Neuro - Fuzzy Modelling

* ANFIS - Adaptive Neuro - Fuzzy Inference Systems

→ a class of adaptive networks that are functionally equivalent to fuzzy inference systems

* ANFIS Architecture

→ Assume that the FIS under consideration has two inputs x and y and one output z.

→ Nodes of the same layer have equivalent functionalities

Layer 1 - Each node i in this layer is an adaptive node with a node function:

$$O_{1,i} = \mu_{A_i}(x) \quad i = 1,2 \quad \text{or}$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \quad i = 3,4$$

(x, y) = input to the i^{th} node

A_i, B_{i-2} = linguistic labels

→ $O_{1,i}$ is the membership grade of a fuzzy set A, and it specifies the degree to which the given input x or y satisfies the quantifier A.

→ A can have any membership fn. If the generalized bell fn:

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

→ Parameters in this layer are called <u>premise parameters</u>.

<u>Layer 2:</u> Every node in this layer is a fixed node labeled Π, whose output is the product of all incoming signals.

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

<u>Layer 3:</u> Every node in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all the rules' firing strengths.

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

→ Outputs of this layer are called <u>normalized firing strengths</u>.

<u>Layer 4:</u> Every node i in this layer is an adaptive node with a node function

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

$\overline{w_i}$ → from Layer 3

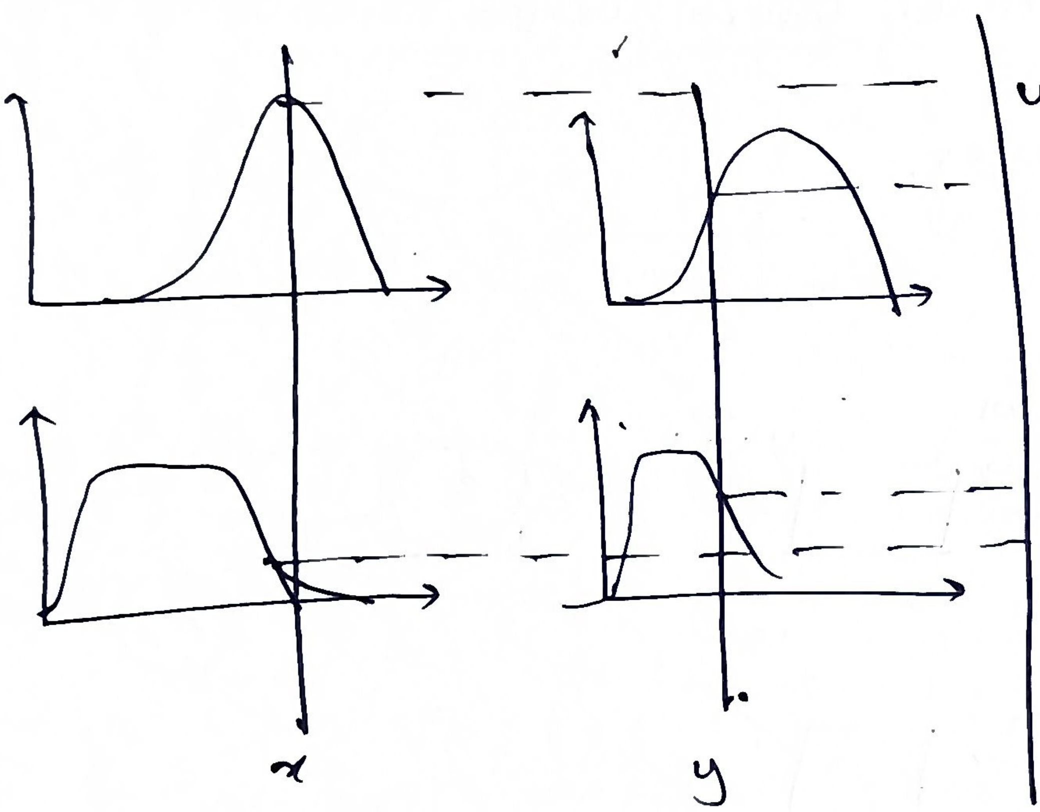
→ Parameters in this layer are called consequent parameters

Layer 5: The single node in this layer is a fixed node labeled Σ, which computes the overall output as the summation of all incoming signals.

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

* ANFIS Models for Different Inference Systems

(1) First order Sugeno Fuzzy model



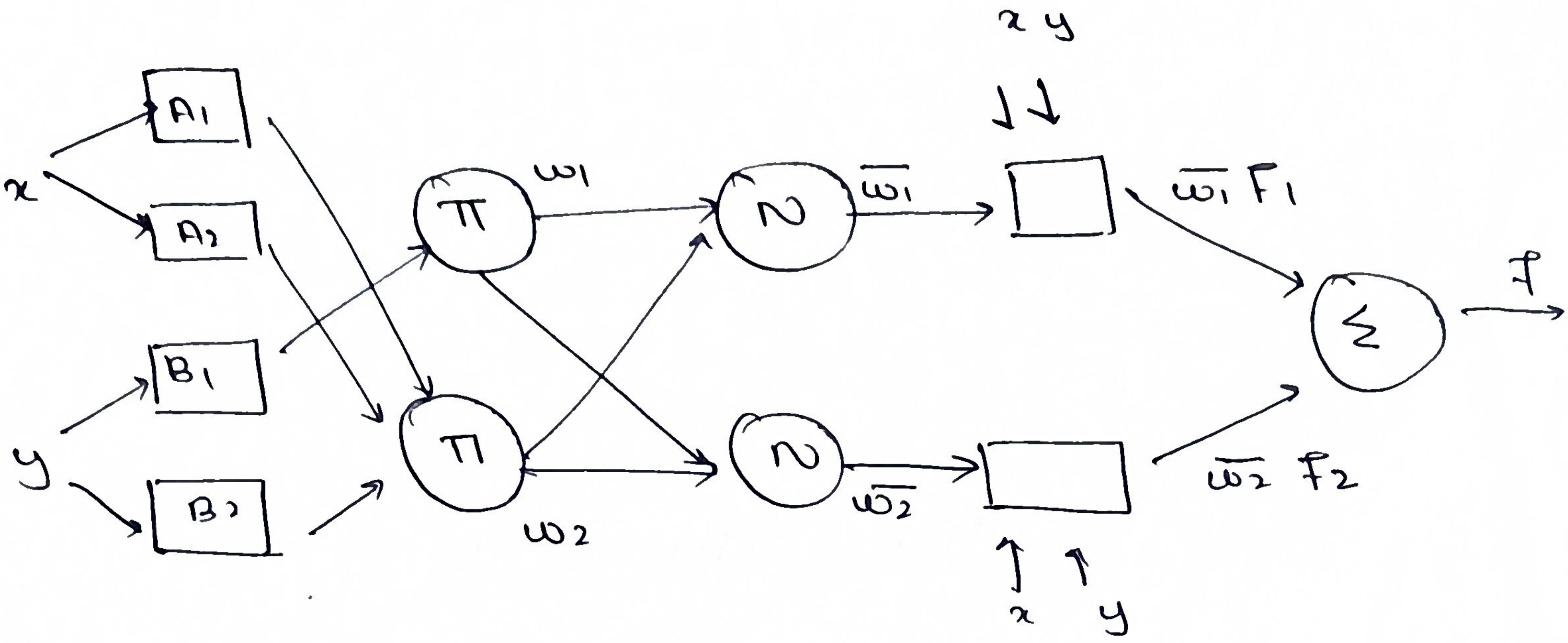
w_1 $f_1 = p_1 x + q_1 y + r_1$

w_2 $f_2 = p_2 x + q_2 y + r_2$

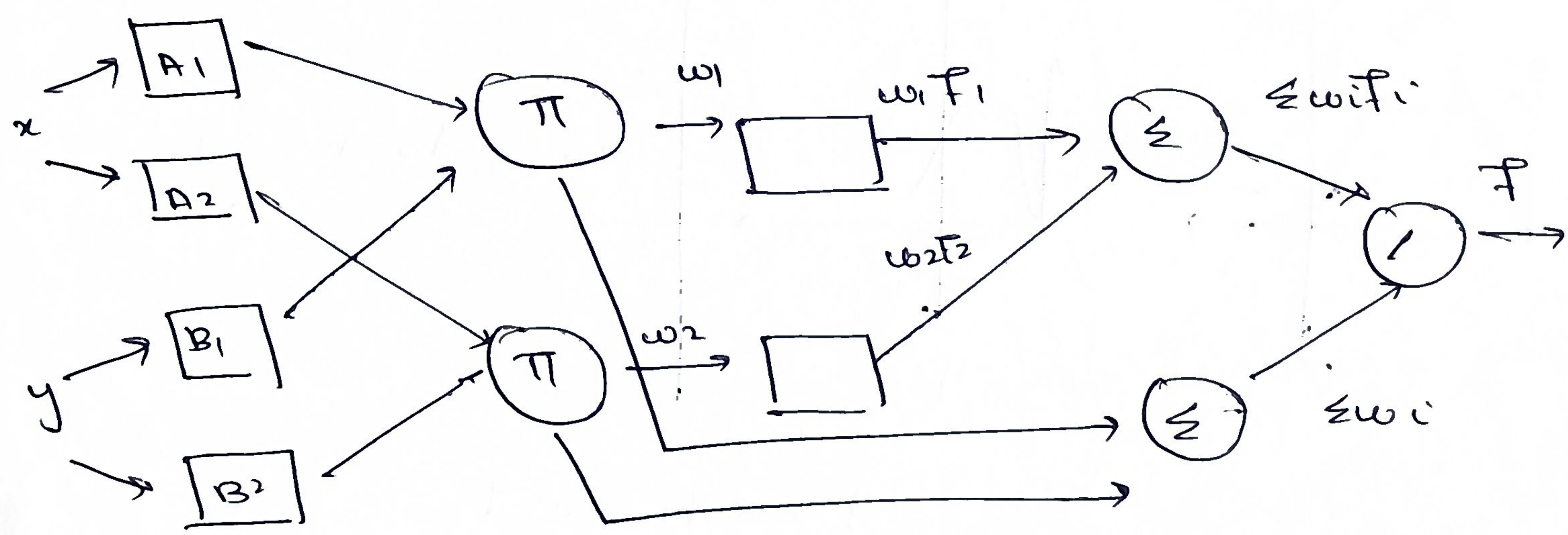
$$\Rightarrow f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

$$= \overline{w_1} f_1 + \overline{w_2} f_2$$

Equivalent Anfis Structure

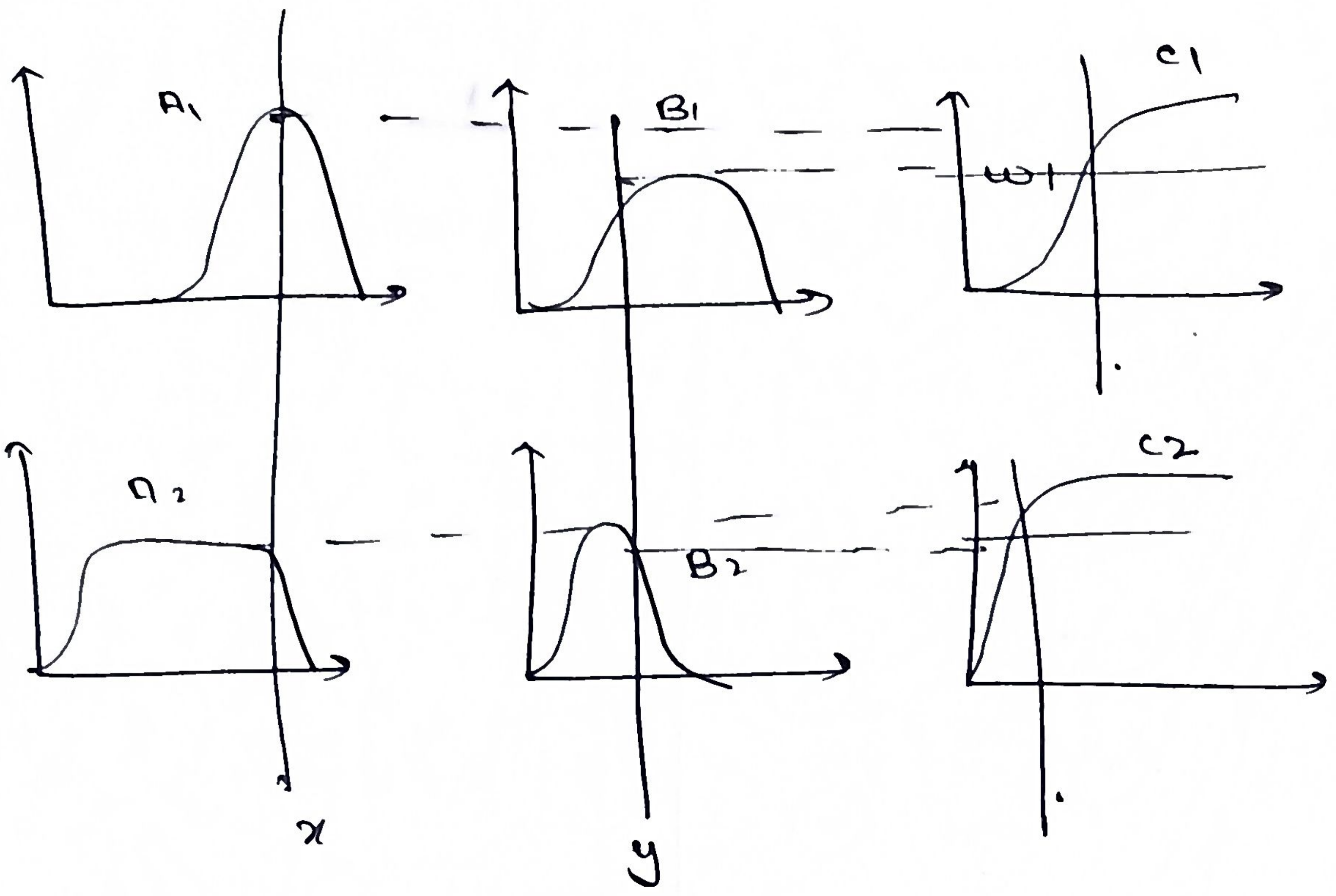


(2) ANFIS for Sugeno Fuzzy Model, where weight normalization happens at the very last layer



(3) ANFIS for Tsukamoto Fuzzy Inference System

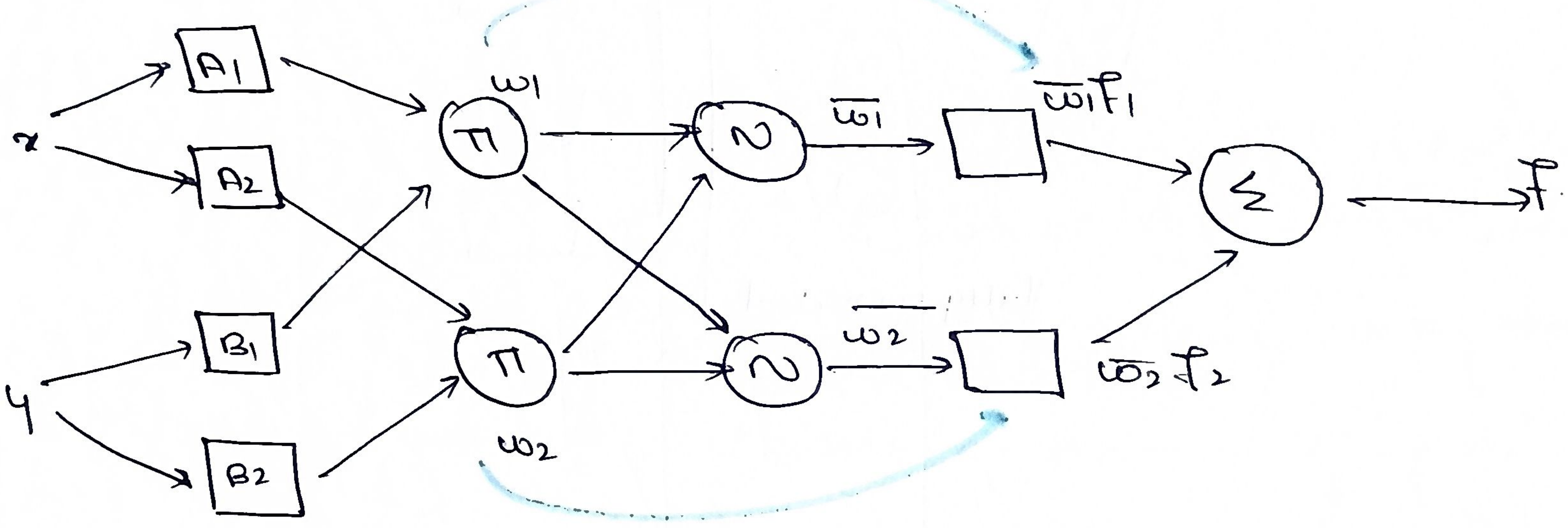
→ Output of each rule is induced jointly by a consequent membership function & a firing strength



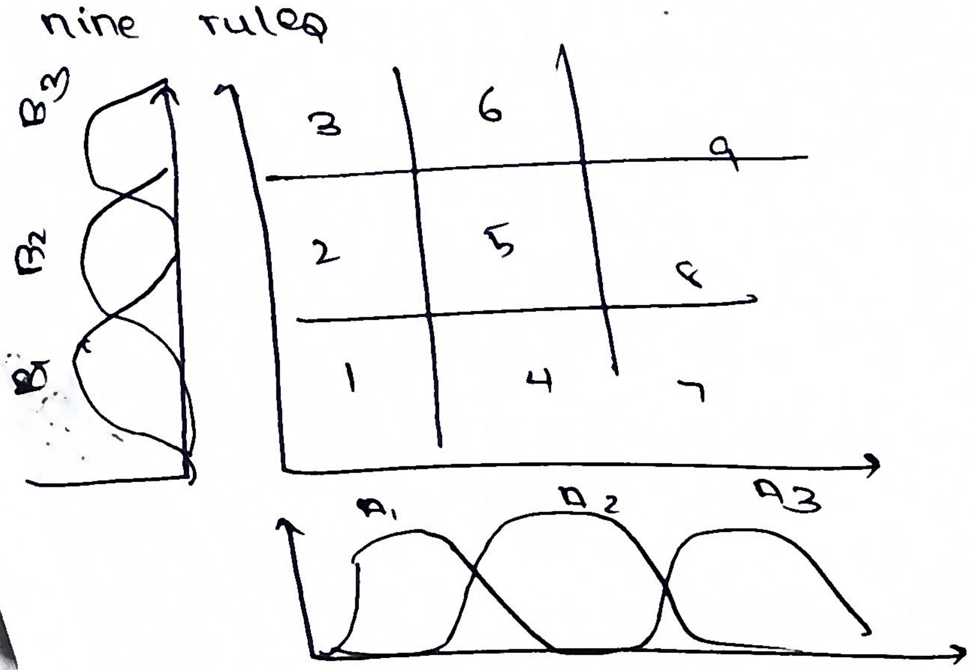
$$\Rightarrow F = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

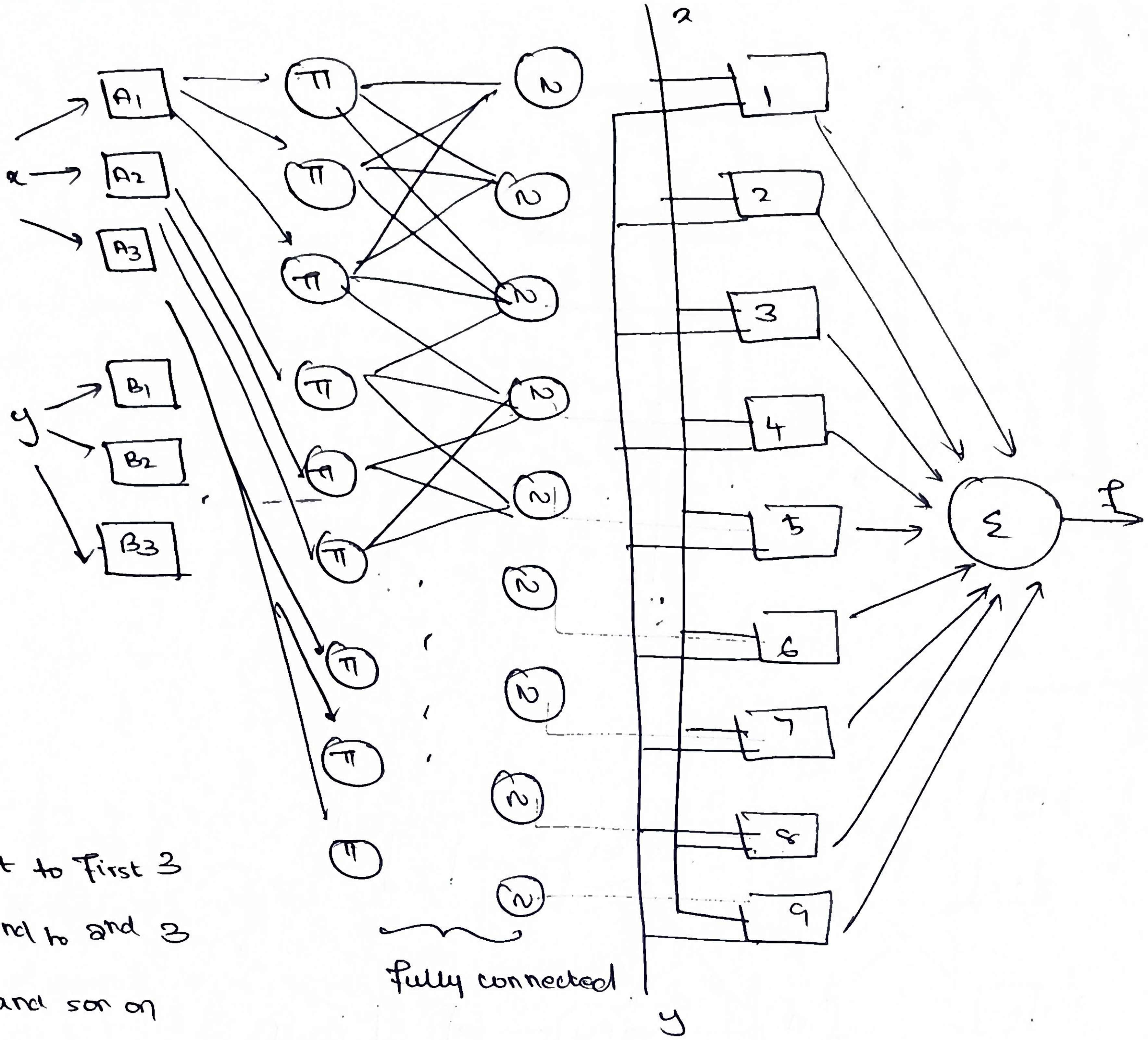
$$= \bar{w}_1 f_1 + \bar{w}_2 f_2$$

Equivalent ANFIS



(4) ANFIS architecture for a 2 input Sugeno Fuzzy model with nine rules





1st to First 3

2nd to 2nd 3

and so on

* Hybrid Learning Algorithm

→ The output of ANFIS can be represented as:

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

$$= \overline{w_1}(p_1 x + q_1 y + r_1) + \overline{w_2}(p_2 x + q_2 y + r_2)$$

$$= (\overline{w_1} x) p_1 + (\overline{w_1} y) q_1 + (\overline{w_1}) r_1 + (\overline{w_2} x) p_2 + (\overline{w_2} y) q_2 + (\overline{w_2}) r_2$$

→ This expression is linear in terms of the consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2.

→ In the hybrid learning algorithm - in the forward pass - node outputs go forward until layer 4 and the consequent parameters are identified by the least squares method.

→ In the backward pass, the error signals propagate backwards and the premise parameters are updated by gradient descent.

	Forward Pass	Backward Pass
Premise parameters	Fixed	gradient descent
consequent parameters	least squares estimator	Fixed
signals	node outputs	error signals

→ The hybrid approach converges much faster, since it reduces the search space dimensions of the original pure back propagation method.

* Learning Methods that cross-fertilize ANFIS and RBFN

→ A radial basis function network is a type of ANN that uses radial basis functions as the activation functions. (unlike the conventionally used hyperbolic tangent fns)

→ Under certain conditions - an RBFN is functionally equivalent to a FIS, as well as ANFIS and CANFIS.

→ The learning schemes used in ANFIS are applicable to RBFNs and vice-versa as well.

→ That is, ANFIS usually has 2 distinct modifiable parts - the antecedent and consequence. These two parts can be adapted by different optimization methods, one of which is the hybrid learning procedure combing GD & LSE - this applicable to RBFN as well.

→ |||ly the analysis and learning algorithms for RBFNs are also applicable to ANFIS & CANFIS.

* Coactive Neuro-Fuzzy Inference Systems

→ CANFIS extends the notion of a single-output system like ANFIS to produce multiple outputs.

→ Multiple outputs can be produced in 2 ways:

(i) MANFIS = multiple ANFIS - no modifiable parameters are shared by the juxtaposed ANFIS models.

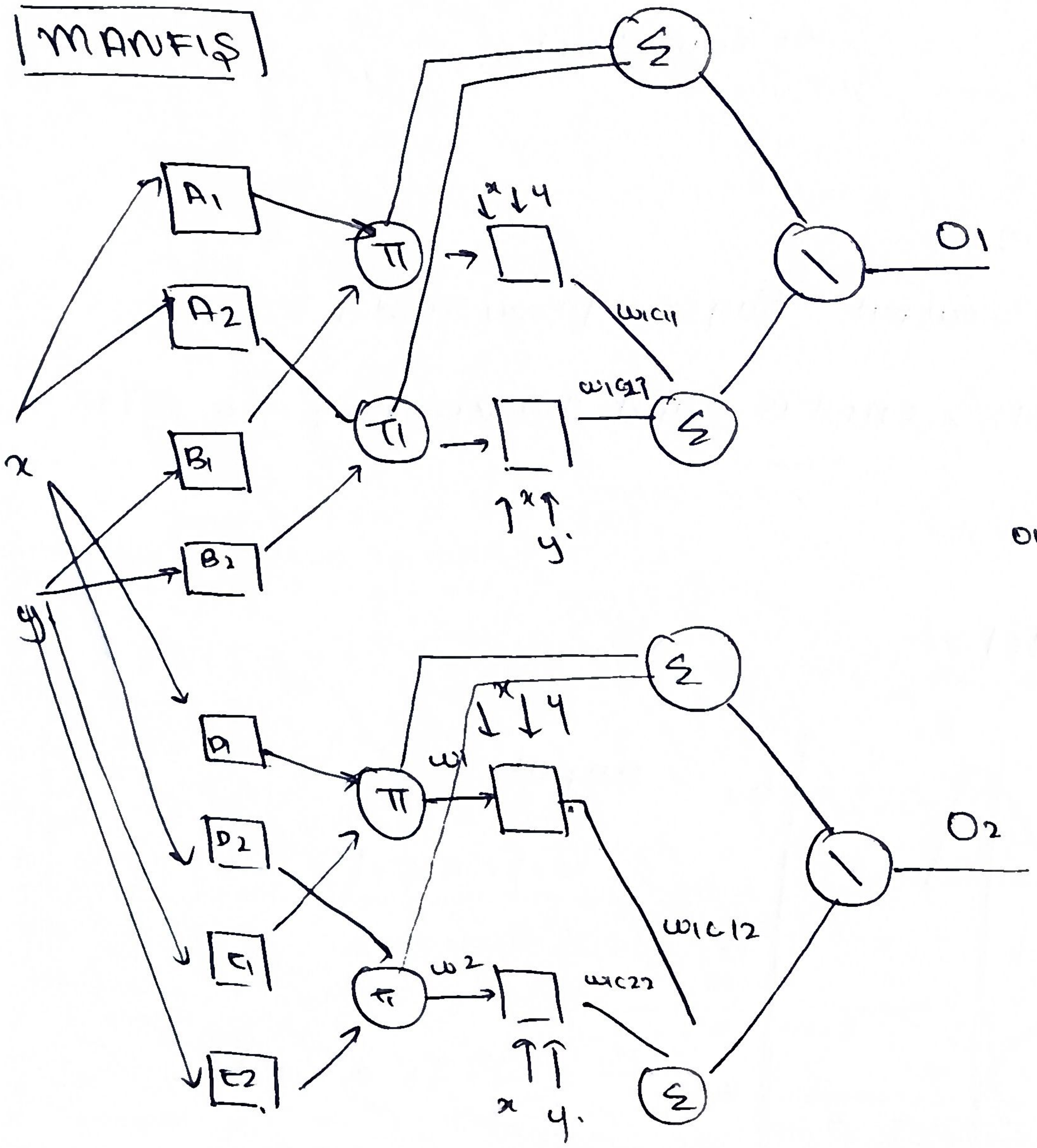
i.e each ANFIS has an independent set of fuzzy rules, which makes it difficult to realize possible correlations between outputs

→ Another concern is in the number of adjustable parameters, which drastically increases as the no. of outputs increases

(ii) CANFIS - generate multiple outputs by maintaining the same antecedents among multiple ANFIS models

→ i.e Fuzzy rules are constructed w/ shared membership values to express correlations between outputs.

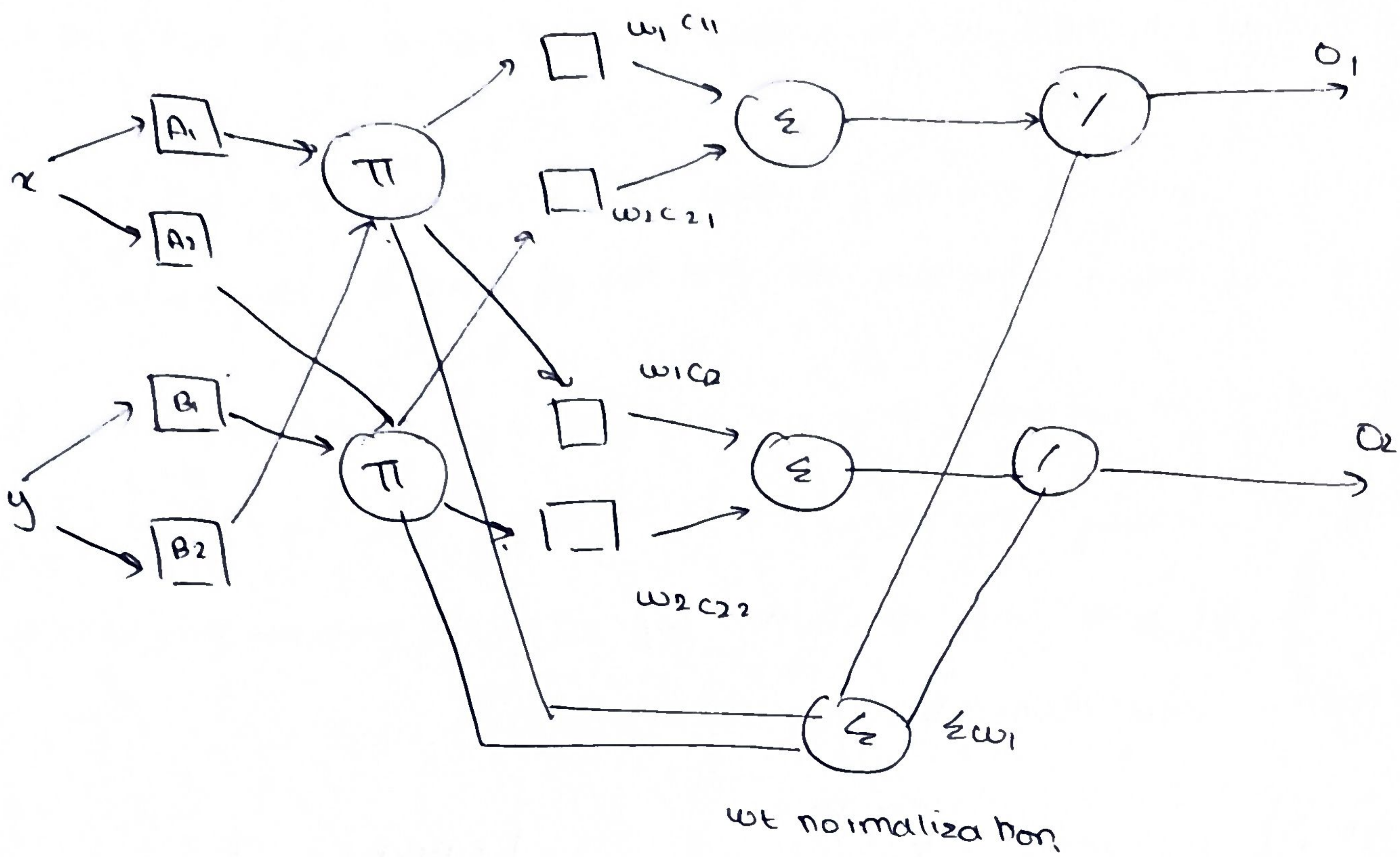
MANFIS



~~CANFIS~~

outputs

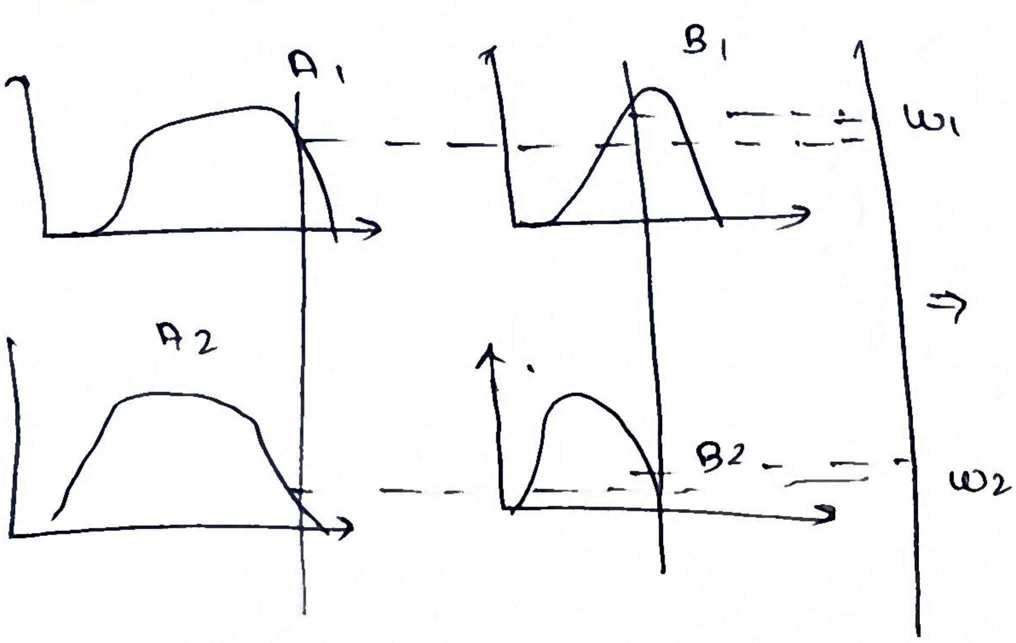
CANFIS



* Architectural Comparisons

between a 2-input, one-output Sugeno fuzzy model, represented in ANFIS, CANFIS and backpropagation MLP models

① Sugeno Fuzzy Model



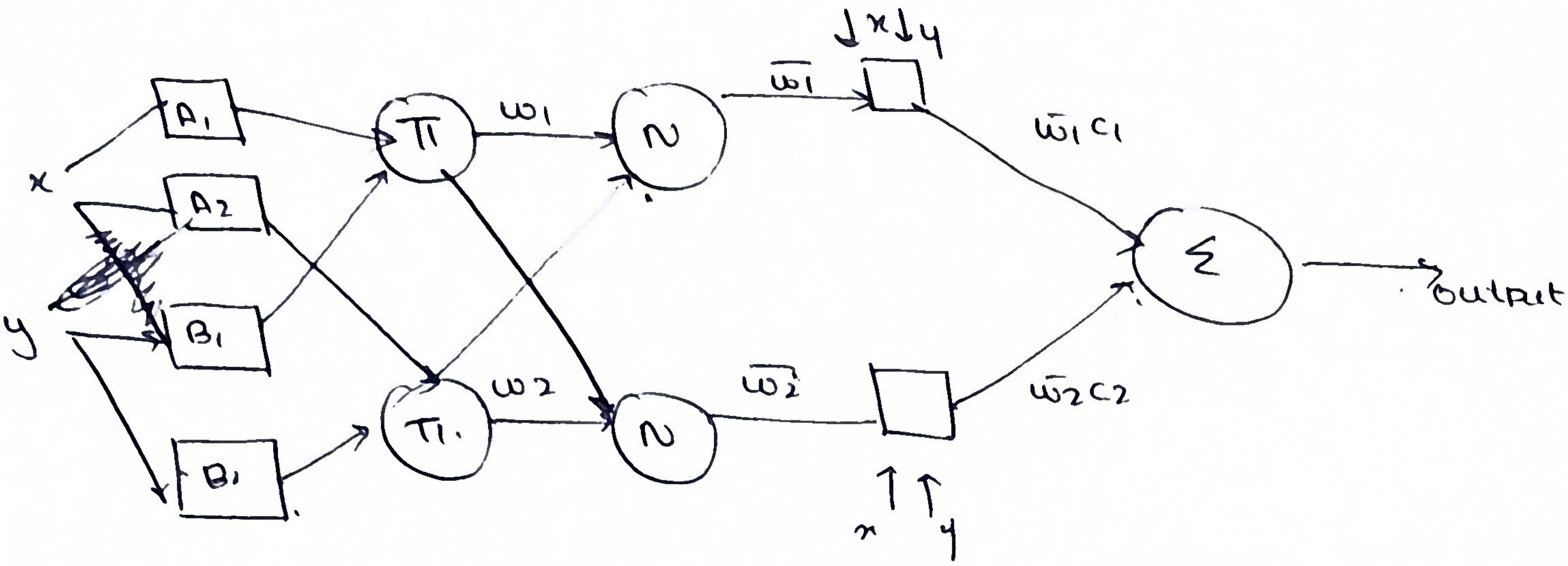
output

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

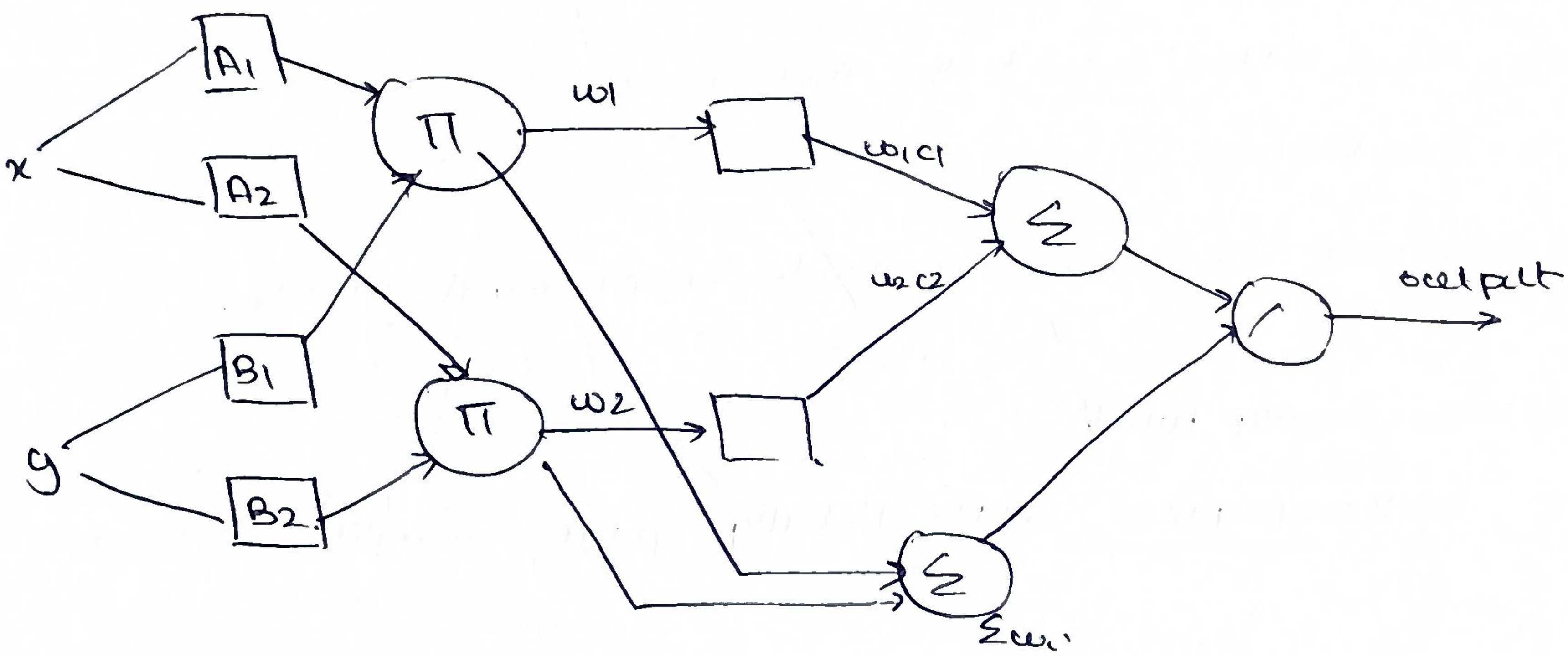
$\Rightarrow$

$$= \bar{w}_1 f_1 + \bar{w}_2 f_2$$

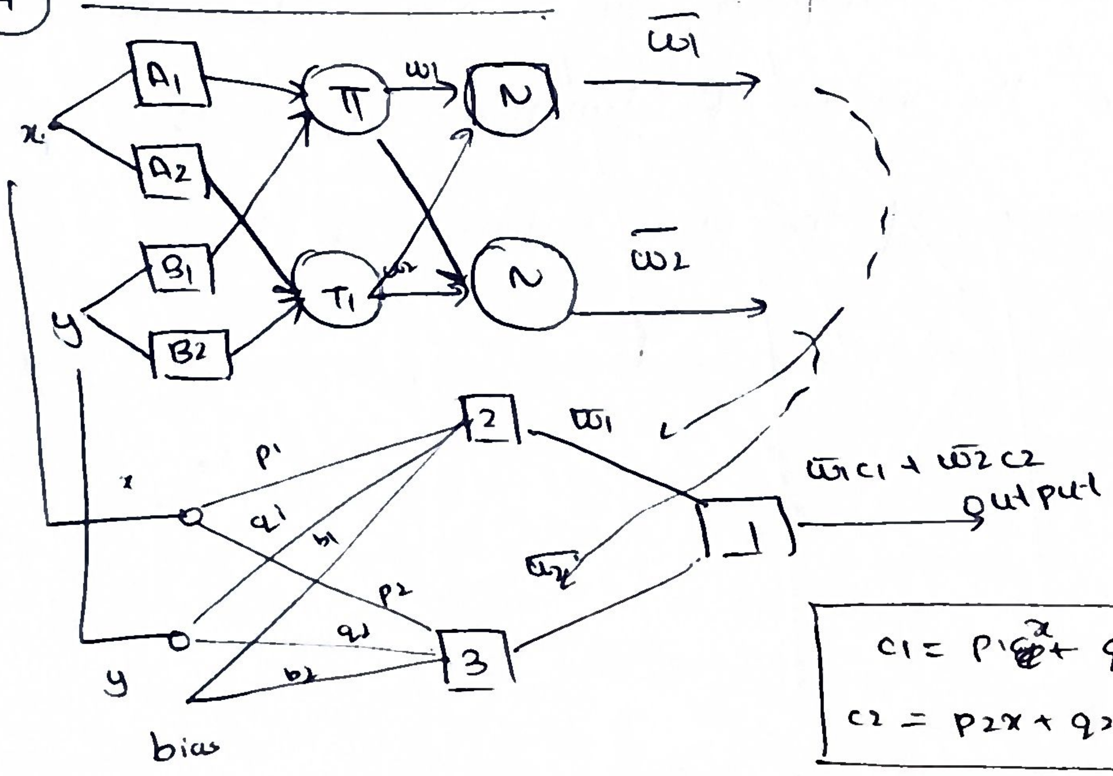
② Equivalent ANFIS model for TSK



③ Equivalent ANFIS, with weight normalization at the last layer.



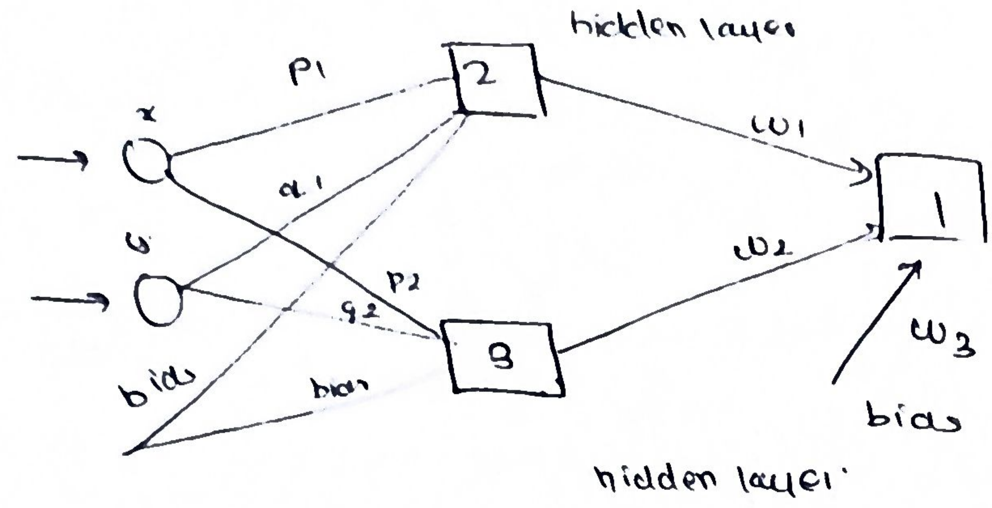
④ CANFIS architecture



$c_1 = p_1 x + q_1 y + r_1$

$c_2 = p_2 x + q_2 y + r_2$

(5) Back propagation MLP



Architectural comparisons

(1) In the black-box MLP, the weights are just numeric connection strengths.

→ Adding more hidden nodes to the MLP is equivalen to adding more rules to CANFIS.

→ This comparison emphasizes the inside transparency of CANFIS.

(2) CANFIS is locally tuned

the back propagation MLP globally updates weight coefficients

(3) back propagation MLP is a better extrapolator due to its global nature, because RBFN fails to estimate the values of functions outside the range of the training data because of the local nature of its hidden & receptive fields (can sometimes be fixed w/ normalization)

* Neuron Functions for Adaptive Networks

→ To make a truly adaptive network, there should be no constraints on neuron functions. (various fns. are used as basis functions as an alternative to Gaussian functions)

2 output CANFIS with 4 rules & a comparable RBFN