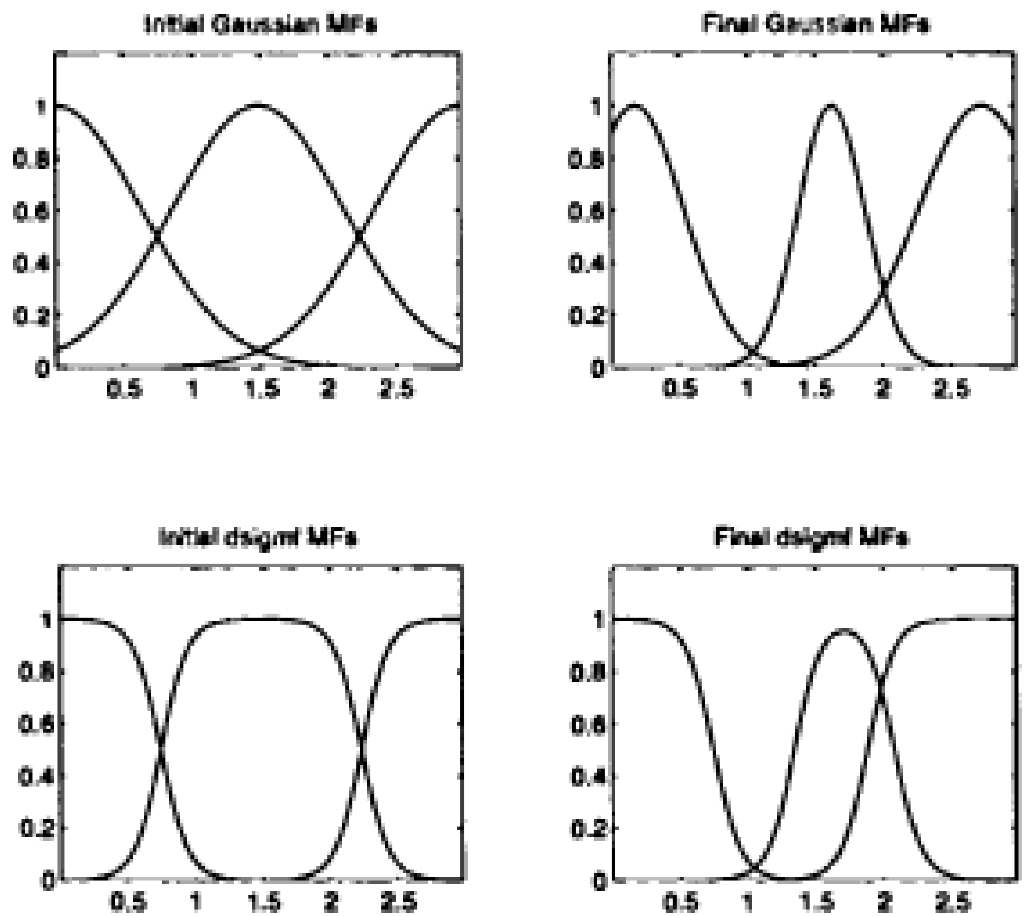
- The hybrid approach converges much faster since it reduces the search space dimensions of the original pure backpropagation method.

# The Neuro-Fuzzy Spectrum



- Trade-offs between input-output mapping precision and MF interpretability.
- Prior knowledge is embedded via fuzzy rules.
- After learning, the resulting model can be understood via these fuzzy rules.
- Black-box NN’s (particularly MLP’s) do not have the same level of ability for knowledge embedding and extraction.
- Ideally the learning of a neuro-fuzzy model should follow the vertical route to the top (mapping precision should be improved while interpretability is maintained)
- In practice, it follows the diagonal route.
- When mapping precision is improved, interpretability is deteriorated.

- Sophisticated learning methods may attain higher input-output precision, but this may lead to meaningless fuzzy rules.



- ANFIS is a **universal approximator.**
- When the number of rules is not restricted, a zero-order Sugeno model has unlimited approximation power for matching any nonlinear function arbitrarily well on a compact set.

  **(Stone-Weierstrass theorem, pg.343-344, (Jang,Sun and Mituzani)**
- Let domain $D$ be a compact space of $N$ dimensions and let $\mathcal{F}$ be a set of continuous real-valued functions on $D$ satisfying identity,separability and algebraic closure properties.
- For any $\epsilon > 0$ and any function $g$ in $C(D)$ (continuous real-valued functions on $D$) there is a funtion $f$ in $\mathcal{F}$ such that $|g(x) - f(x)| < \epsilon$ for all $x \in D$