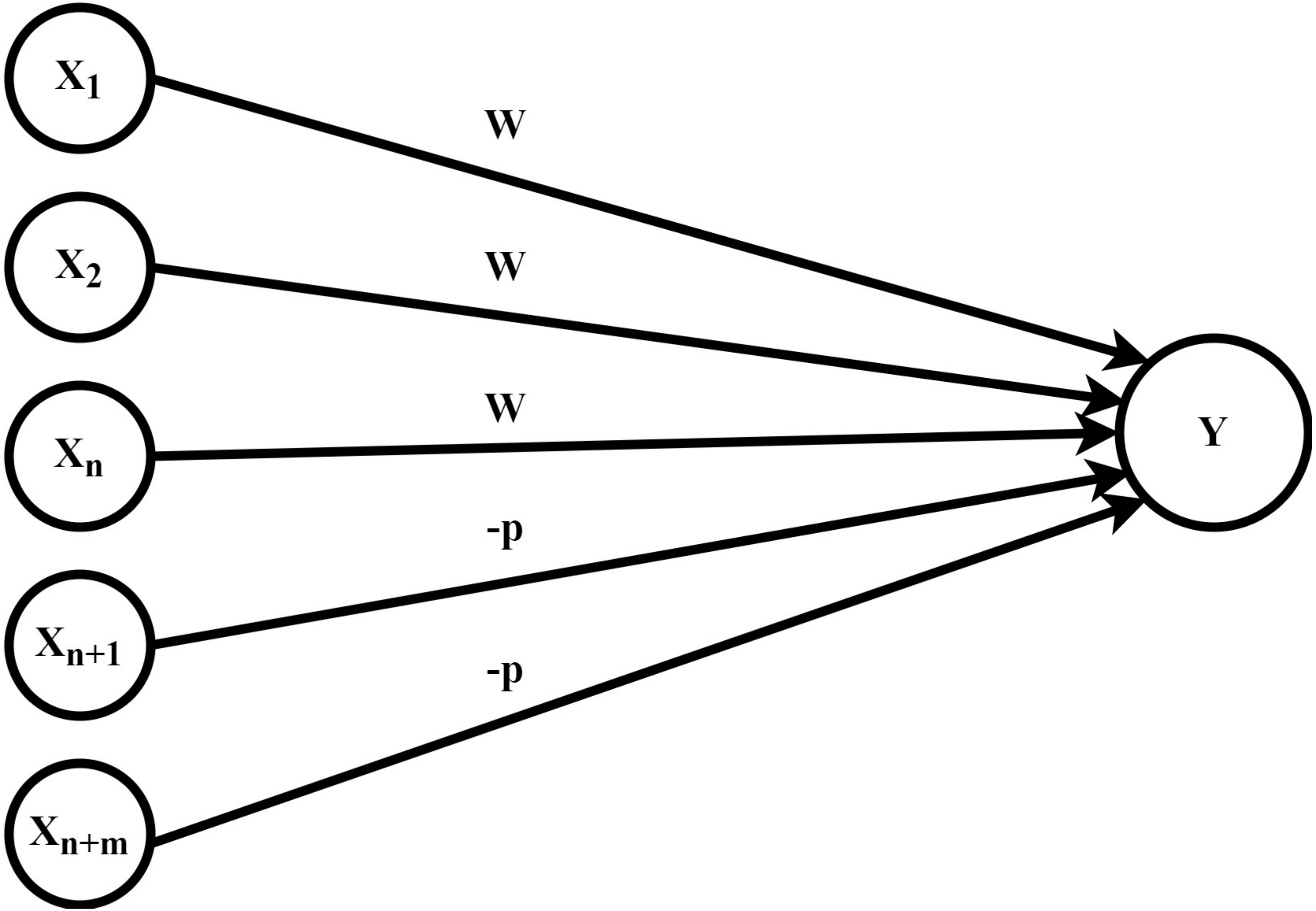
McCulloch-Pitts Neuron Model

- Mc Culloch – Pitts Neuron Model was formulated in the year by Warren Mc Culloch and Walter Pitts in the year 1943.
- It is characterized by its formalism, elegant and precise mathematical definition.
- The neuron allows only the binary states i.e., ‘0’s and ‘1’s, so it is called as a binary activated neuron.
- These neurons are connected by direct weighted path. The connected path can be excitatory or inhibitory.

McCulloch-Pitts Neuron Model



McCulloch-Pitts Neuron Model

- Excitatory connections have positive weights and inhibitory connections have negative weights.
- There will be same weight for the excitatory connections entering into a particular neuron.
- The neuron is associated with the Threshold value. The neuron fires if the net input to the neuron is greater than the Threshold.
- The threshold is set, so that the inhibition is absolute, because the non-zero inhibitory output will prevent the neuron from firing.

McCulloch-Pitts Neuron Model

The Mc Culloch – Pitts Neuron has an activation function

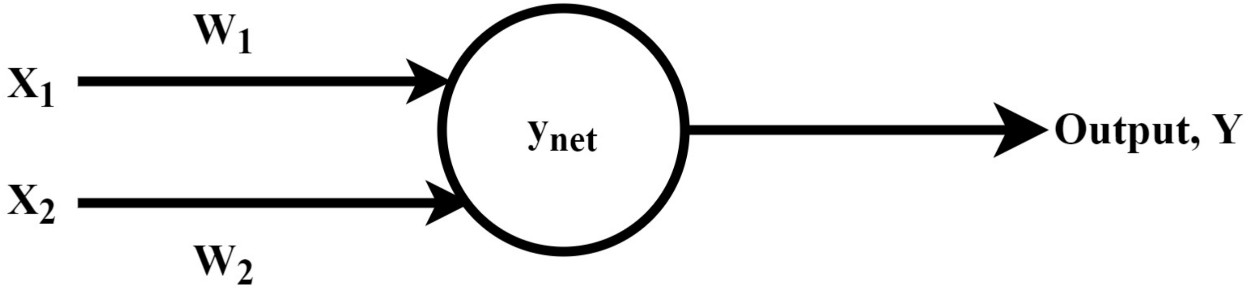
$$f(y_{in}) = 1, \text{ if } y_{net} >= \theta$$

$$= 0, \text{ if } y_{net} < \theta$$

Implementation of AND gate using McCulloch-Pitts Neuron Model

Truth table of AND Gate:

x_1	x_2	Output $Y = x_1 * x_2$
0	0	0
0	1	0
1	0	0
1	1	1



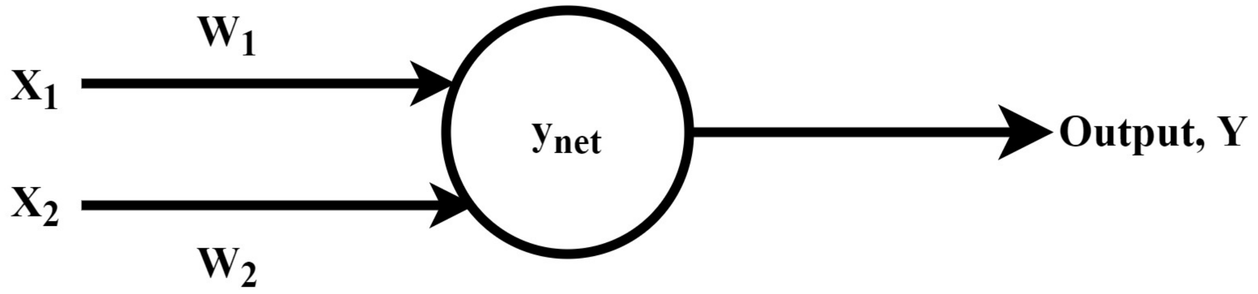
Implementation of AND gate using McCulloch-Pitts Neuron Model

Net input P = $\sum$ (Inputs * Weights) = $[(\mathbf{x_1} * w_1) + (\mathbf{x_2} * w_2)]$

Assuming the weights are excitatory, $w_1 = 1$ and $w_2 = 1$

Net input P= $(\mathbf{x_1} * 1) + (x_2 * 1) = \mathbf{x_1 + x_2}$

Here, the **Threshold θ = 2**



Implementation of AND gate using McCulloch-Pitts Neuron Model

- Output $Y=f(P)= 1, \quad \text{if} \quad P>= \theta$

  $= 0, \quad \text{if} \quad P< \theta$

Case 1: X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0 <2, => Y= 0

Case 2: X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1 <2, => Y= 0

Case 3: X1 = 1, X2 = 0, then P = (1*0)+(0*1) = 1 <2, => Y= 0

Case 4: X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2=2, => Y= 1

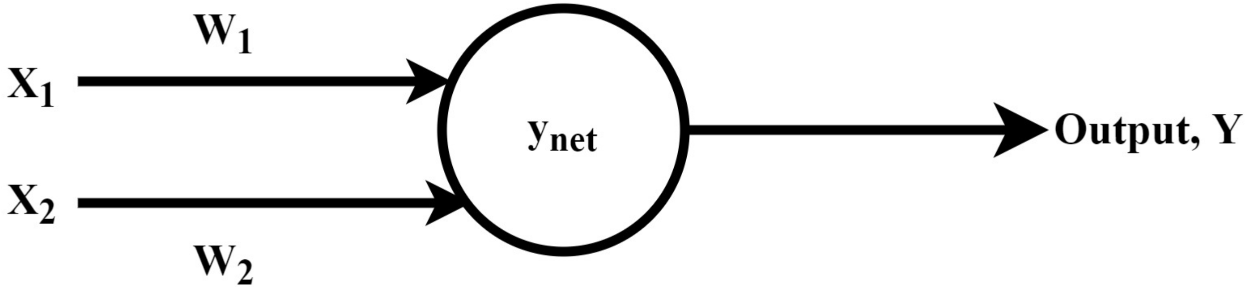
Implementation of AND gate using McCulloch-Pitts Neuron Model

x_1	x_2	Net Input P	Output Y=f(P)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

Implementation of OR gate using McCulloch-Pitts Neuron Model

Truth Table of OR Gate:

INPUT X1	INPUT X2	OUTPUT Y= X1 +X2
0	0	0
0	1	1
1	0	1
1	1	1



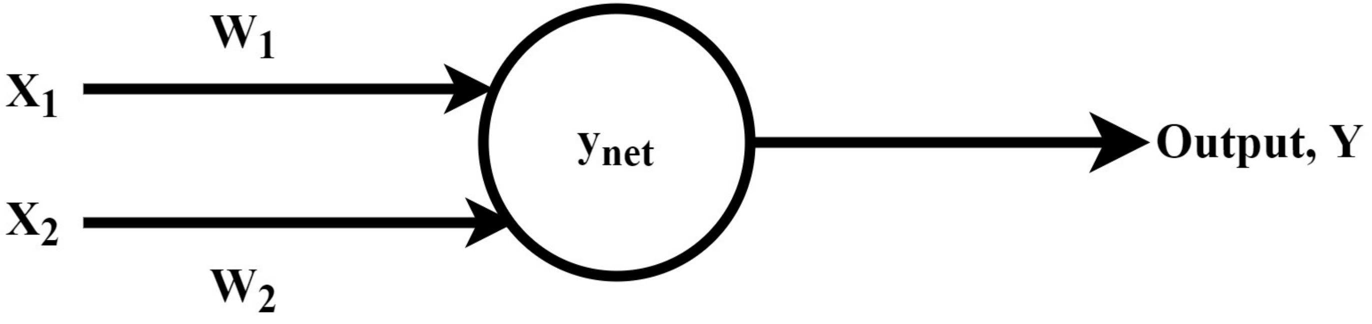
Implementation of AND gate using McCulloch-Pitts Neuron Model

Net input P = $\sum$ (Inputs * Weights) = $[(\mathbf{x_1}*w_1) + (\mathbf{x_2}*w_2)]$

Assuming the weights are excitatory, $w_1 = 1$ and $w_2 = 1$

Net input P= $(\mathbf{x_1}*1) + (x_2*1) = \mathbf{x_1 + x_2}$

Here, the **Threshold θ = 1**



Implementation of OR gate using McCulloch-Pitts Neuron Model

- Output $Y=f(P)= 1, \quad \text{if} \quad P>= \theta$

  $= 0, \quad \text{if} \quad P< \theta$

Case 1: X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0<1, => Y= 0

Case 2: X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1>=1, => Y= 1

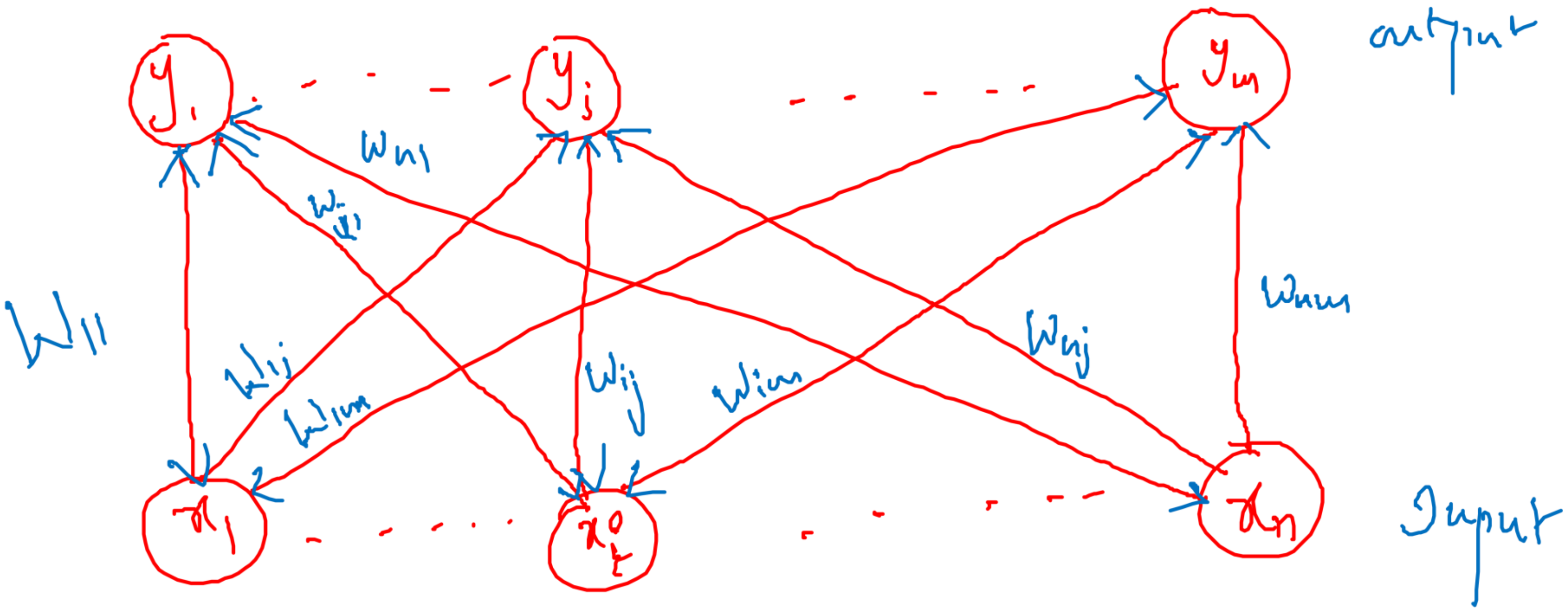
Case 3: X1 = 1, X2 = 0, then P = (1*0)+(0*1) = 1>=1, => Y= 1

Case 4: X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2>1, => Y= 1

Implementation of OR gate using McCulloch-Pitts Neuron Model

x_1	x_2	Net Input P	Output Y=f(P)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	1

Bidirectional Associative Memory :-



① Discrete BAM ② Continuous BAM

For pattern (E), $t = (-1, 1)$

pattern (H), $t = (1, 1)$

(i) Find the weight matrix with input pattern E and H.

Take $* = 1$, $\bullet = -1$

Input,

$$E = [1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ 1]$$

$$H = [1\ -1\ 1\ 1\ -1\ 1\ 1\ 1\ 1\ 1\ -1\ 1\ 1\ -1\ 1]$$

$$W_{ij} = \sum_{p} s^T(p) \cdot t(p)$$

Pattern "E" :-

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}_{15\times 1} \begin{bmatrix} -1 & 1 \end{bmatrix}_{1\times 2} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix}_{15\times 2}$$

Pattern "H" :-

$$W_2 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \end{bmatrix}_{15\times 1} \begin{bmatrix} 1 & 1 \end{bmatrix}_{1\times 2} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix}_{15\times 2}$$

$$W = W_1 + W_2$$

$$= \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix}_{15 \times 2}$$

ii, Test the response of the network

Pattern E :-

$$y_{net} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}}_{1\times 15} \underbrace{\begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix}}_{15\times 2}$$

$$\Rightarrow y_{net} = [-2-2-2-2 \quad 2+2+2+2+2+2+2+2+2+2+2]$$

$$\Rightarrow y_{net} = [-8 \quad 22]$$

$$Y_1 = f(y_{net}) = [-1 \quad 1]$$

As the target for E-pattern is $t_1 = [-1 \quad 1]$,

$$Y_1 = t_1$$

The response is correct.

Pattern H :-

$$y_{net} = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix}_{1\times 15} \begin{bmatrix} 0 & 2 \\ -2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & 2 \\ -2 & 0 \\ 0 & 2 \end{bmatrix}_{15\times 2}$$

$$y_{net} = \begin{bmatrix} 2+2+2+2 & 2+2+2+2+2+2+2+2+2+2+2 \end{bmatrix}$$

$$\Rightarrow y_{net} = \begin{bmatrix} 8 & 22 \end{bmatrix}$$

$$Y_2 = f(y_{net}) = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$t_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\Rightarrow \quad Y_2 = t_2$$

The response is correct.

Classical Sets (or) Crisp Sets :-

Universe of Discourse :-

It is a set with which reference to a particular context, contains all possible elements having the same characteristics and from which sets can be formed. The universal set is denoted by "E".

Eg :- The universal set of all students in a university

Set :-

A set is a well defined collection of objects.

$$\underset{\text{set}}{A} = \{ \underbrace{a_1, a_2, a_3, \dots a_n}_{\text{Members}} \}$$

Eg:- $A = \{ \text{Sachin, Dravid, Ganguly} \}$

$B = \{ \text{cow, buffalo, calf} \}$

A set is also defined based on the properties the members have to satisfy. In such a case, a set "A" is defined as,

$$A = \{ x \mid p(x) \}$$

Where,

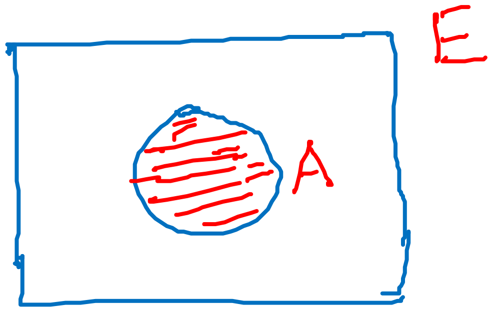
$p(x)$ stands for the property "P" to be satisfied by the member "x".

i.e., A is the set of all x such that $p(x)$ is satisfied.

Eg:- $A = \{x \mid x \text{ is an odd number}\}$

Venn diagram :-

Venn diagrams are pictorial representation to denote a set. Given a set "A" defined over a universal set "E", the venn diagram for A and E is shown in the figure.



Eg:- In the Venn diagram, If "E" represents the set of university students, then "A" may represent the set of female students.

i.e. $E = \{$ students of university $\}$

$A = \{$ female students of university $\}$

Membership :-

An element "x" is said to be a member of a set "A", if "x" belongs to set "A".

The membership is denoted by "$\in$" and is pronounced "belongs to".

$x \in A \longrightarrow x$ belongs to A

$x \notin A \longrightarrow x$ doesnot belongs to A

Eg :- $A = \{1, 2, 3, 4, 6, 8, 9, 10\}$

for $x = 5$ and $y = 8$

We have $x \notin A$

and $y \in A$

Cardinality :-

The number of elements in a set is called its cardinality.

Cardinality of set A is denoted as

$n(A)$ (or) $|A|$ (or) $\#A$

Eg:- $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$n(A) = |A| = \#A = \underline{\underline{10}}.$

Family of Sets :-

A set whose members are sets themselves, is referred to as a family of sets.

Eg:- $A = \{\{1,2,3\}, \{4,1,9\}, \{5,6,4\}\}$.

Null set (or) Empty set :-

A set is said to be a null set (or) empty set, if it has no members.

A null set is indicated as " ϕ " (or) " $\{\ \}$ "

and indicates an impossible event. Also $|\phi| = 0$.

$$A = \{\ \}$$

Eg:- The set of all prime ministers who are below 15 years of age.

Singleton Set :-

A set with a single element is called a "Singleton set".

A Singleton set has Cardinality of 1.

Eg :- $A = \{a\}$ i.e. Number of elements = 1

Cardinality, $|A| = 1$.

Subset :-

Given sets A and B defined over a universal set "E". "A" is said to be a subset of "B" if "A" is fully contained in "B" i.e., every element of "A" is in "B".

It is denoted as "$A \subset B$" and we say that "A" is subset of "B" (or) A is a proper set of "B".

On the other hand, if A is contained in (or) equivalent to that of "B", then we denote the subset relation as "$A \subseteq B$". In such a case, "A" is called the improper subset of B.

Superset :-

Given sets A and B on "E", the universal set, A is said to be a superset of B, if every element of B is contained in A.

It is denoted as "$A \supset B$", we say A is a superset of B (or) A contains B. If A contains B and equivalent to B, then we denote it as "$A \supseteq B$".

Eg:- $A = \{3, 4\}$, $B = \{3, 4, 5\}$, $C = \{4, 5, 3\}$

Here $A \subset B$ and $B \supset A$

$C \subseteq B$ and $B \supseteq C$

Power Set :-

A power set of set A is the set of all possible subsets that are derivable from A including null set.

A powerset is indicated as $P(A)$ and has cardinality of $|P(A)| = 2^{|A|}$.

Classical logic :-

Let P and Q be the propositions belongs to the sets A, B, then the truthness of propositions P, Q can be denoted as $T(P)$ and $T(Q)$ where $T(P) = 1$ and $T(Q) = 1$

$$P : x \in A$$

$$Q : x \in B$$

Logical connectives :-

These are used to connect two simple propositions and to make a simple compound proposition. Some of the logical connectives are given as follows.

① Disjunction $(\vee)$:-

It is denoted by $P \vee Q \Big|_{x \in A \,(\text{or})\, x \in B}$

Truthness,

$$T(P \vee Q) = \text{Max}\left[T(P), T(Q)\right]$$

② Conjunction ($\wedge$) :-

$p \wedge Q$: $x \in A$ and $x \in B$

$T(p \wedge Q) = Min[T(p), T(Q)]$

③ Negation ($\sim$) :-

If, $T(p) = 1$, then $T(\tilde{p}) = 0$

④ <u>Implication ($\longrightarrow$) :-</u>

It is denoted by $P \longrightarrow Q$ | $x \notin A$ (or) $x \in B$

i.e, $P \longrightarrow Q = (\bar{P} \cup Q)$

Truthness, $T(P \to Q) = T(\bar{P} \cup Q)$

In Implication, If the first proposition is true and second proposition is false, then the result is false.

Let P and Q be two propositions. The truth table for Implication is shown.

P	Q	$P \to Q$
T	T	T
T	F	F
F	T	T
F	F	T

Let $T = 1$, $F = 0$

$$T(P \longrightarrow Q) = T(\bar{P} \cup Q) = \mathrm{Max}[T(\bar{P}), T(Q)]$$

For $P = T$, $Q = T$, $T(\bar{P} \cup Q) = \mathrm{Max}[0, 1] = 1 = T$

$P = T$, $Q = F$, $T(\bar{P} \cup Q) = \mathrm{Max}[0, 0] = 0 = F$

$P = F$, $Q = T$, $T(\bar{P} \cup Q) = \mathrm{Max}[1, 1] = 1 = T$

$P = F$, $Q = F$, $T(\bar{P} \cup Q) = \mathrm{Max}[1, 0] = 1 = T$

In Disjunction ($\vee$),

$$T(p \vee Q) = Max[T(p), T(Q)]$$

Let $T = 1$, $F = 0$

For $p = T, Q = T$, $T(p \vee Q) = Max[1, 1] = 1 = T$

$p = T, Q = F$, $T(p \vee Q) = Max[1, 0] = 1 = T$

$p = F, Q = T$, $T(p \vee Q) = Max[0, 1] = 1 = T$

$p = F, Q = F$, $T(p \vee Q) = Max[0, 0] = 0 = F$

In conjunction ($\wedge$),

$$T(p \wedge Q) = \text{Min}[T(p), T(Q)]$$

for $p = T$, $Q = T$, $T(p \wedge Q) = \text{Min}[1, 1] = 1 = T$

$p = T$, $Q = F$, $T(p \wedge Q) = \text{Min}[1, 0] = 0 = F$

$p = F$, $Q = T$, $T(p \wedge Q) = \text{Min}[0, 1] = 0 = F$

$p = F$, $Q = F$, $T(p \wedge Q) = \text{Min}[0, 0] = 0 = F$

5 Equivalance ($\longleftrightarrow$) :-

$$T(p \longleftrightarrow Q) = \begin{cases} 1, & \text{if } T(p) = T(Q) \\ 0, & \text{if } T(p) \neq T(Q) \end{cases}$$

Truth table of all logic Connectives :-

P	Q	P → Q	P ∨ Q	P ∧ Q	P ↔ Q
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	T	F	F
F	F	T	F	F	T

Logical connectives operation in Fuzzy logic :-

1. Negation :-

$$T(\tilde{P}^c) = 1 - T(\tilde{P})$$

② Disjunction ($\vee$) :-

$$T(\tilde{p} \vee \tilde{Q}) = \mathrm{Max}\left[T(\tilde{p}), T(\tilde{Q})\right]$$

③ Conjunction ($\wedge$) :-

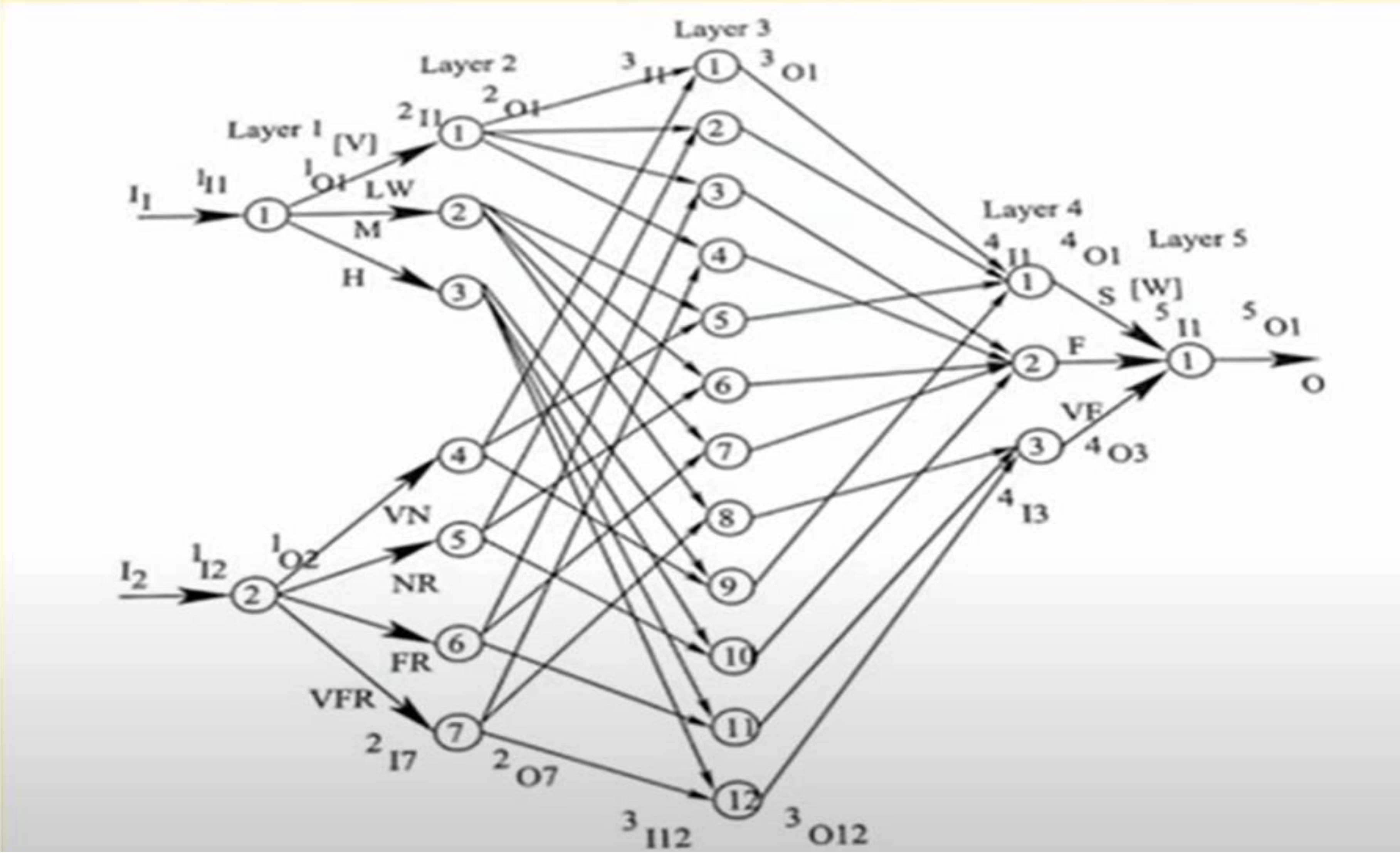
$$T(\tilde{p} \wedge \hat{Q}) = \mathrm{Min}\left[T(\tilde{p}), T(\tilde{Q})\right]$$

4 <u>Implication $(\rightarrow)$:-</u>

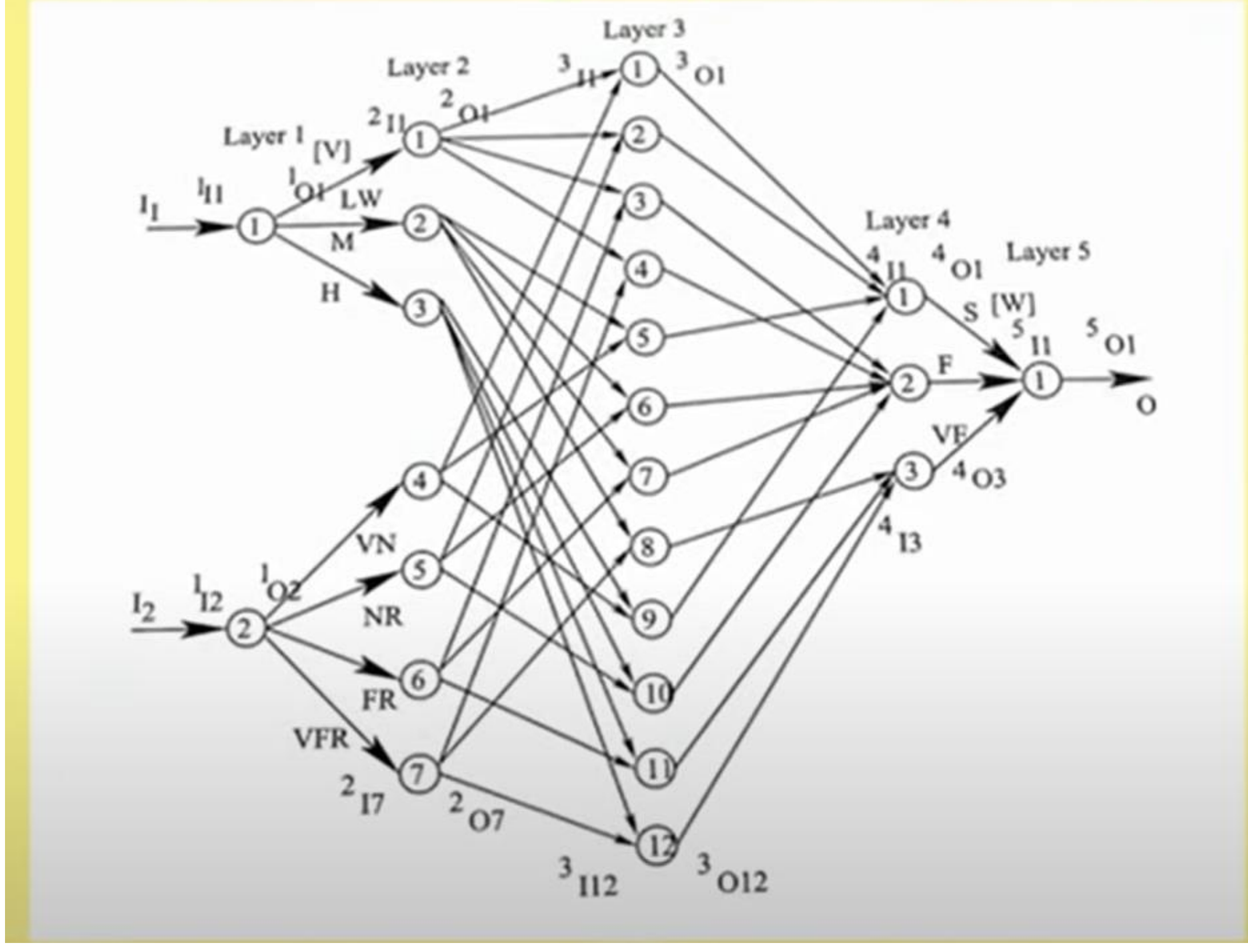
$$T(\tilde{P} \rightarrow \tilde{Q}) = T(\tilde{P}^c \cup Q)$$

$$= Max\left[T(\tilde{P}^c), T(\tilde{Q})\right]$$

Neuro-Fuzzy System (NFS) based on Mamdani Approach



Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
[V]
LW
M
H
VN
NR
FR
VFR
S
F
VF
[W]
O



Layer 1: Input Layer
Layer 2: Fuzzification
Layer 3: AND operation
Layer 4: Fuzzy Inference
Layer 5: Defuzzification

Neuro Fuzzy System based on Mamdani Approach

- The main purpose of developing this neuro-fuzzy system is to design or evolve rather one very efficient fuzzy reasoning tool, which can perform the input output modeling.
- This shows actually the schematic view of one neuro-fuzzy system, very simple neuro-fuzzy system having only 2 inputs and 1 output, and this particular network consists of 5 layers.
- The first layer is the input layer, the second layer is actually the fuzzification layer, third layer is known as the AND operation layer, then comes the fourth layer is nothing, but fuzzy inference and the fifth layer is nothing, but is your defuzzification.
- Now, look-wise this is similar to a particular network, but truly speaking, this is nothing, but the Mamdani approach of fuzzy reasoning tool.

Layer 1 : Input Layer

- we have got two inputs like **I_1 and I_2.**
- So, on the first layer, **I_I1**, that is the input of the first neuron lying on the input layer and **I_I2,** that is the input of the second neuron lying on the first layer and here, I have got actually **1_O1** that is the output of the first neuron lying on the first layer and then **1_O2,** that is the output of the second neuron lying on the first layer.
- So, the first layer is nothing, but the input layer.

Layer 2 : Fuzzification Layer

- The purpose of fuzzification is to find out the normalized value corresponding to these particular inputs.
- Once you have got the fuzzified output and so, here you will be getting the outputs of layer 2 is nothing, but the values of the memberships.

Layer 3 : AND Operation

- Now these membership values for the two inputs: I_1 and I_2, I will have to use here as input of the layer 3. And, on **layer 3**, actually we will have to carry out the **AND operation**
- We try to **compare** the two **membership values** and we try to find out the **minimum**.
- Now, **this minimum** of these two values will be **considered** as **output** of **layer 3** or the third layer.

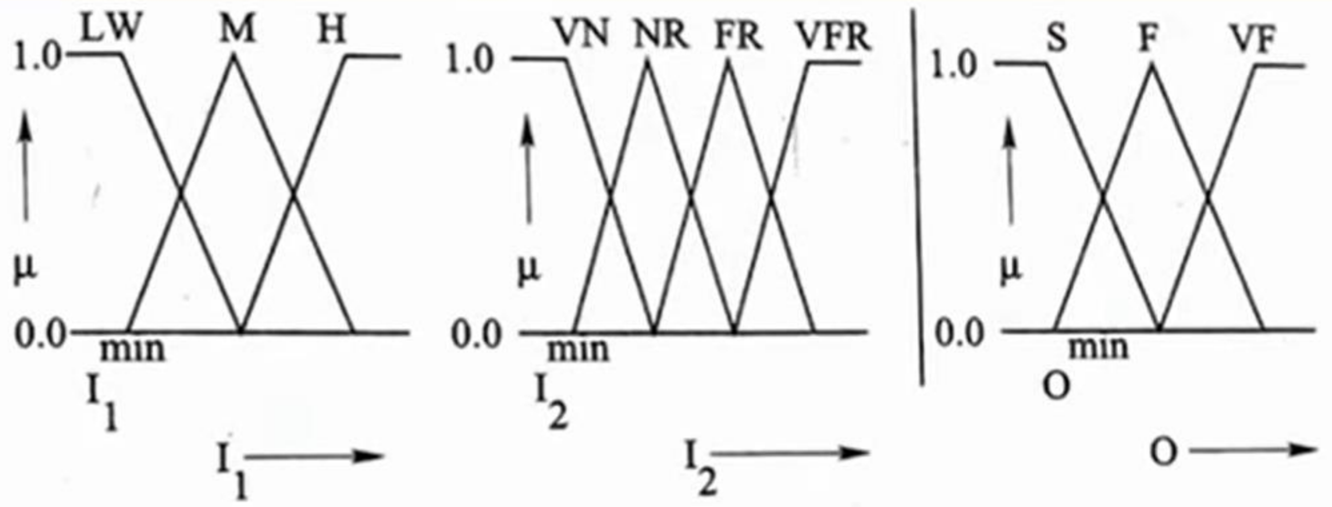
Layer 4 : Fuzzy Inference

- The purpose of fuzzy inference is to select the set of fired rules corresponding to one set of input parameters.
- Now, in the rule base, we have got a large number of rules. So, out of these large number of rules, only a few will be fired depending on the set of inputs.
- Now, fuzzy inference is going to decide, which rules are going to be fired out of the maximum number of rules present in that particular the rule base.

Layer 4 : Fuzzy Inference

- So, these outputs are denoted by actually 4_O1 that is nothing, but the output of the first neuron lying on the fourth layer, and so on.
- Here, those things will enter as input to the **fifth layer** or the **defuzzification layer** and as output of this defuzzification layer.
- So, here we will be getting some crisp output.

Membership Function Distributions

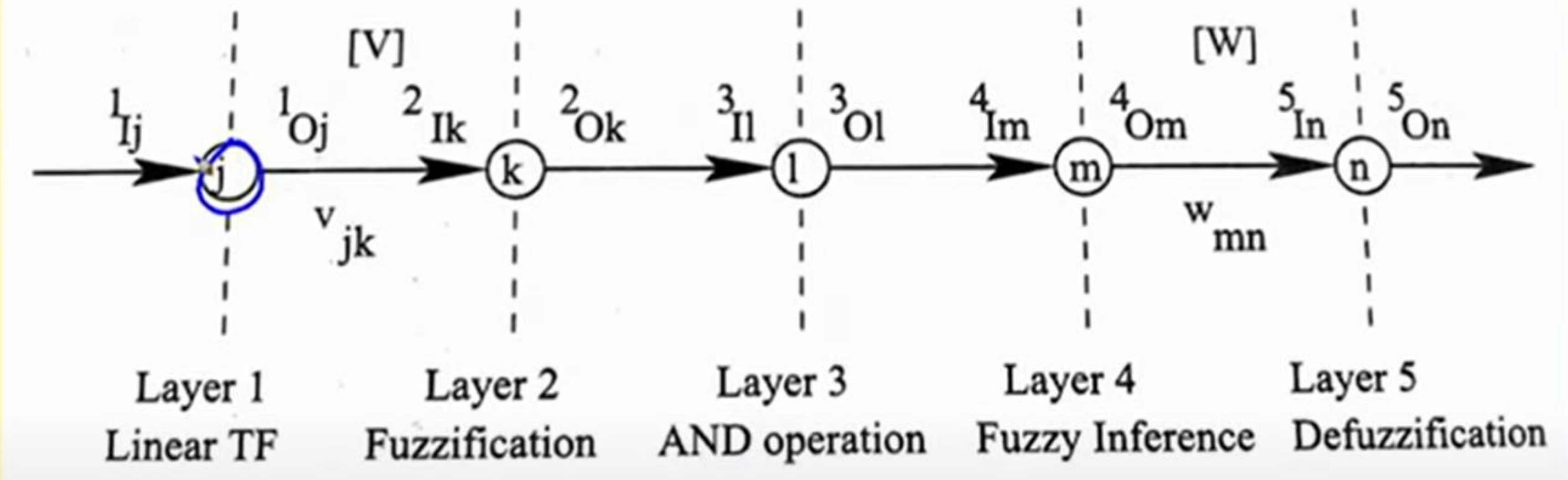


LW: Low
M: Medium
H: High
VN: Very Near
NR: Near
FR: Far
VFR: Very Far
S: Slow
F: Fast
VF: Very Fast

Rule Base

I_1 \ I_2	VN	NR	FR	VFR
LW	S	S	F	F
M	S	F	F	VF
H	S	F	VF	VF

A Specific Neuron at Each Layer of the NN



Assumptions:

$$v_{1,av} = \frac{v_{11} + v_{12} + v_{13}}{3}; \quad Say, v_{11} = v_{12} = v_{13} = v_{1,av}$$

$$v_{2,av} = \frac{v_{24} + v_{25} + v_{26} + v_{27}}{4}; Say, v_{24} = v_{25} = v_{26} = v_{27} = v_{2,av}$$

$$w_{av} = \frac{w_{11} + w_{21} + w_{31}}{3}; Say, w_{11} = w_{21} = w_{31} = w_{av}$$

Layer 3: Performs logical AND operations

Layer 4: Carries out the task of fuzzy inference

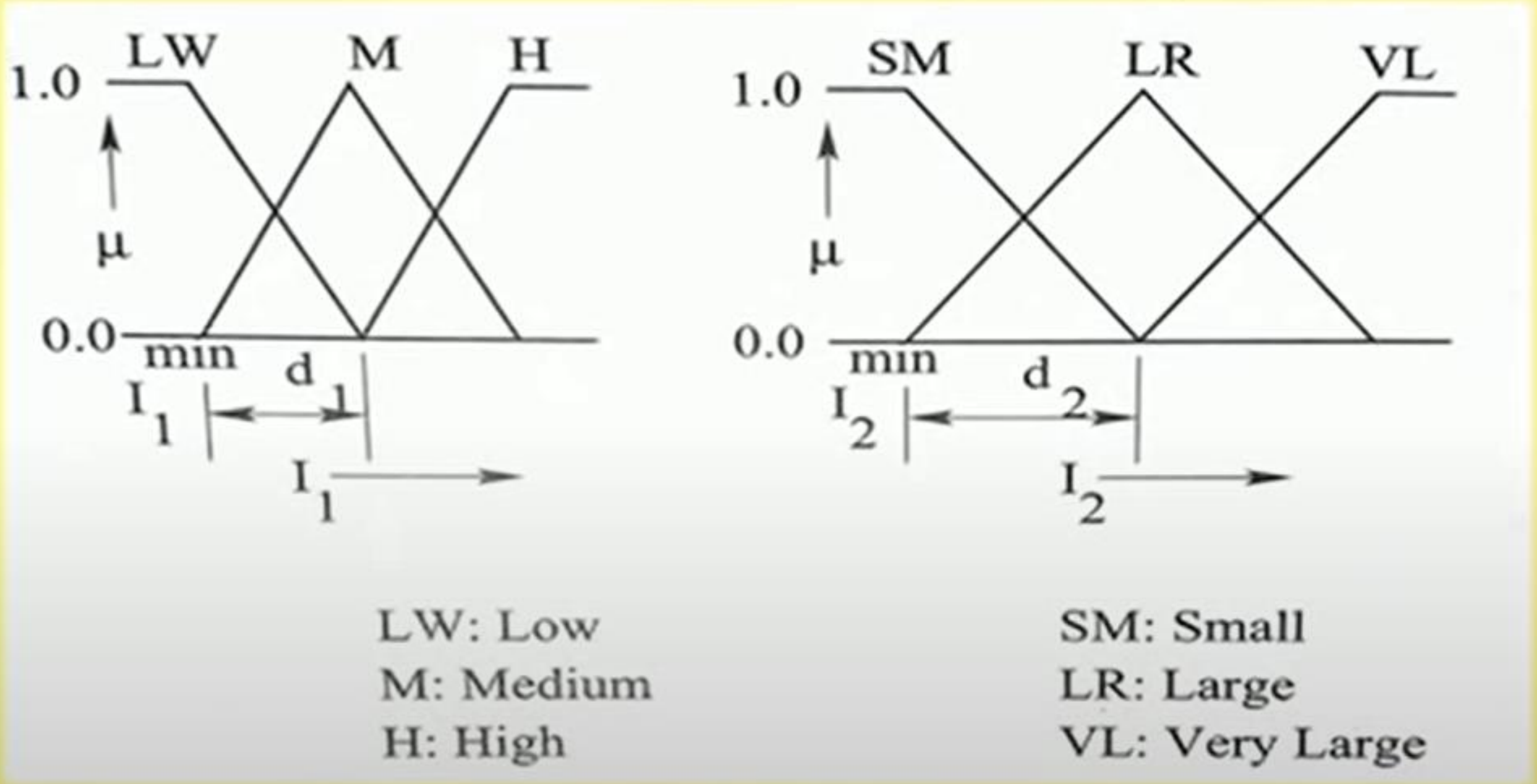
Layer 5: Defuzzification

Center of Sums Method

$$\text{Crisp Output} = \frac{\sum_{i=1}^{r} A_i f_i}{\sum_{i=1}^{r} A_i}$$

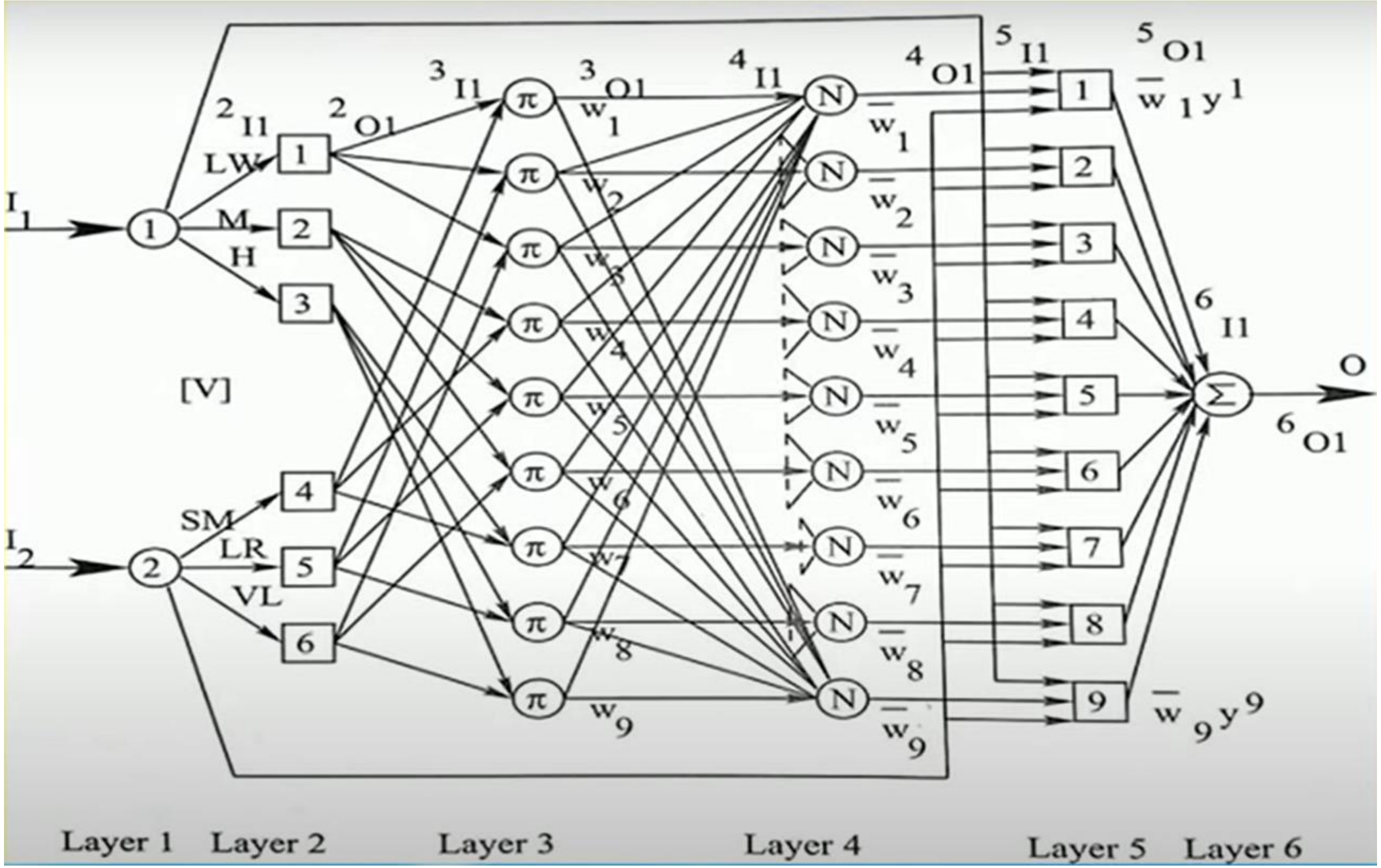
Neuro-Fuzzy System Based On Takagi and Sugeno's Approach

- It is also known as Adaptive Neuro-Fuzzy Inference System (ANFIS)
- Proposed by Jang, 1993
- Let us consider a process with two inputs: I_1 and I_2 and one output: O



Membership function distributions of input variables

ANFIS ARCHITECTURE



ANFIS ARCHITECTURE

Let us assume that the following 4 rules are fired, corresponding to a set of inputs: I_1^* and I_2^*

If I_1 is LW and I_2 is SM the $y_1 = a_1I_1+b_1I_2+c_1$
If I_1 is LW and I_2 is LR then $y_2 = a_2I_1+b_2I_2+c_2$
If I_1 is M and I_2 is SM then $y_4 = a_4I_1+b_4I_2+c_4$
If I_1 is M and I_2 is LR then $y_5 = a_5I_1+b_5I_2+c_5$

ANFIS LAYERS

- **Layer 1:** Linear TF (output = input)

$$1_{O1} = 1_{I1} = I_1^*$$
$$1_{O2} = 1_{I2} = I_2^*$$

- **Layer 2:** Fuzzification
- **Layer 3:** Firing strengths of the rules are calculated considering the products of μ values

$$w_1 = \mu_{LW}(I_1^*) \text{ X } \mu_{SM}(I_2^*)$$

Similarly, we get w_2, w_4 and w_5

- **Layer 4:** Calculate normalized firing strength of each rule

$$\overline{w_1} = \frac{w_1}{w_1 + w_2 + w_4 + w_5}$$

Similarly, we get $\overline{w_2}, \overline{w_4}, \overline{w_5}$

ANFIS LAYERS

- **Layer 5:** Output is calculated as the product of normalized firing strength and output of the corresponding fired rule y

$$5_{01} = \bar{w}_1 \times y^1,$$

$$5_{02} = \bar{w}_2 \times y^2,$$

$$5_{04} = \bar{w}_4 \times y^4,$$

$$5_{05} = \bar{w}_5 \times y^5$$

- **Layer 6:** Overall output $6_{O1} = \overline{w_1}\, y^1 + \overline{w_2}\, y^2 + \overline{w_4}\, y^4 + \overline{w_5}\, y^5$