Soft Computing

Dr.Priya S

Two Techniques

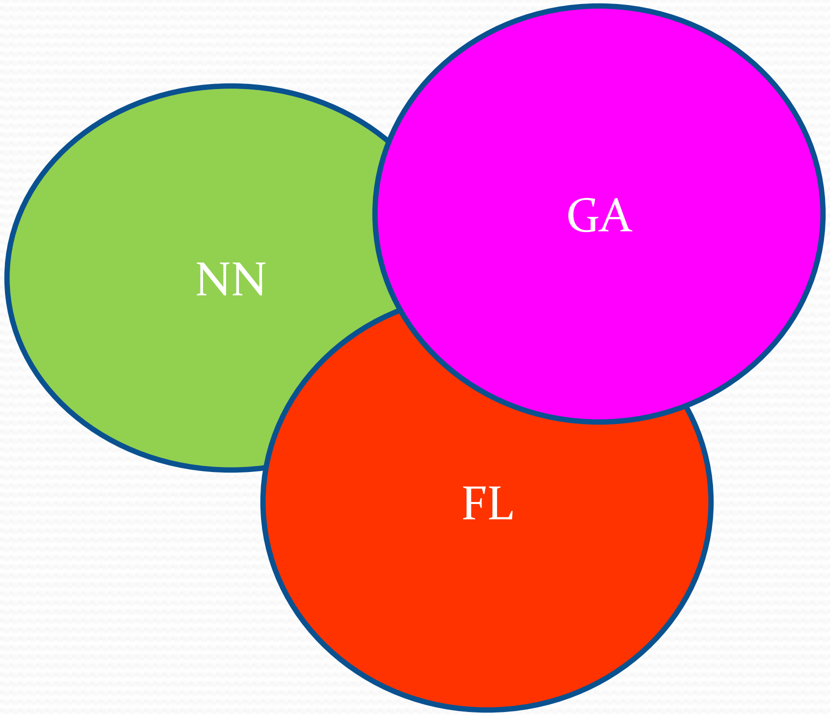
- 1 Hard Computing: deals with precise models where accurate solutions are achieved quickly.
- 2 Soft Computing: deals with approximate models and gives solutions to complex problems

What is soft computing?

- Soft computing is a term used in computer science to refer to problems, whose solutions are unpredictable, uncertain and between 0 and 1.
- Soft computing is used in situations where problems are unsolvable or too time consuming to solve with current hardware.
- Soft computing is the use of approximate calculations to provide imprecise solutions to complex computational real-life problems.

- The role model for soft computing is human mind.
- The term soft computing was introduced in 1994 Prof. Lotfi. A. Zadeh of University of California
- Soft computing is based on techniques such as artificial neural networks, fuzzy logic, genetic algorithm, machine learning and expert systems

Components of Soft Computing



Why Soft Computing ?

The aim of Soft Computing is to exploit tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve close resemblance with human-like decision making.

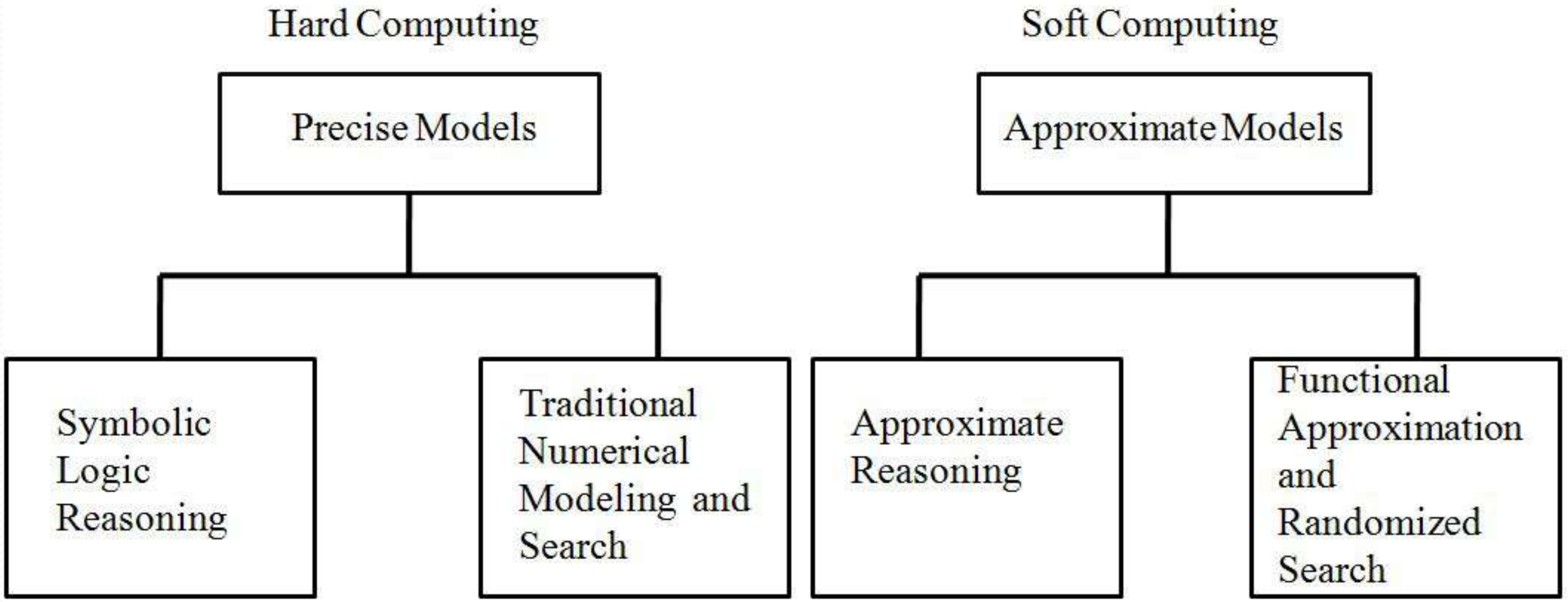
Soft Computing is a new multidisciplinary field, to construct a new generation of Artificial Intelligence, known as *Computational Intelligence*.

The main goal of Soft Computing is to develop intelligent machines and to solve nonlinear and mathematically unmodelled system problems

- Approximation: here the model features are similar to the real ones, but not the same.
- Uncertainity: here we are not sure that the features of the model are the same as that of the real ones.
- Imprecision: here the model features (quantities) are not the same as that of the real ones, but close to them.

Hard Computing vs Soft Computing

- 1 Hard Computing: deals with precise models where accurate solutions are achieved quickly.
- Our conventional algorithms are termed as hard computing.(greedy algorithm,dynamic programming etc)
- 2 Soft Computing: deals with approximate models and gives solutions to complex problems.



Hard Computing

Precise Models

Symbolic Logic Reasoning

Traditional Numerical Modeling and Search

Soft Computing

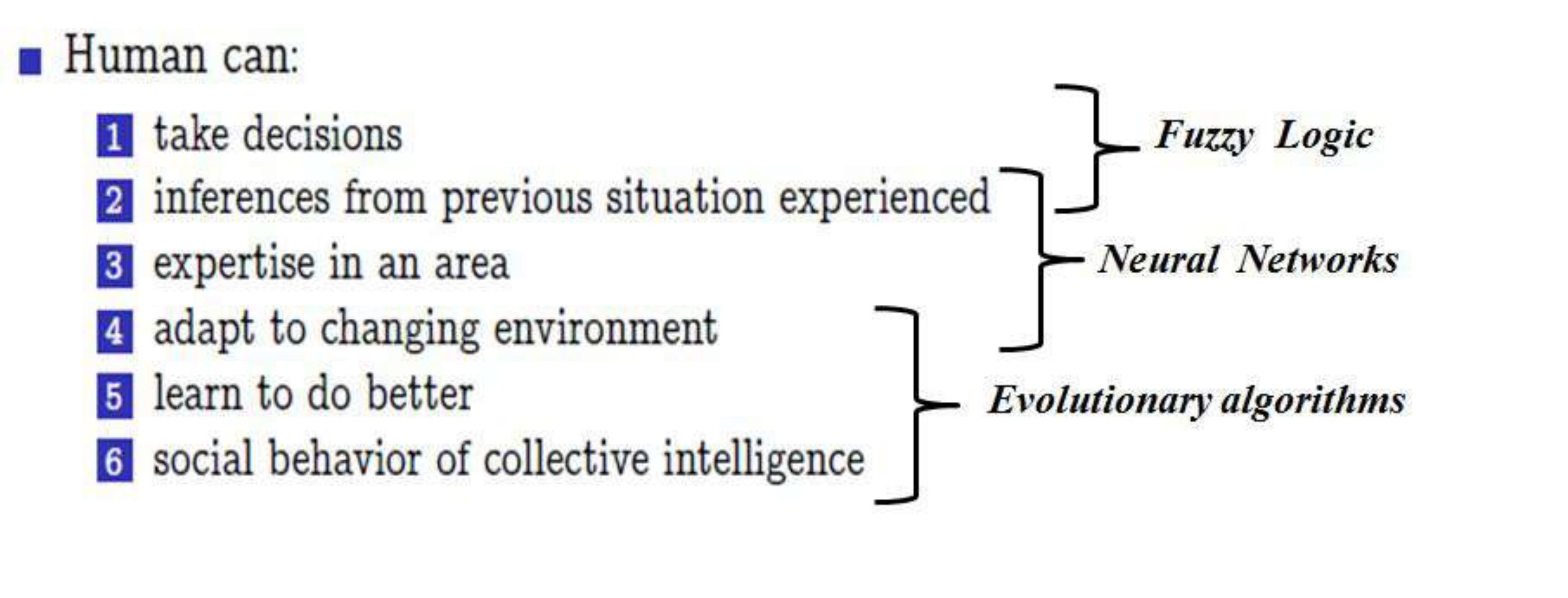
Approximate Models

Approximate Reasoning

Functional Approximation and Randomized Search

Hard Computing	Soft Computing
Works well for simple problems	Well suited for real-life problems
Requires precisely stated analytical model	Tolerant to imprecision, approximation, partial truth and uncertainty
Needs full truth	Works with partial truth
Precision / Accuracy	Approximation
Uses two valued logic	Can use multi valued logic
Requires more computation time and is serial	Requires less computation time and is often parallel
Deterministic	Stochastic
Requires exact input data	Works with ambiguous/ noisy data
Produces precise answers	Can yield approximate answers

- Human beings are capable of:
- 1 Taking decisions
- 2 Taking inferences from previous situation experienced
- 3 Achieving expertise in an area
- 4 Adapting to changing environment
- 5 Learning to do better
- 6 Showing social behavior / collective intelligence



- Human can:
  1. take decisions
  2. inferences from previous situation experienced
  3. expertise in an area
  4. adapt to changing environment
  5. learn to do better
  6. social behavior of collective intelligence

Fuzzy Logic

Neural Networks

Evolutionary algorithms

Advantages of Soft Computing

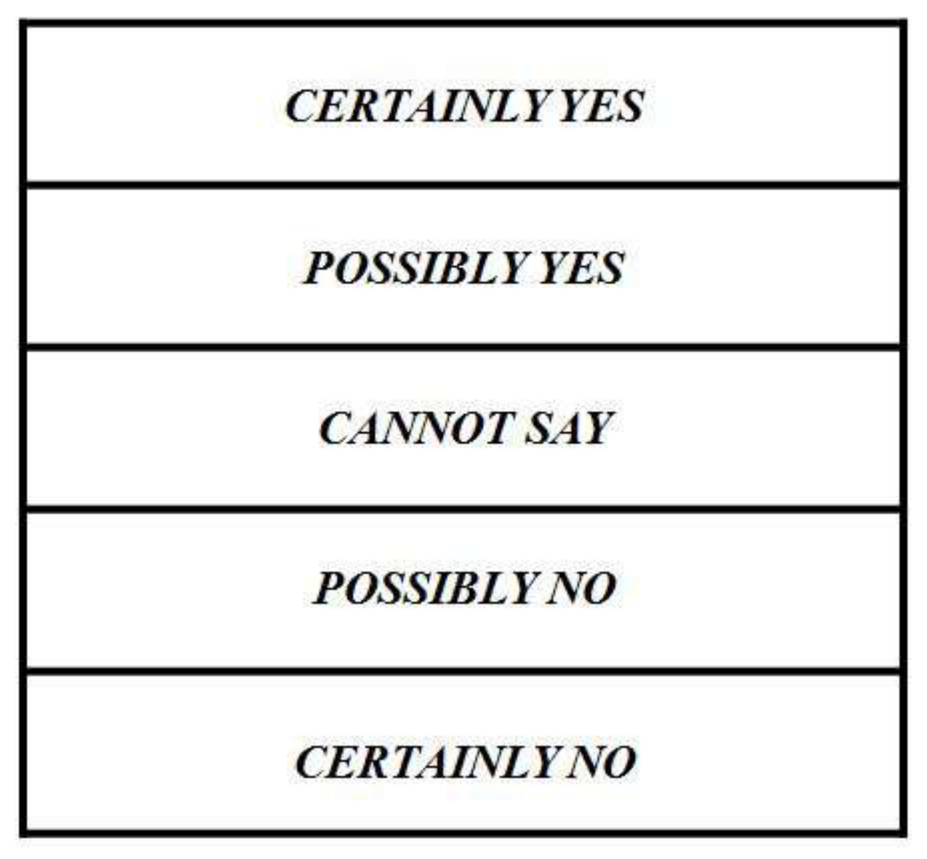
- *First*, it made solving nonlinear problems, in which mathematical models are not available/ possible.
- *Second*, it introduced the human knowledge such as cognition, recognition, understanding, learning, and others into the fields of computing.
- This resulted in the possibility of constructing intelligent systems.

Soft Computing Techniques

- FUZZY LOGIC
- ARTIFICIAL NEURAL NETWORK
- GENETIC ALGORITHMS

Fuzzy Logic

- A method of reasoning that resembles human reasoning.
- FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO(0 and 1).
- The human decision making includes a range of possibilities between YES and NO such as:



CERTAINLY YES
POSSIBLY YES
CANNOT SAY
POSSIBLY NO
CERTAINLY NO

Artificial Neural Network

- It is an efficient information processing system which resembles in characteristics with a biological neural network.
- It is implemented to model the human brain.
- ANN is a collection of highy interconnected processing elements called nodes or units or neurons
- ANN can perform various tasks such as pattern-matching and classification,optimization function,approximation,vector quantization and data clustering.

ANN

- NNs are constructed and implemented to model the human brain.
- The main objective is to develop computational device for modeling the brain to perform various tasks.
- ANNs are implemented using high speed digital computers which makes the simulation of neural processes feasible.

- ANNs possess large number of highly interconnected processing elements called nodes or units or neurons.
- Operates in parallel.
- ANNs collective behavior is characterized by their,
- 1 Ability to learn
- 2 Recall
- 3 Generalize training patterns or data

Genetic Algorithms

- Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics.
- GAs are a subset of a much larger branch of computation known as **Evolutionary Computation.**
- GAs were developed by John Holland and his students and colleagues at the University of Michigan.
- Genetic Algorithm (GA) is a search-based optimization technique based on the principles of **Genetics and Natural Selection**.

Soft Computing- Applications

- Handwritten Script Recognition.
- Image Processing and Data Compression.
- Automotive Systems and Manufacturing.
- Soft computing based Architecture.
- Decision Support Systems.
- Power System Analysis.
- Bioinformatics.
- Investment and Trading.

Artificial Neural Networks

- NNs are constructed and implemented to model the human brain.
- The main objective is to develop computational device for modeling the brain to perform various tasks.
- ANNs are implemented using high speed digital computers which makes the simulation of neural processes feasible.

Artificial Neural Networks

- **Artificial Neural Network**– is a biologically inspired network of artificial neurons configured to perform specific tasks.
- Neural network, in general, is a highly interconnected network of billions of neurons with trillions of interconnections between them.

Artificial Neural Networks

- A digital computer can do everything that an artificial neural network can do. Why ANN?
- It allows to use very simple computational operations (additions, multiplication and fundamental logic elements) to solve complex, mathematically ill-defined problems, nonlinear problems or stochastic problems.
- A conventional algorithm uses complex sets of equations, and can be applied only to the given problem .
- ANN is highly parallel
- ANN is computationally and algorithmically very simple
- It has self-organizing feature to allow it to hold for a wide range of problems.

- A *serial computer* has a central processor. In this system, computational steps are deterministic, sequential.
- *Neural networks* are not sequential or necessarily deterministic. There are no central processor

Advantages of neural network

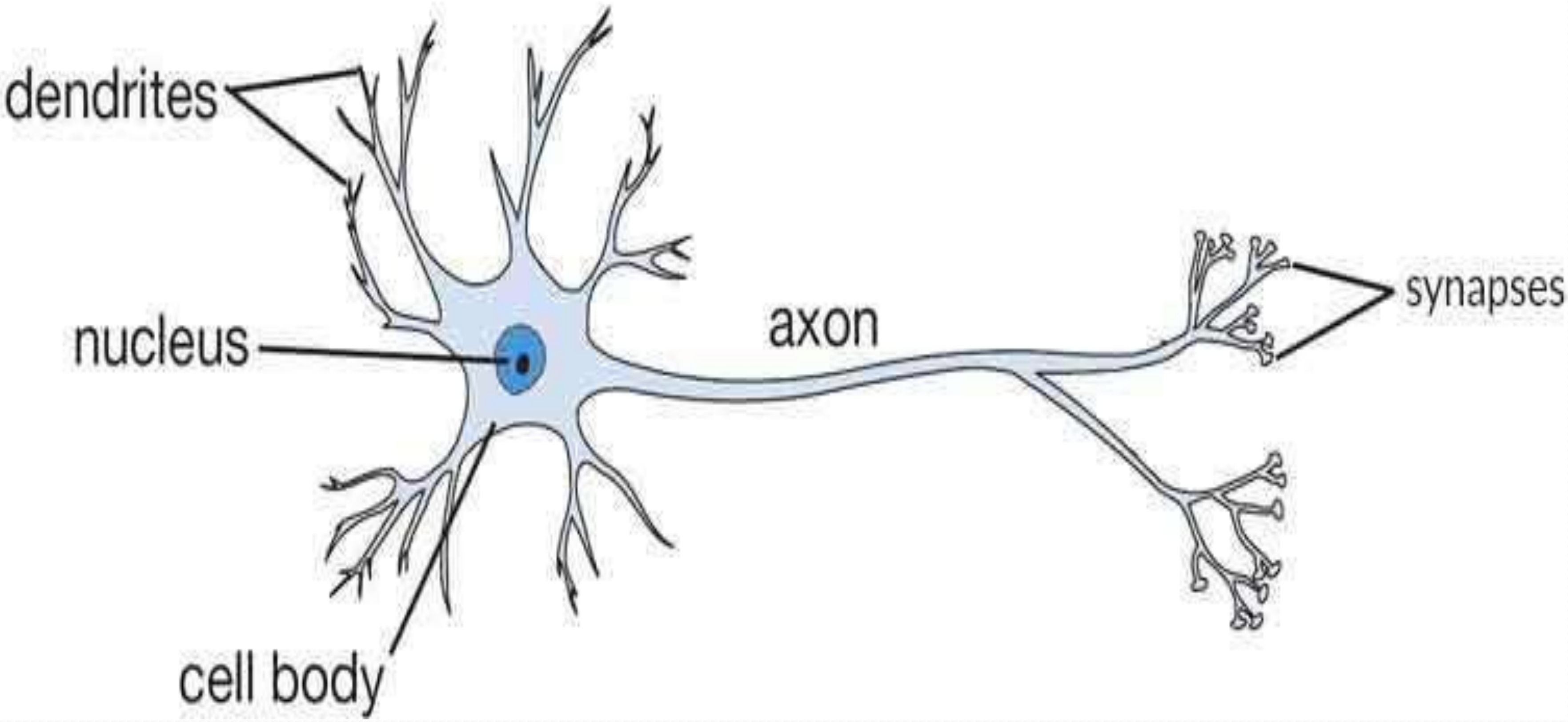
- Adaptive learning
- Self organization
- Real – time operation
- Fault tolerance

Application scope of NN

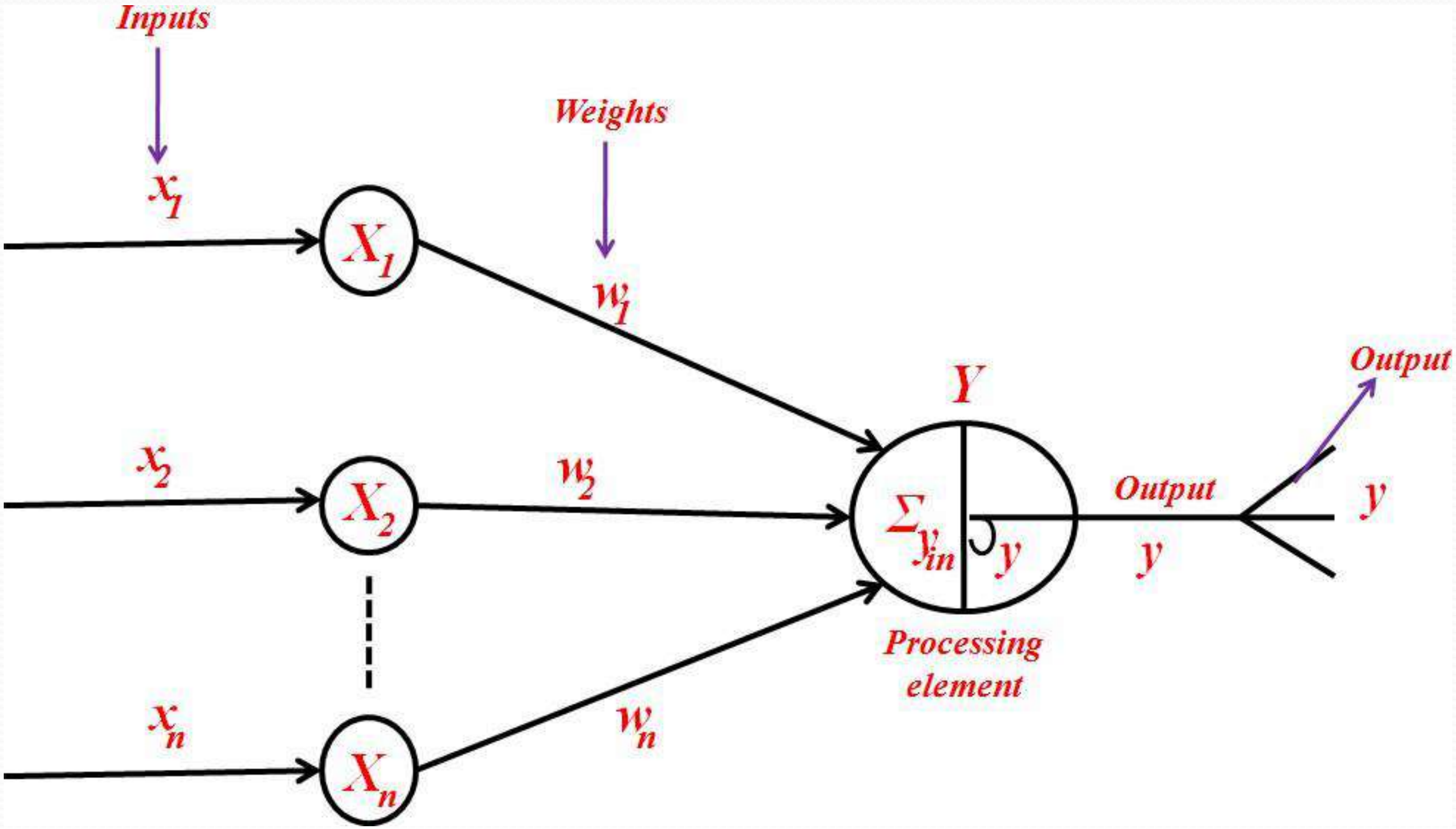
- Air traffic control
- Animal behavior
- Betting on stock market
- Criminal sentencing
- Data mining
- Employee hiring
- Fraud detection
- Hand writing recognition

- Medical diagonosis
- Photos and fingerprint recognition
- Recipes and chemical formulation
- Scheduling of various transports
- Traffic flow prediction
- Voice recognition
- Weather prediction

Biological Neuron



Mathematical model of artificial neuron

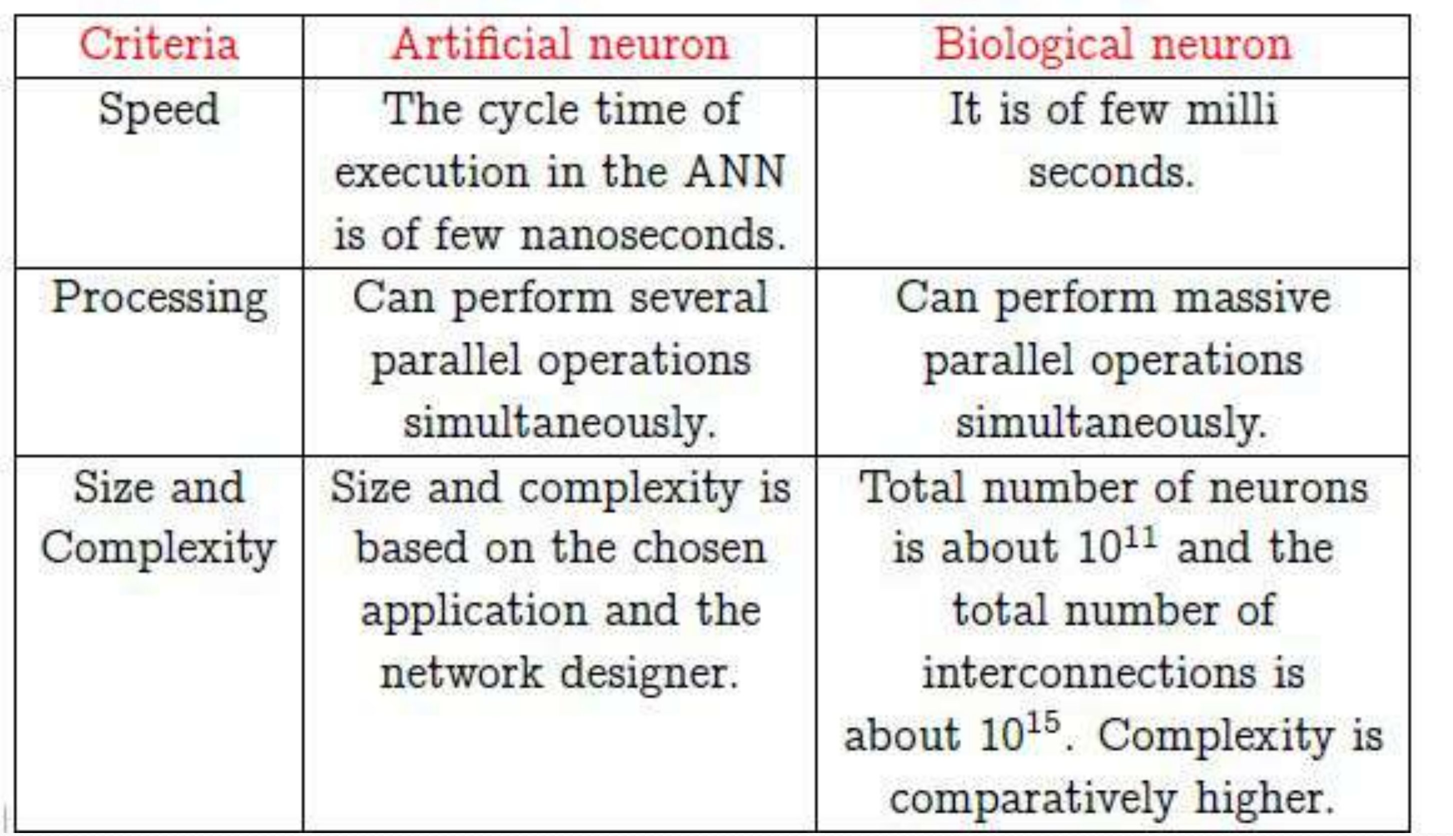


$$\text{Net input, } y_{in} = x_1 w_1 + x_2 w_2 + \ldots . + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

- Where i represents the i th processing element
- Activation function is applied over it to calculate the output
- Weight represents the strength of synapse connecting the input and output neuron
- Weight can be positive or negative
- Positive wt. corresponds to excitatory synapse
- Negative wt. corresponds to inhibitory synapse

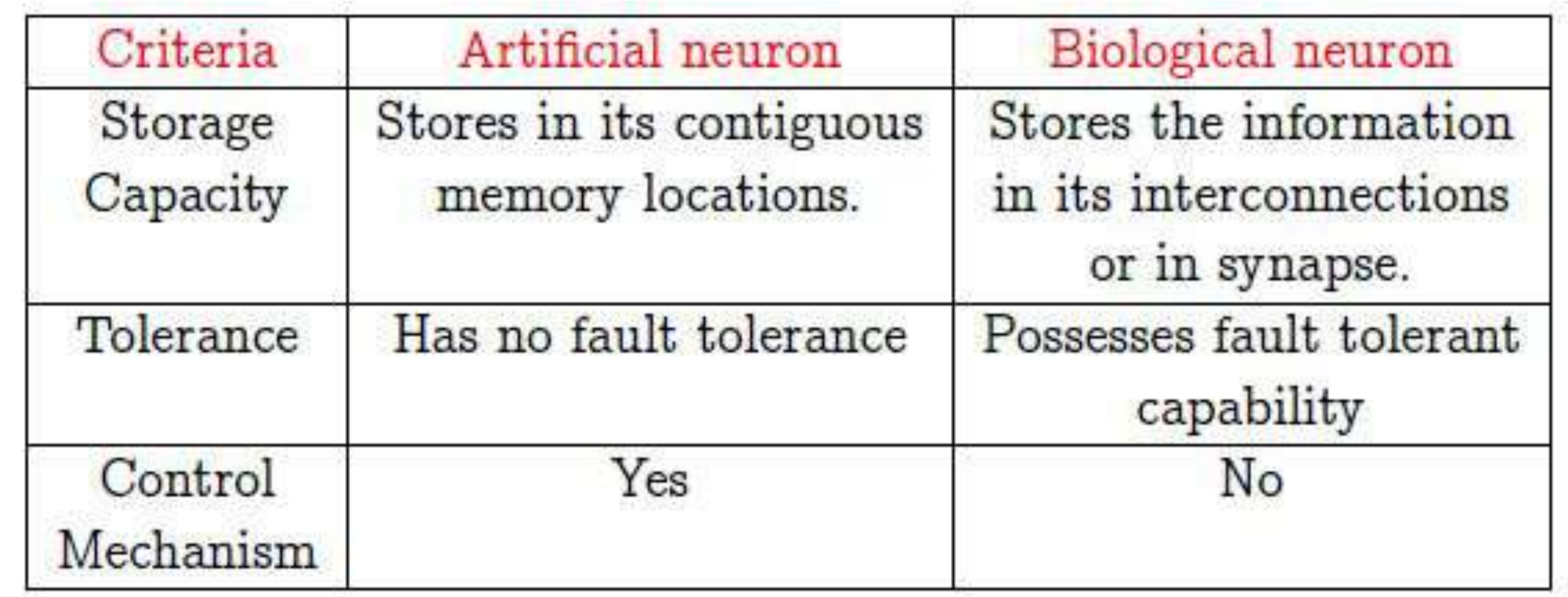
Biological neuron	Artificial neuron
Cell	Neuron
Dendrites	Weights or interconnections
Soma	Net input
Axon	Output

Comparison Between Biological Neuron and Artificial Neuron
(Brain Vs Computer)



Criteria	Artificial neuron	Biological neuron
Speed	The cycle time of execution in the ANN is of few nanoseconds.	It is of few milli seconds.
Processing	Can perform several parallel operations simultaneously.	Can perform massive parallel operations simultaneously.
Size and Complexity	Size and complexity is based on the chosen application and the network designer.	Total number of neurons is about 10^{11} and the total number of interconnections is about 10^{15}. Complexity is comparatively higher.

Contd…



Criteria	Artificial neuron	Biological neuron
Storage Capacity	Stores in its contiguous memory locations.	Stores the information in its interconnections or in synapse.
Tolerance	Has no fault tolerance	Possesses fault tolerant capability
Control Mechanism	Yes	No

Characteristics of ANN

1. It is a neurally implemented mathematical model.
2. There exist a large number of highly interconnected processing elements called *neurons* in an ANN.
3. The interconnections with their weighted linkages hold the informative knowledge.
4. The input signals arrive at the processing elements through connections and connecting weights.
5. The processing elements of the ANN have the *ability to learn, recall, and generalize*. *from the given data by suit- able assignment or adjustment of weights*
6. No single neuron carries specific information.

Basic Models of ANN

The models of ANN are specified by 3 basic entities

- The model's synaptic inter connections
- Training or learning rules adopted for updating and adjusting the connection weights
- Their activation function

- Based on the connection pattern, Neural Networks are classified as

1. Feed Forward Networks
2. Feed Back Networks or Recurrent Networks

- Based on the no of layers, Neural Networks are classified as

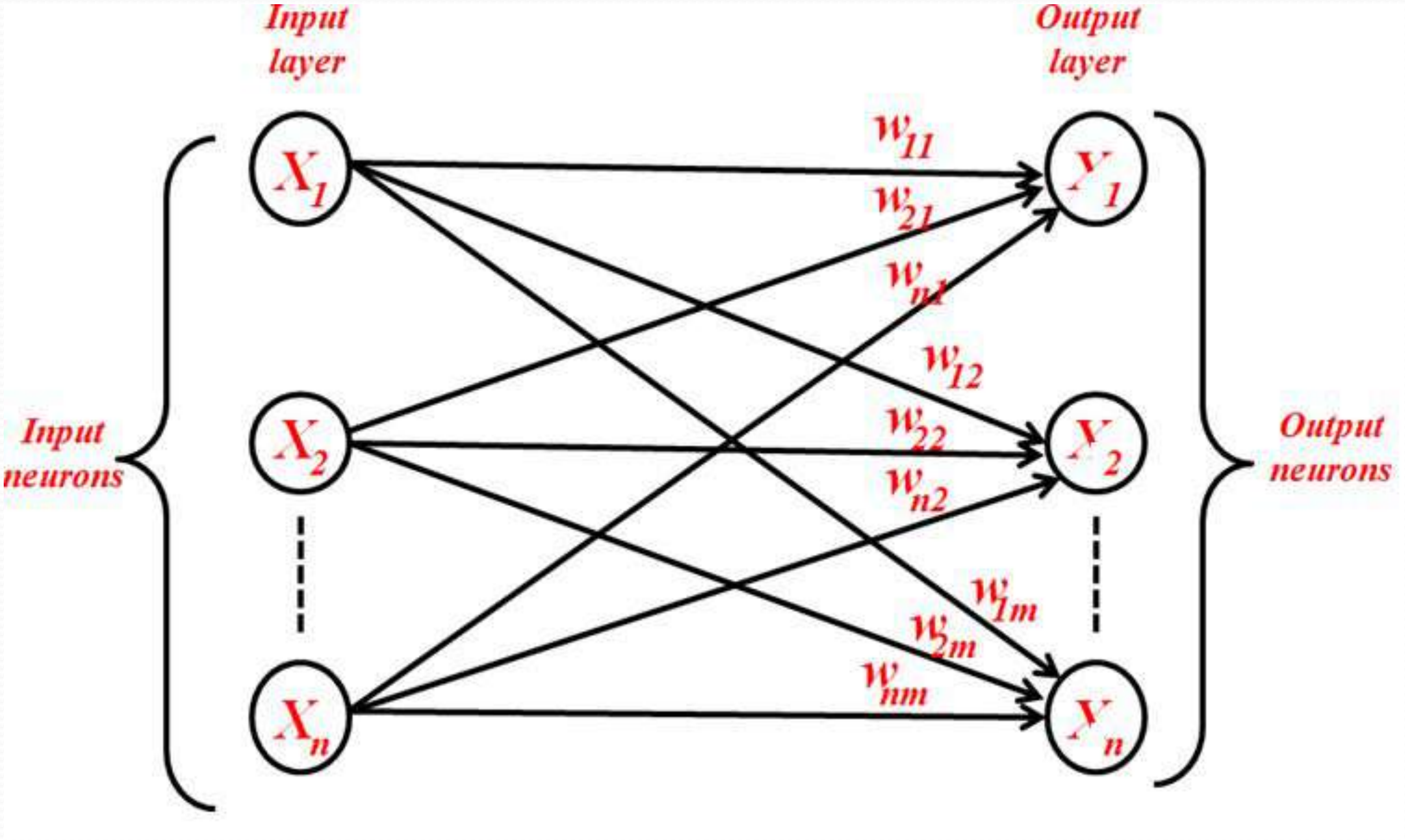
1. Single Layer Networks
2. Multilayer Networks

Connections

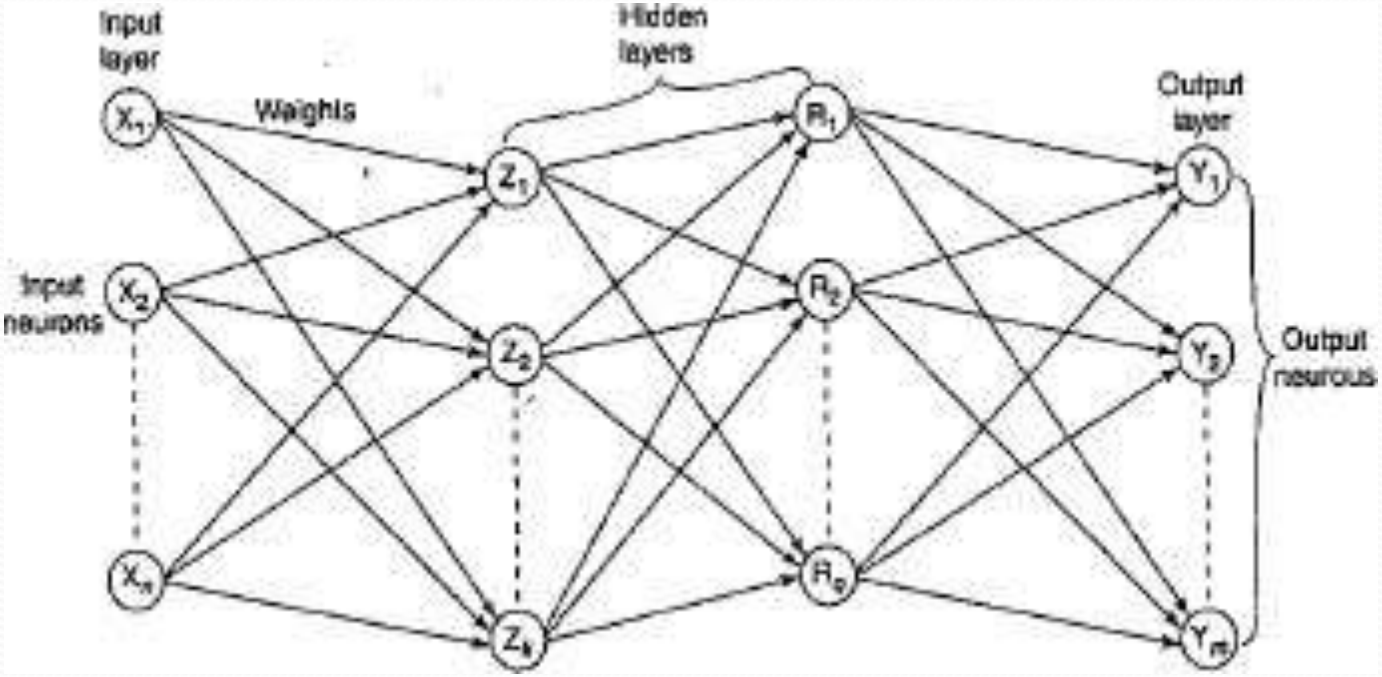
- The ANN consists of highly interconnected processing elements called neurons
- The arrangements of neurons to form layers and the connection pattern formed within and between layers is called the *network architecture*.

1. single layer feed forward network
2. multilayer feed forward network
3. single node with its own feedback
4. single layer recurrent network
5. multilayer recurrent network

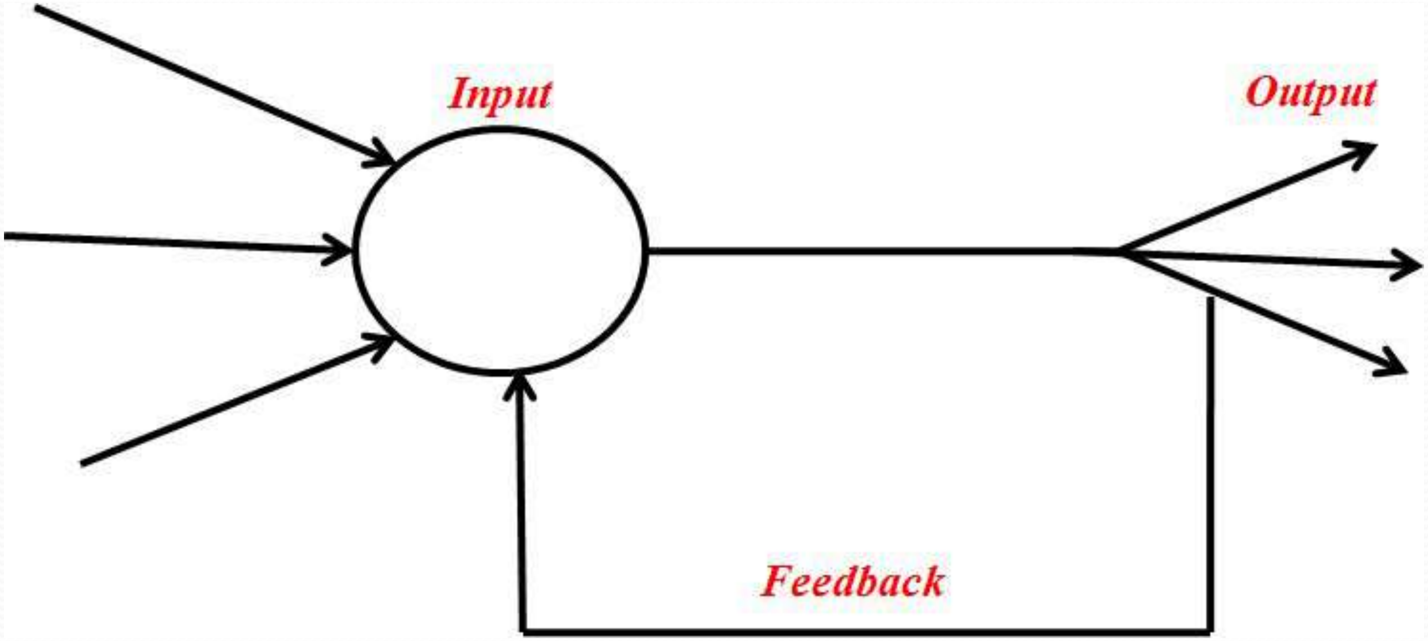
Single layer feed forward network



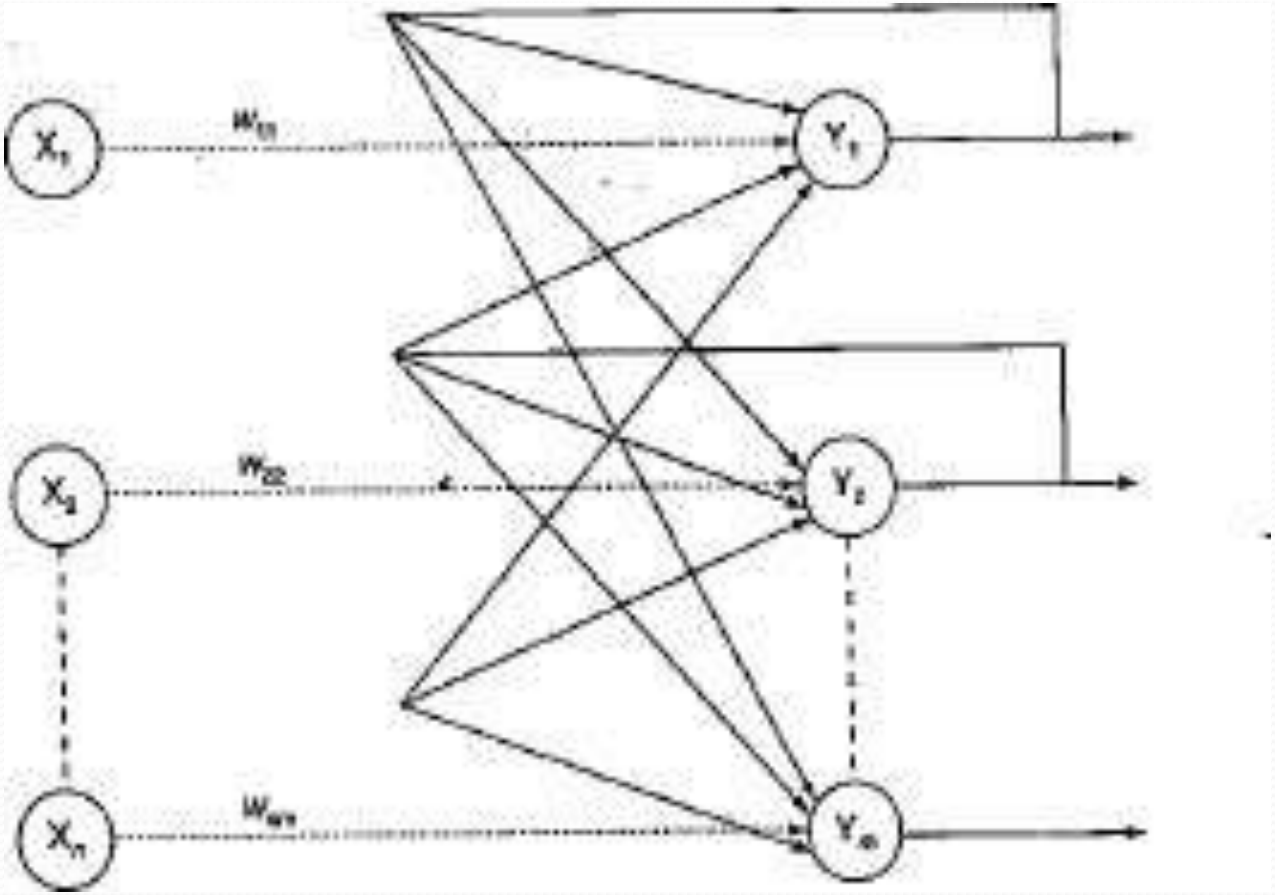
Multilayer feed forward network



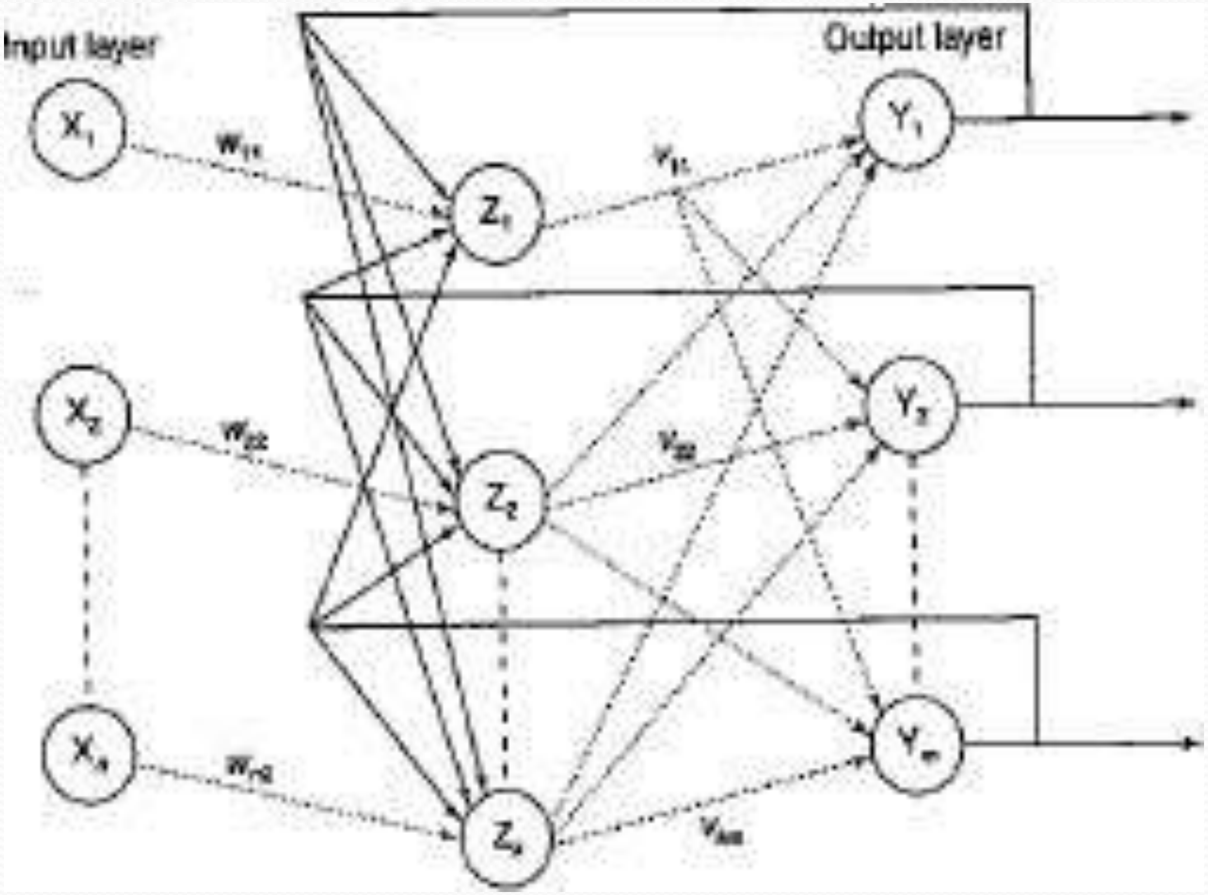
Single node with its own feedback



Single layer recurrent network

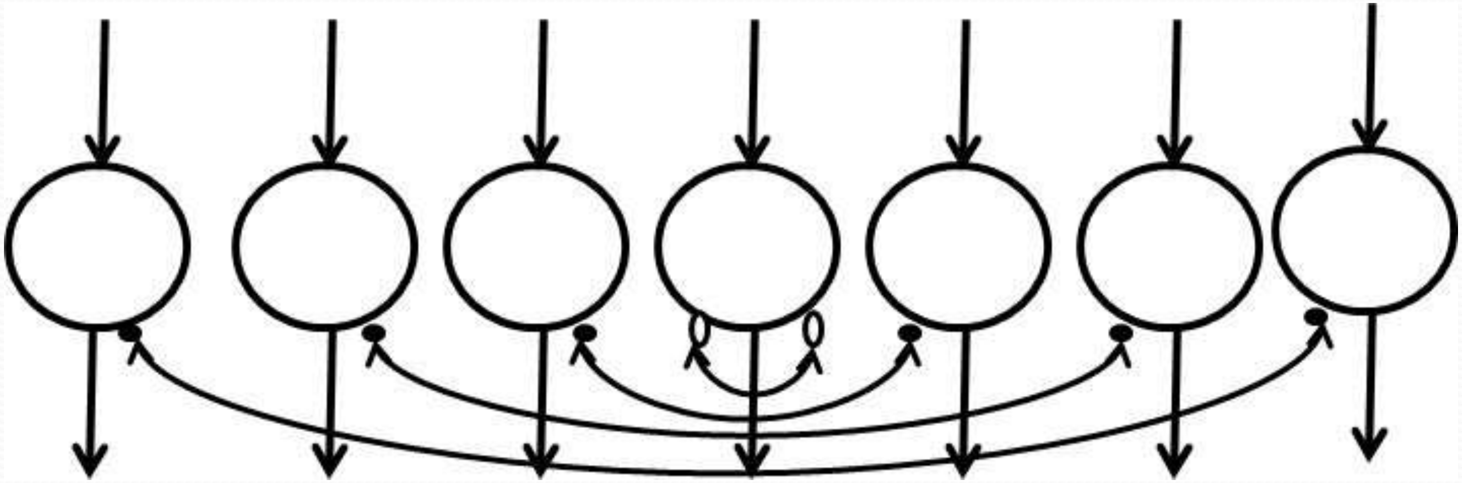


Multilayer recurrent network



- Lateral Feedback Network
- Output of the neuron is directed back to itself or to other neurons in the same layer
- Special type lateral feedback network is called on-centre- off -surround or lateral inhibition structure

On center off surrounded or Lateral inhibition structure

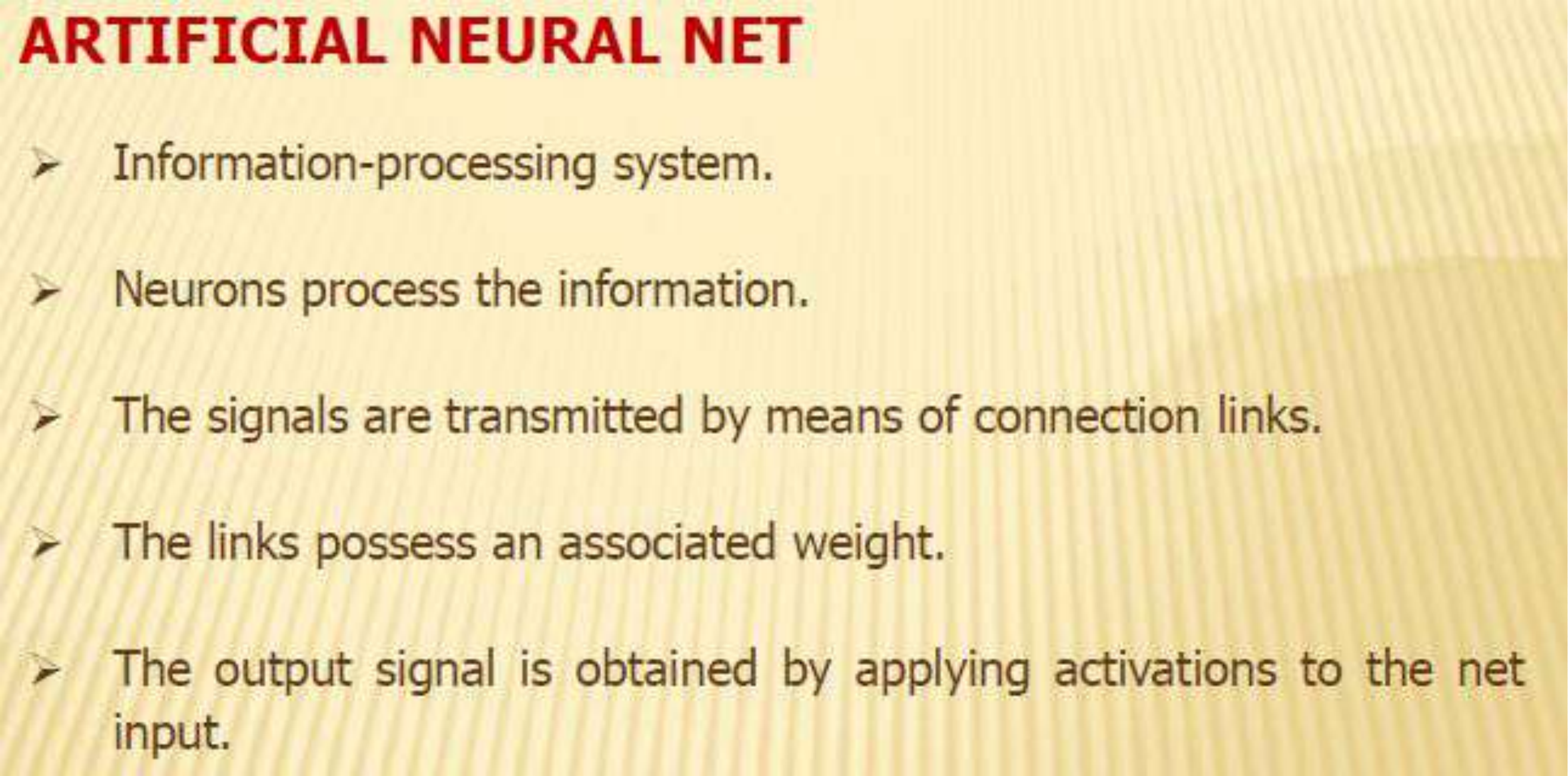


- Special type of lateral f/b network
- Each neuron receives two different types of input

1. Excitatory- Input from nearby processing elements
2. Inhibitory- Inputs from distant processing element

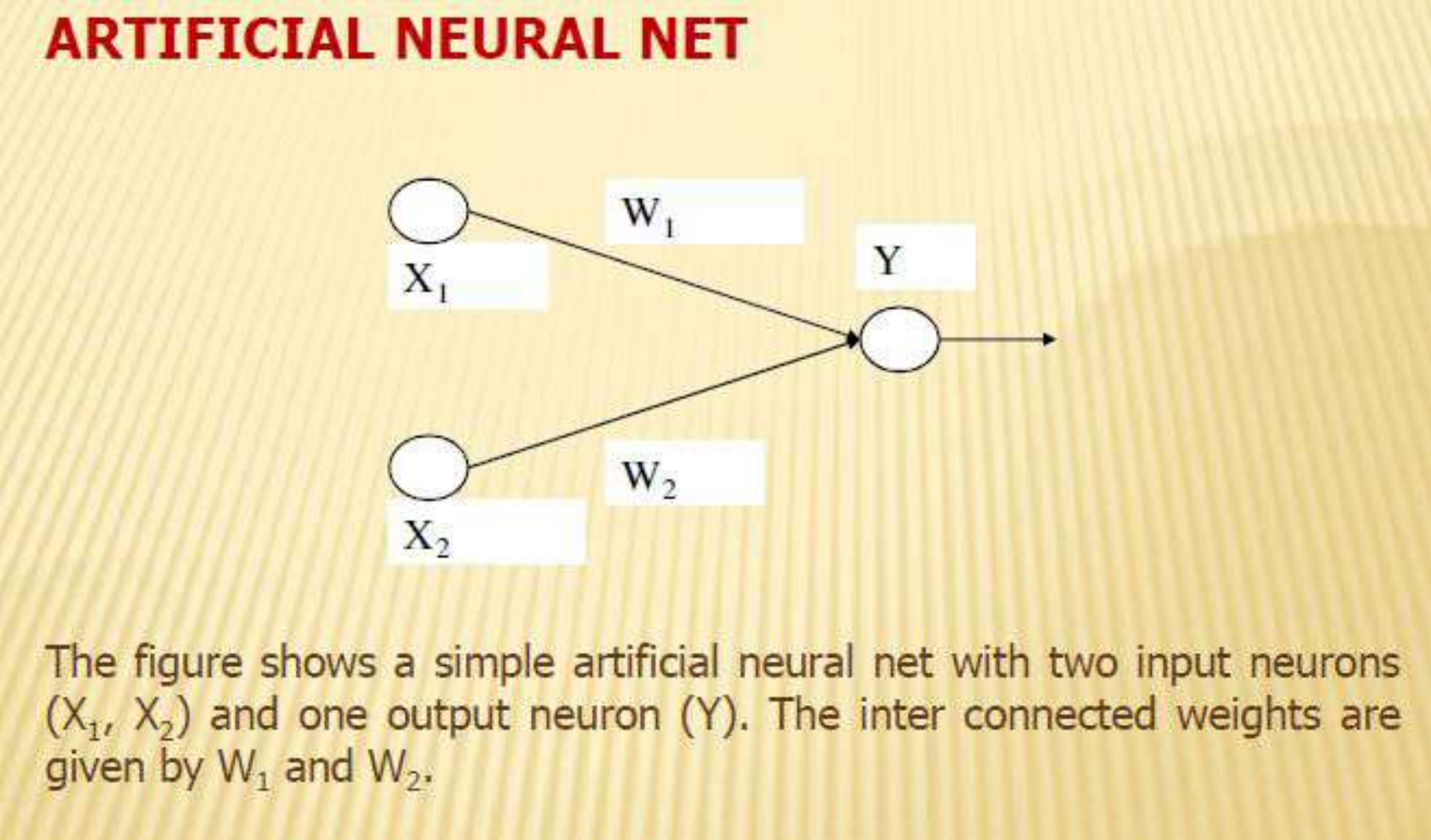
- Open circle represents excitatory
- Closed circle represents inhibitory

ANN (Recap)



ARTIFICIAL NEURAL NET

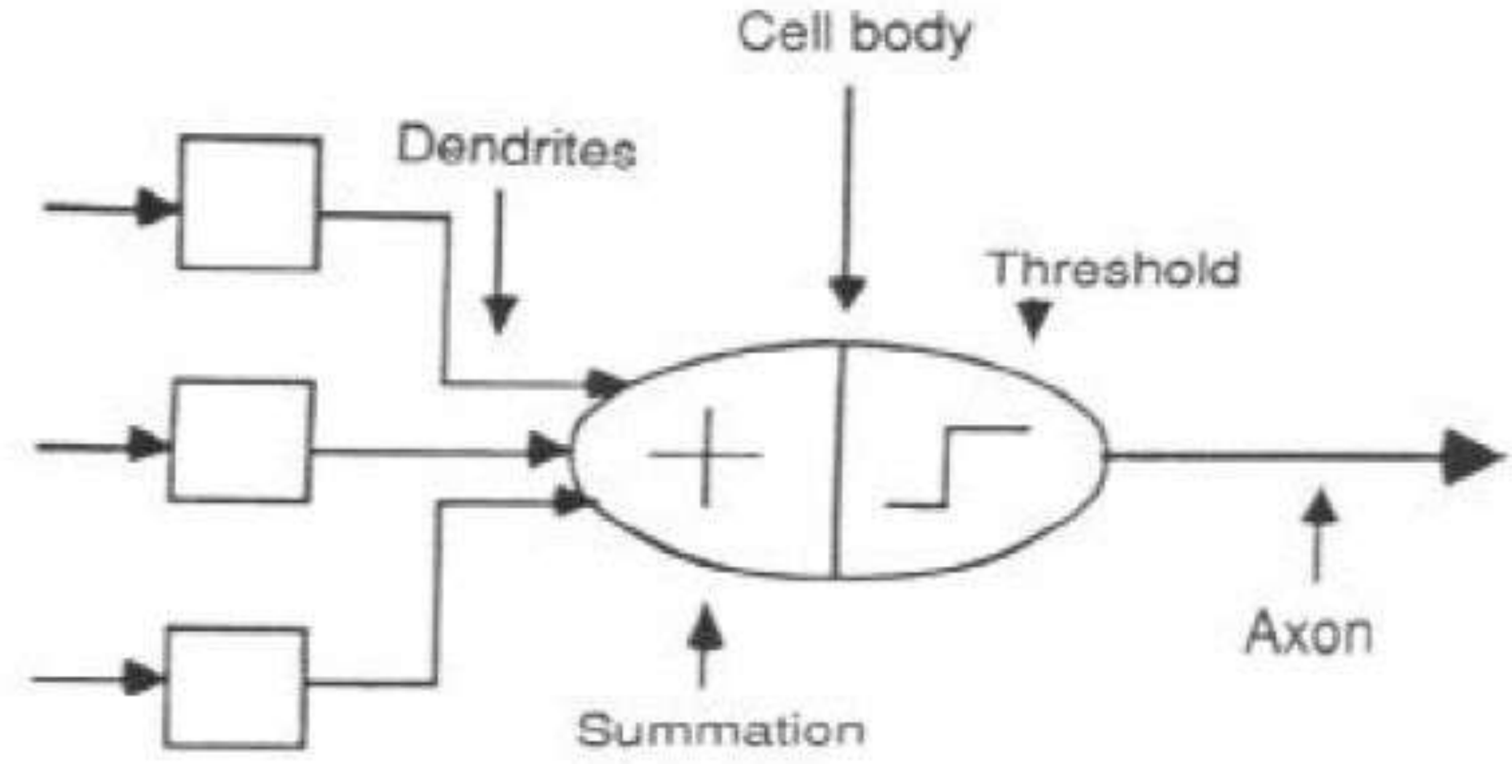
- Information-processing system.
- Neurons process the information.
- The signals are transmitted by means of connection links.
- The links possess an associated weight.
- The output signal is obtained by applying activations to the net input.



ARTIFICIAL NEURAL NET

The figure shows a simple artificial neural net with two input neurons (X_1, X_2) and one output neuron (Y). The inter connected weights are given by W_1 and W_2.

Association of Artificial net with Biological net



Characteristics of Neural Networks

1. Ability to learn
2. Recall
3. Generalize training patterns or data
4. Adapt to changing environment
5. Process information in parallel and distributed manner
6. Fault tolerent

Learning

- *Learning or training* is a process by which a NN adapts it-self to a stimulus by making proper parameter adjustments, resulting in the production of desired response.
- It is a method of setting the appropriate weight values or by changing the network structure
- Two kinds of learning:
- Parameter learning: It updates the connecting weights in a neural net.
- Structure learning: It focuses on the change in network structure.

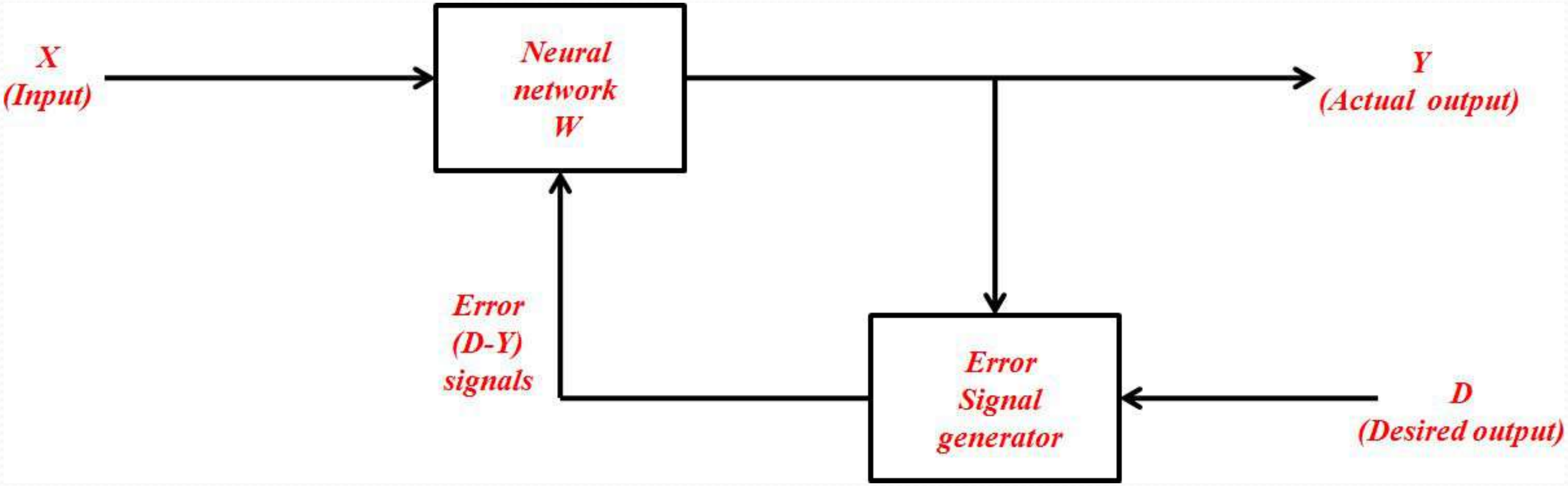
Categories of Learning

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

Supervised Learning

- Learning with the help of a Supervisor/Teacher
- Provide the network with a series of sample inputs and compare the obtained the output with the expected output.
- The difference between these two o/p s is the error
- Making use of the error network parameters are adjusted so as to make the error minimum
- This results in performance improvement
- Eg: Perceptron Networks, Adaline, Madaline, BPN, RBFN etc.

Supervised Learning

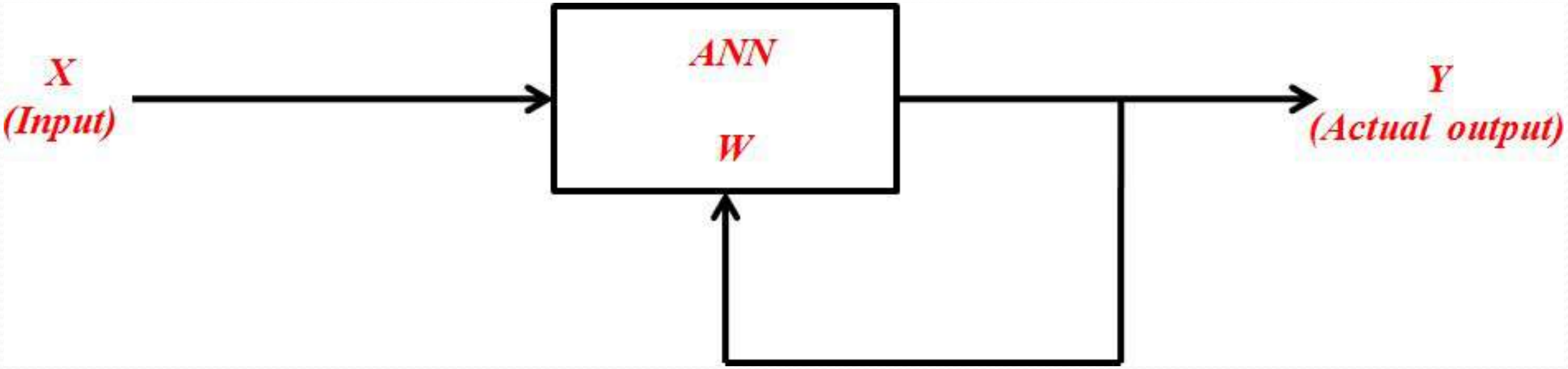


- Input vector along with the target vector is called training pair
- During training, input is presented to the network which results in an output vector (actual output vector- Y)
- Actual output is compared with Desired output-D
- Error is calculated (D-Y)
- Error is used for weight adjustment
- Repeat the process until the error is minimum

Unsupervised Learning

- Learning without the help of a teacher/ supervisor
- Output is not known
- Error information can not be used to improve network performance
- Input vectors of similar type are grouped without the use of training data to form clusters/groups/class
- When a new input pattern is applied to NN , it gives an output response indicating the class to which it belongs

Unsupervised Learning

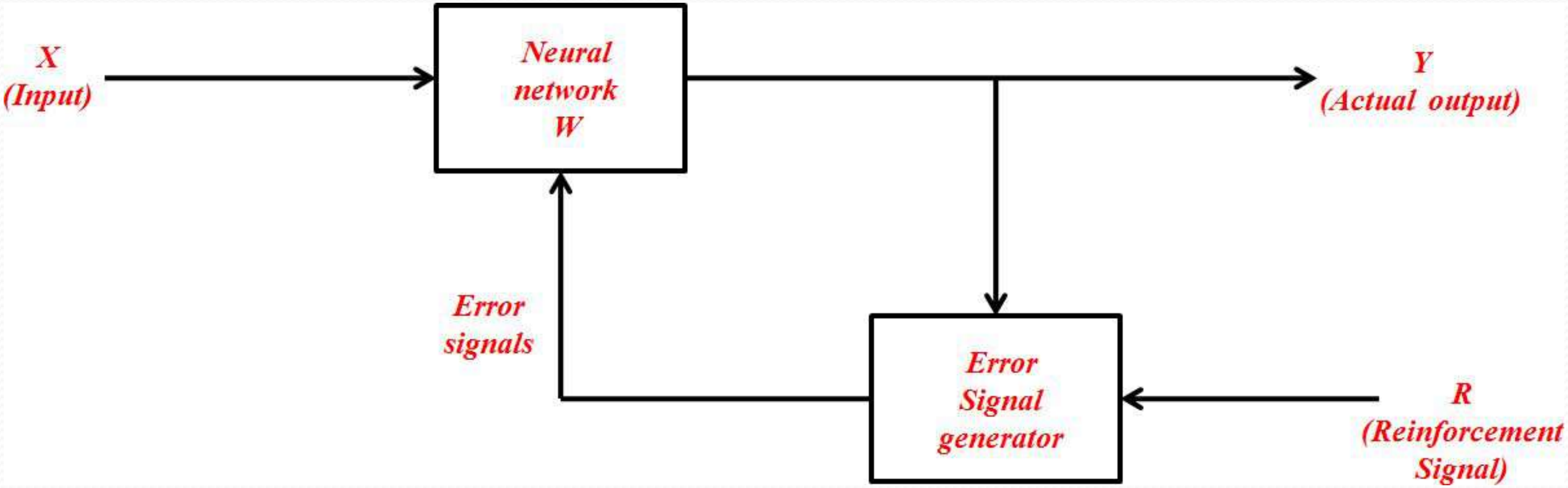


- The network learns on its own by discovering the patterns , regularities, features or categories from the input data and relations for the input data over the output
- While discovering these features, the network undergoes change in its parameter called Self Organization
- Eg: Hebbian learning , Competitive laerning

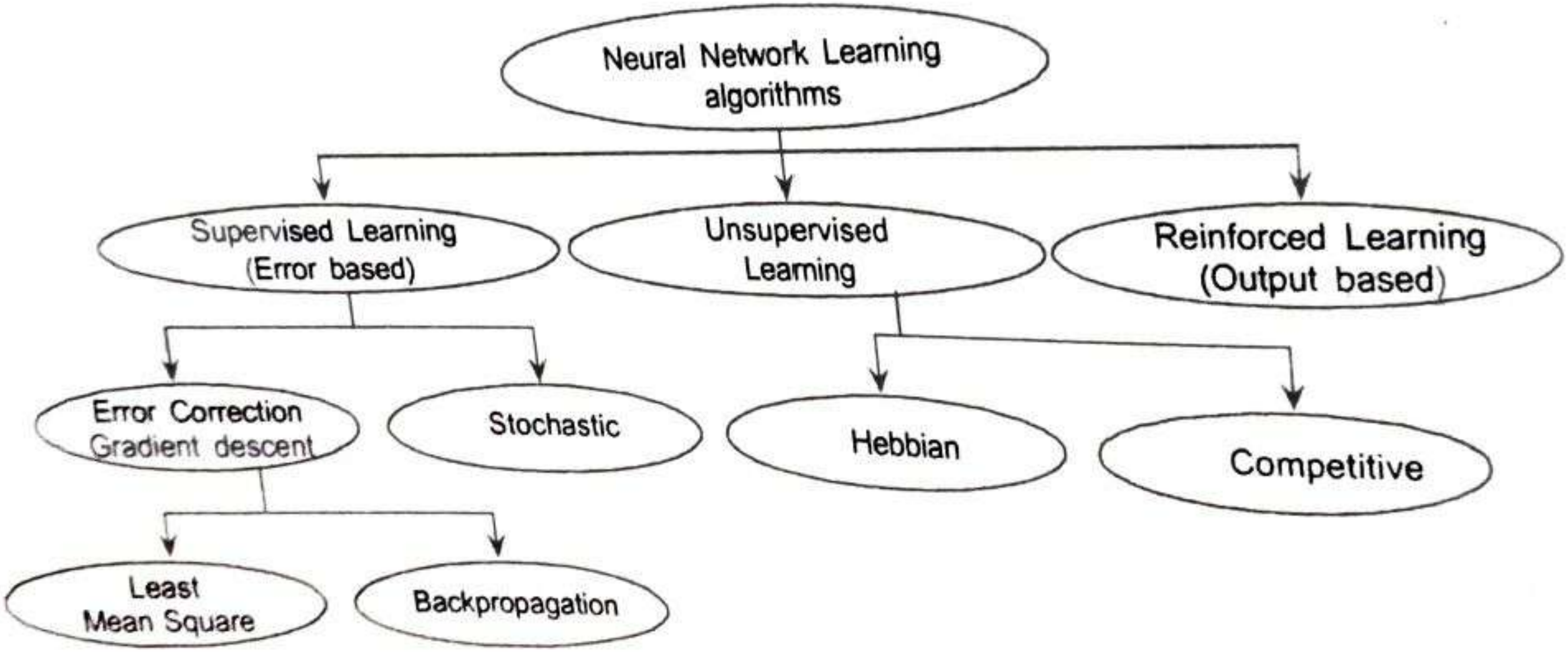
Reinforcement Learning

- It is a form of supervised learning
- In SL correct target outputs are known for each input
- In RL less information is available about target output
- It tells whether the output is correct or not
- Reward- for correct output
- Penalty – for wrong output
- SL- exact information about the output is available
- RL- only critic information about the output is available

Reinforcement Learning



Classification of Learning Algorithms



Neural Network Learning algorithms
Supervised Learning (Error based)
Unsupervised Learning
Reinforced Learning (Output based)
Error Correction Gradient descent
Stochastic
Hebbian
Competitive
Least Mean Square
Backpropagation

Gradient descent learning

- **Gradient Descent learning :**
  This learning technique is based on the minimization of error *E* defined in terms of weights and the activation function of the network.
- Also, it is required that the activation function employed by the network is differentiable, as the weight update is dependent on the gradient of the error *E* .
- Thus, if ΔW_{ij} denoted the weight update of the link connecting the *i*-th and *j*-th neuron of the two neighboring layers then

  $$\Delta W_{ij} = \eta \frac{\partial E}{\partial W_{ij}}$$

  where η is the **learning rate parameter** and $\frac{\partial E}{\partial W_{ij}}$ is the **gradient** with reference to the weight W_{ij}
- The least mean square and back propagation are two variations of this learning technique.

Stochastic learning

- **Stochastic learning**

  In this method, weights are adjusted in a probabilistic fashion. Simulated annealing is an example of such learning (proposed by Boltzmann and Cauch)

Hebbian learning

Hebbian learning was developed Donald Hebb in 1949

- This learning is based on correlative weight adjustment. This is, in fact, the learning technique inspired by biology.
- Here, the input-output pattern pairs (x_i, y_i) are associated with the weight matrix W. W is also known as the correlation matrix.
- This matrix is computed as follows.

$$W = \sum_{i=1}^{n} X_i Y_i^T$$

where Y_i^T is the transpose of the associated vector y_i

Competitive learning

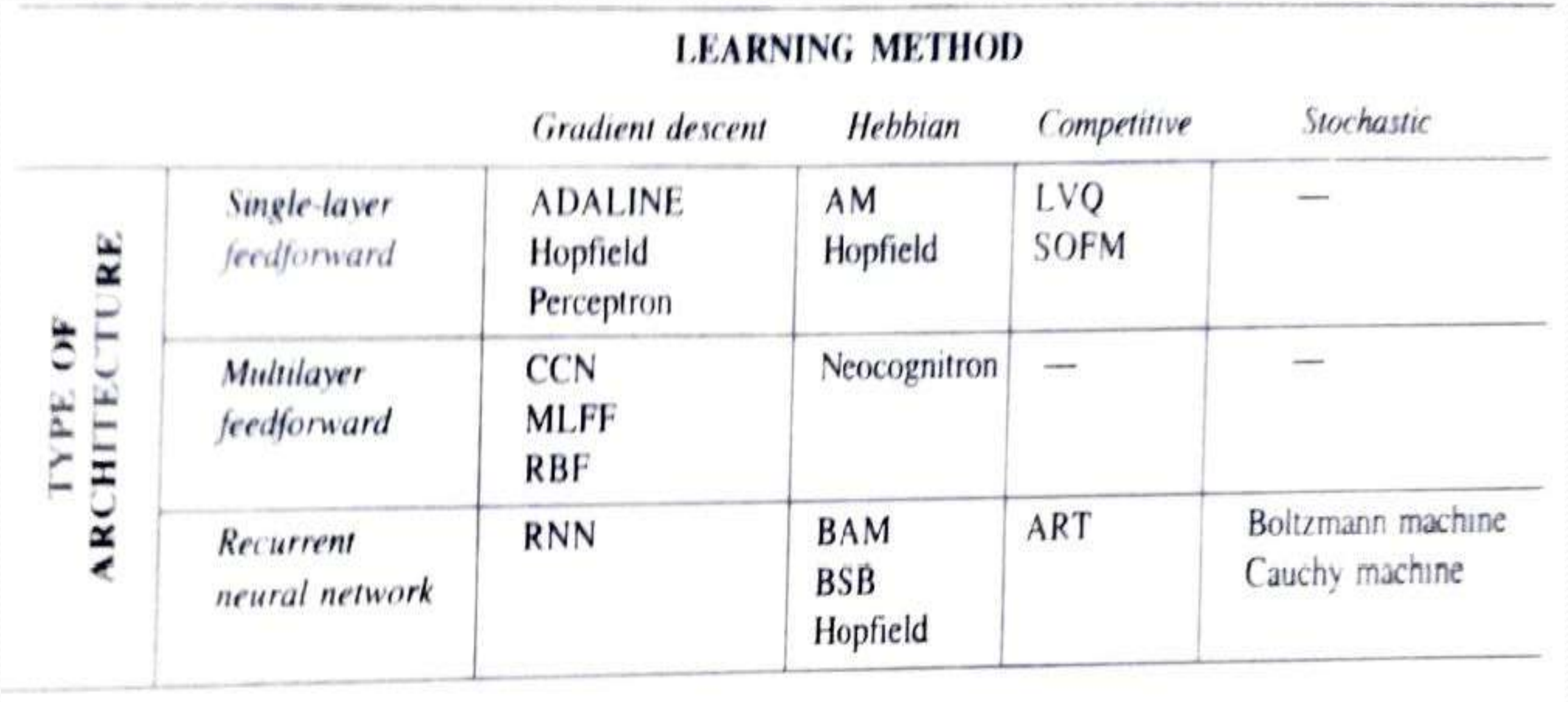
- **Competitive learning**

  In this learning method, those neurons which responds strongly to input stimuli have their weights updated.

- When an input pattern is presented, all neurons in the layer compete and the winning neuron undergoes weight adjustment.

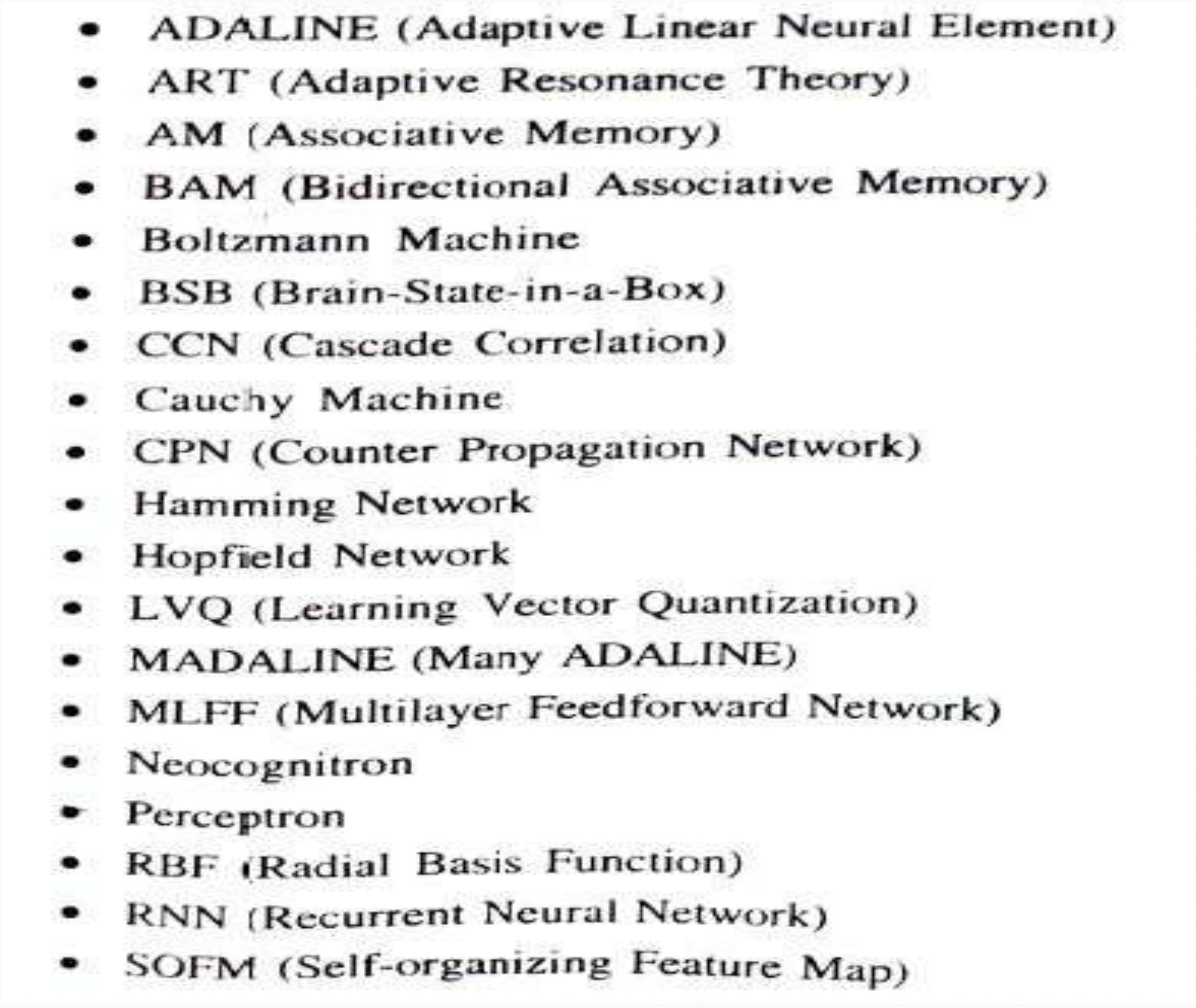
- This is why it is called a Winner-takes-all strategy.

Classification of NN



		LEARNING METHOD			
		Gradient descent	*Hebbian*	*Competitive*	*Stochastic*
TYPE OF ARCHITECTURE	*Single-layer feedforward*	ADALINE Hopfield Perceptron	AM Hopfield	LVQ SOFM	—
	Multilayer feedforward	CCN MLFF RBF	Neocognitron	—	—
	Recurrent neural network	RNN	BAM BSB Hopfield	ART	Boltzmann machine Cauchy machine

Taxonomy of NN Architectures



- ADALINE (Adaptive Linear Neural Element)
- ART (Adaptive Resonance Theory)
- AM (Associative Memory)
- BAM (Bidirectional Associative Memory)
- Boltzmann Machine
- BSB (Brain-State-in-a-Box)
- CCN (Cascade Correlation)
- Cauchy Machine
- CPN (Counter Propagation Network)
- Hamming Network
- Hopfield Network
- LVQ (Learning Vector Quantization)
- MADALINE (Many ADALINE)
- MLFF (Multilayer Feedforward Network)
- Neocognitron
- Perceptron
- RBF (Radial Basis Function)
- RNN (Recurrent Neural Network)
- SOFM (Self-organizing Feature Map)

Activation Functions

- Mathematical equations that determine the output of NN
- It decides whether a neuron is fired(activated) or not
- Two types 1. Linear 2. Non-linear
- Activation functions

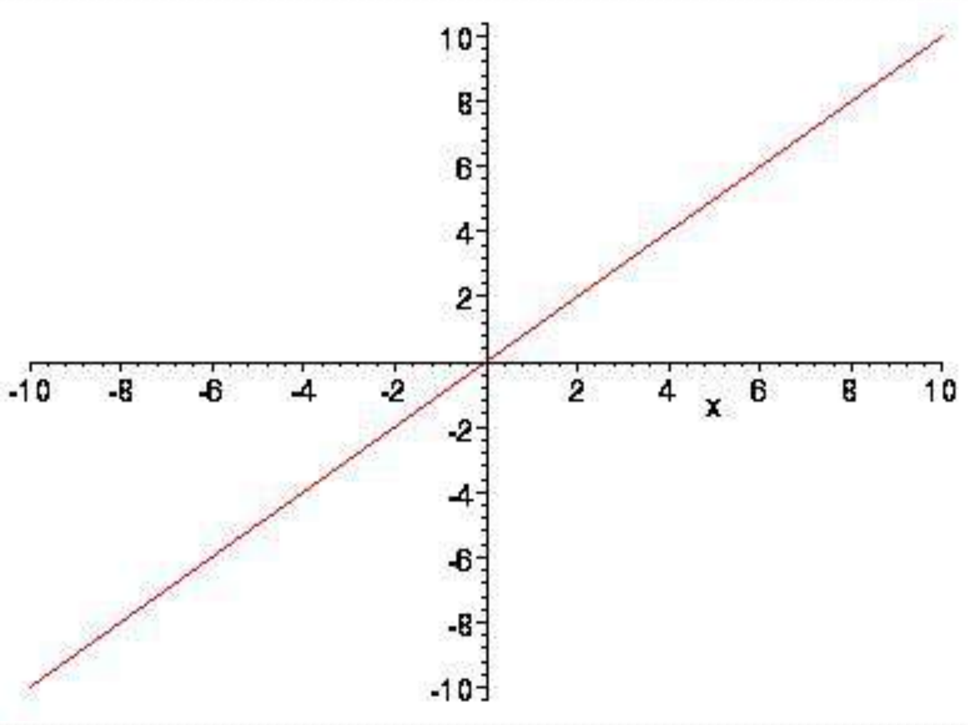
1. Identity Function
2. Step Function--- Binary & Bipolar
3. Sigmoid Function--- Binary & Bipolar
4. Hyperbolic Tangent Function
5. Ramp Function etc.

Binary step function is known as threshold function or Heaviside function

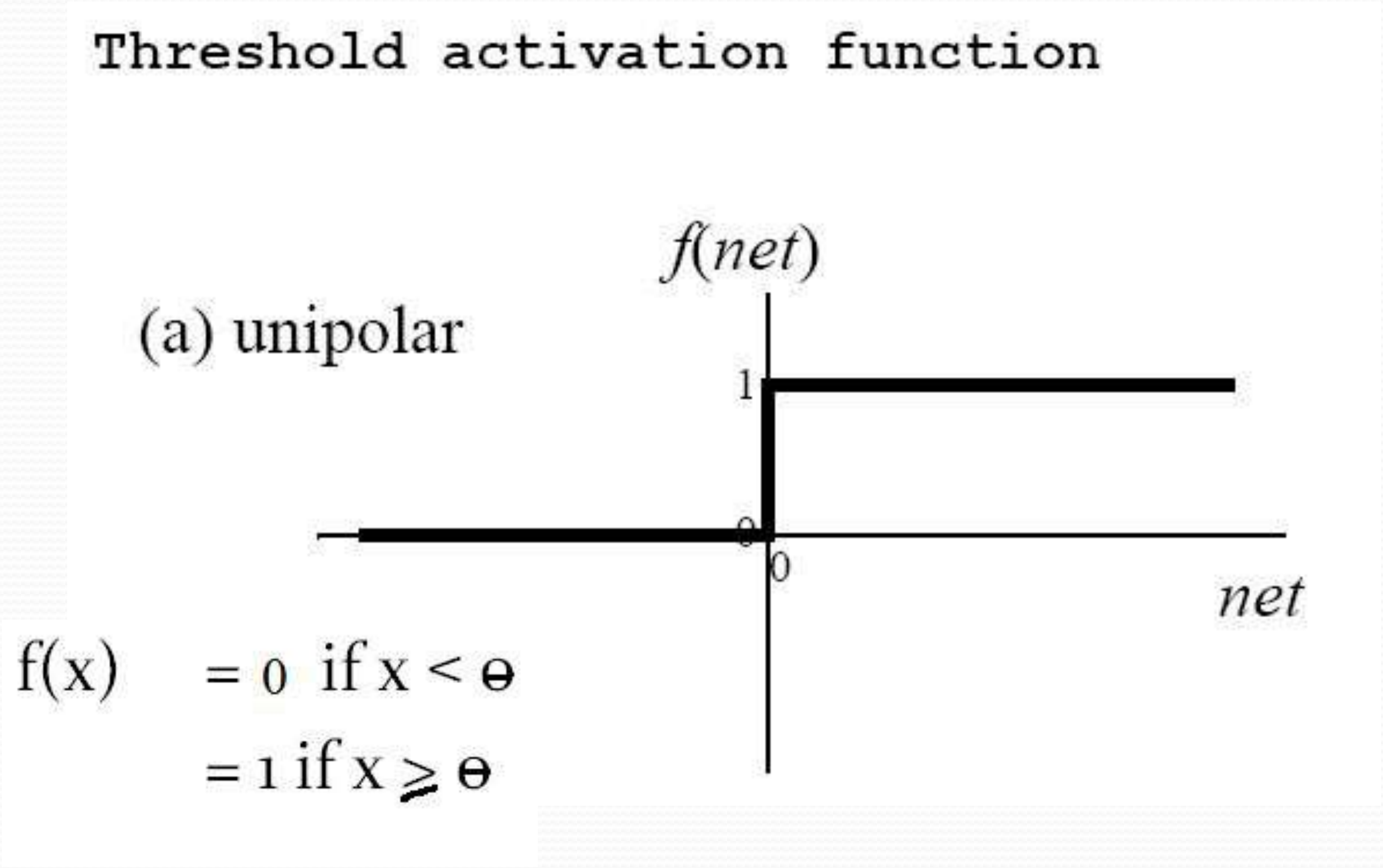
Identity Function

The identity function is given by

$$f(x) = x$$



Step Function



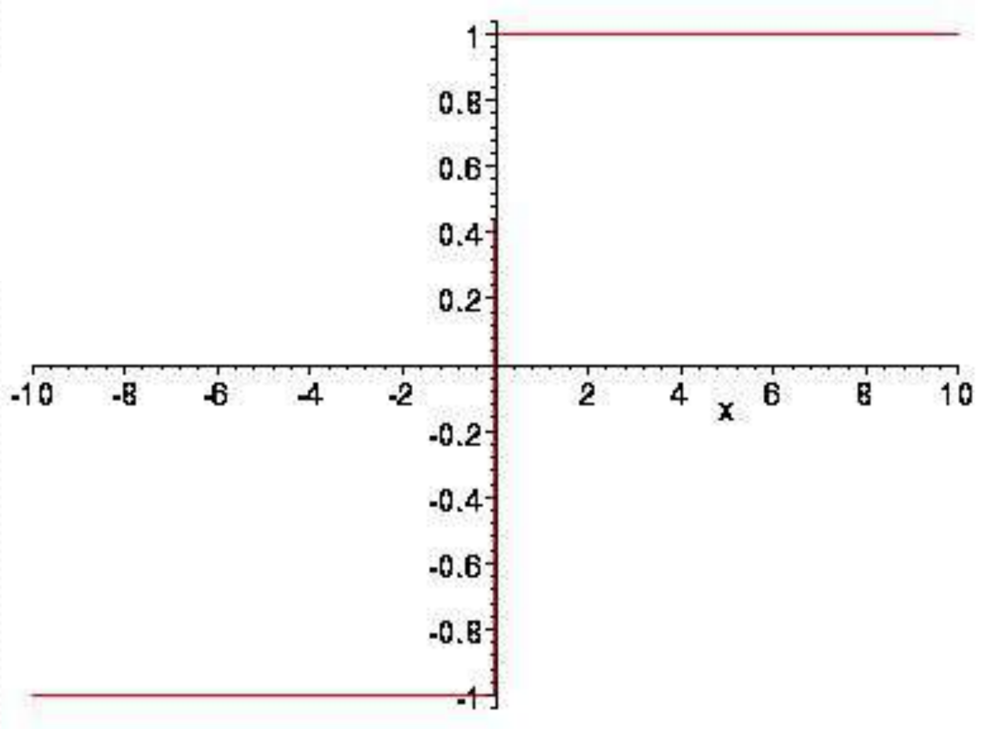
Threshold activation function

(a) unipolar

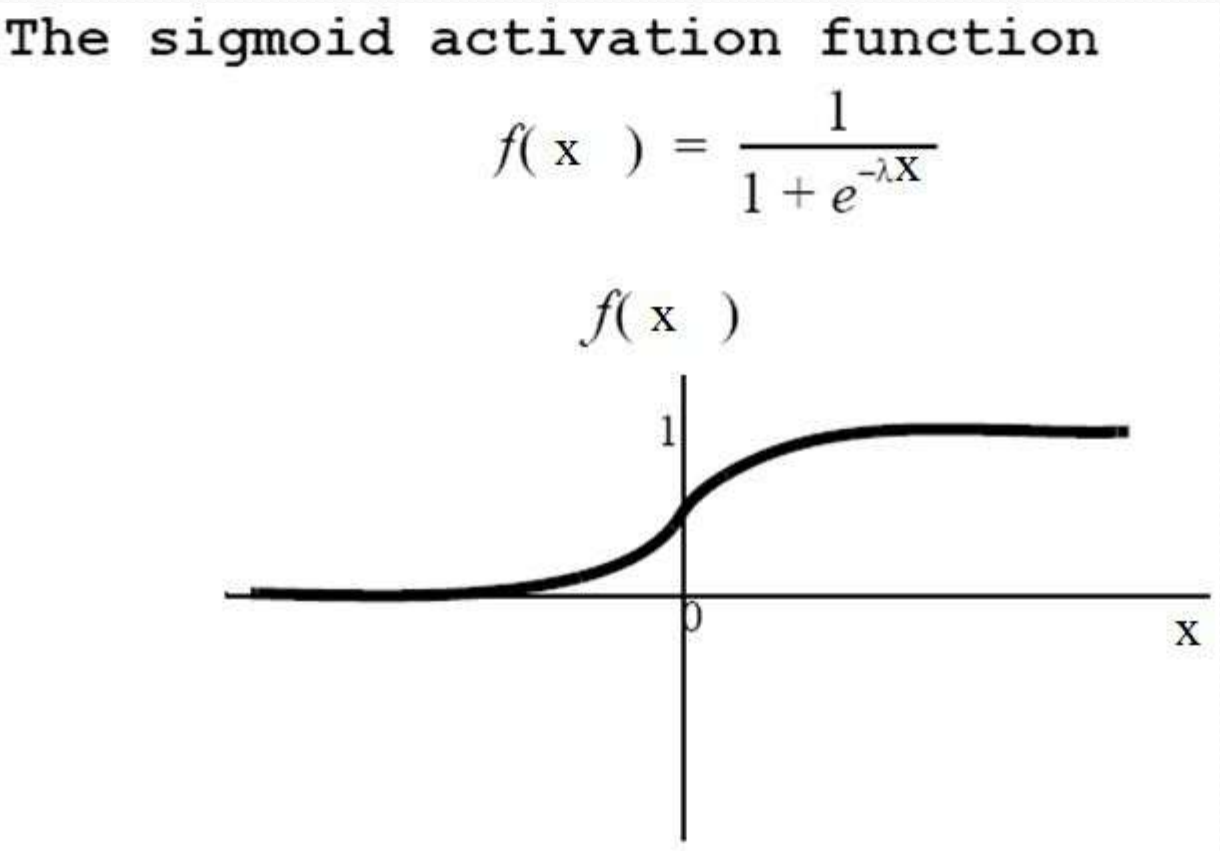
$$f(x) \quad = 0 \ \text{if } x < \theta$$
$$= 1 \ \text{if } x \geq \theta$$

Bipolar Step Function

$$f(x) \quad = -1 \text{ if } x < \theta$$
$$\quad = 1 \text{ if } x \geq \theta$$



Binary Sigmoid Activation Function



The sigmoid activation function

$$f(x) = \frac{1}{1 + e^{-\lambda x}}$$

Bipolar Sigmoid Function

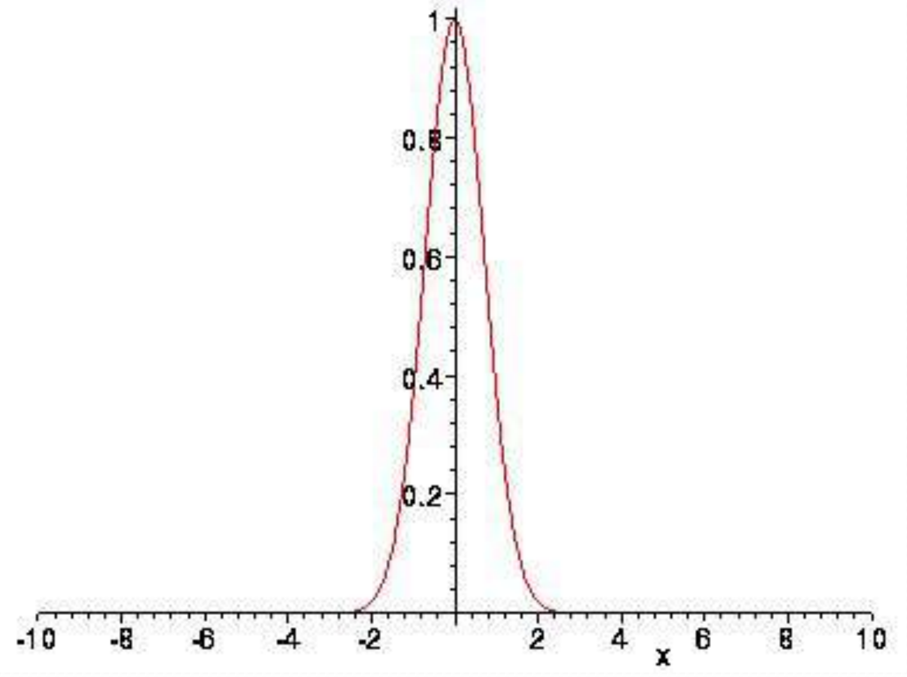
- It is given by $f(x) = \frac{2}{1+e^{-\lambda x}} - 1$
- It is closely related to hyperbolic tangent function

Radial Basis Functions

- A radial basis function is simply a

Gaussian,

$$f(x) = e^{(-\alpha x^2)}$$



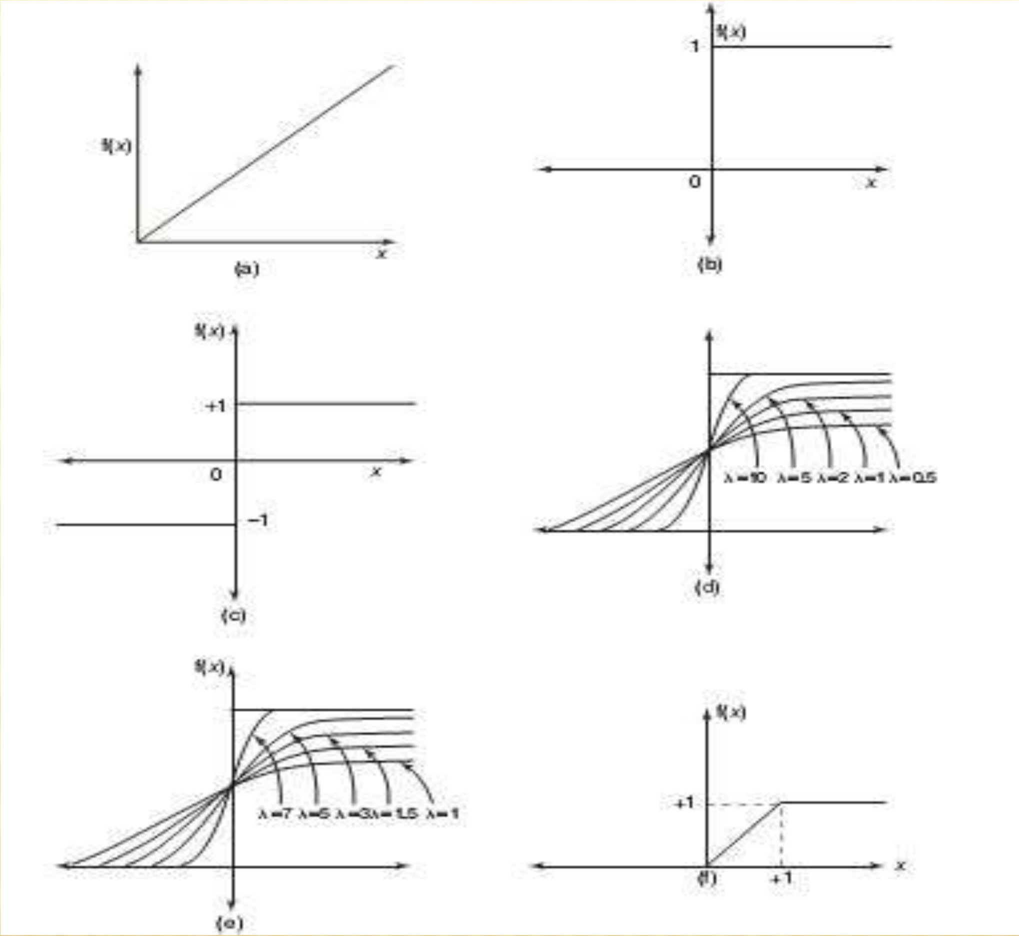
ACTIVATION FUNCTIONS (Recap)

- **ACTIVATION LEVEL – DISCRETE OR CONTINUOUS**

- **HARD LIMIT FUCNTION (DISCRETE)**
  - Binary Activation function
  - Bipolar activation function
  - Identity function

- **SIGMOIDAL ACTIVATION FUNCTION (CONTINUOUS)**
  - Binary Sigmoidal activation function
  - Bipolar Sigmoidal activation function

ACTIVATION FUNCTIONS (Recap)



Activation functions:

(A) Identity

(B) Binary step

(C) Bipolar step

(D) Binary sigmoidal

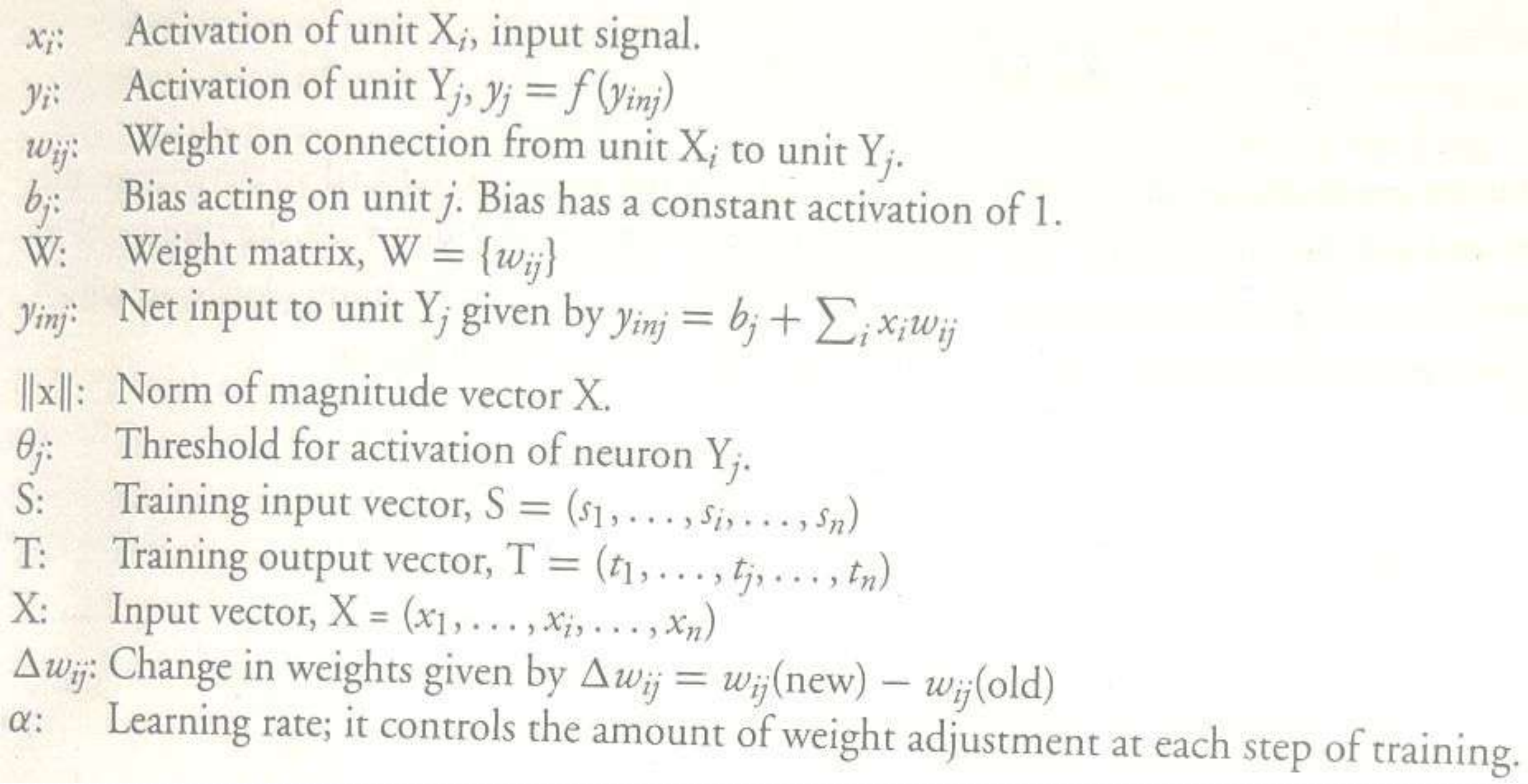
(E) Bipolar sigmoidal

(F) Ramp

Important Terminologies of ANN

- Weights
- Bias
- Threshold
- Learning Rate
- Momentum Factor
- Vigilance Parameter

Common notations used in NN



x_i: Activation of unit X_i, input signal.

y_i: Activation of unit Y_j, $y_j = f(y_{inj})$

w_{ij}: Weight on connection from unit X_i to unit Y_j.

b_j: Bias acting on unit j. Bias has a constant activation of 1.

W: Weight matrix, $W = \{w_{ij}\}$

y_{inj}: Net input to unit Y_j given by $y_{inj} = b_j + \sum_i x_i w_{ij}$

$\|x\|$: Norm of magnitude vector X.

θ_j: Threshold for activation of neuron Y_j.

S: Training input vector, $S = (s_1, \ldots, s_i, \ldots, s_n)$

T: Training output vector, $T = (t_1, \ldots, t_j, \ldots, t_n)$

X: Input vector, $X = (x_1, \ldots, x_i, \ldots, x_n)$

Δw_{ij}: Change in weights given by $\Delta w_{ij} = w_{ij}(\text{new}) - w_{ij}(\text{old})$

α: Learning rate; it controls the amount of weight adjustment at each step of training.

McCulloch Pitts Neuron

- Usually called as *M P neuron*. Connected
- by directed weighted paths. Activation of
- a M P neuron is binary.
- The weights associated with the communication links may be *excitatory(weight is positive)* or *inhibitory(weight is negative)*.

  The threshold plays a major role in the M P neuron.
-

- Neurons are sparsely and randomly connected

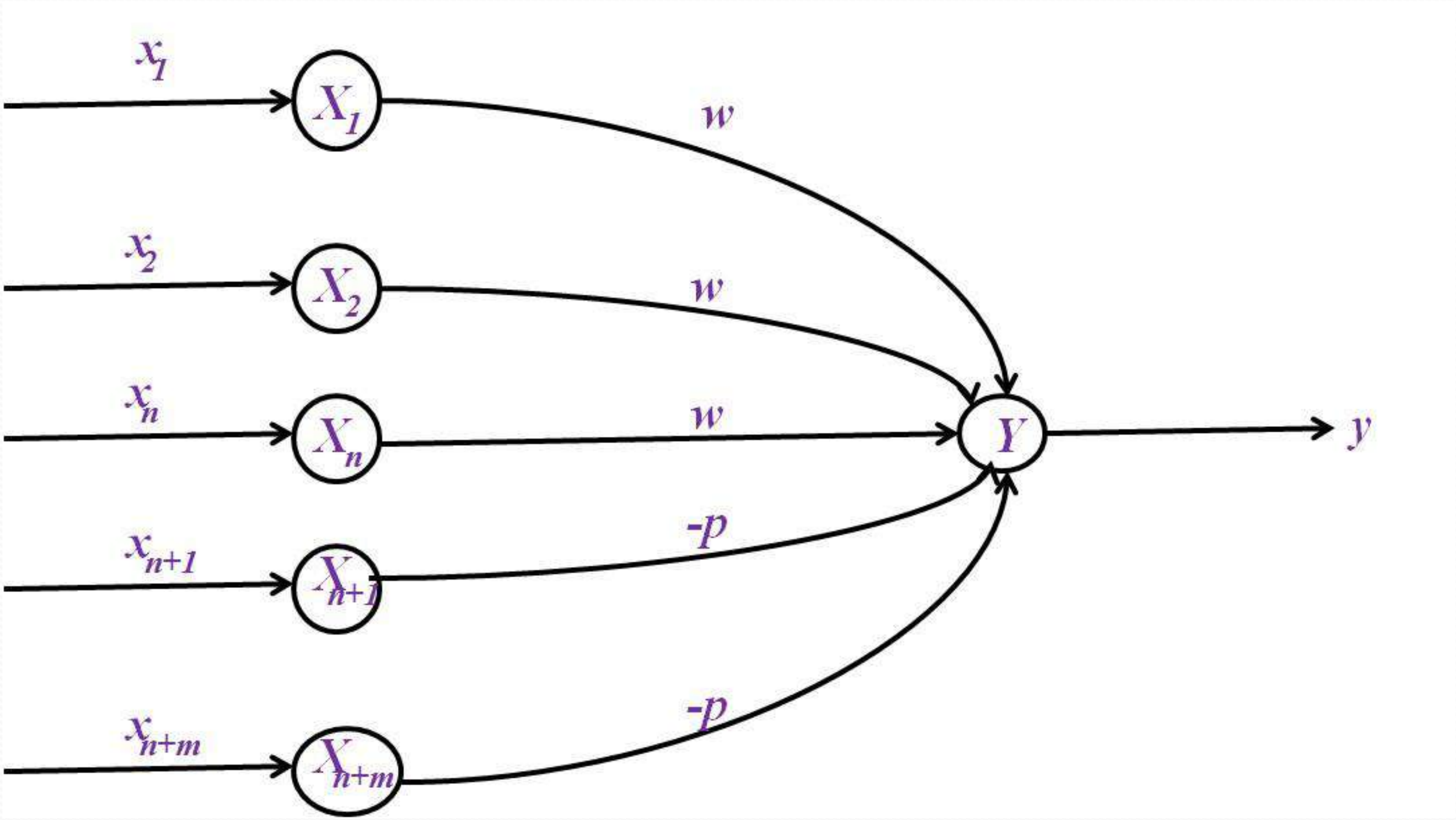
- Firing state is binary (1 = firing, 0 = not firing)

- All but one neuron are excitatory (tend to increase voltage of other cells)

  - One inhibitory neuron connects to all other neurons
  - It functions to regulate network activity (prevent too many firings)

McCulloch Pitts Neuron

Architecture



Firing of the neuron is based on the threshold. Activation function is applied on net input to get the output which is $y= f(y_{in})$.

Activation function is defined as

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

Threshold is obtained by satisfying the following condition

$$\theta > nw - p$$

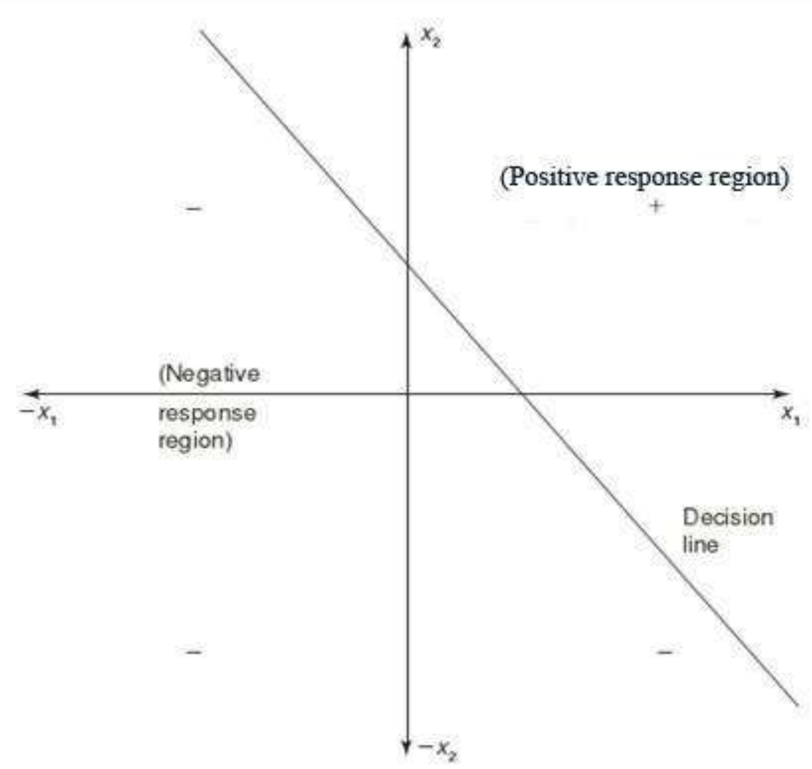
Neuron will fire if it receives k or more excitatory input but no inhibitory input where,

$$kw \geq \theta > (k - 1)w$$

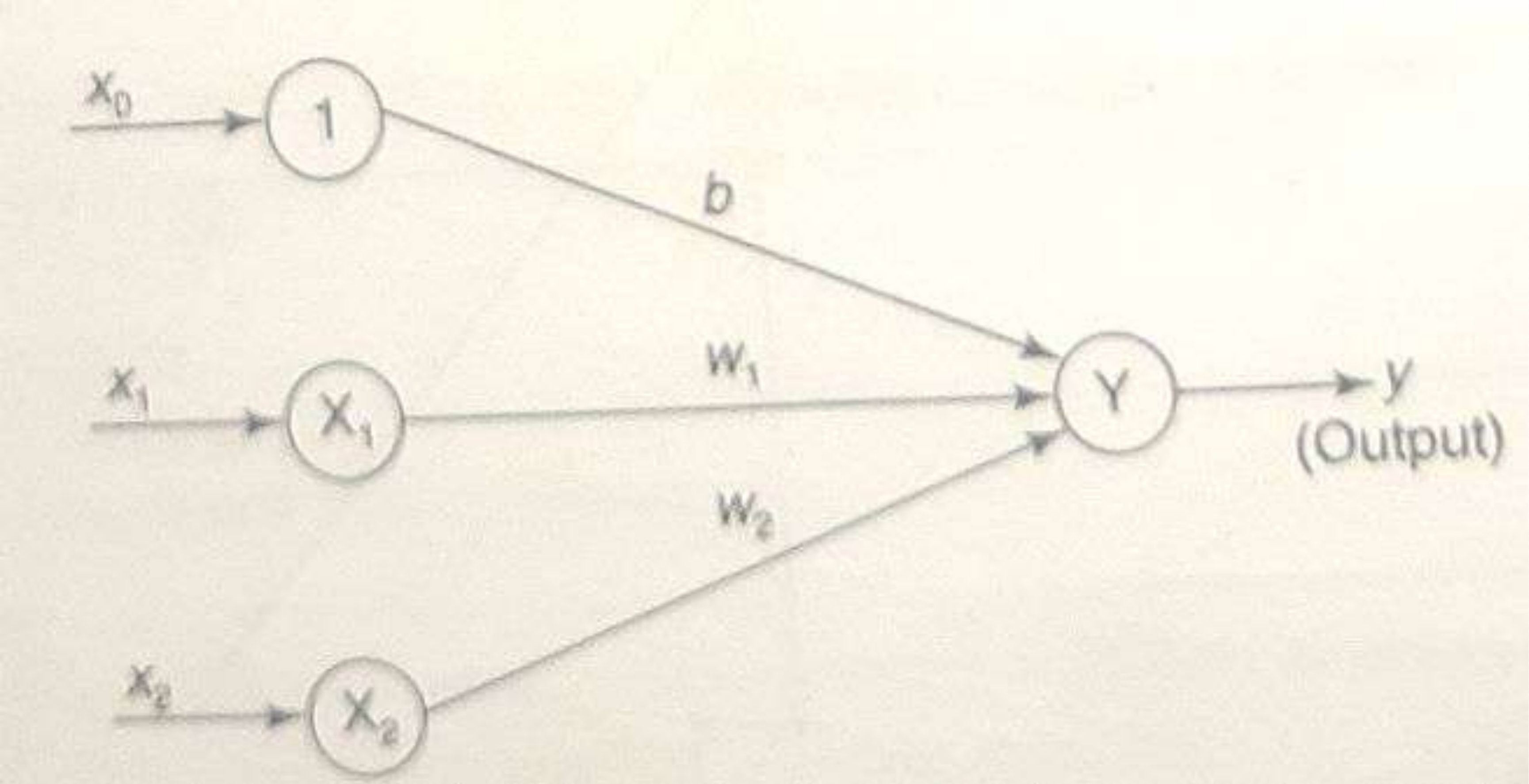
Linear Separability

- Linear separability is the concept wherein the separation of the input space into regions is based on whether the network response is positive or negative.

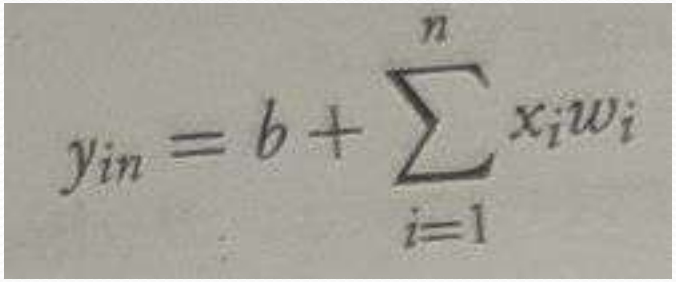
- Consider a network having positive response in the first quadrant and negative response in all other quadrants (AND function) with either binary or bipolar data, then the decision line is drawn separating the positive response region from the negative response region.



A single layer NN

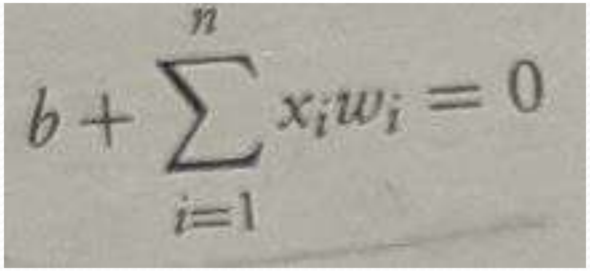


Generally, the net input calculated to the output unit is



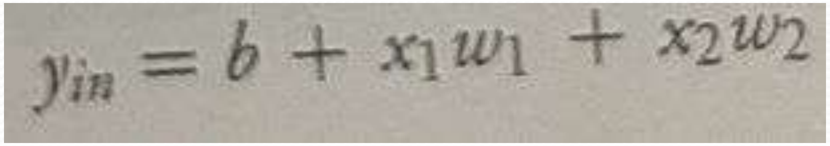
$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

The decision boundary is determined by the equation



$$b + \sum_{i=1}^{n} x_i w_i = 0$$

The net input for the network in Fig. shown with bias is

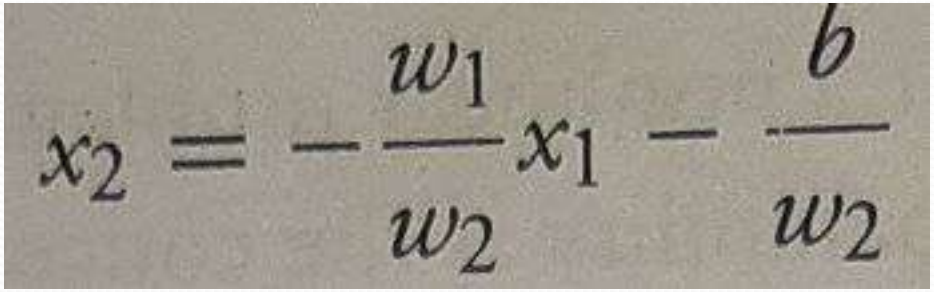


$$y_{in} = b + x_1 w_1 + x_2 w_2$$

The decision boundary is determined by the equation

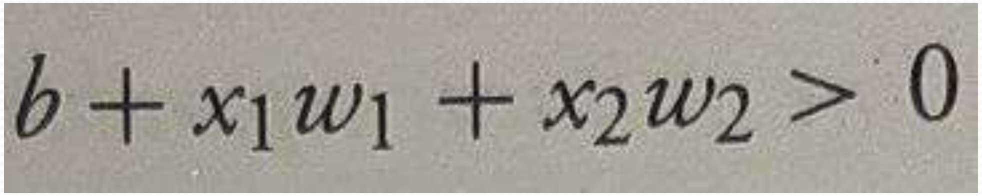
$$b + x_1 w_1 + x_2 w_2 = 0$$

X2 is calculated as



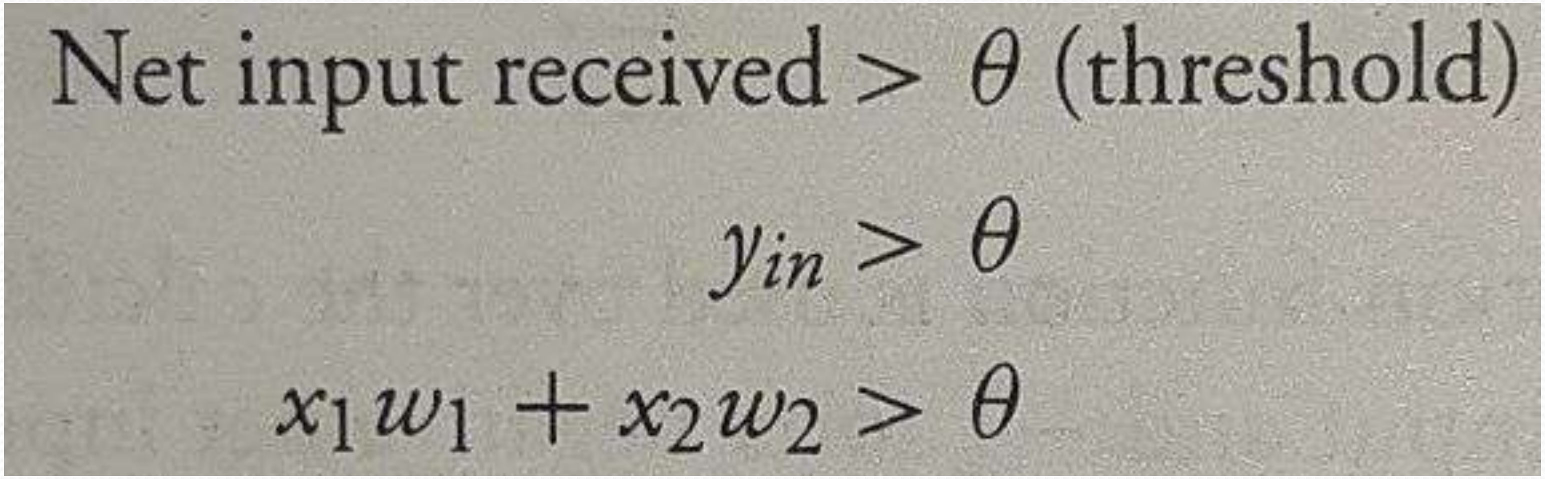
$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{b}{w_2}$$

Requirement for a positive response of the net if bias is used



$$b + x_1w_1 + x_2w_2 > 0$$

If threshold is used, Requirement for a positive response is

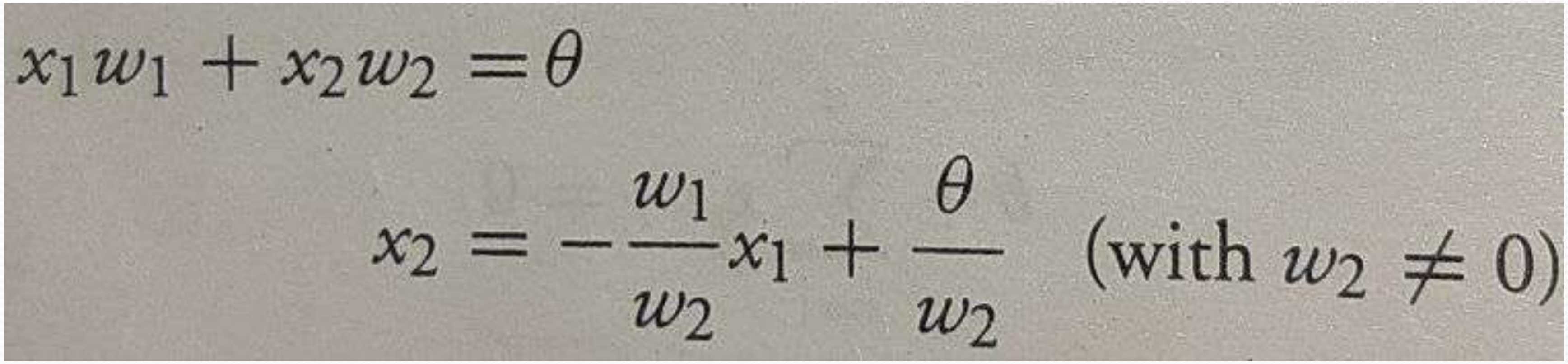


Net input received > θ (threshold)

$$y_{in} > \theta$$

$$x_1w_1 + x_2w_2 > \theta$$

The separating line equation will be



$$x_1 w_1 + x_2 w_2 = \theta$$

$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{\theta}{w_2} \quad (\text{with } w_2 \neq 0)$$

HEBB NETWORK

Donald Hebb stated in 1949 that in the brain, the learning is performed by the change in the synaptic gap. Hebb explained it:

"When an axon of cell A is near enough to excite cell B, and repeatedly or permanently takes place in firing it, some growth process or metabolic change takes place in one or both the cells such that A's efficiency, as one of the cells firing B, is increased."

HEBB LEARNING

- The weights between neurons whose activities are positively correlated are increased:

$$\frac{dw_{ij}}{dt} \sim \text{correlation}(x_i , x_j)$$

- Associative memory is produced automatically

- The Hebb rule can be used for pattern association, pattern categorization, pattern classification and over a range of other areas.

Hebb Network

Theory

- The weight vector is found to increase proportionally to the product of the input and learning signal(learning signal is equal to the neuron◦s output).
- In Hebb learning, if the two interconnected neurons are *on* simultaneously, then the weight associated with these neu- rons can be increased by the modification made in their weight.

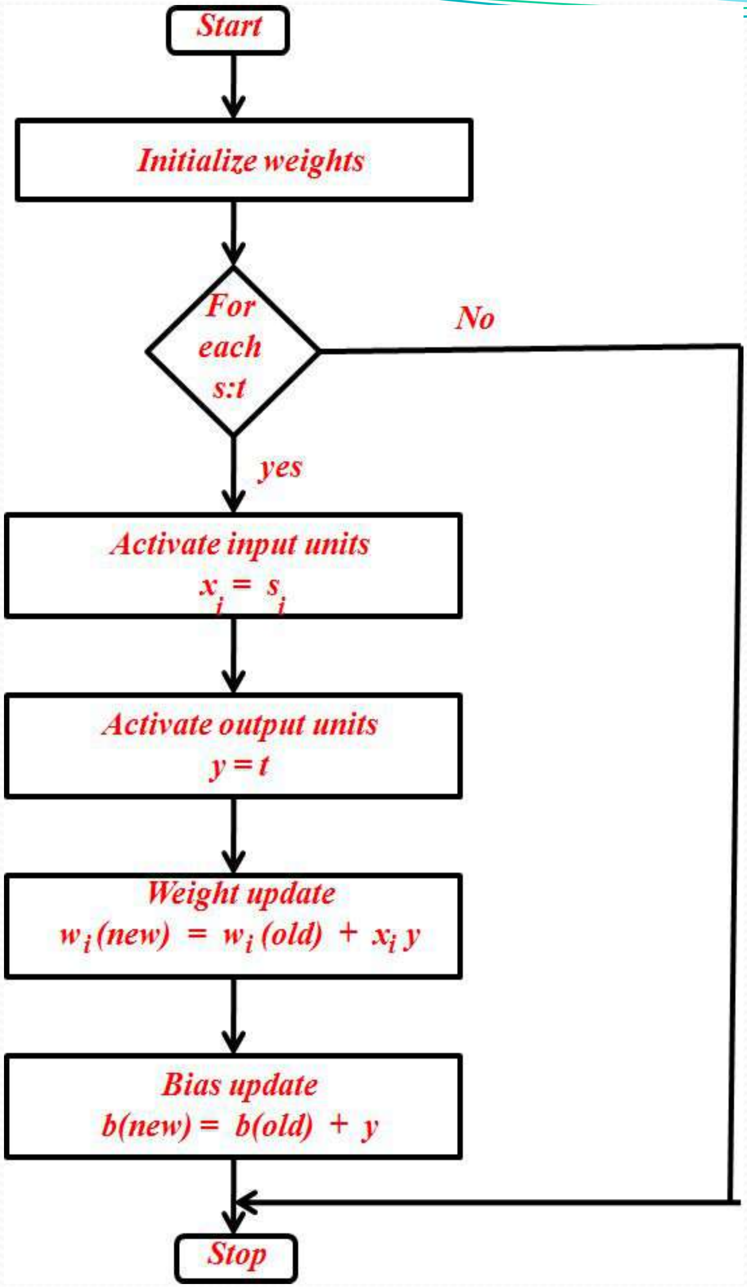
The weight update is given by,

- $$w_i(new) = w_i(old) + x_i y$$

The Hebb rule is more suited for bipolar data than binary data.

-

Flowchart of Hebb training algorithm



Training Algorithm

- *Step 0* : Initialize the weights.

$$w_i = 0 \text{ for } i = 1 \text{ to } n$$

- *Step 1* : Steps 2 4 have to be performed for each input training vector and target output pair, $s : t$.

- *Step 2* : Input units activations are set.

$$x_i = s_i \text{ for } i = 1 \text{ to } n$$ *Step 3* :

- Output units activations are set.$y = t$

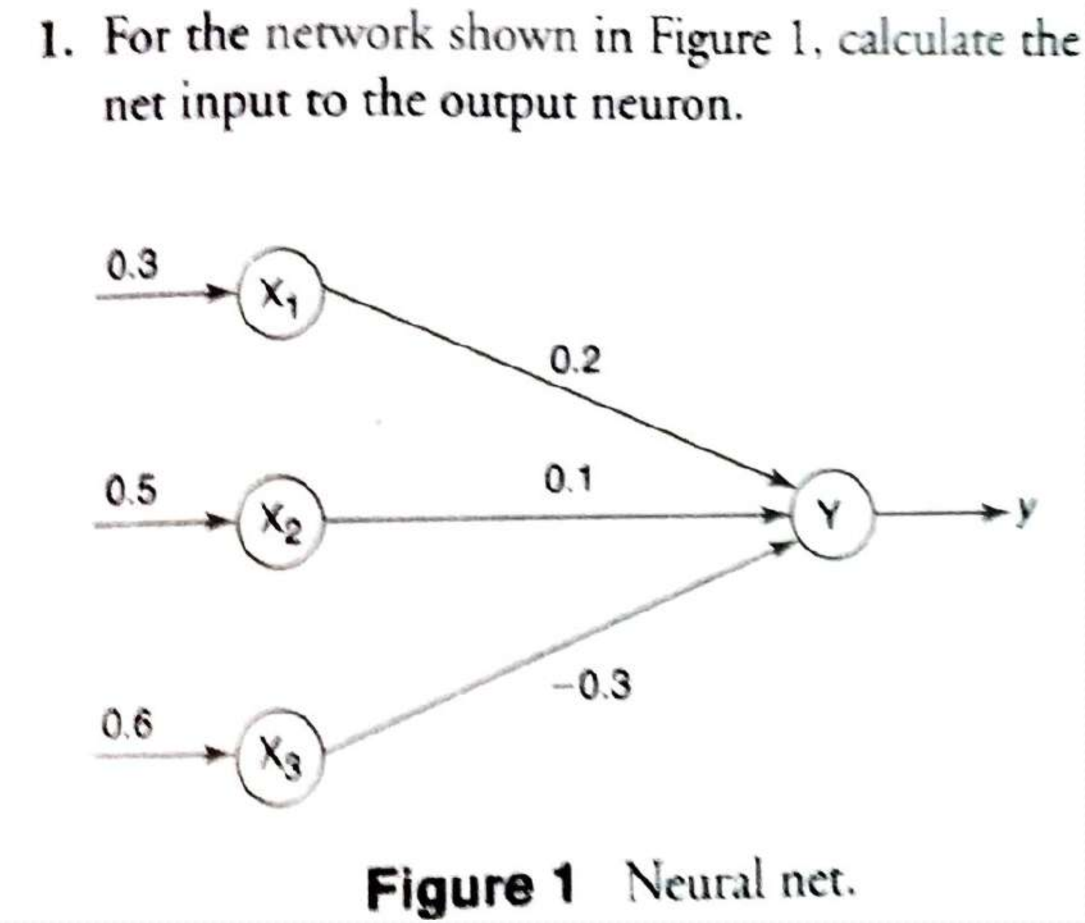
- *Step 4* : Weight adjustments and bias adjustments are per- formed.

$$w_i(new) = w_i(old) + x_i y$$

$$b(new) = b(old) + y$$ Change

in weight, $w = xy$

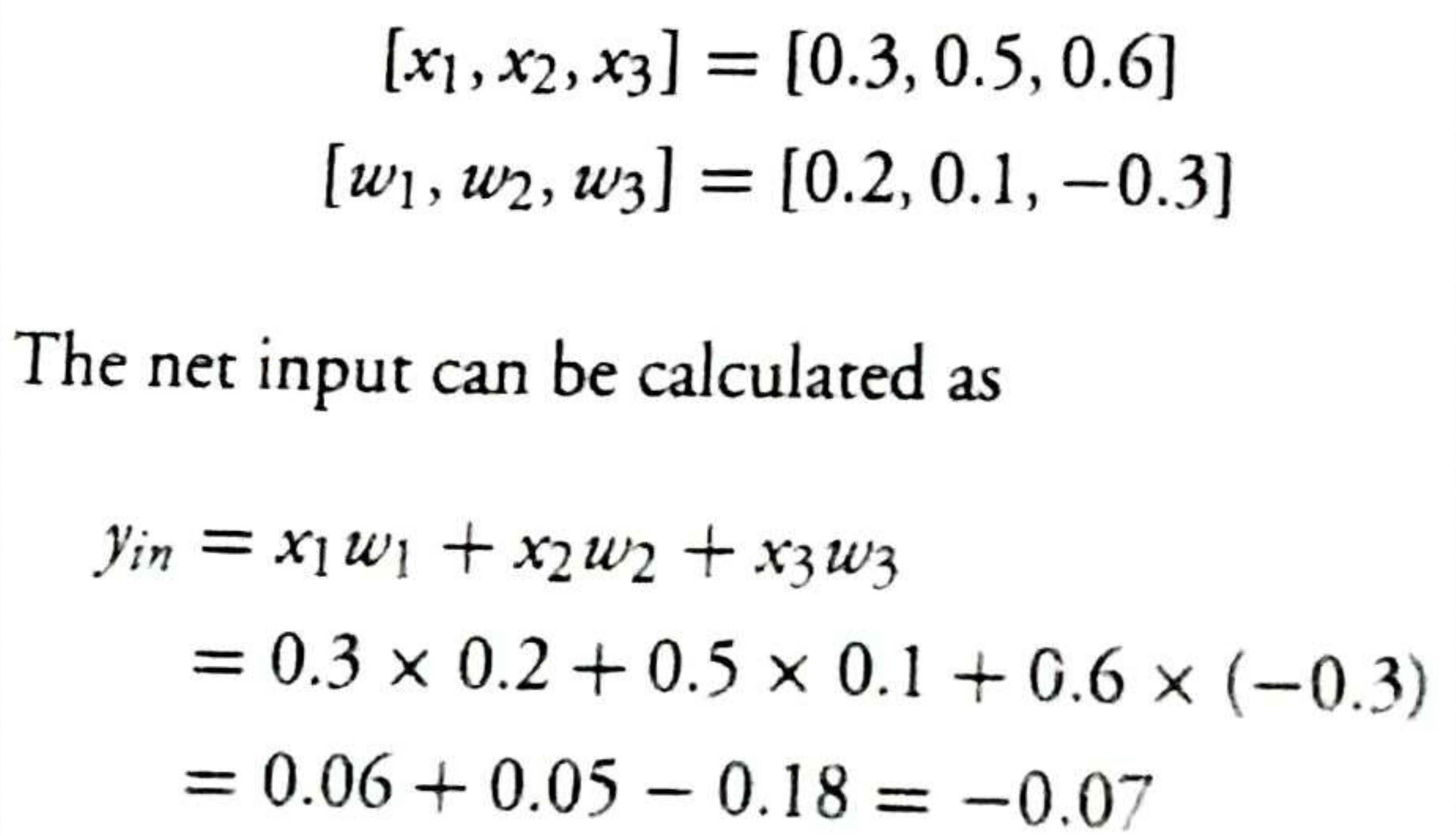
Problems



1. For the network shown in Figure 1, calculate the net input to the output neuron.

Figure 1 Neural net.

Solution: **The given NN consists of 3 i/p neurons and 1 o/p neuron .The i/p s and weights are**



$$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$$

$$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$$

The net input can be calculated as

$$\begin{aligned} y_{in} &= x_1 w_1 + x_2 w_2 + x_3 w_3 \\ &= 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times (-0.3) \\ &= 0.06 + 0.05 - 0.18 = -0.07 \end{aligned}$$

2. Calculate the net input for the network shown in Figure 2 with bias included in the network.

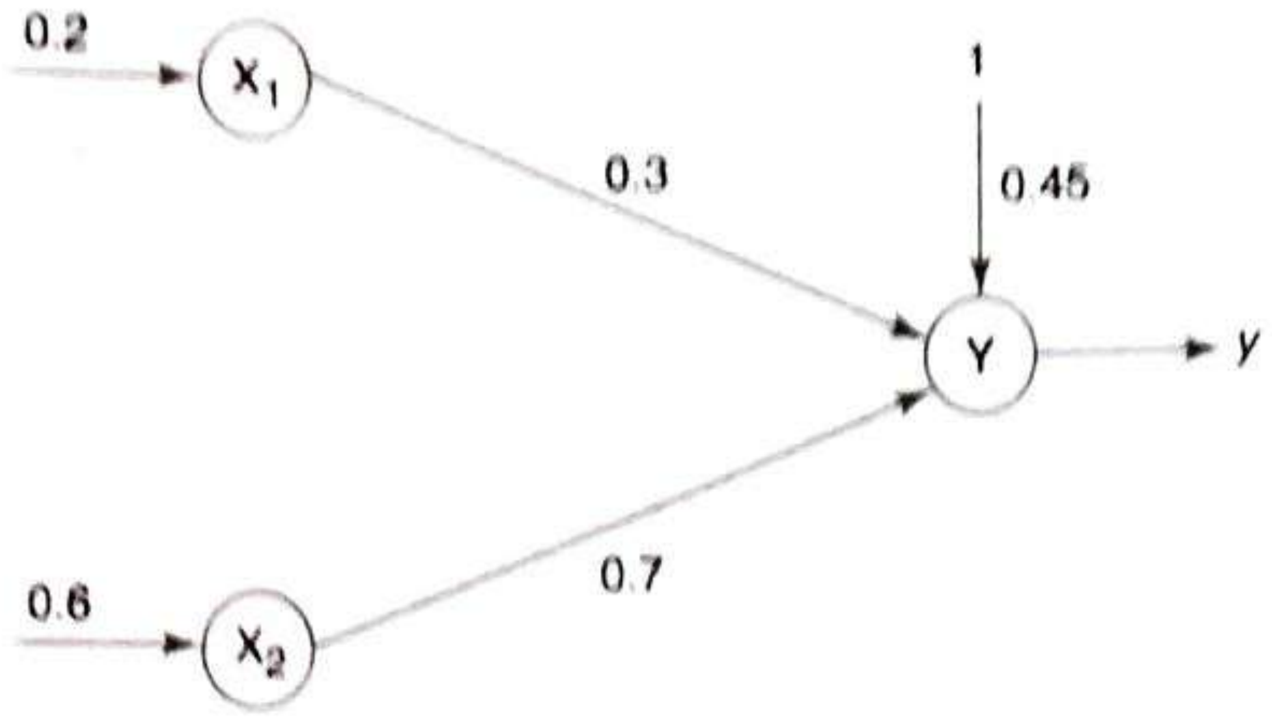
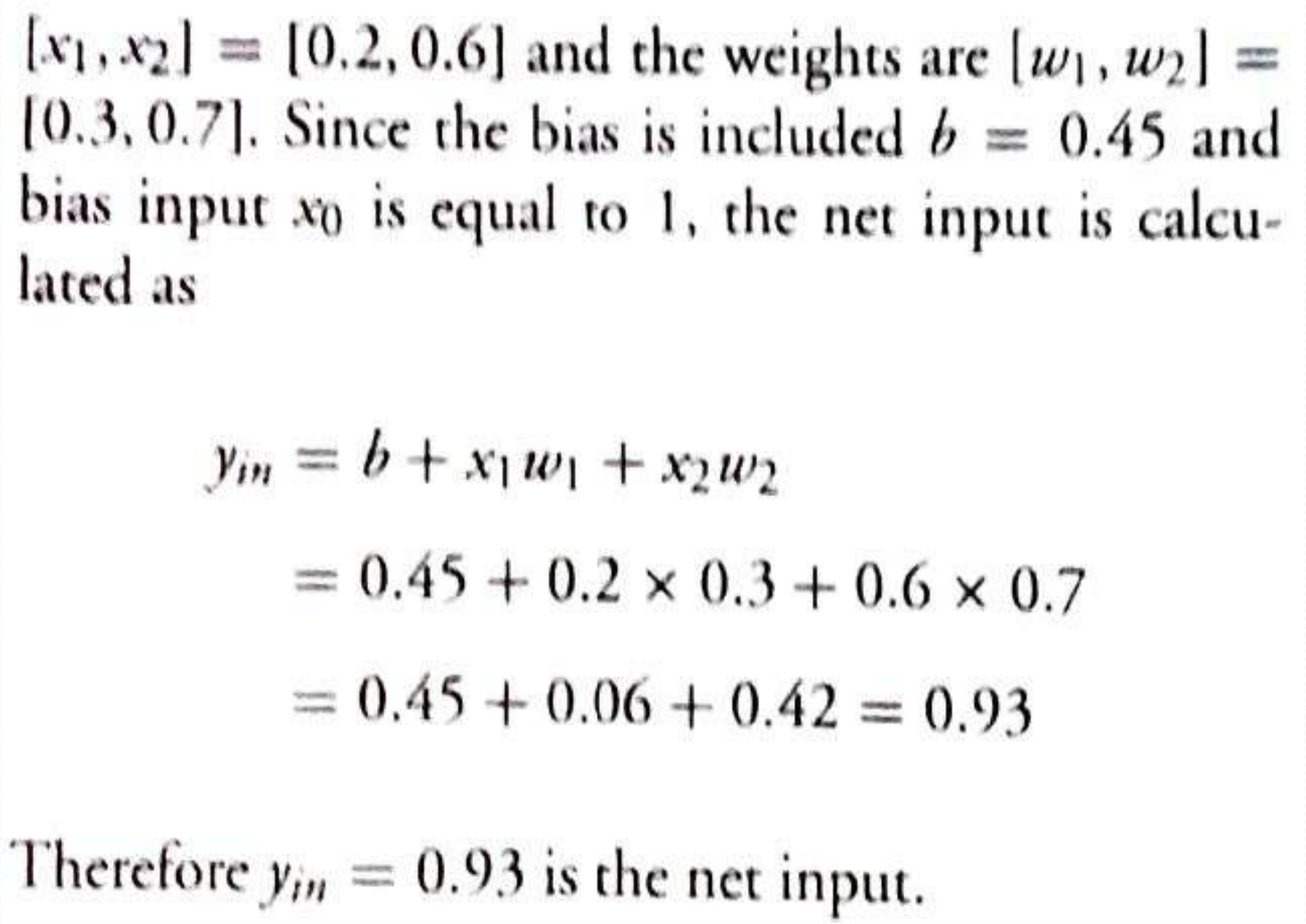


Figure 2 Simple neural net.

Solution: **The given NN consists of 2 i/p neurons, bias and 1 o/p neuron .The i/p s and weights are given as**



$[x_1, x_2] = [0.2, 0.6]$ and the weights are $[w_1, w_2] = [0.3, 0.7]$. Since the bias is included $b = 0.45$ and bias input x_0 is equal to 1, the net input is calculated as

$$
\begin{aligned}
y_{in} &= b + x_1 w_1 + x_2 w_2 \\
&= 0.45 + 0.2 \times 0.3 + 0.6 \times 0.7 \\
&= 0.45 + 0.06 + 0.42 = 0.93
\end{aligned}
$$

Therefore $y_{in} = 0.93$ is the net input.

3. Obtain the output of the neuron Y for the network shown in Figure 3 using activation functions as: (i) binary sigmoidal and (ii) bipolar sigmoidal.

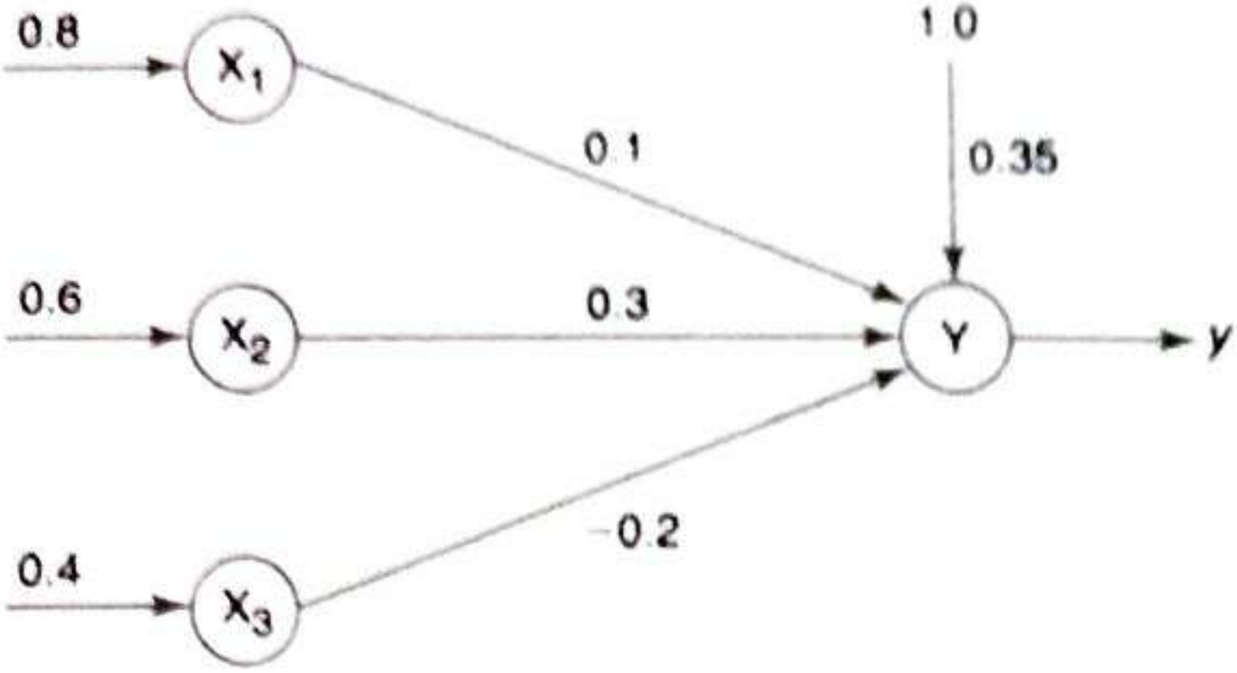
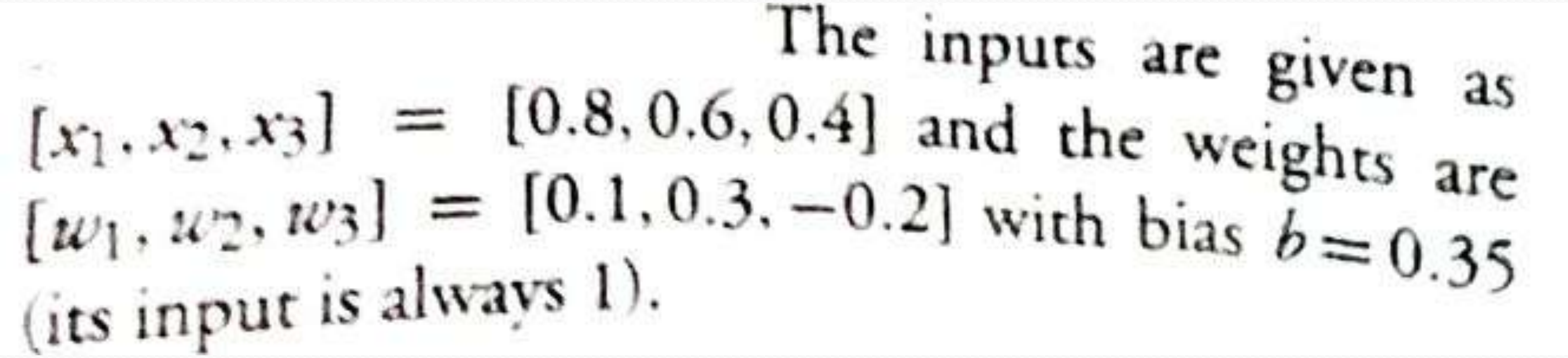
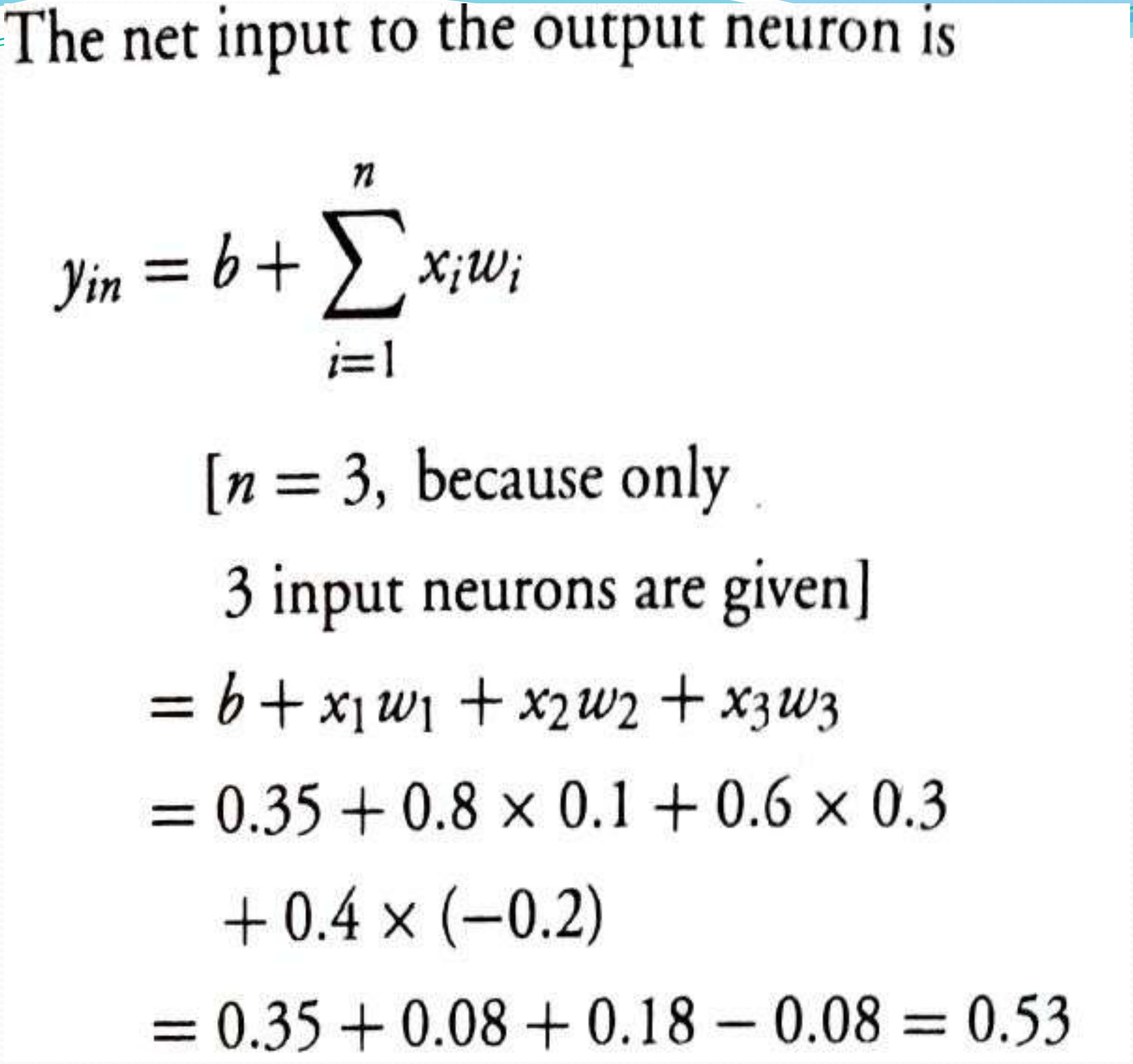


Figure 3 Neural net.

Solution: **The given NN consists of 3 i/p neurons, bias and 1 o/p neuron .The i/p s and weights are given as**



The inputs are given as $[x_1, x_2, x_3] = [0.8, 0.6, 0.4]$ and the weights are $[w_1, w_2, w_3] = [0.1, 0.3, -0.2]$ with bias $b = 0.35$ (its input is always 1).



The net input to the output neuron is

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

$[n = 3$, because only

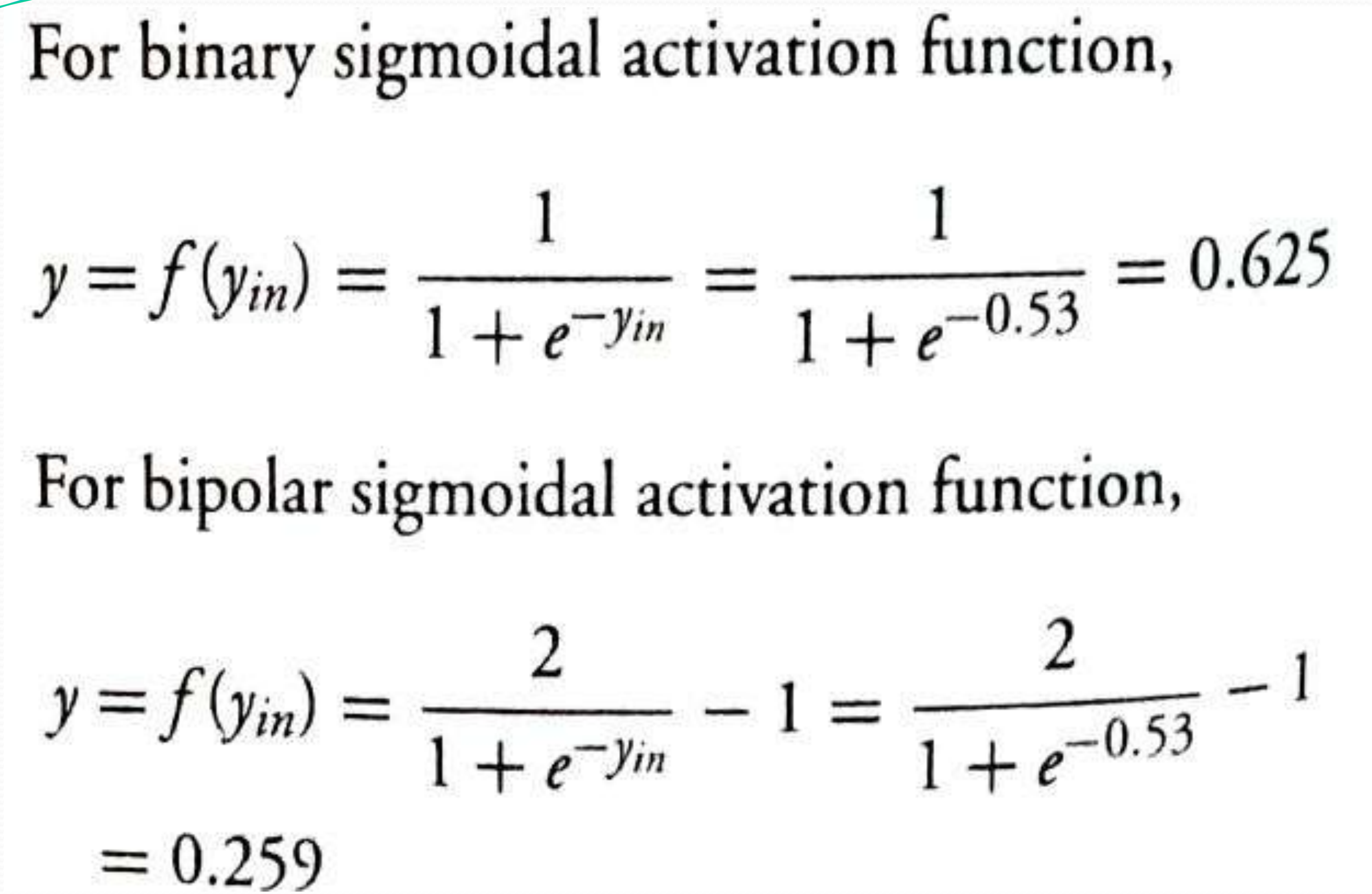
3 input neurons are given]

$$= b + x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3$$

$$+ 0.4 \times (-0.2)$$

$$= 0.35 + 0.08 + 0.18 - 0.08 = 0.53$$



For binary sigmoidal activation function,

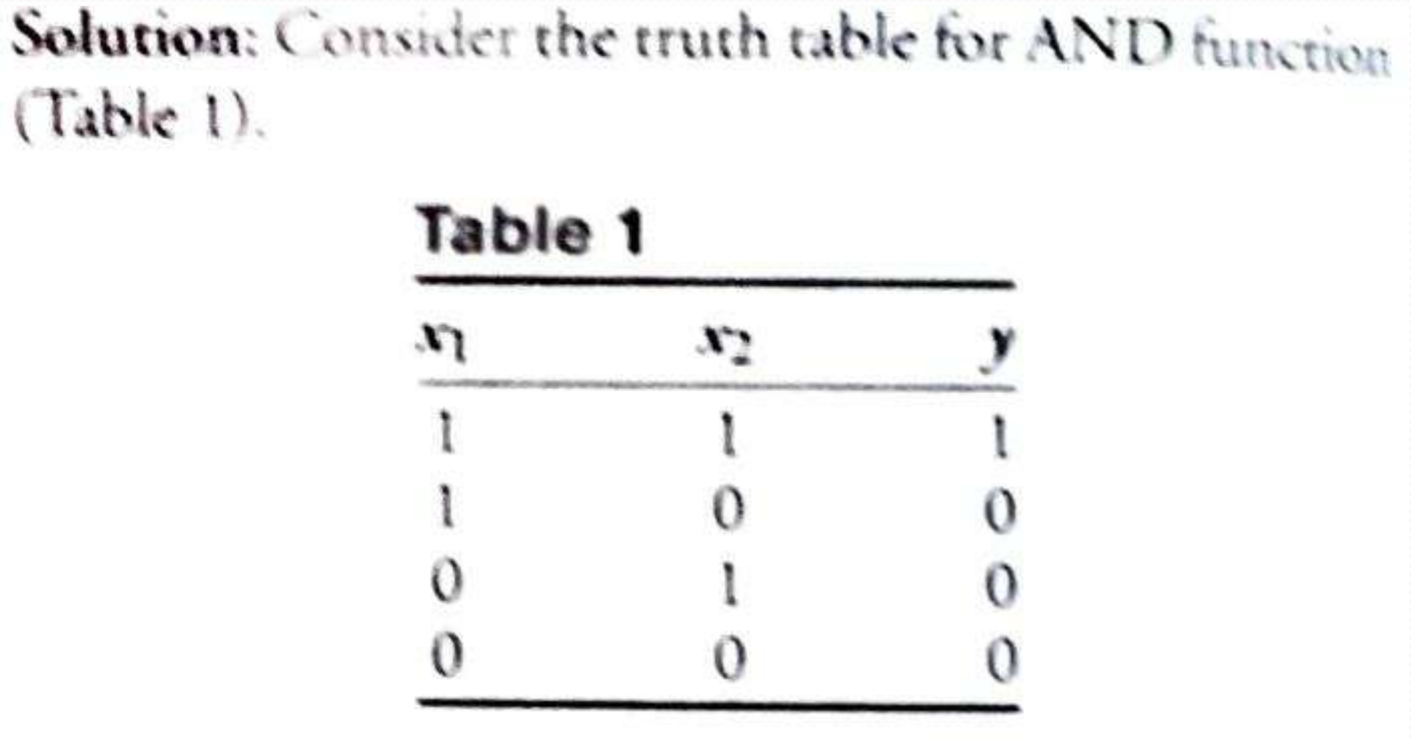
$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.53}} = 0.625$$

For bipolar sigmoidal activation function,

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1 = \frac{2}{1+e^{-0.53}} - 1$$

$$= 0.259$$

4. **Implement AND function using McCulloch-Pitts Neuron (Take Binary data)**



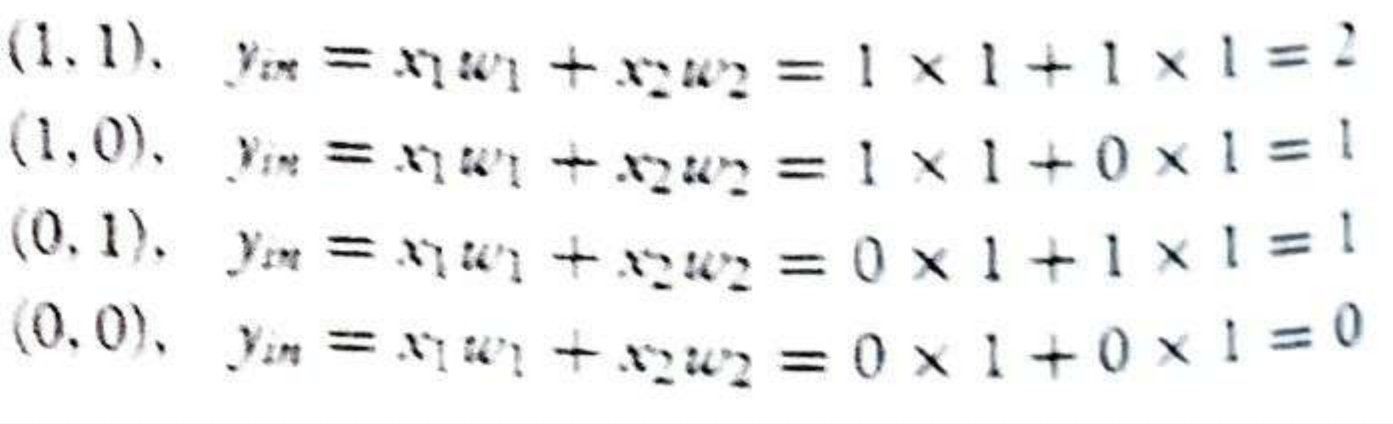
Solution: Consider the truth table for AND function (Table 1).

Table 1

x_1	x_2	y
1	1	1
1	0	0
0	1	0
0	0	0

For each input pair, calculate net input y_{in}

Assume the weights be w1=w2=1



$(1, 1)$, $y_{in} = x_1 w_1 + x_2 w_2 = 1 \times 1 + 1 \times 1 = 2$

$(1, 0)$, $y_{in} = x_1 w_1 + x_2 w_2 = 1 \times 1 + 0 \times 1 = 1$

$(0, 1)$, $y_{in} = x_1 w_1 + x_2 w_2 = 0 \times 1 + 1 \times 1 = 1$

$(0, 0)$, $y_{in} = x_1 w_1 + x_2 w_2 = 0 \times 1 + 0 \times 1 = 0$

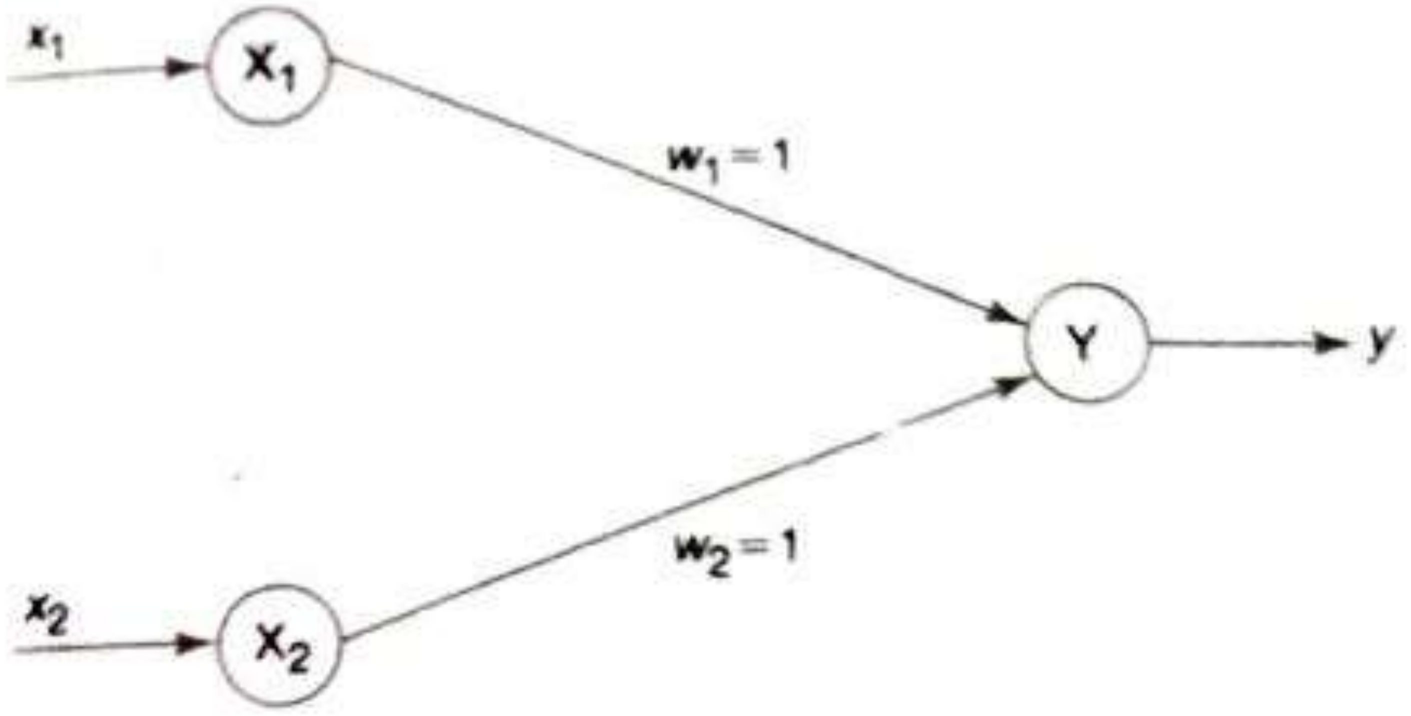
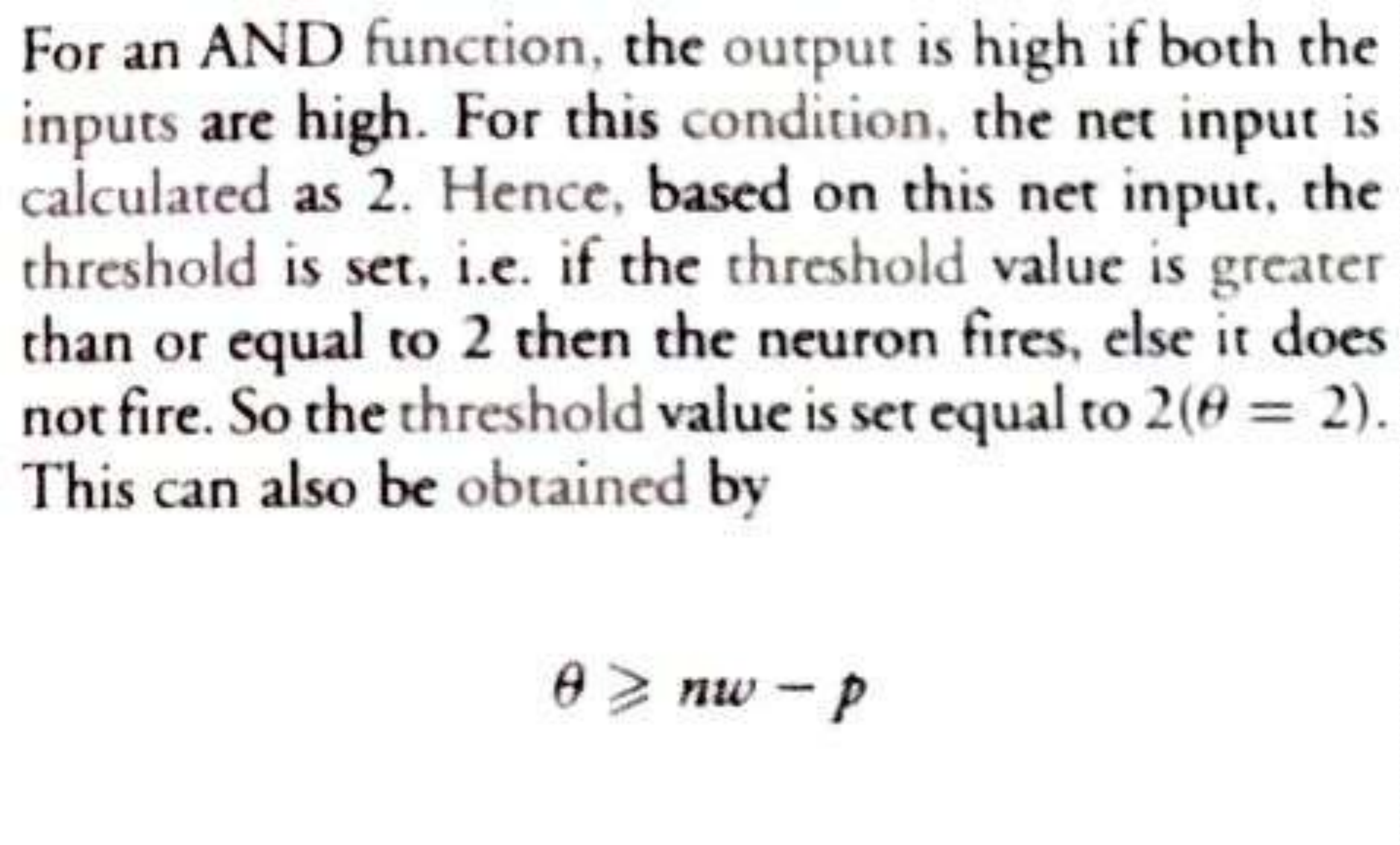
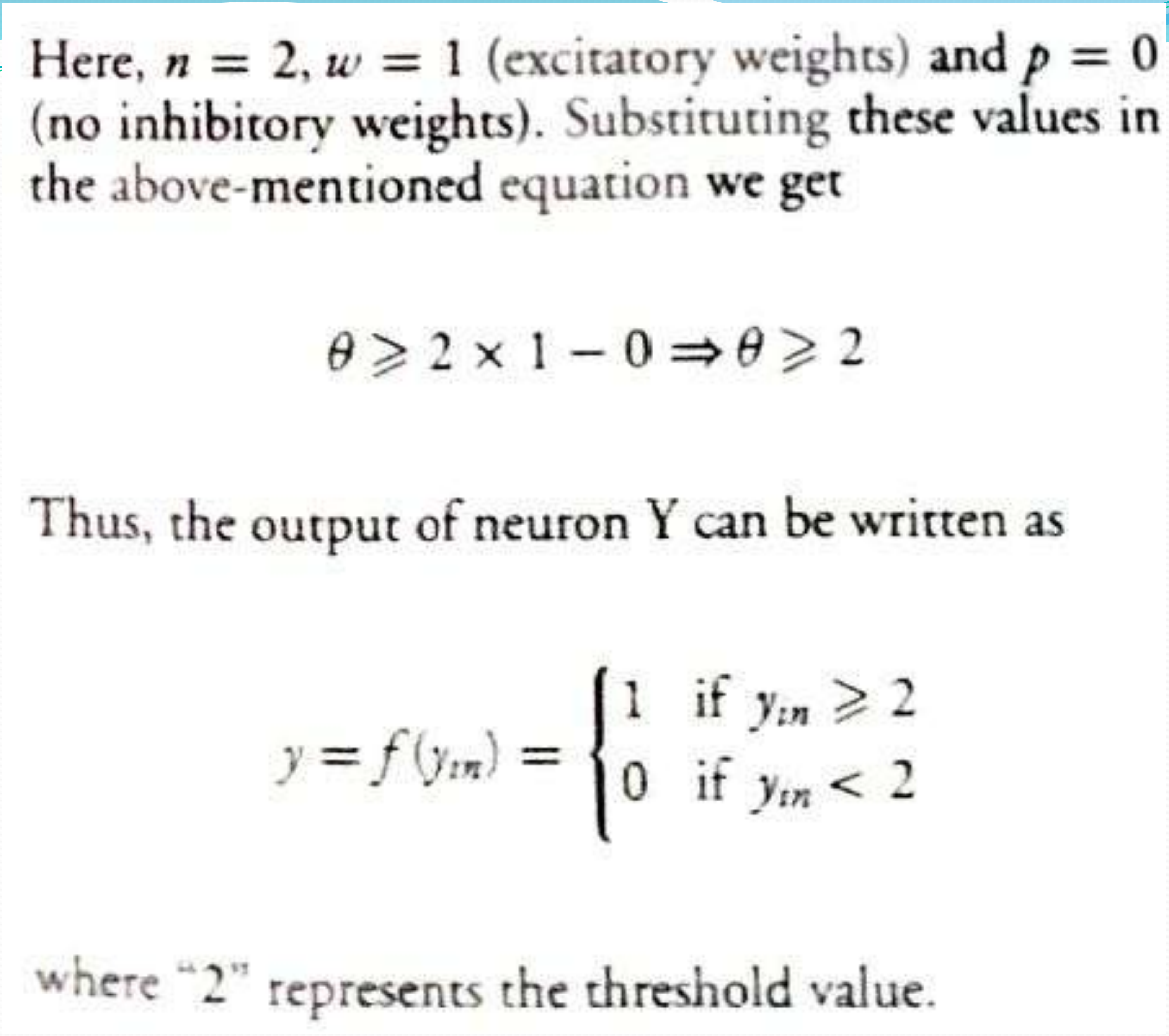


Figure 4 Neural net.



For an AND function, the output is high if both the inputs are high. For this condition, the net input is calculated as 2. Hence, based on this net input, the threshold is set, i.e. if the threshold value is greater than or equal to 2 then the neuron fires, else it does not fire. So the threshold value is set equal to 2($\theta = 2$). This can also be obtained by

$$\theta \geqslant nw - p$$



Here, $n = 2$, $w = 1$ (excitatory weights) and $p = 0$ (no inhibitory weights). Substituting these values in the above-mentioned equation we get

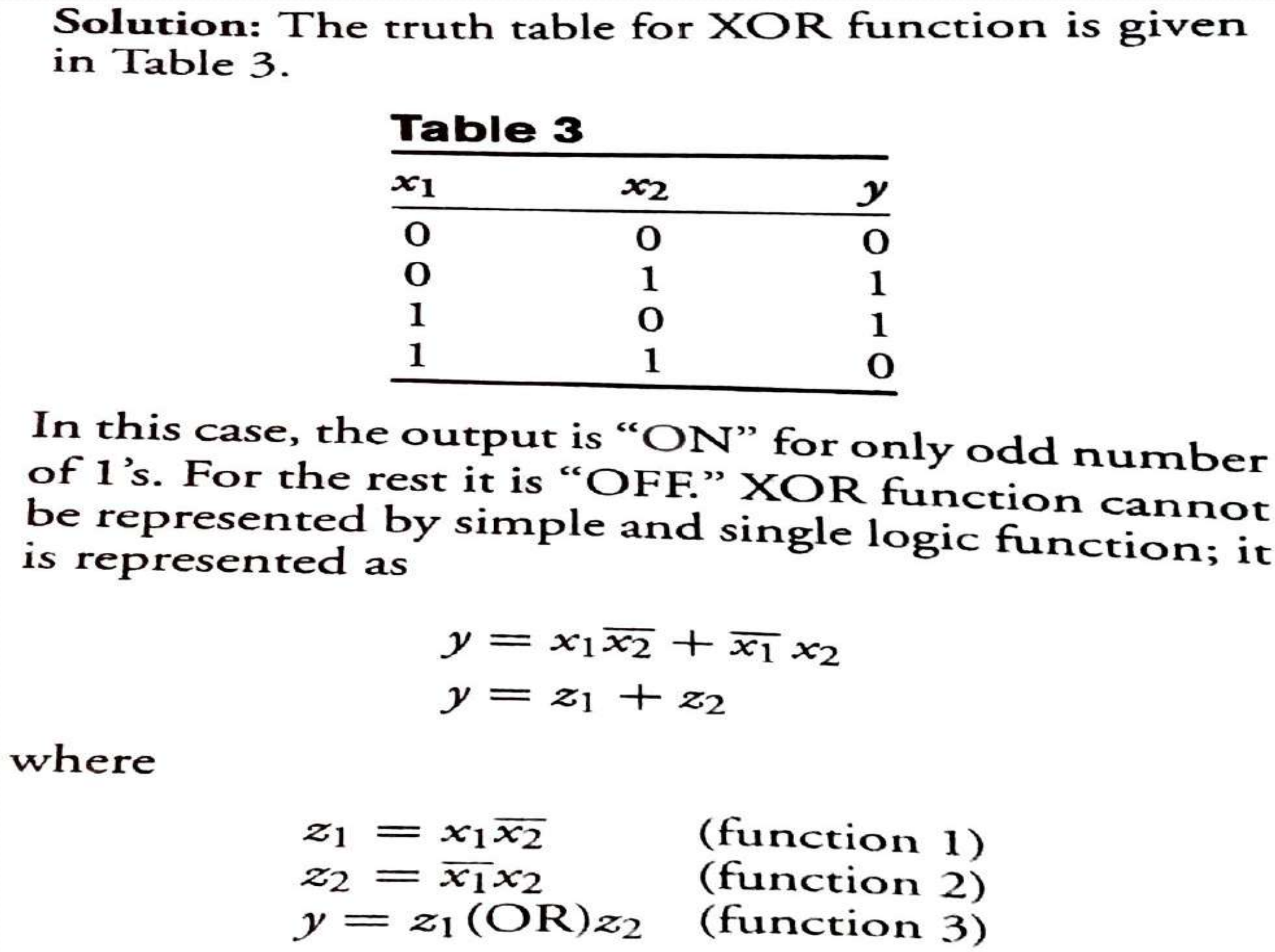
$$\theta \geqslant 2 \times 1 - 0 \Rightarrow \theta \geqslant 2$$

Thus, the output of neuron Y can be written as

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geqslant 2 \\ 0 & \text{if } y_{in} < 2 \end{cases}$$

where "2" represents the threshold value.

5. **Implement XOR function using McCulloch-Pitts Neuron (Take Binary data)**



Solution: The truth table for XOR function is given in Table 3.

Table 3

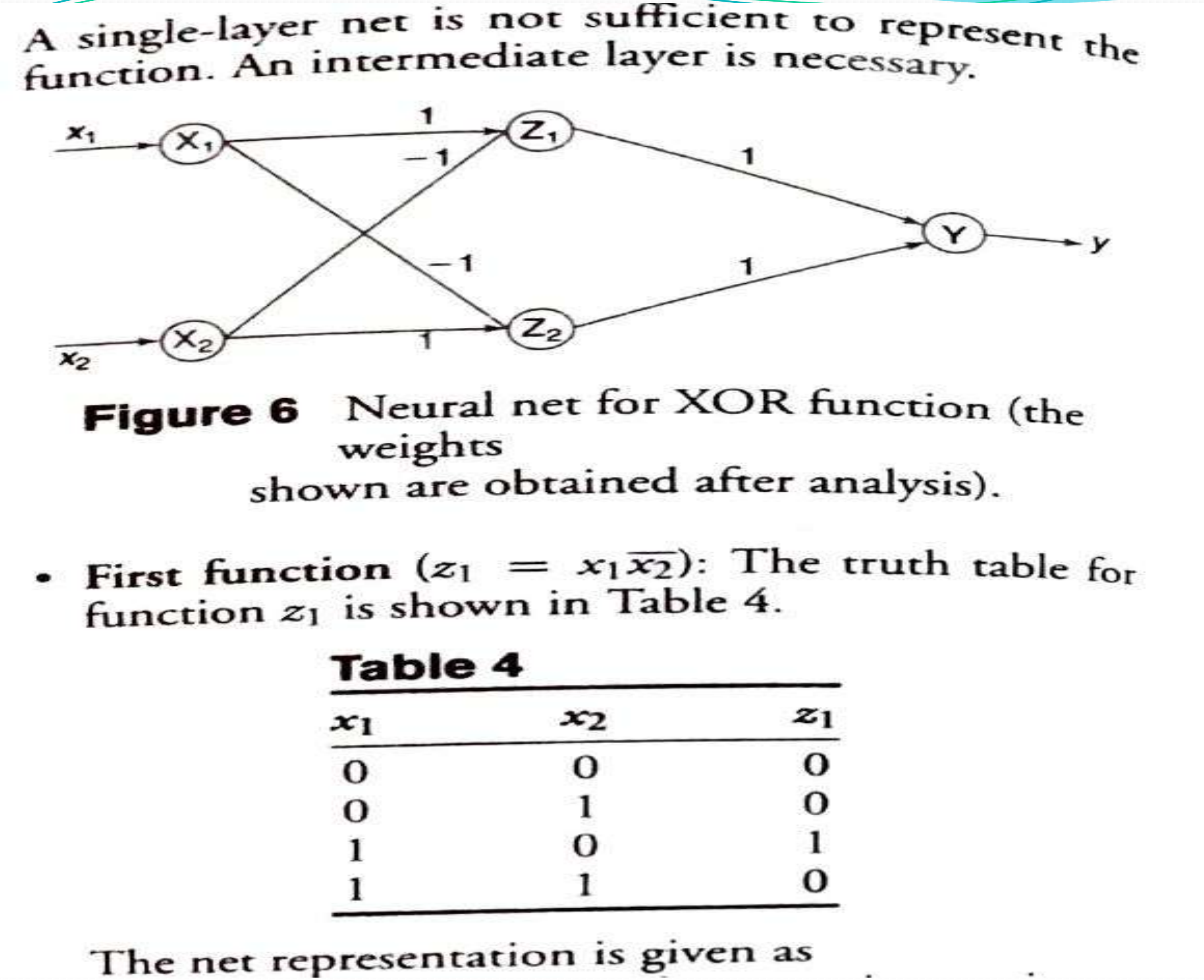
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

In this case, the output is "ON" for only odd number of 1's. For the rest it is "OFF." XOR function cannot be represented by simple and single logic function; it is represented as

$$y = x_1\overline{x_2} + \overline{x_1}\,x_2$$
$$y = z_1 + z_2$$

where

$$z_1 = x_1\overline{x_2} \qquad \text{(function 1)}$$
$$z_2 = \overline{x_1}x_2 \qquad \text{(function 2)}$$
$$y = z_1(\mathrm{OR})z_2 \qquad \text{(function 3)}$$



A single-layer net is not sufficient to represent the function. An intermediate layer is necessary.

x_1
X_1
1
-1
Z_1
1
Y
y
-1
1
x_2
X_2
1
Z_2

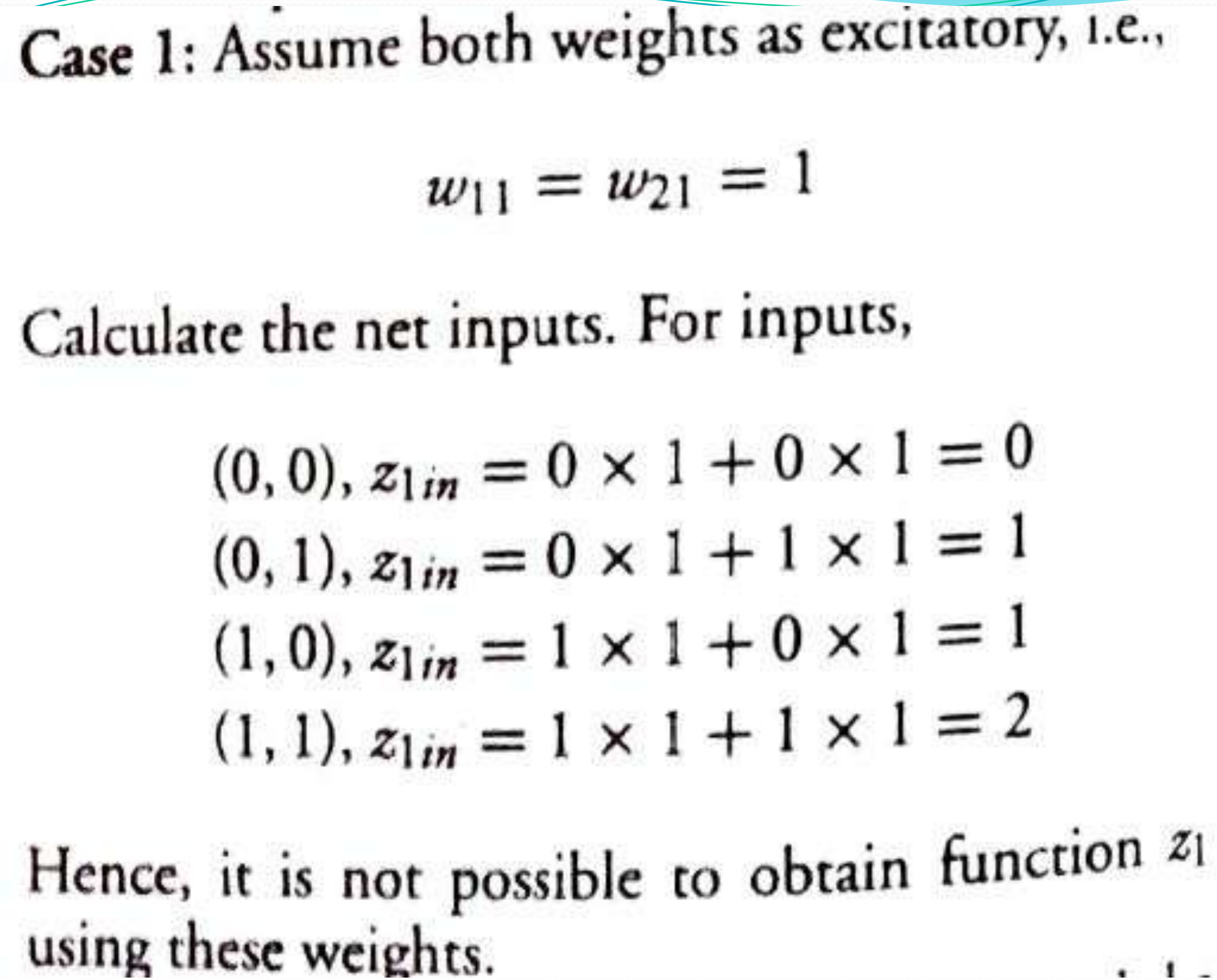
Figure 6 Neural net for XOR function (the weights shown are obtained after analysis).

- **First function** ($z_1 = x_1\overline{x_2}$): The truth table for function z_1 is shown in Table 4.

Table 4

x_1	x_2	z_1
0	0	0
0	1	0
1	0	1
1	1	0

The net representation is given as



Case 1: Assume both weights as excitatory, i.e.,

$$w_{11} = w_{21} = 1$$

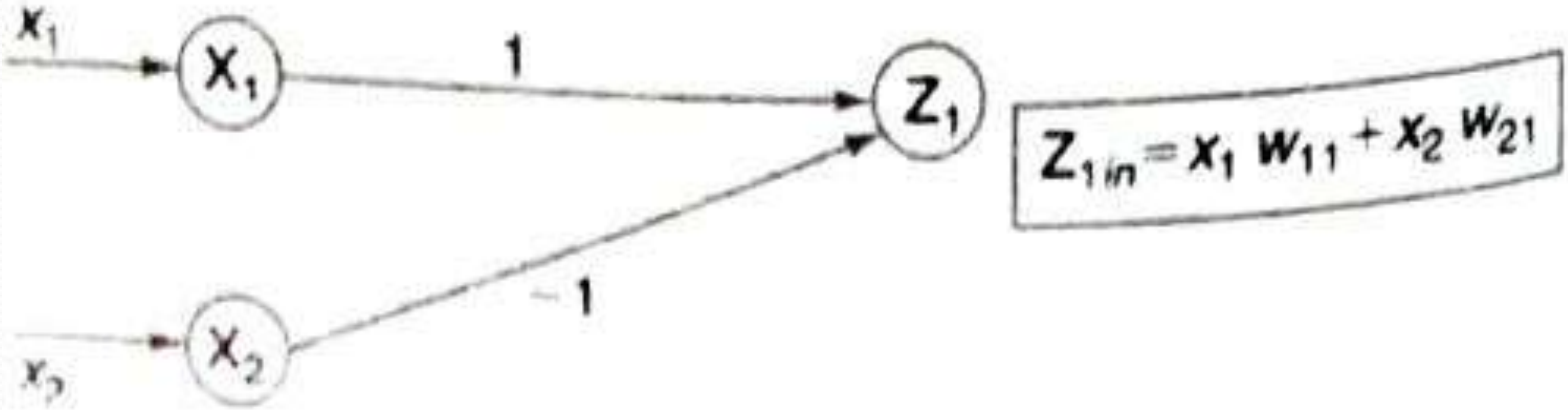
Calculate the net inputs. For inputs,

$$\begin{aligned}
(0, 0),\ z_{1in} &= 0 \times 1 + 0 \times 1 = 0 \\
(0, 1),\ z_{1in} &= 0 \times 1 + 1 \times 1 = 1 \\
(1, 0),\ z_{1in} &= 1 \times 1 + 0 \times 1 = 1 \\
(1, 1),\ z_{1in} &= 1 \times 1 + 1 \times 1 = 2
\end{aligned}$$

Hence, it is not possible to obtain function z_1 using these weights.

Case 2: Assume one weight as excitatory and the other as inhibitory, i.e.,

$$w_{11} = 1; \quad w_{21} = -1$$



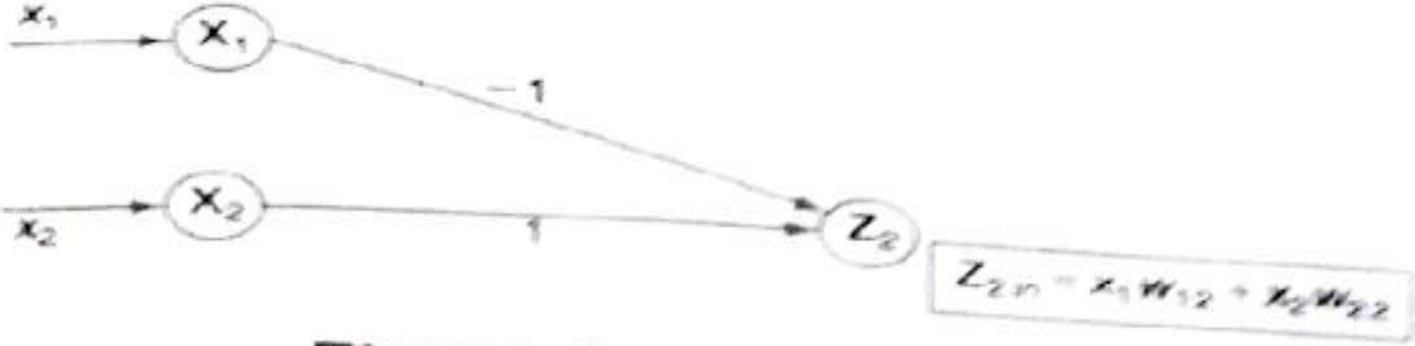


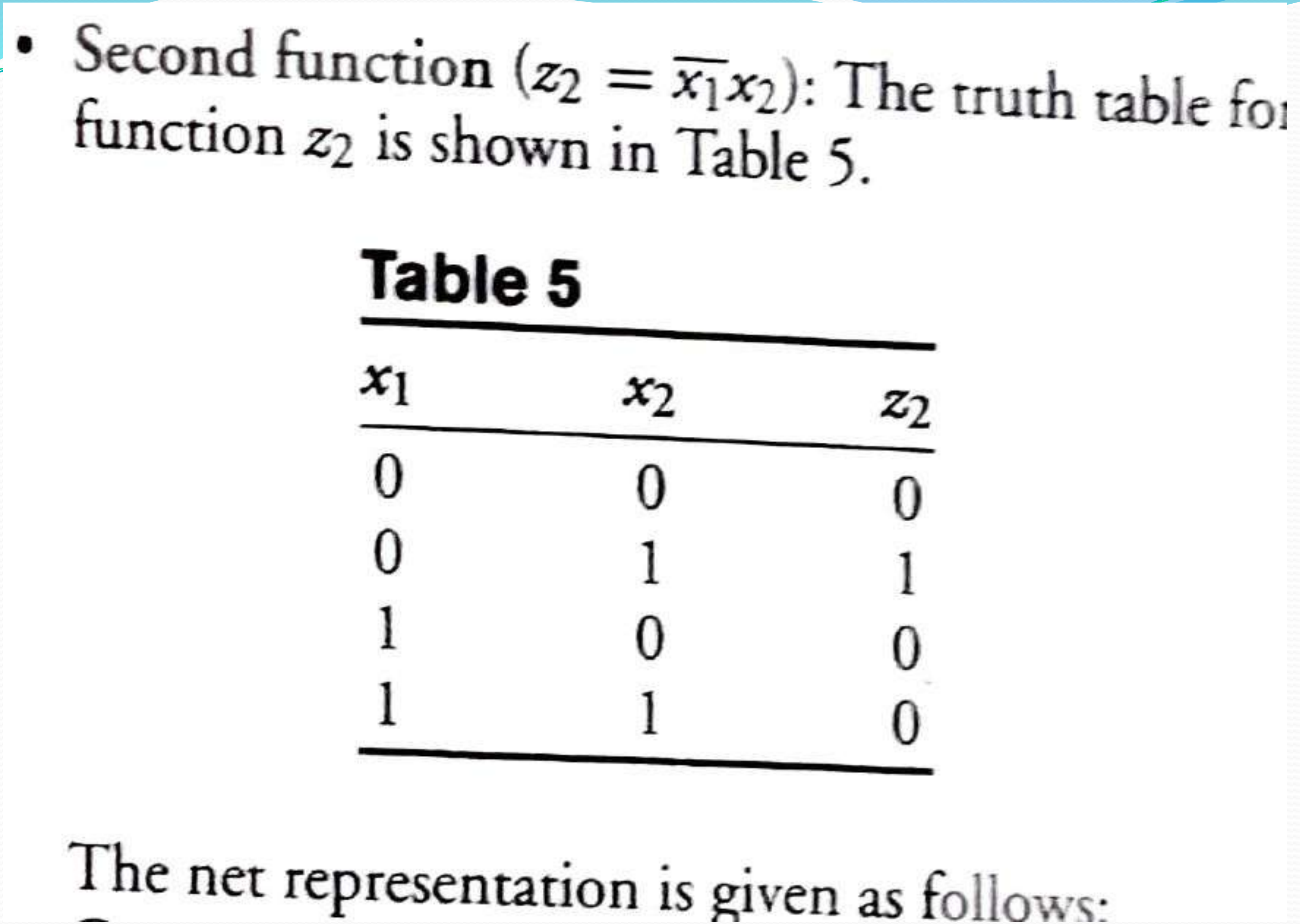
Figure 8 Neural net for Z_2.

Calculate the net inputs. For inputs

$$
\begin{aligned}
&(0, 0), z_{1in} = 0 \times 1 + 0 \times -1 = 0\\
&(0, 1), z_{1in} = 0 \times 1 + 1 \times -1 = -1\\
&(1, 0), z_{1in} = 1 \times 1 + 0 \times -1 = 1\\
&(1, 1), z_{1in} = 1 \times 1 + 1 \times -1 = 0
\end{aligned}
$$

On the basis of this calculated net input, it is possible to get the required output. Hence,

$$
\begin{aligned}
w_{11} &= 1\\
w_{21} &= -1\\
\theta &\geq 1 \quad \text{for the } Z_1 \text{ neuron}
\end{aligned}
$$

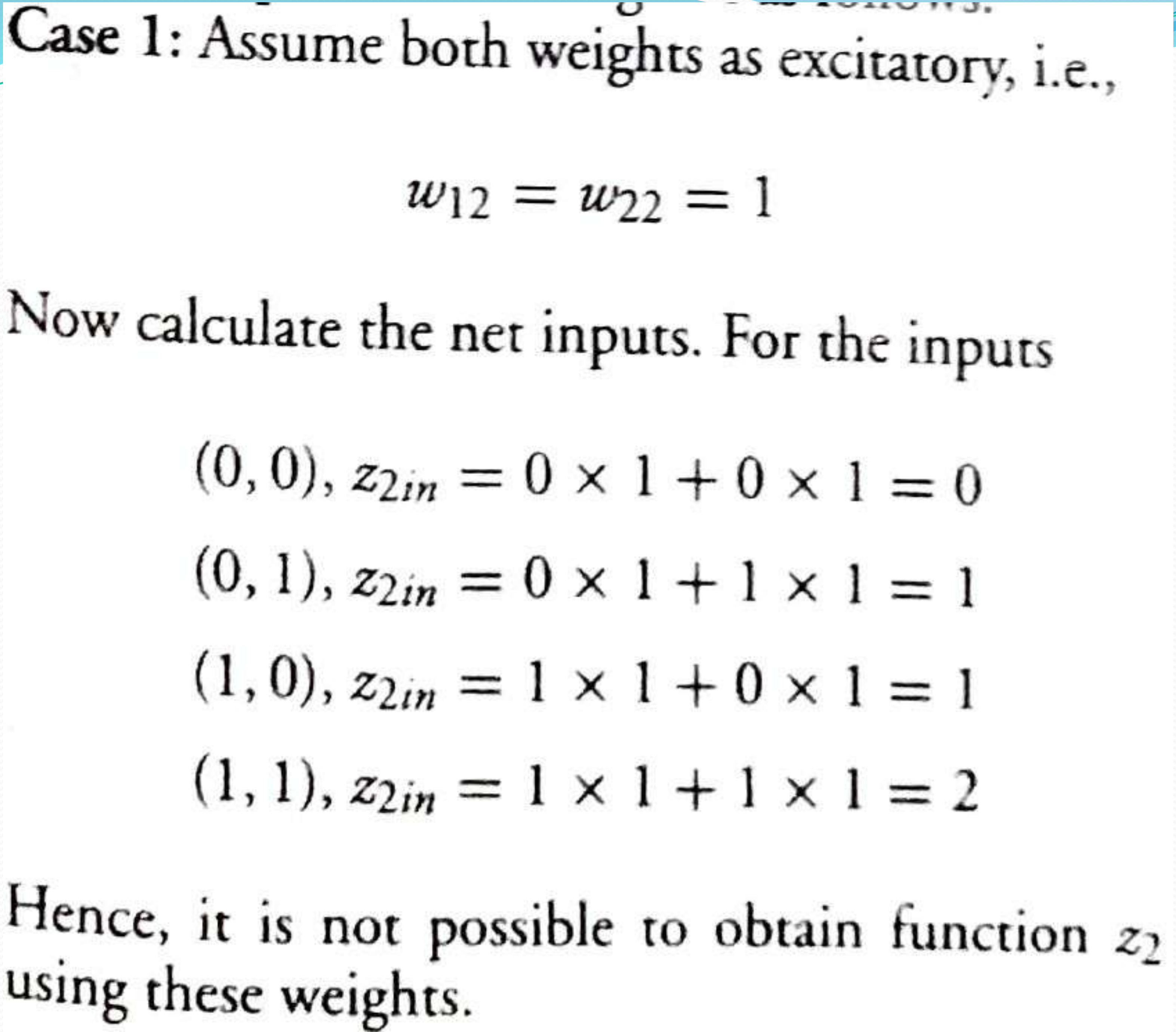


- Second function ($z_2 = \overline{x_1}x_2$): The truth table for function z_2 is shown in Table 5.

Table 5

x_1	x_2	z_2
0	0	0
0	1	1
1	0	0
1	1	0

The net representation is given as follows:



Case 1: Assume both weights as excitatory, i.e.,

$$w_{12} = w_{22} = 1$$

Now calculate the net inputs. For the inputs

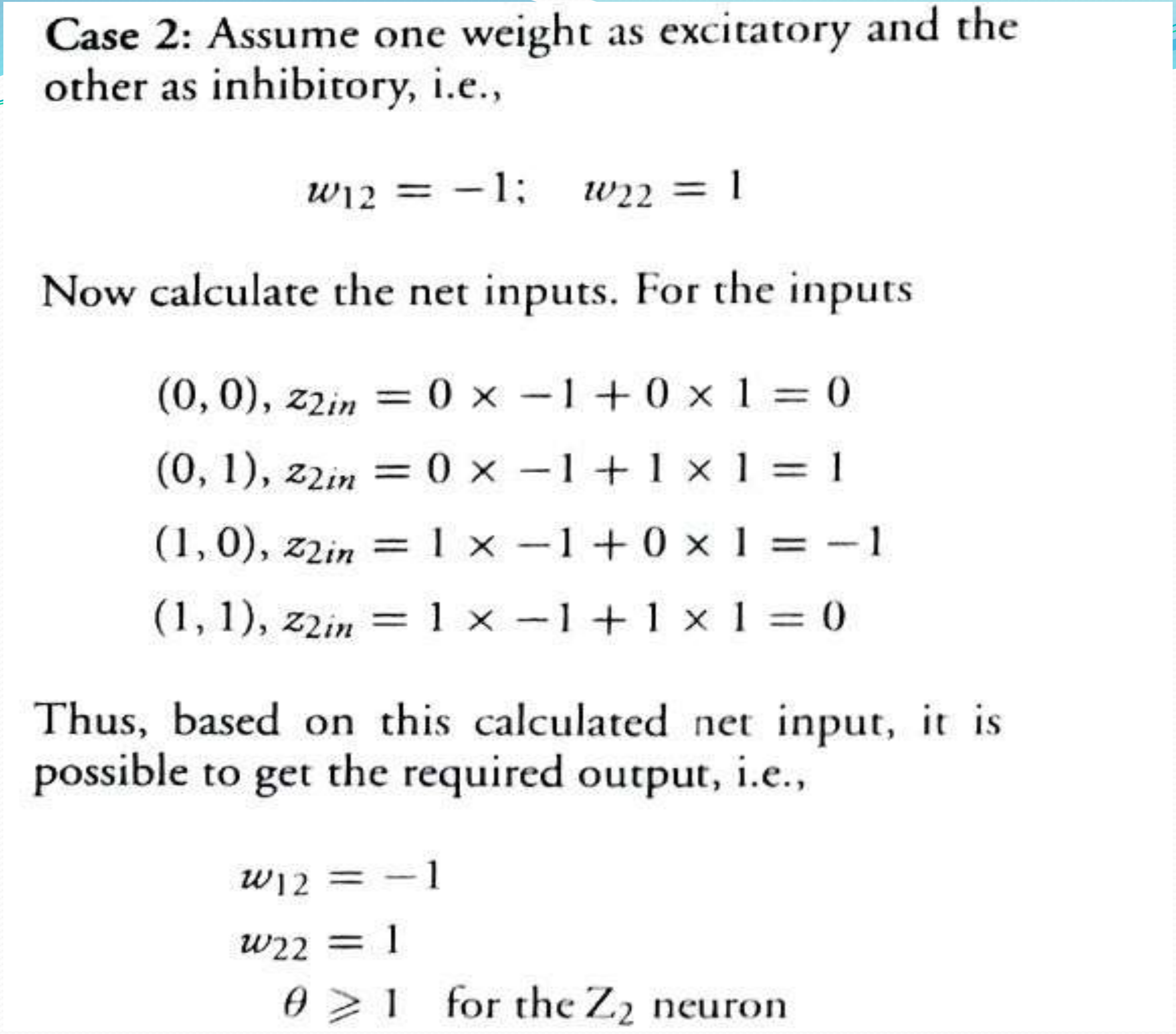
$$(0,0),\ z_{2in} = 0 \times 1 + 0 \times 1 = 0$$

$$(0,1),\ z_{2in} = 0 \times 1 + 1 \times 1 = 1$$

$$(1,0),\ z_{2in} = 1 \times 1 + 0 \times 1 = 1$$

$$(1,1),\ z_{2in} = 1 \times 1 + 1 \times 1 = 2$$

Hence, it is not possible to obtain function z_2 using these weights.



Case 2: Assume one weight as excitatory and the other as inhibitory, i.e.,

$$w_{12} = -1; \quad w_{22} = 1$$

Now calculate the net inputs. For the inputs

$$(0,0),\ z_{2in} = 0 \times -1 + 0 \times 1 = 0$$

$$(0,1),\ z_{2in} = 0 \times -1 + 1 \times 1 = 1$$

$$(1,0),\ z_{2in} = 1 \times -1 + 0 \times 1 = -1$$

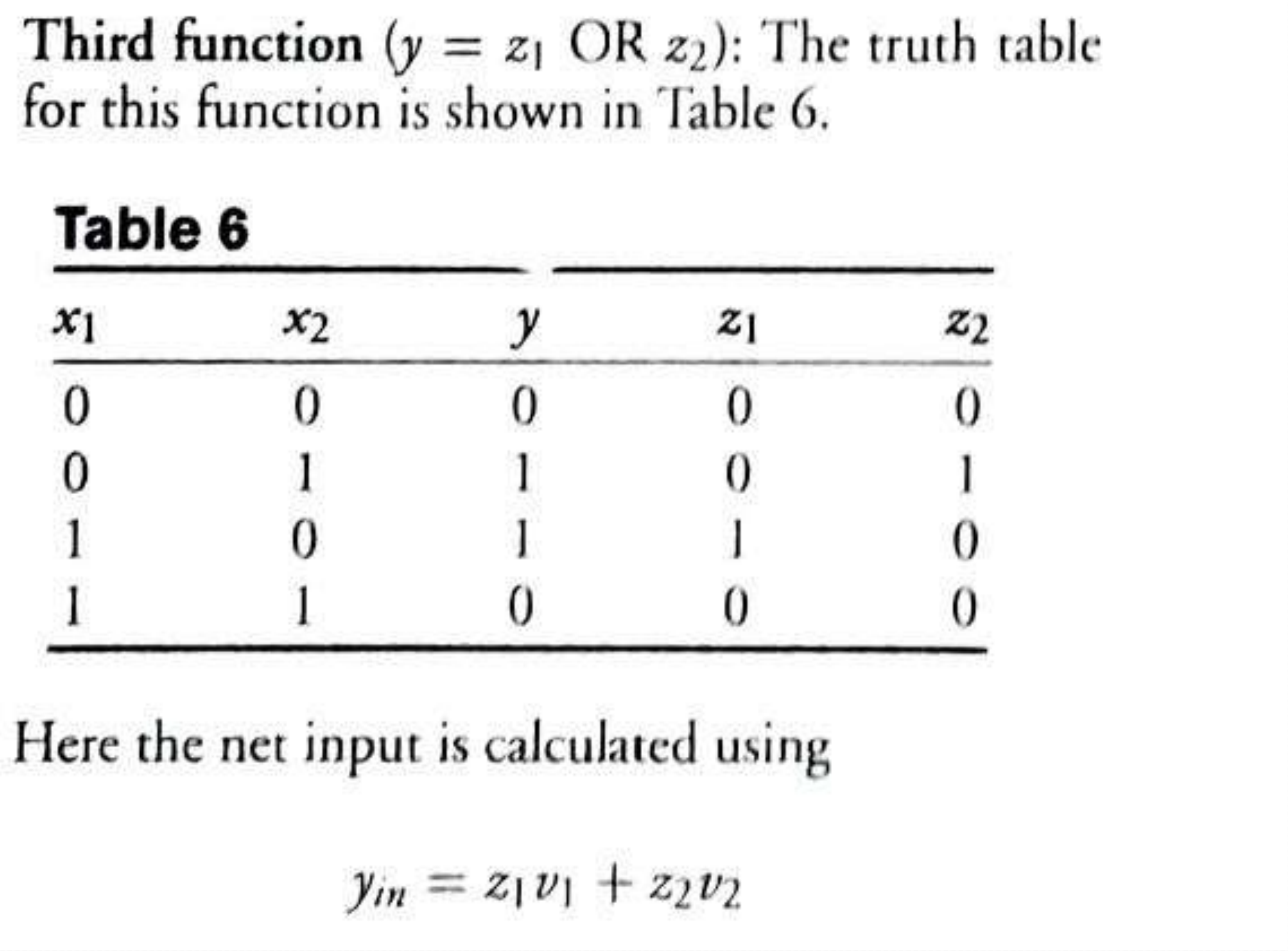
$$(1,1),\ z_{2in} = 1 \times -1 + 1 \times 1 = 0$$

Thus, based on this calculated net input, it is possible to get the required output, i.e.,

$$w_{12} = -1$$

$$w_{22} = 1$$

$$\theta \geqslant 1 \quad \text{for the } Z_2 \text{ neuron}$$



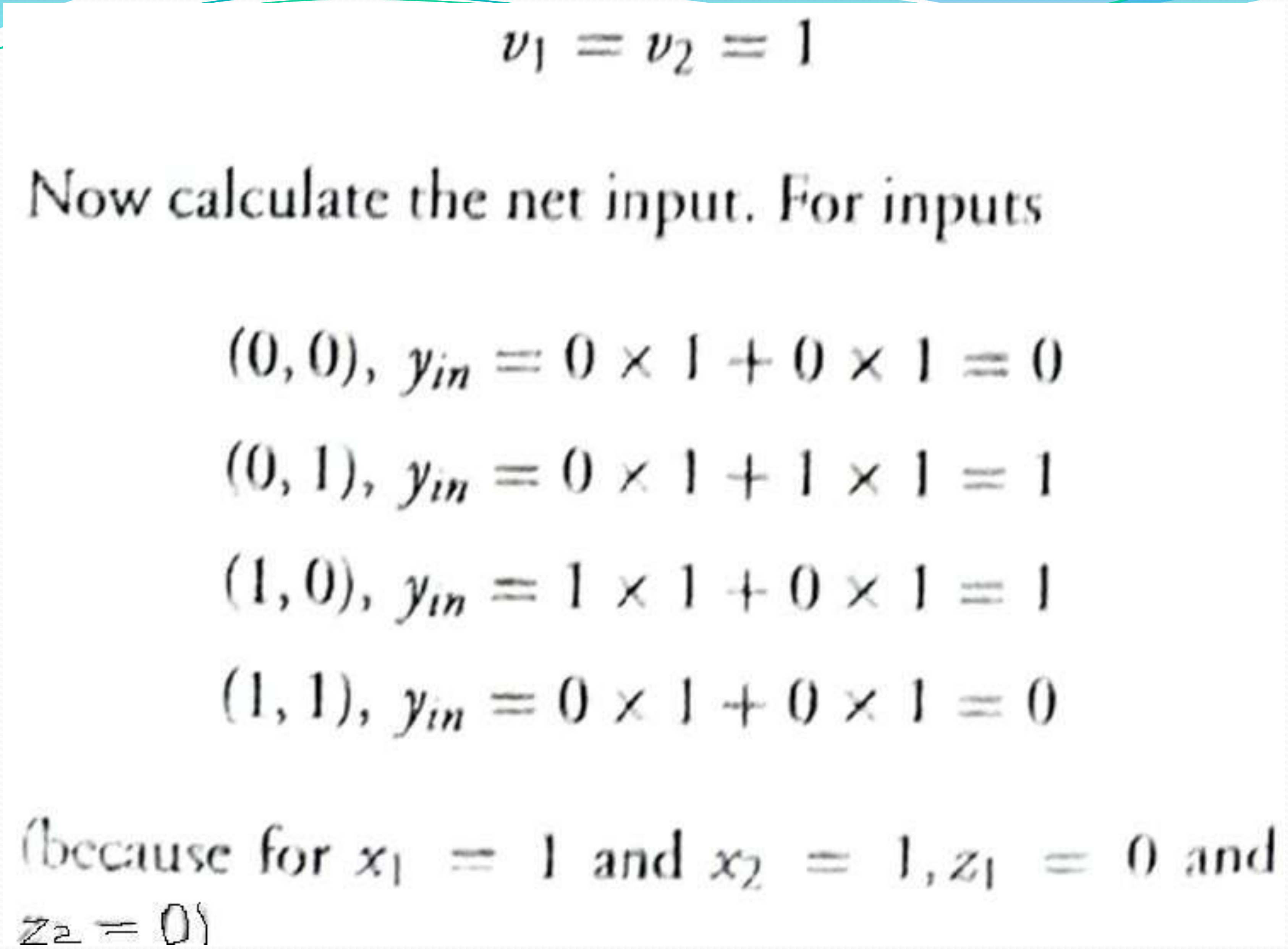
Third function ($y = z_1$ OR z_2): The truth table for this function is shown in Table 6.

Table 6

x_1	x_2	y	z_1	z_2
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	0	0

Here the net input is calculated using

$$y_{in} = z_1 v_1 + z_2 v_2$$



$$v_1 = v_2 = 1$$

Now calculate the net input. For inputs

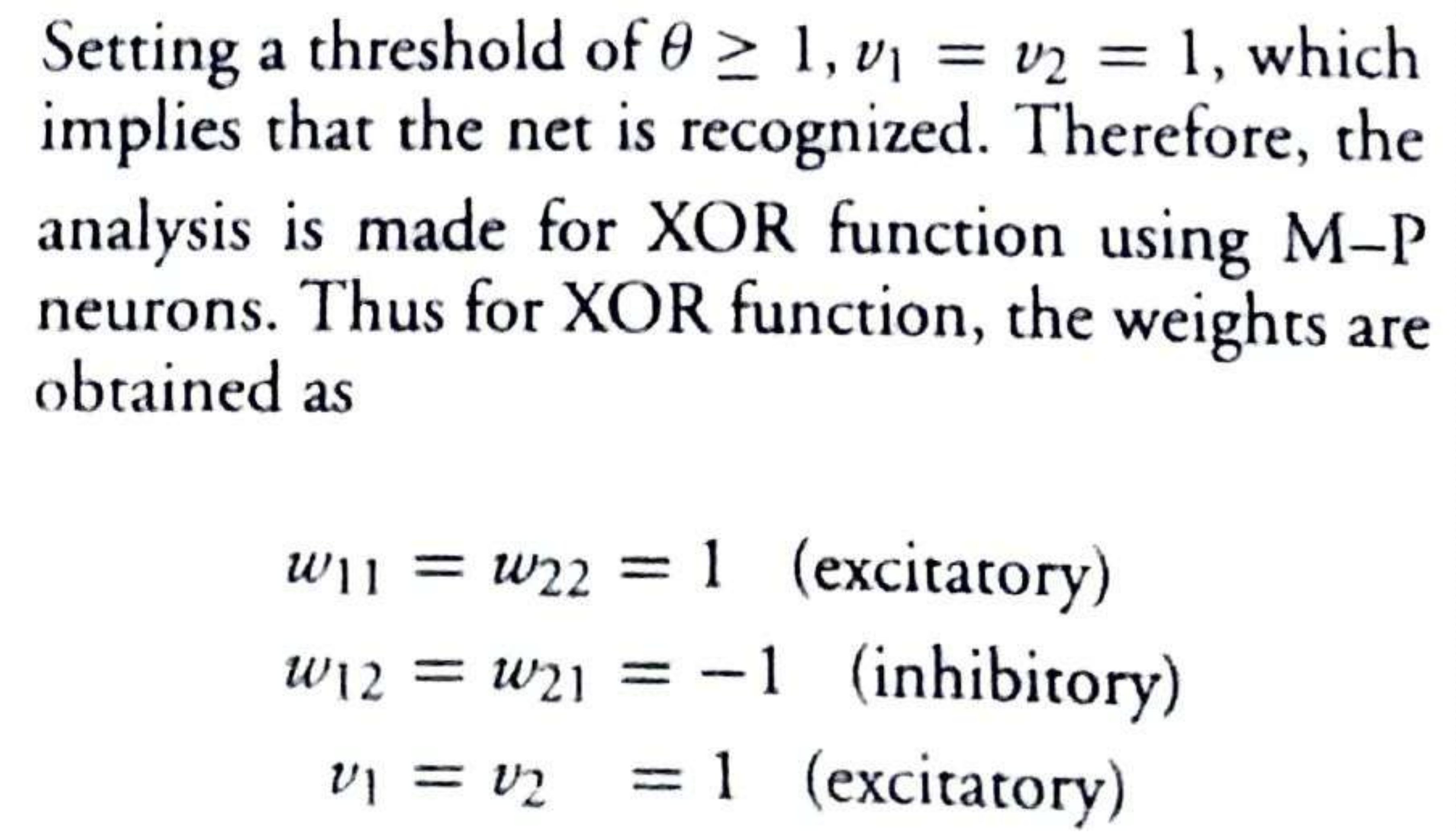
$$(0,0),\ y_{in} = 0 \times 1 + 0 \times 1 = 0$$

$$(0,1),\ y_{in} = 0 \times 1 + 1 \times 1 = 1$$

$$(1,0),\ y_{in} = 1 \times 1 + 0 \times 1 = 1$$

$$(1,1),\ y_{in} = 0 \times 1 + 0 \times 1 = 0$$

(because for $x_1 = 1$ and $x_2 = 1, z_1 = 0$ and $z_2 = 0$)



Setting a threshold of $\theta \geq 1$, $v_1 = v_2 = 1$, which implies that the net is recognized. Therefore, the analysis is made for XOR function using M–P neurons. Thus for XOR function, the weights are obtained as

$$w_{11} = w_{22} = 1 \quad \text{(excitatory)}$$

$$w_{12} = w_{21} = -1 \quad \text{(inhibitory)}$$

$$v_1 = v_2 \quad = 1 \quad \text{(excitatory)}$$

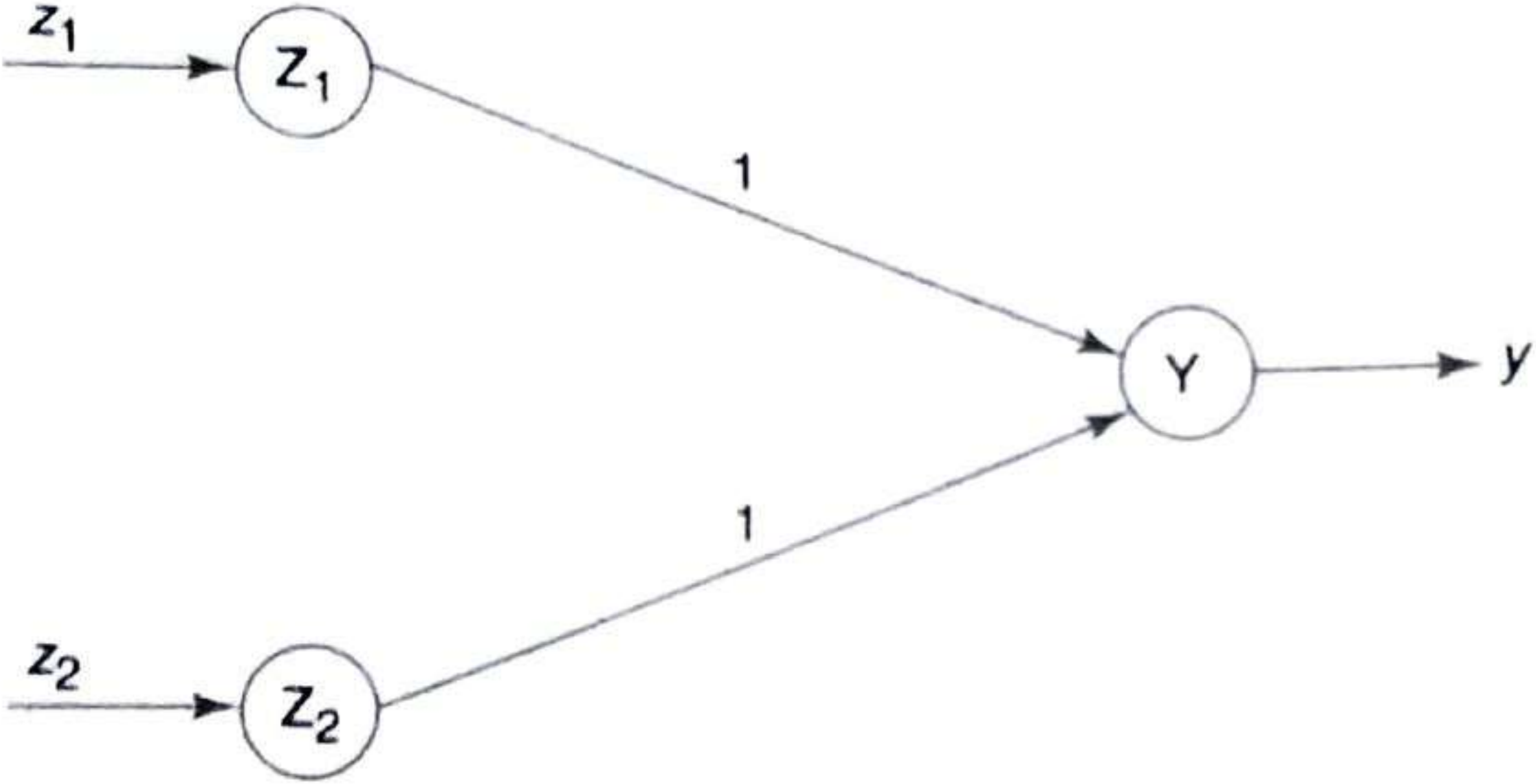
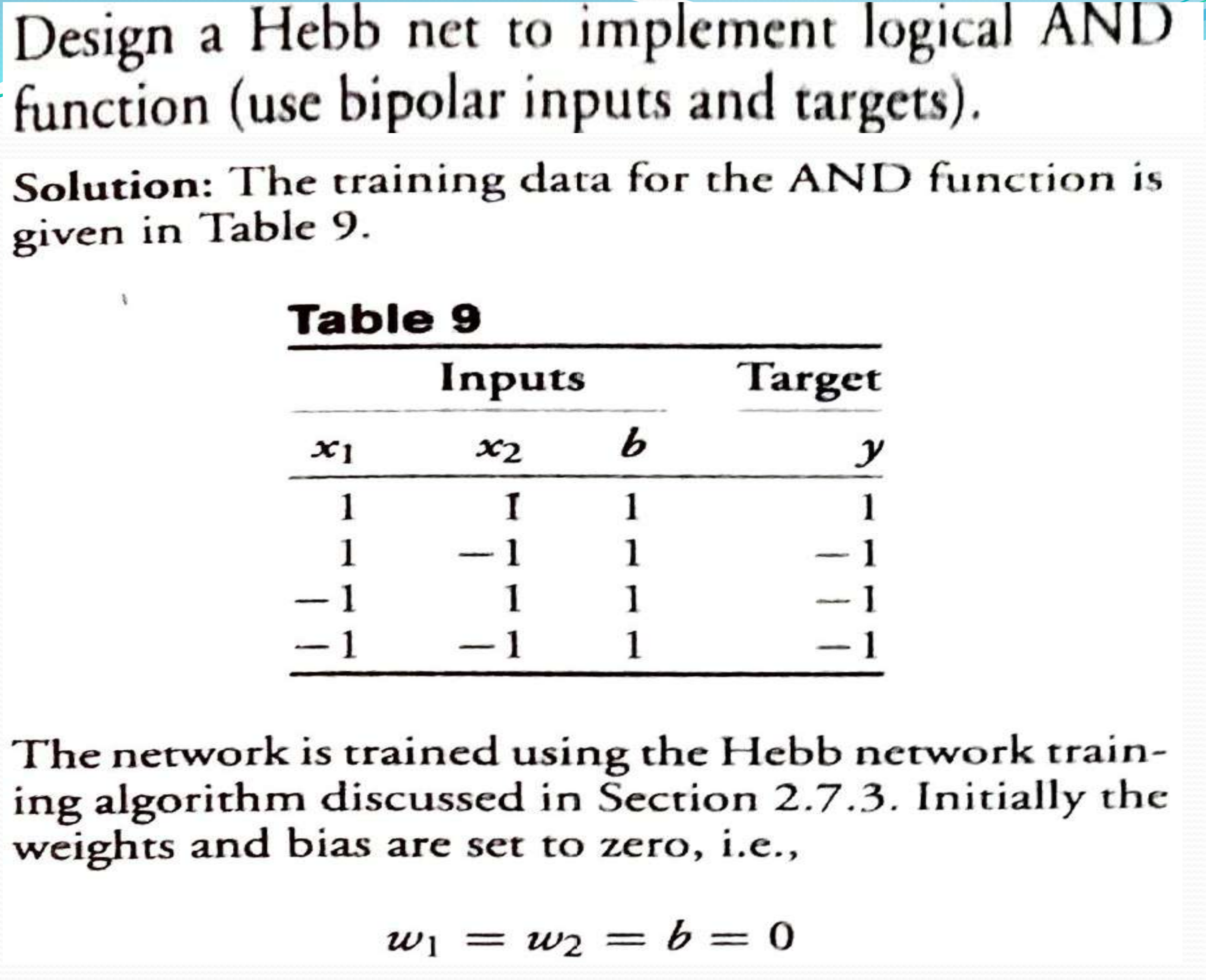


Figure 9 Neural net for Y (Z_1 OR Z_2).



Design a Hebb net to implement logical AND function (use bipolar inputs and targets).

Solution: The training data for the AND function is given in Table 9.

Table 9

Inputs			Target
x_1	x_2	b	y
1	1	1	1
1	−1	1	−1
−1	1	1	−1
−1	−1	1	−1

The network is trained using the Hebb network training algorithm discussed in Section 2.7.3. Initially the weights and bias are set to zero, i.e.,

$$w_1 = w_2 = b = 0$$

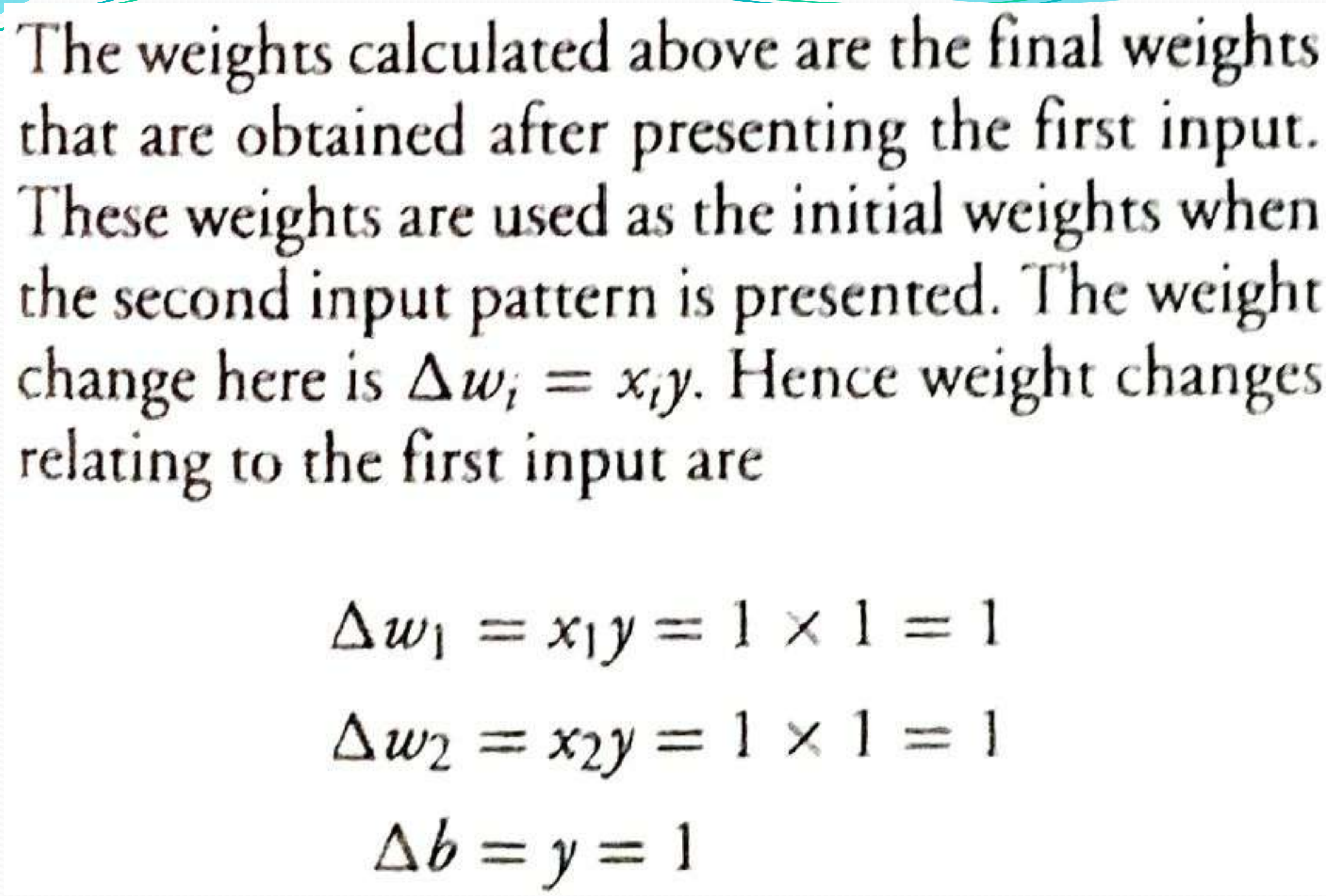
- **First input $[x_1\ x_2\ b] = [1\ 1\ 1]$ and target = 1** [i.e., $y = 1$]: Setting the initial weights as old weights and applying the Hebb rule, we get

$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

$$w_1(\text{new}) = w_1(\text{old}) + x_1 y = 0 + 1 \times 1 = 1$$

$$w_2(\text{new}) = w_2(\text{old}) + x_2 y = 0 + 1 \times 1 = 1$$

$$b(\text{new}) = b(\text{old}) + y = 0 + 1 = 1$$

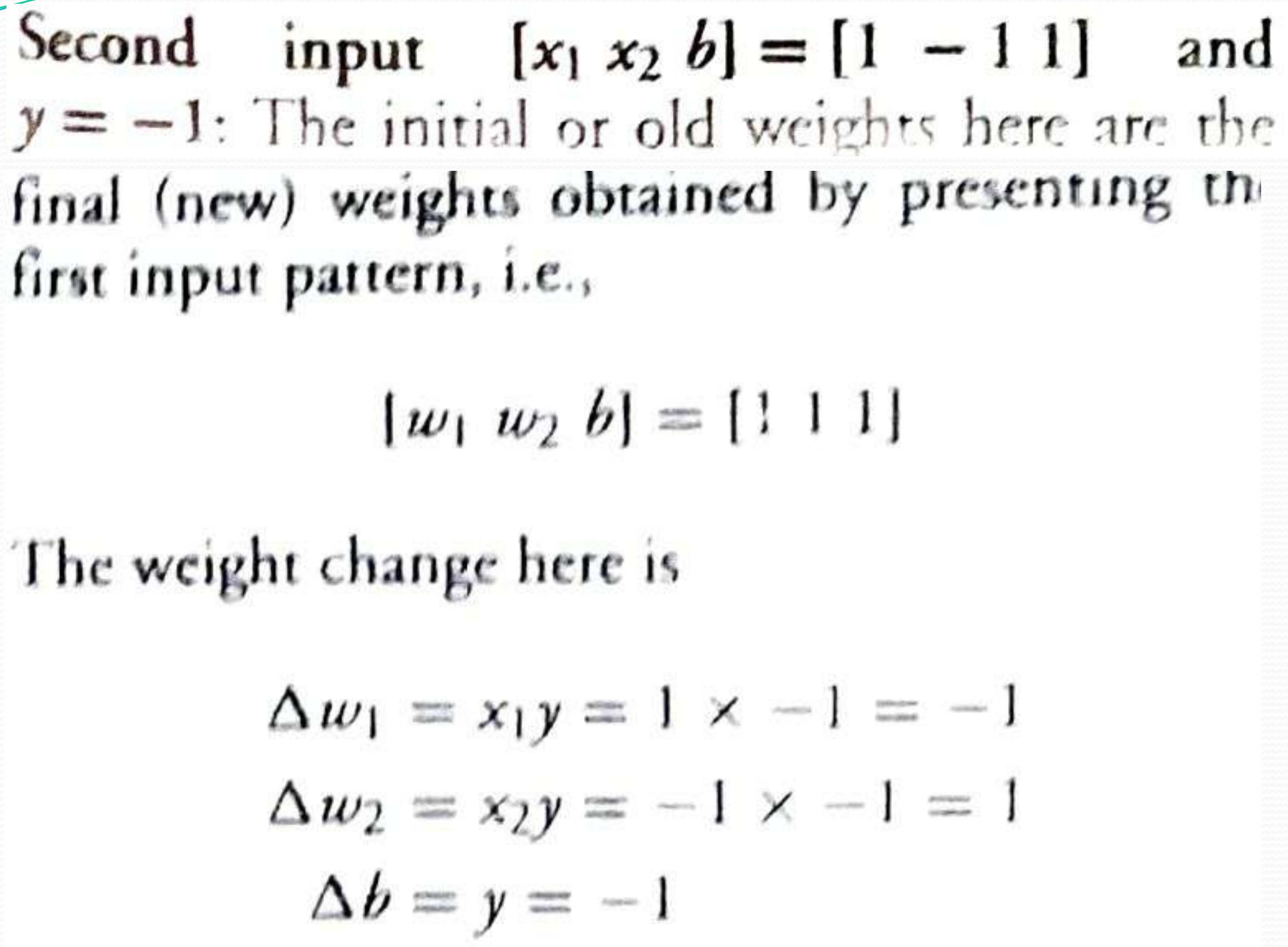


The weights calculated above are the final weights that are obtained after presenting the first input. These weights are used as the initial weights when the second input pattern is presented. The weight change here is $\Delta w_i = x_i y$. Hence weight changes relating to the first input are

$$\Delta w_1 = x_1 y = 1 \times 1 = 1$$

$$\Delta w_2 = x_2 y = 1 \times 1 = 1$$

$$\Delta b = y = 1$$

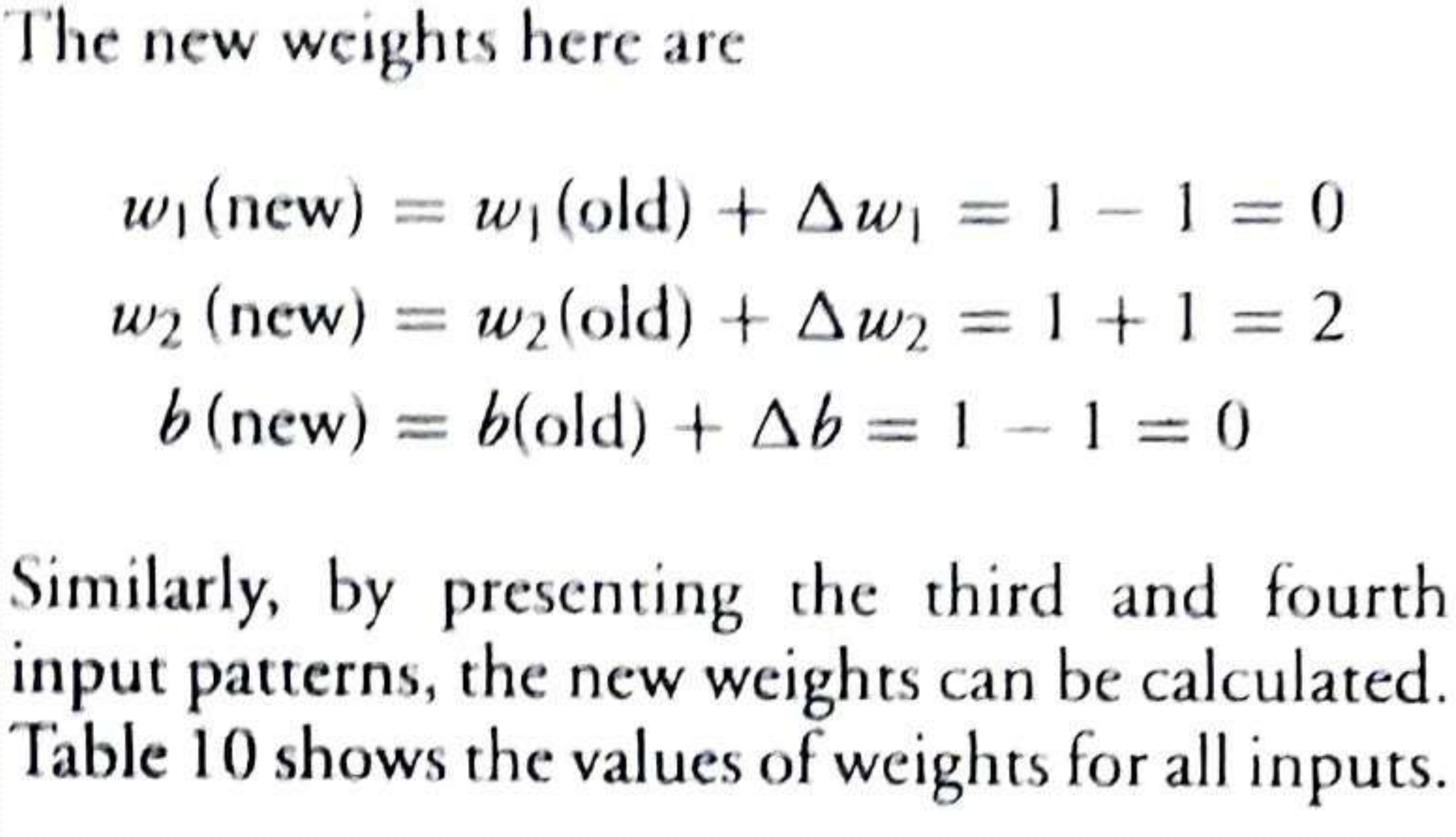


Second input $[x_1\ x_2\ b] = [1\ -1\ 1]$ and $y = -1$: The initial or old weights here are the final (new) weights obtained by presenting th first input pattern, i.e.,

$$[w_1\ w_2\ b] = [1\ 1\ 1]$$

The weight change here is

$$\Delta w_1 = x_1 y = 1 \times -1 = -1$$
$$\Delta w_2 = x_2 y = -1 \times -1 = 1$$
$$\Delta b = y = -1$$



The new weights here are

$$w_1(\text{new}) = w_1(\text{old}) + \Delta w_1 = 1 - 1 = 0$$

$$w_2(\text{new}) = w_2(\text{old}) + \Delta w_2 = 1 + 1 = 2$$

$$b(\text{new}) = b(\text{old}) + \Delta b = 1 - 1 = 0$$

Similarly, by presenting the third and fourth input patterns, the new weights can be calculated. Table 10 shows the values of weights for all inputs.

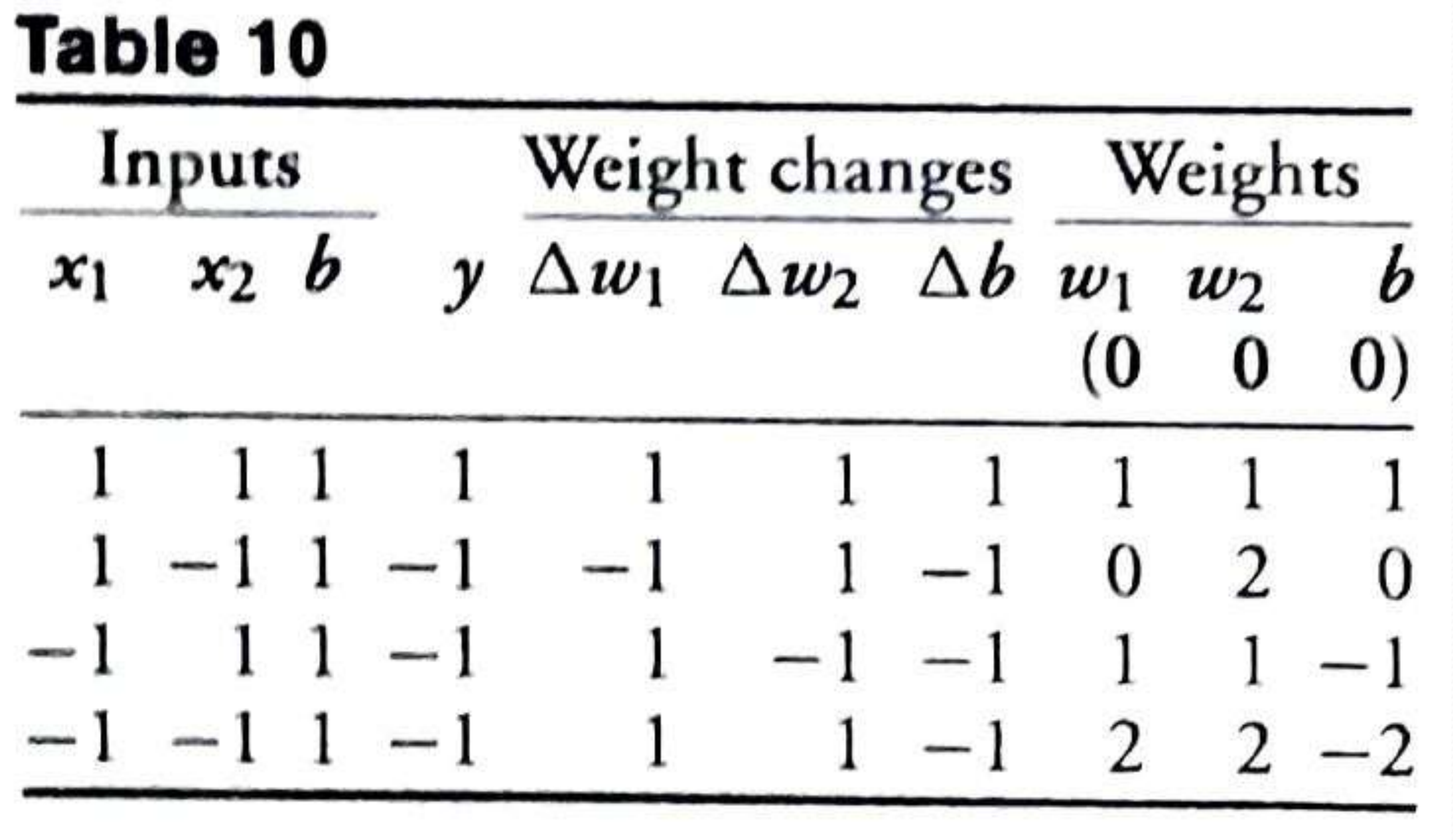
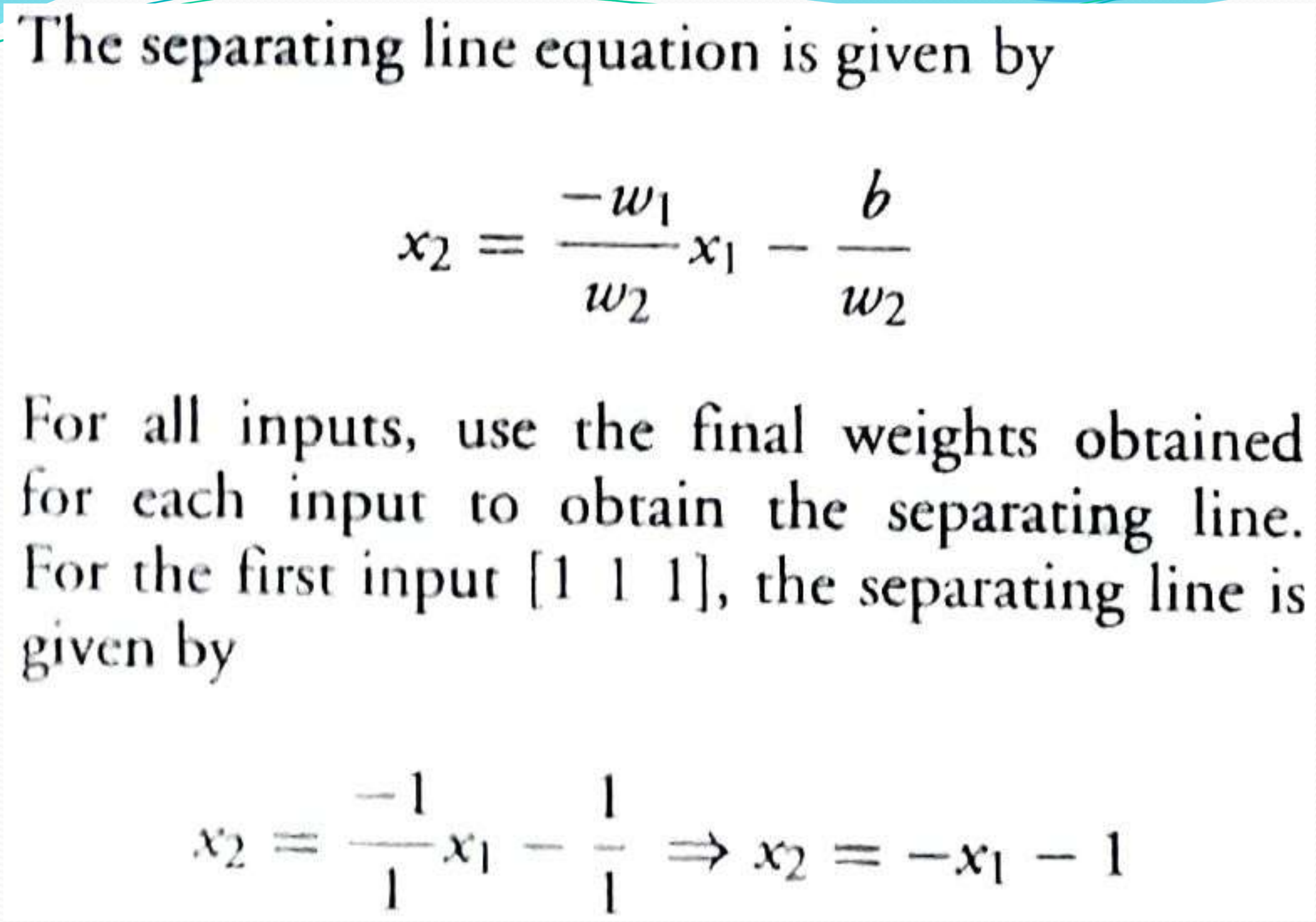


Table 10

Inputs				Weight changes			Weights		
x_1	x_2	b	y	Δw_1	Δw_2	Δb	w_1	w_2	b
							(0	0	0)
1	1	1	1	1	1	1	1	1	1
1	−1	1	−1	−1	1	−1	0	2	0
−1	1	1	−1	1	−1	−1	1	1	−1
−1	−1	1	−1	1	1	−1	2	2	−2

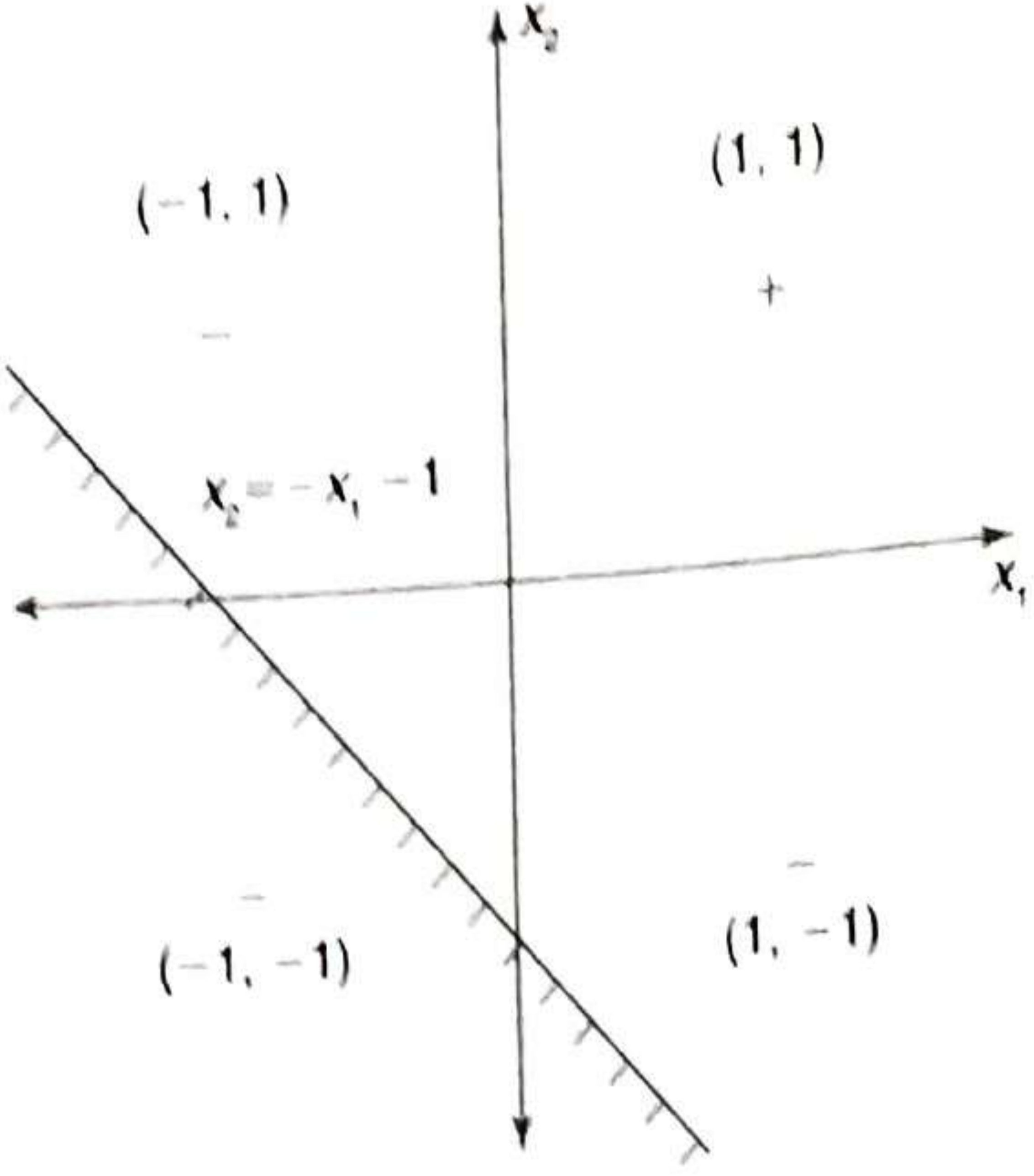


The separating line equation is given by

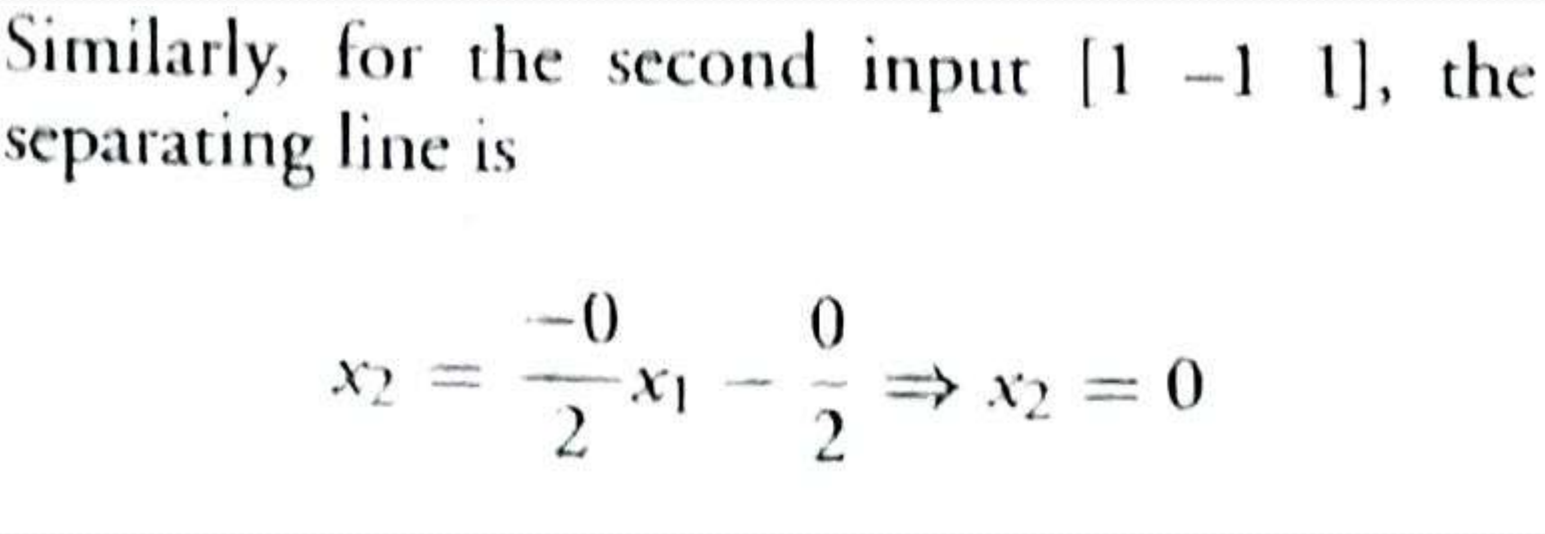
$$x_2 = \frac{-w_1}{w_2}x_1 - \frac{b}{w_2}$$

For all inputs, use the final weights obtained for each input to obtain the separating line. For the first input [1 1 1], the separating line is given by

$$x_2 = \frac{-1}{1}x_1 - \frac{1}{1} \Rightarrow x_2 = -x_1 - 1$$

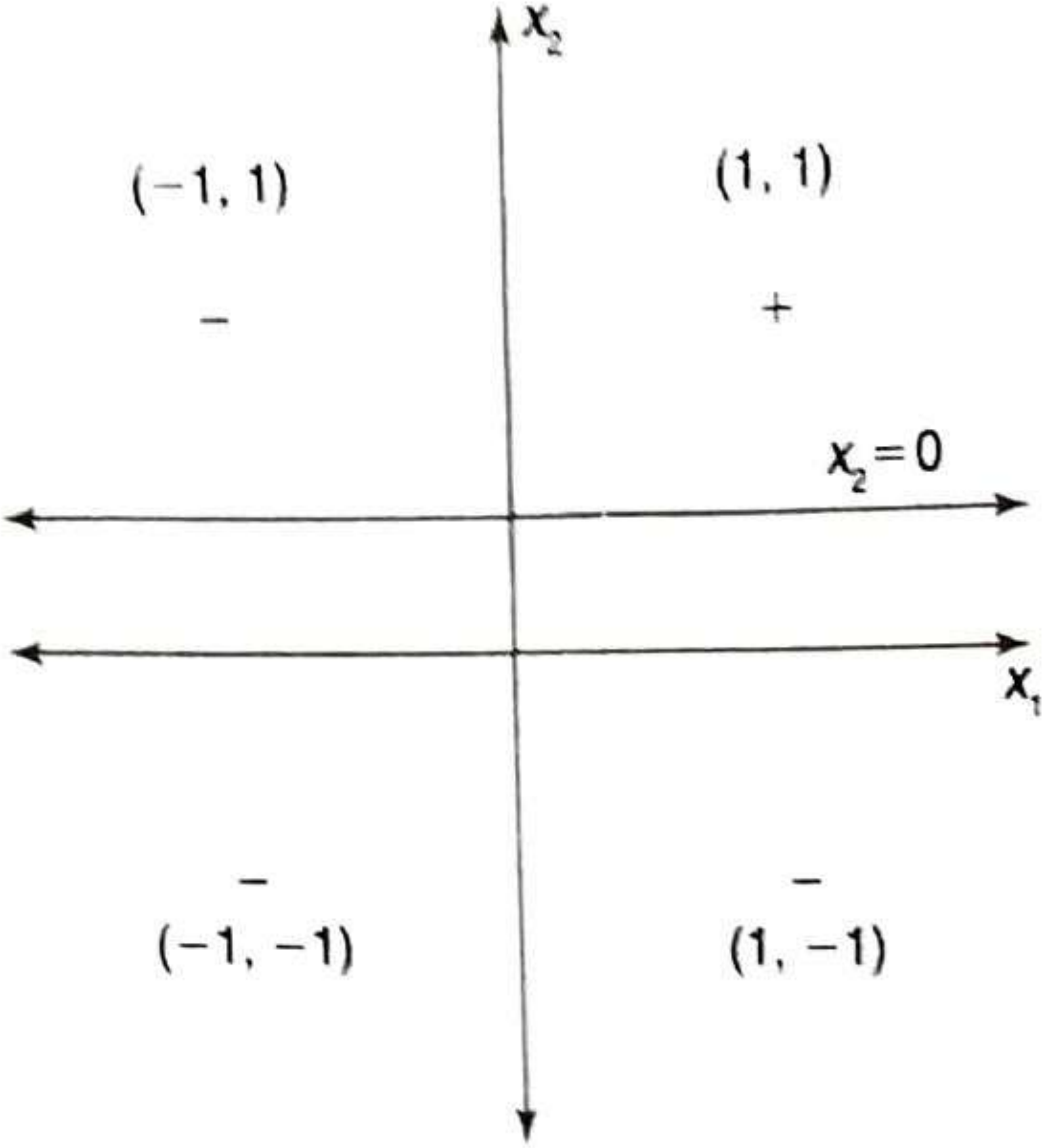


(A) First input



Similarly, for the second input [1 −1 1], the separating line is

$$x_2 = \frac{-0}{2}x_1 - \frac{0}{2} \Rightarrow x_2 = 0$$



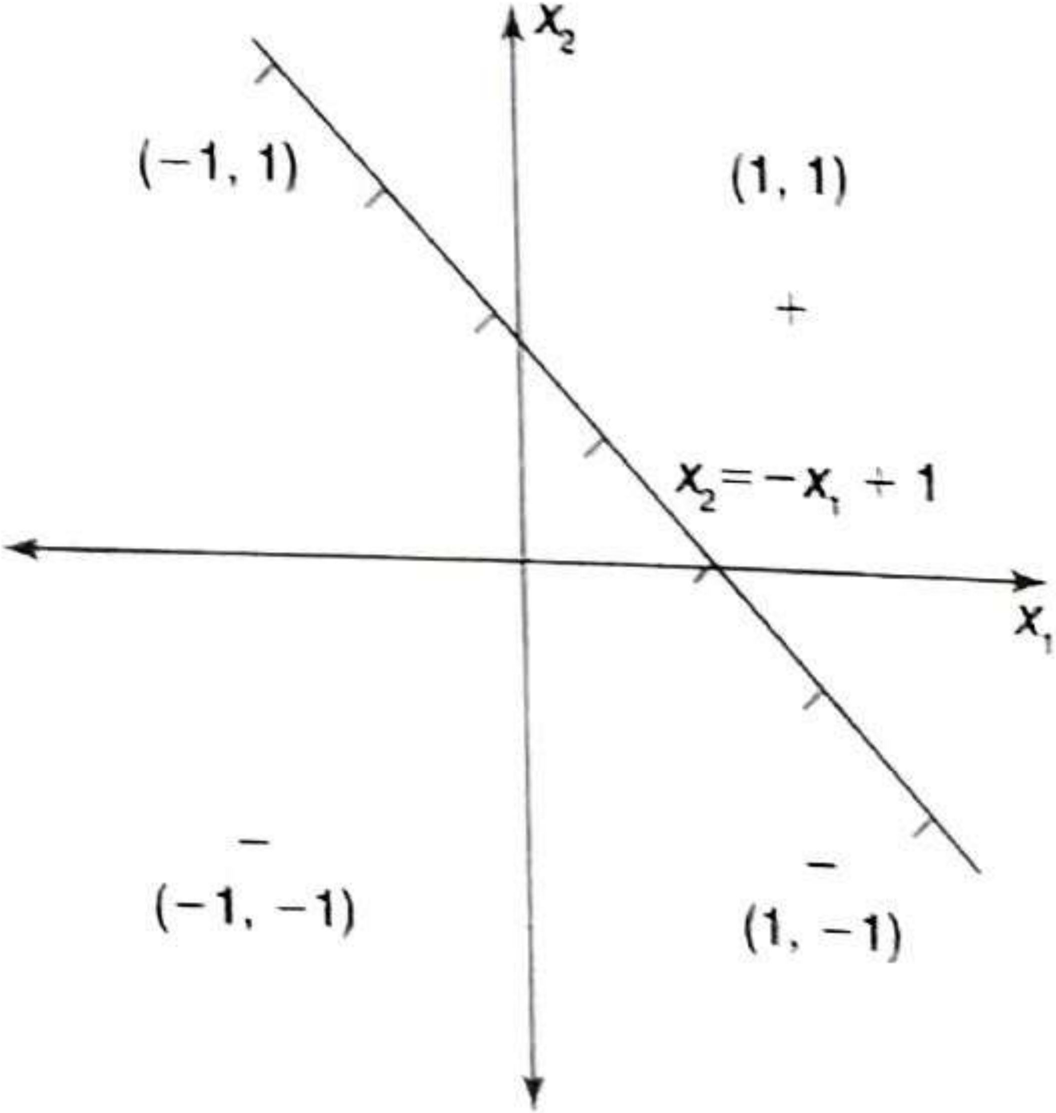
(B) Second input

For the third input [−1 1 1], it is

$$x_2 = \frac{-1}{1}x_1 + \frac{1}{1} \Rightarrow x_2 = -x_1 + 1$$

Finally, for the fourth input [−1 − 1 1], the separating line is

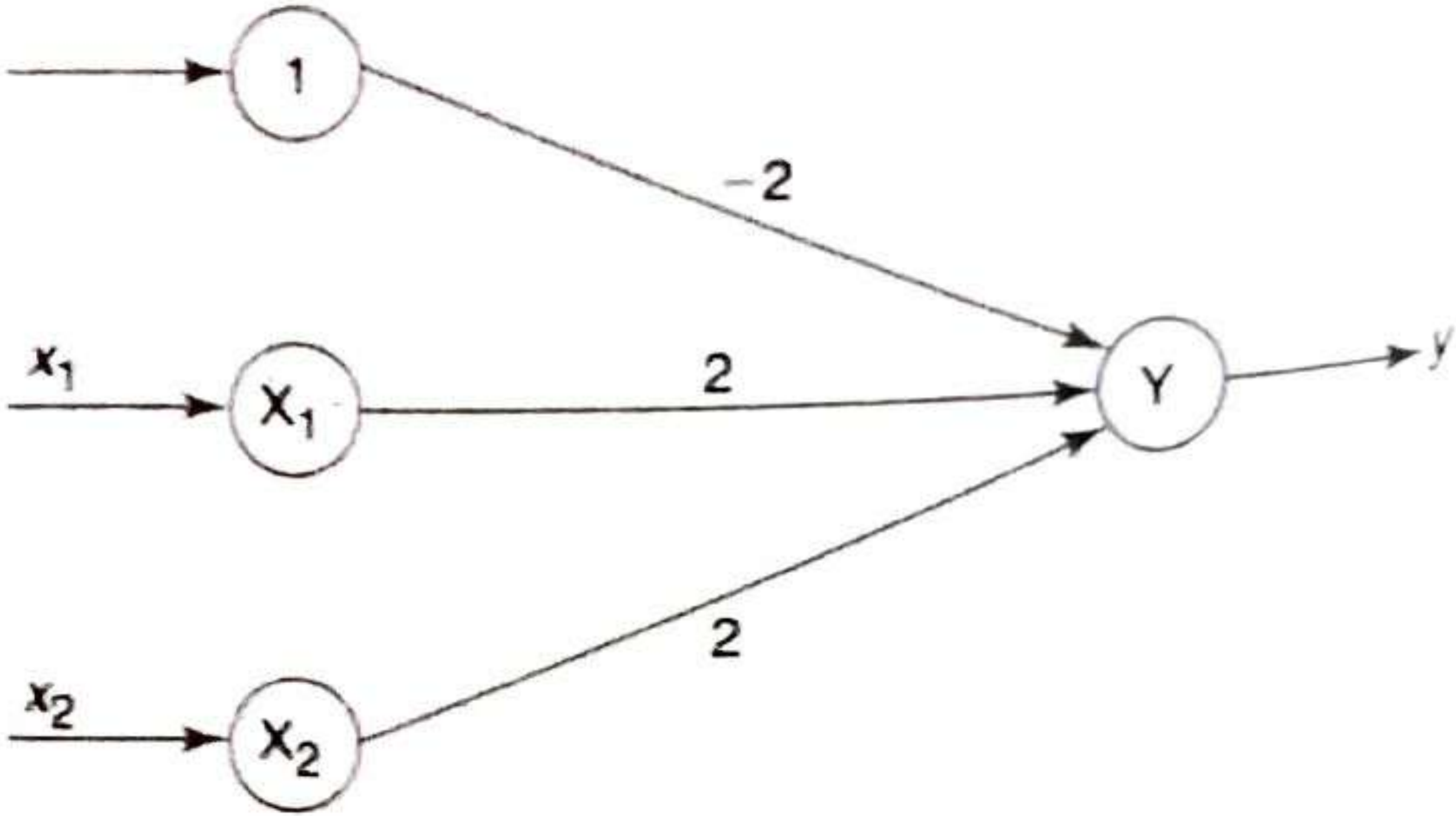
$$x_2 = \frac{-2}{2}x_1 + \frac{2}{2} \Rightarrow x_2 = -x_1 + 1$$



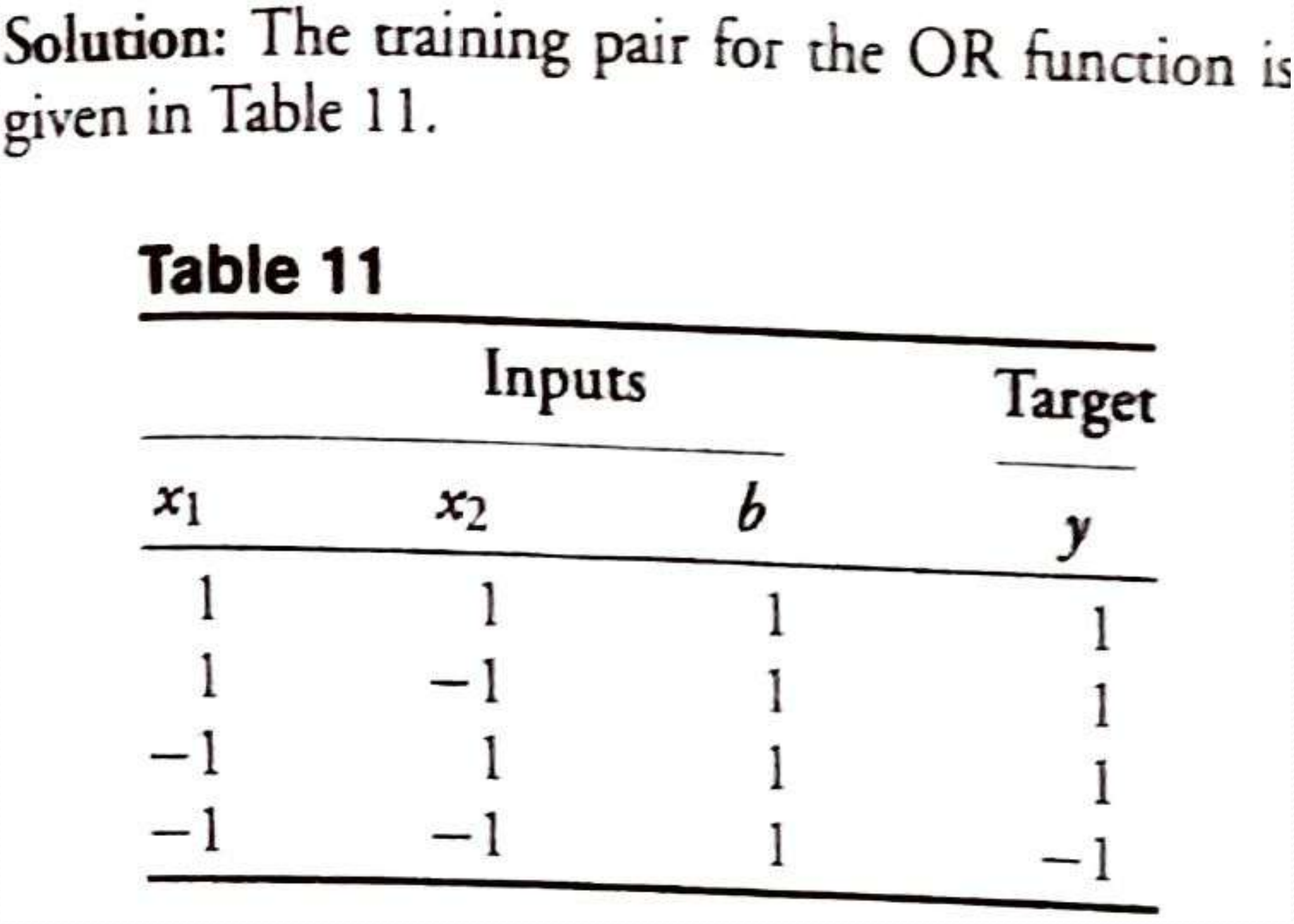
(C) Third and fourth inputs

Hebb net for AND Function

The Final weights are w1=2, w2=2,b=-2



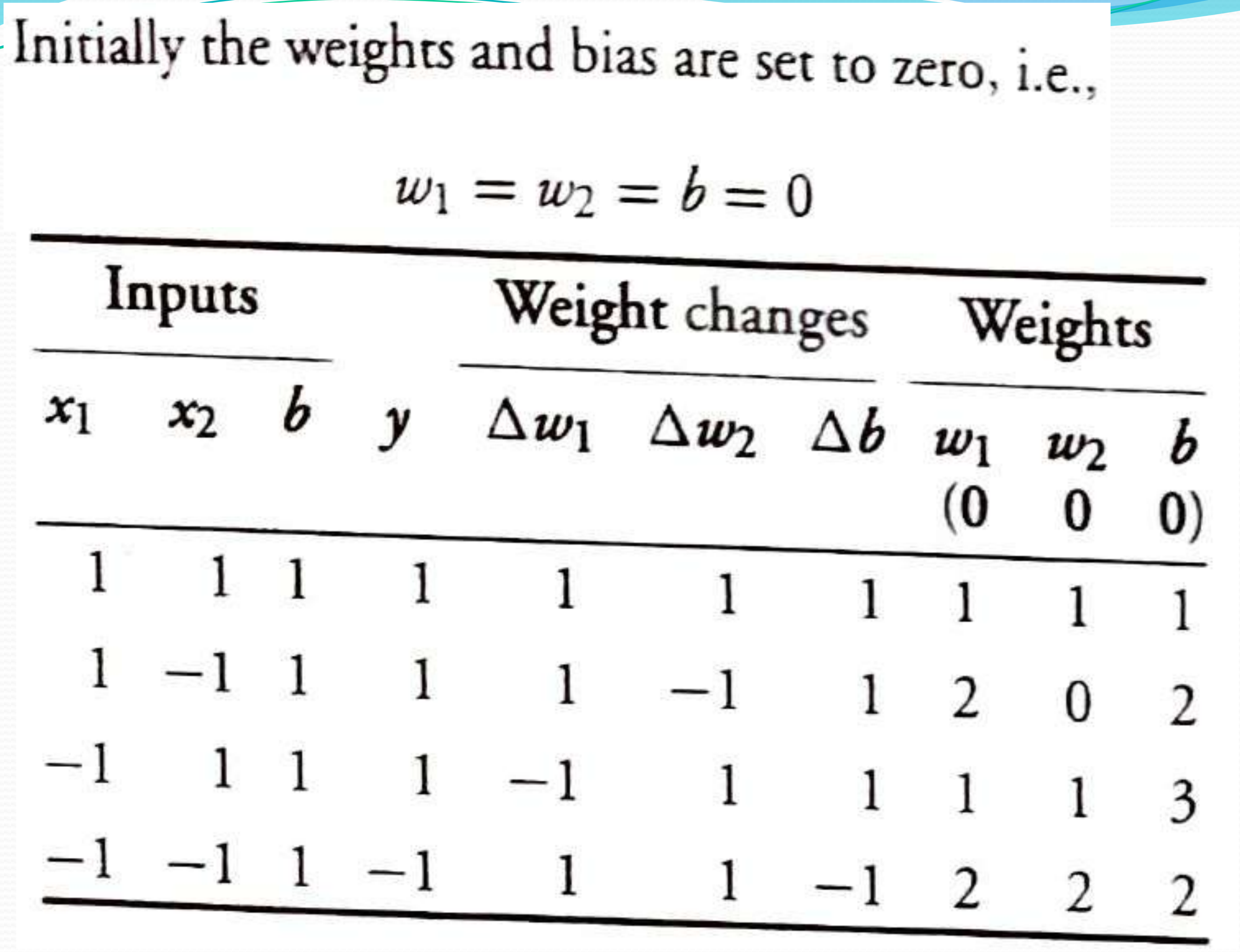
Hebb net for OR function using bipolar inputs and targets



Solution: The training pair for the OR function is given in Table 11.

Table 11

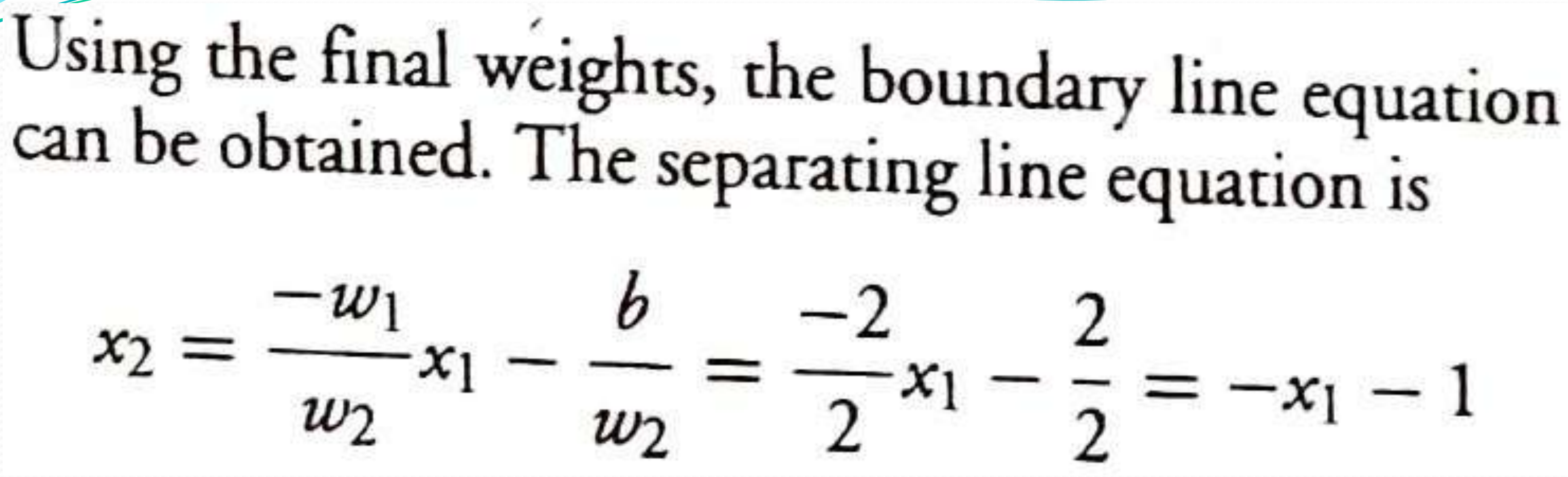
Inputs			Target
x_1	x_2	b	y
1	1	1	1
1	−1	1	1
−1	1	1	1
−1	−1	1	−1



Initially the weights and bias are set to zero, i.e.,

$$w_1 = w_2 = b = 0$$

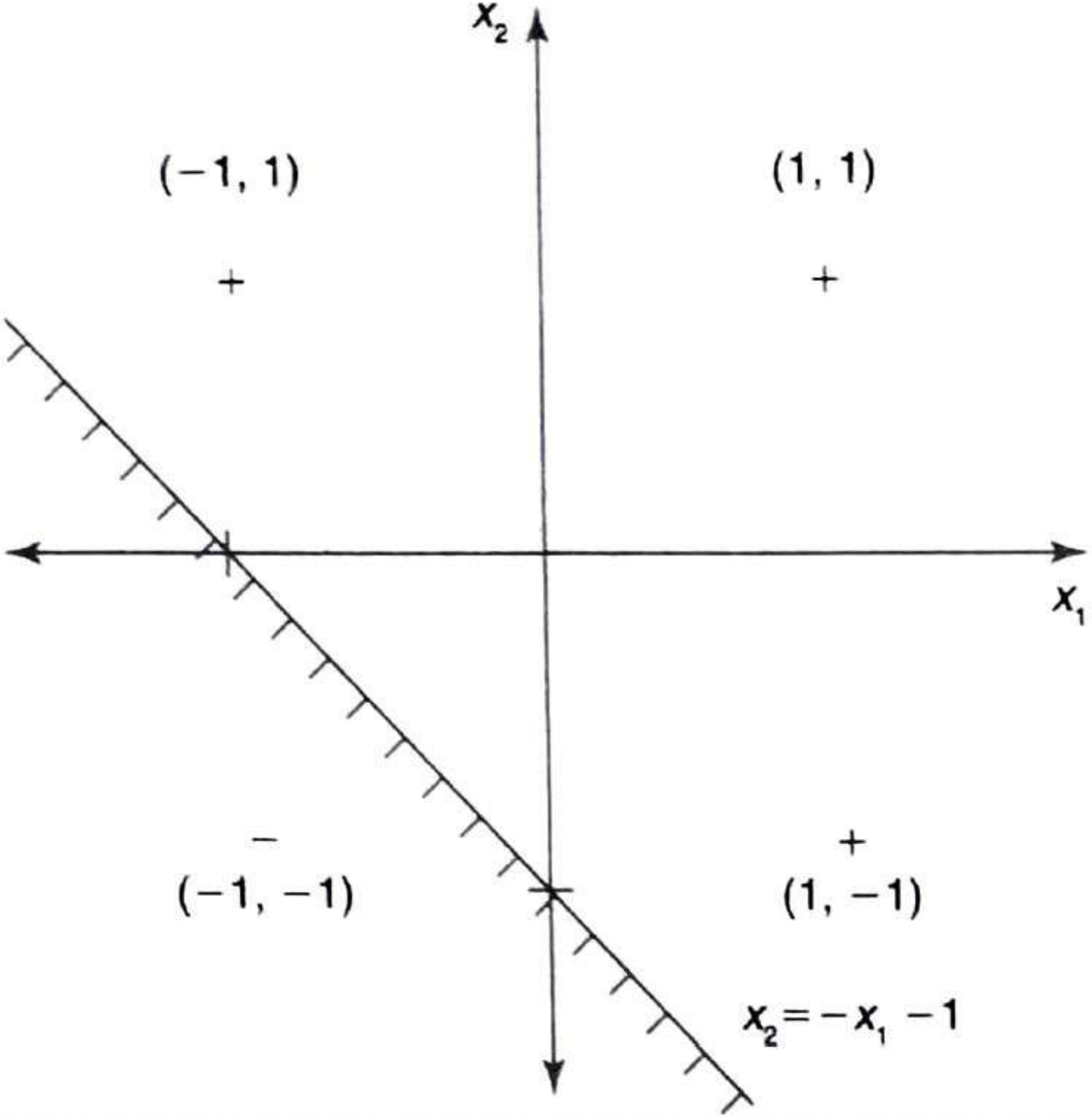
Inputs				Weight changes			Weights		
x_1	x_2	b	y	Δw_1	Δw_2	Δb	w_1	w_2	b
							(0	0	0)
1	1	1	1	1	1	1	1	1	1
1	−1	1	1	1	−1	1	2	0	2
−1	1	1	1	−1	1	1	1	1	3
−1	−1	1	−1	1	1	−1	2	2	2



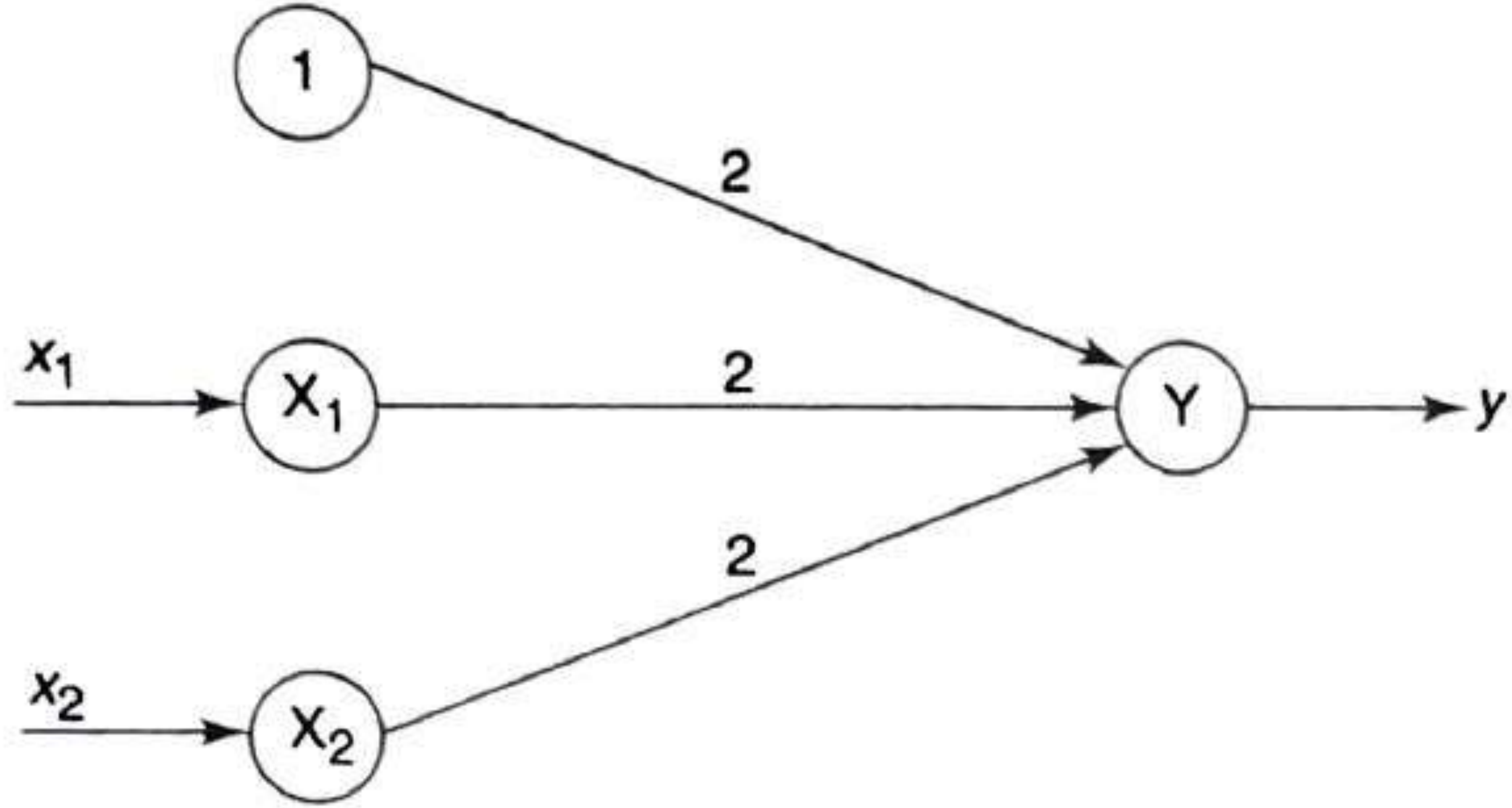
Using the final weights, the boundary line equation can be obtained. The separating line equation is

$$x_2 = \frac{-w_1}{w_2}x_1 - \frac{b}{w_2} = \frac{-2}{2}x_1 - \frac{2}{2} = -x_1 - 1$$

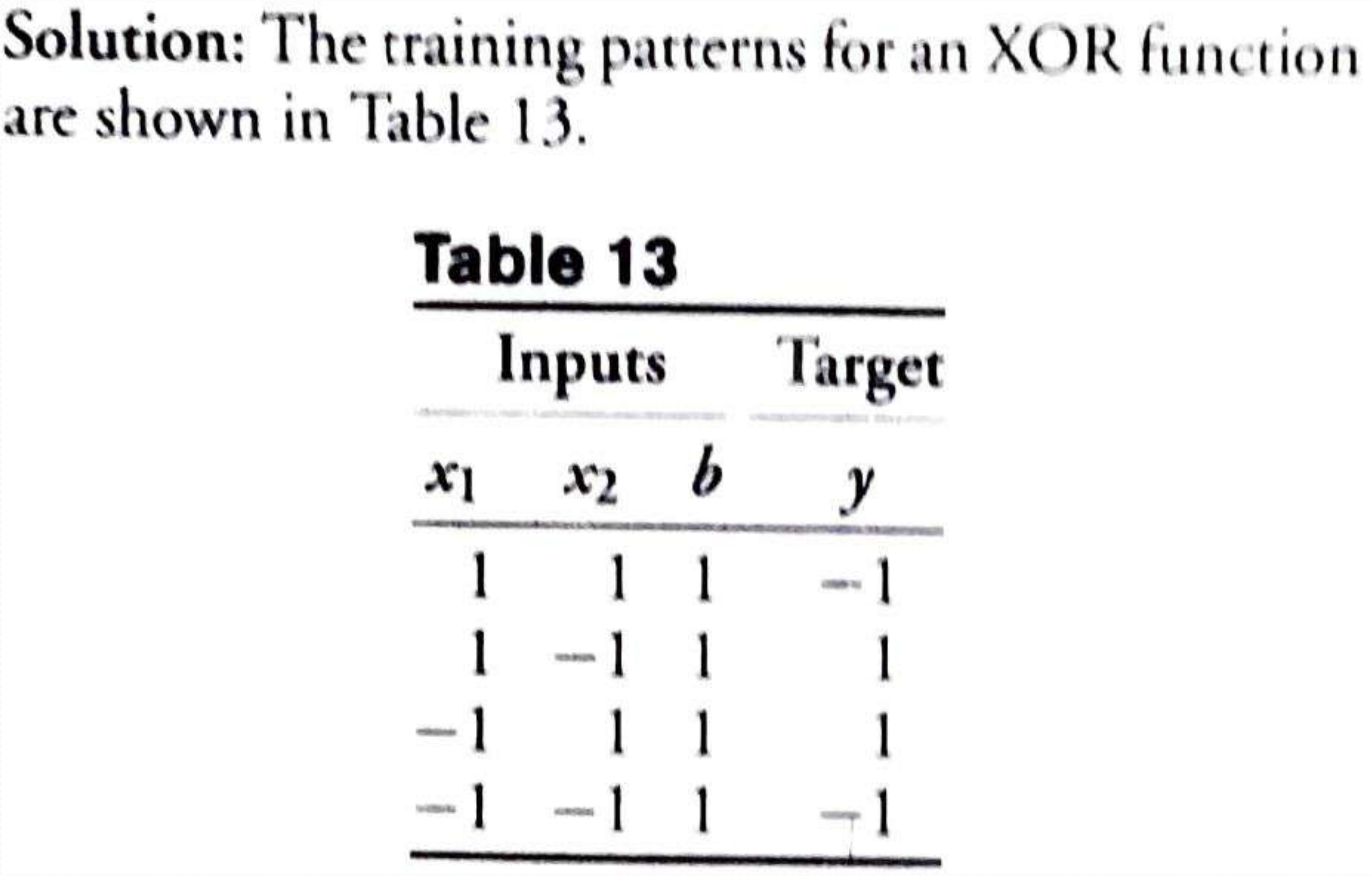
The Final weights are w1=2, w2=2,b=-2


x_2
(−1, 1)
(1, 1)
+
+
x_1
−
+
(−1, −1)
(1, −1)
$x_2 = -x_1 - 1$

Hebb net for OR function



Hebb net for XOR functionusing bipolar inputs and targets

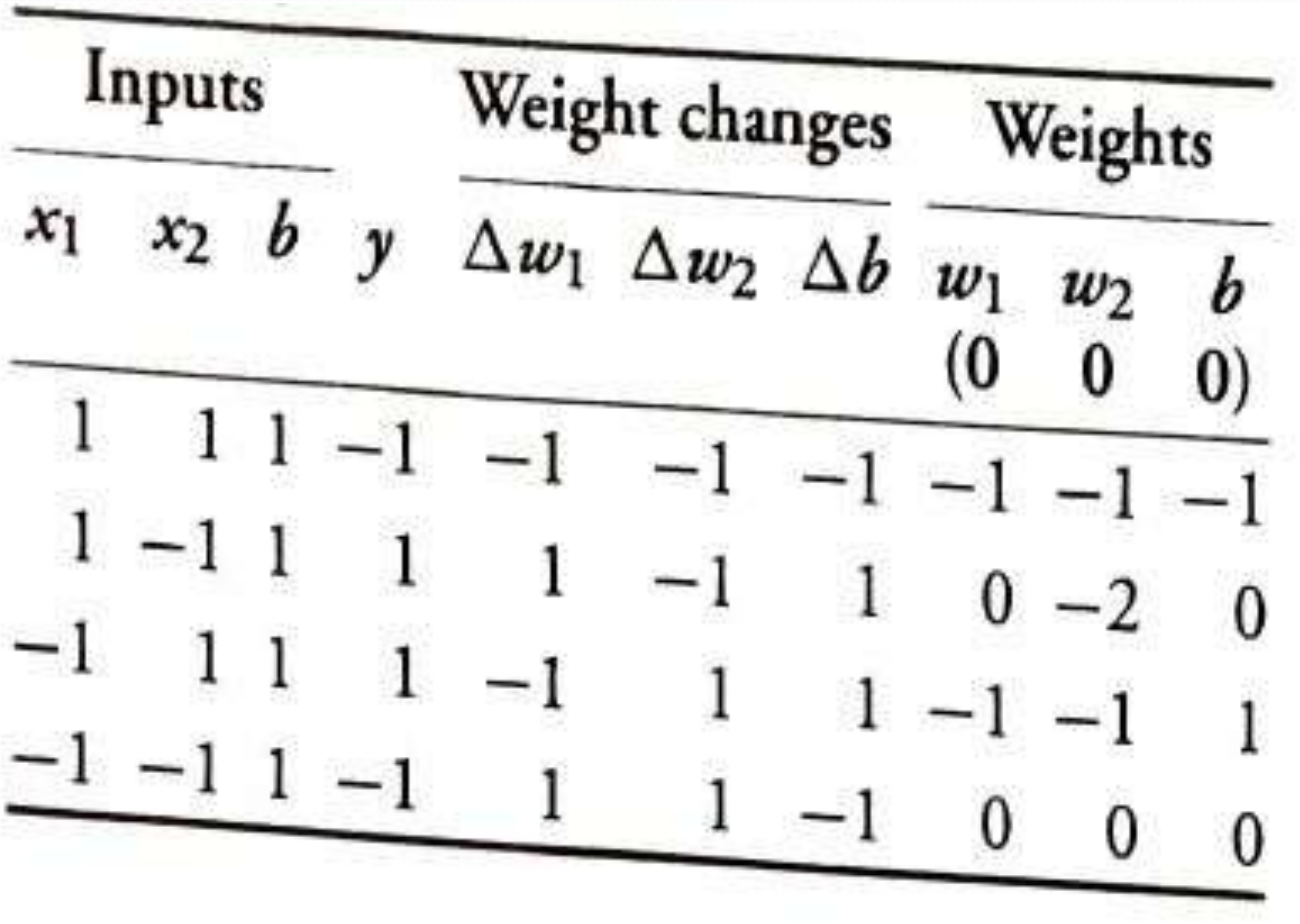


Solution: The training patterns for an XOR function are shown in Table 13.

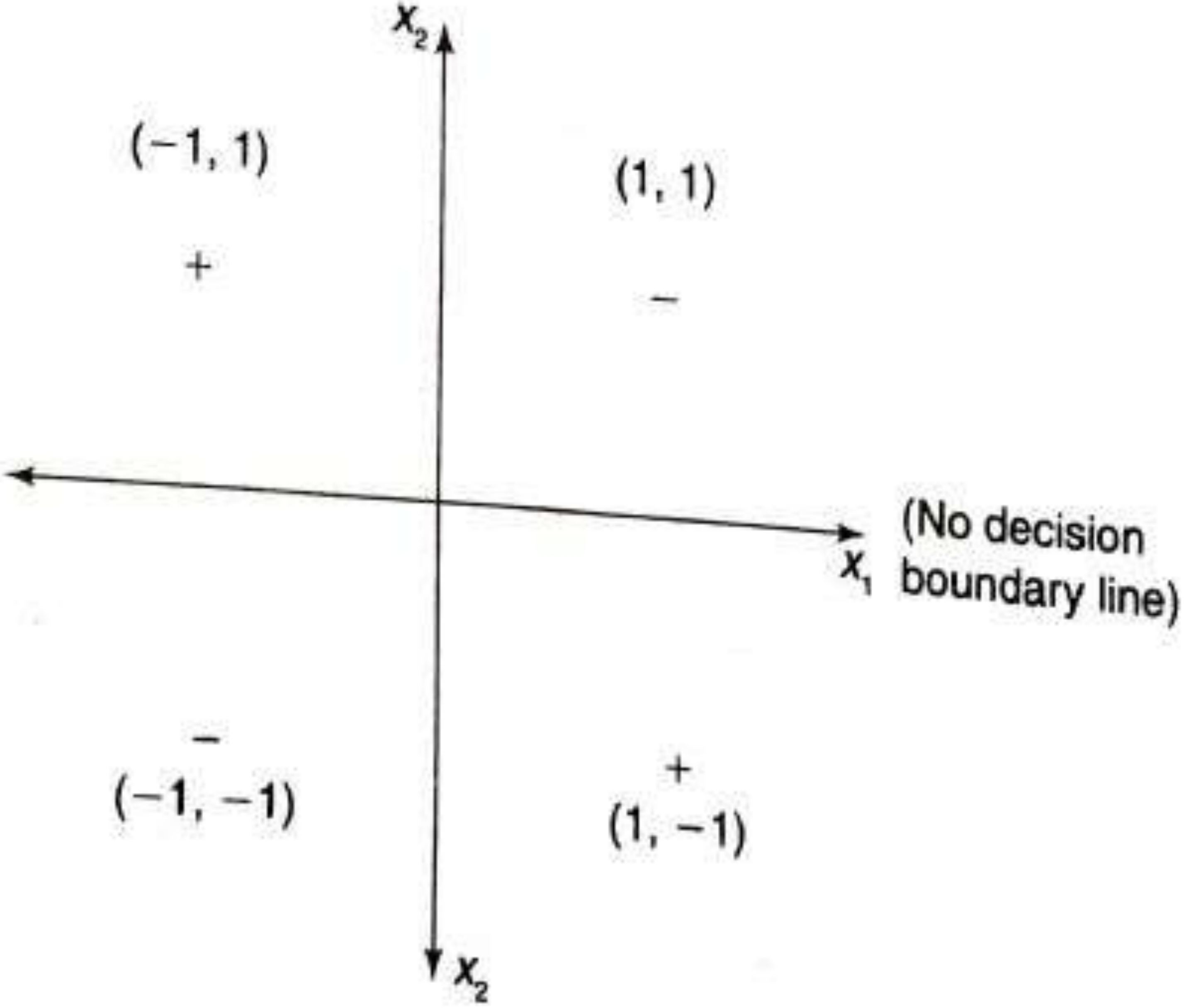
Table 13

Inputs			Target
x_1	x_2	b	y
1	1	1	−1
1	−1	1	1
−1	1	1	1
−1	−1	1	−1

$$w_1 = w_2 = b = 0$$



Inputs				Weight changes			Weights		
x_1	x_2	b	y	Δw_1	Δw_2	Δb	w_1	w_2	b
							(0	0	0)
1	1	1	−1	−1	−1	−1	−1	−1	−1
1	−1	1	1	1	−1	1	0	−2	0
−1	1	1	1	−1	1	1	−1	−1	1
−1	−1	1	−1	1	1	−1	0	0	0


x_2
(−1, 1)
+
(1, 1)
−
(No decision boundary line)
x_1
−
(−1, −1)
+
(1, −1)
x_2

MODULE 2

- Perceptron Networks
- ADALINE
- Back Propagation Networks

Perceptron Networks

Characteristics

- Perceptron network consists of 3 units: *sensory unit (input unit)*, *associator unit (hidden unit)*, and *response unit (output unit)*.
- Sensory units are connected to associator units with fixed weights having values *1,0 or -1* .
- The binary activation function is used in sensory unit and associator unit.
- The response unit has an activation of *1,0 or -1* .

- The output of the perceptron network is given by;

$$y = f(y_{in})$$

where $f(y_{in})$ is activation function and is defined as;

$$y = f(y_{in}) = \begin{cases} 1 \text{ if } y_{in} \geq \theta \\ 0 \text{ if } -\theta \leq y_{in} \leq \theta \\ -1 \text{ if } y_{in} < \theta \end{cases}$$

- The perceptron learning rule is used in the weight updation between *associator unit and response unit*.
- The error calculation is based on the comparison of the val- ues of targets with those of the calculated outputs.

The weights will be adjusted on the basis of the learning rule if an error has occurred for a particular training pattern.

$$w_i(new) = w_i(old) + \alpha t x_i$$
$$b(new) = b(old) + \alpha t$$

where,

t= target value(+1*or* 1)

α= learning rate

If no error occurs, there is no weight updation and training process may be stopped

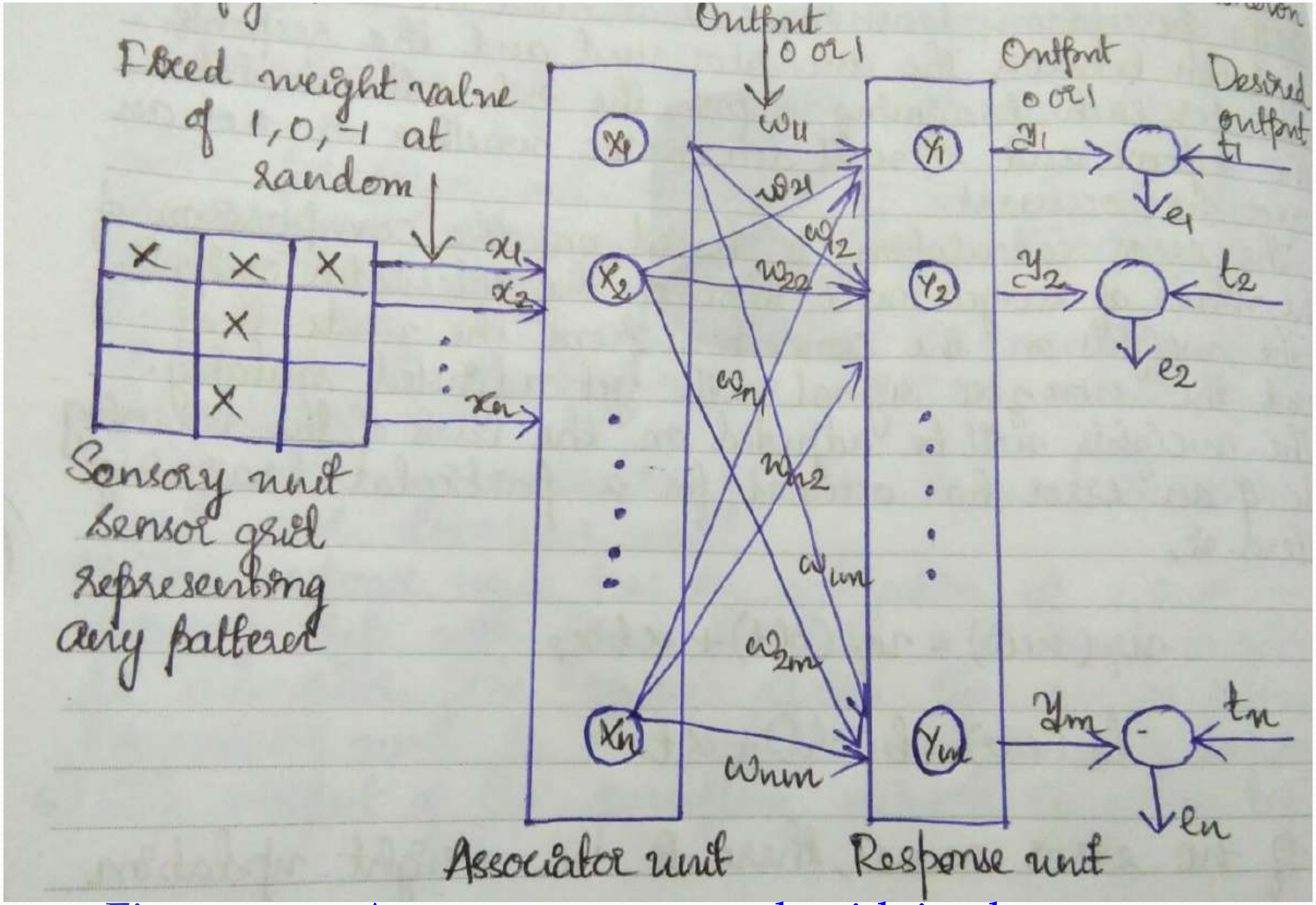
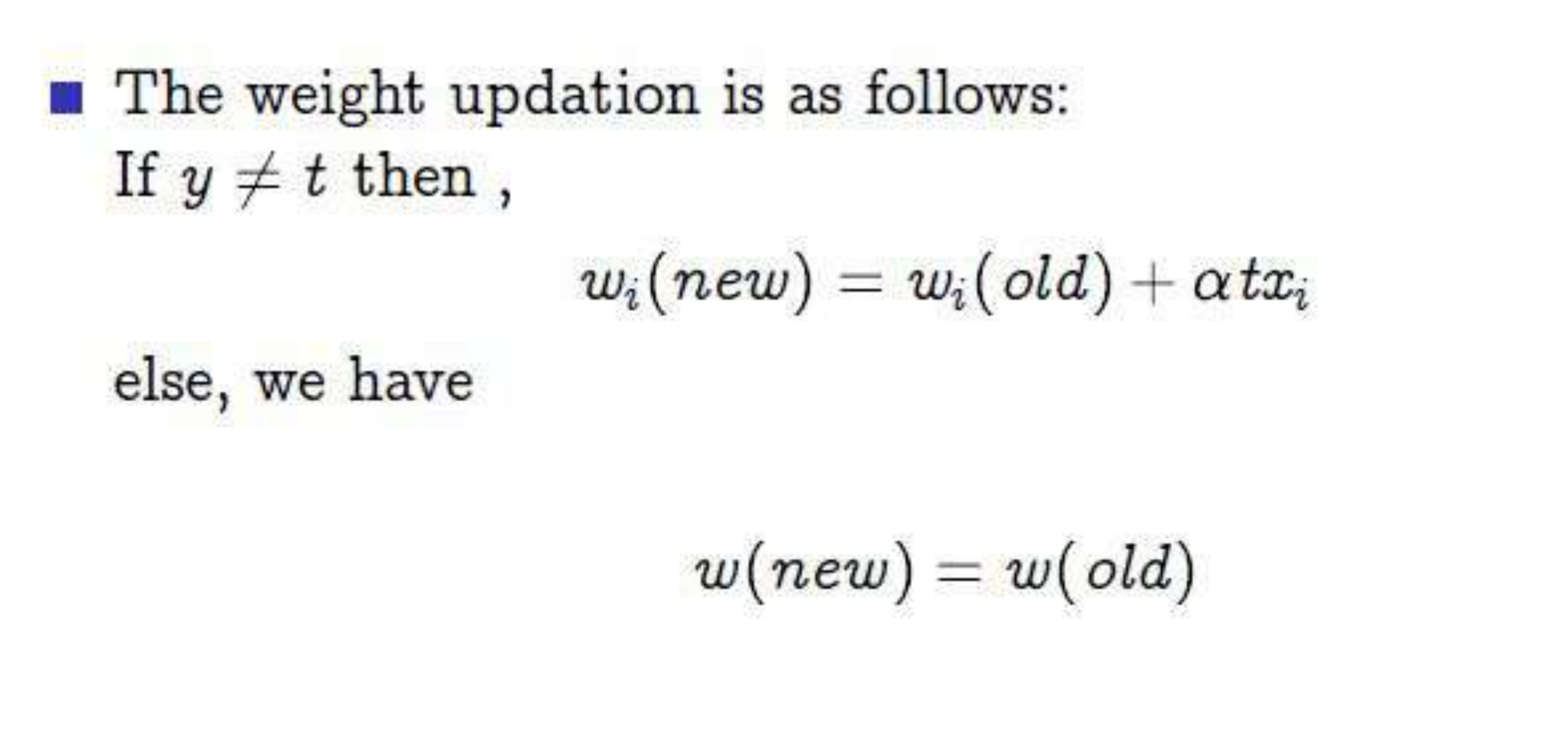


Figure : A perceptron network with its three units

Learning Rule

- A finite *n* number of input training vectors with their associated target values**;** *x(n) and t(n)*.
- The output *y* is obtained on the basis of the net input calculated and activation function being applied over the net input.

$$y = f(y_{in}) = \begin{cases} 1 \text{ if } y_{in} \geq \theta \\ 0 \text{ if } -\theta \leq y_{in} \leq \theta \\ -1 \text{ if } y_{in} < \theta \end{cases}$$



- The weight updation is as follows:
  If $y \neq t$ then ,

$$w_i(new) = w_i(old) + \alpha t x_i$$

else, we have

$$w(new) = w(old)$$

Perceptron Learning Rule Convergence Theorem

- “If there is a weight vector W, such that f(x (n) W)= t(n), then for any starting vector w_1, the perceptron learning rule will converge to a weight vector that gives the correct response for all training patterns, and this learning takes place within a finite number of steps provided that the solution exists”

Architecture

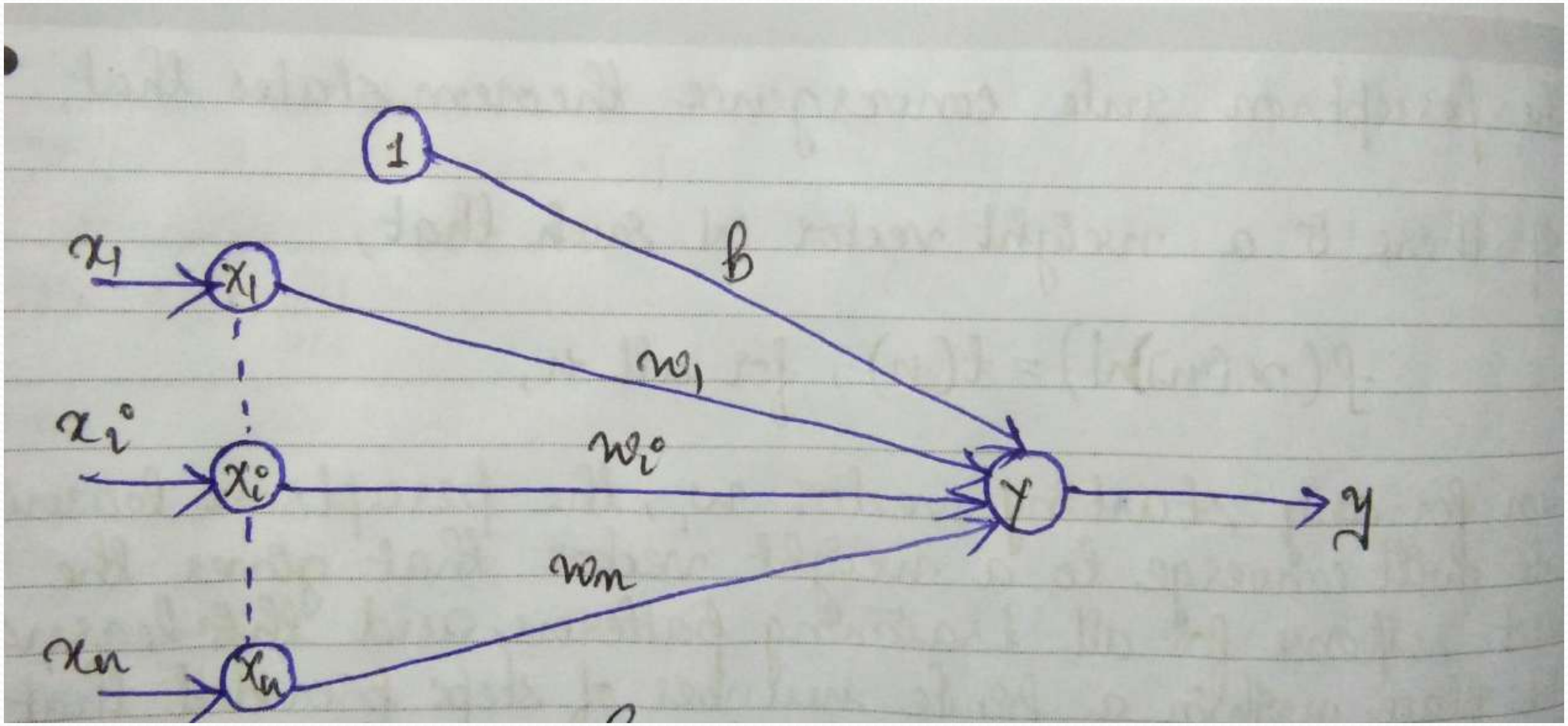


Figure : Single classification perceptron network

- Perceptron has sensory, associator and response unit
- In this architecture only the associator and response unit is shown and sensory unit is hidden because only the weights between the associator and the response unit are adjusted
- Input layer consists of input neurons from X1…Xi…Xn
- There always exist a common bias of 1
- This is a single layer network

Flowchart

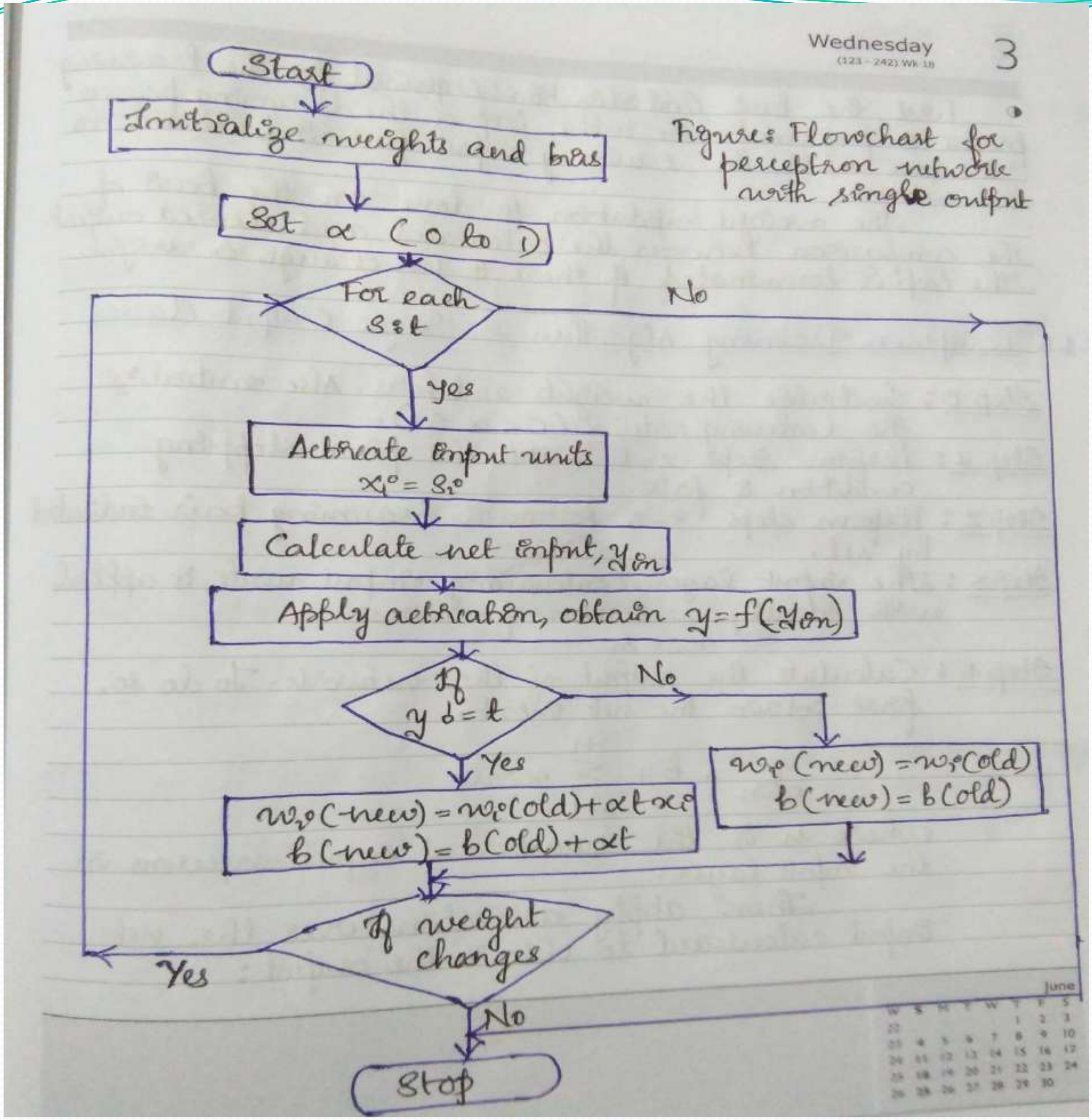
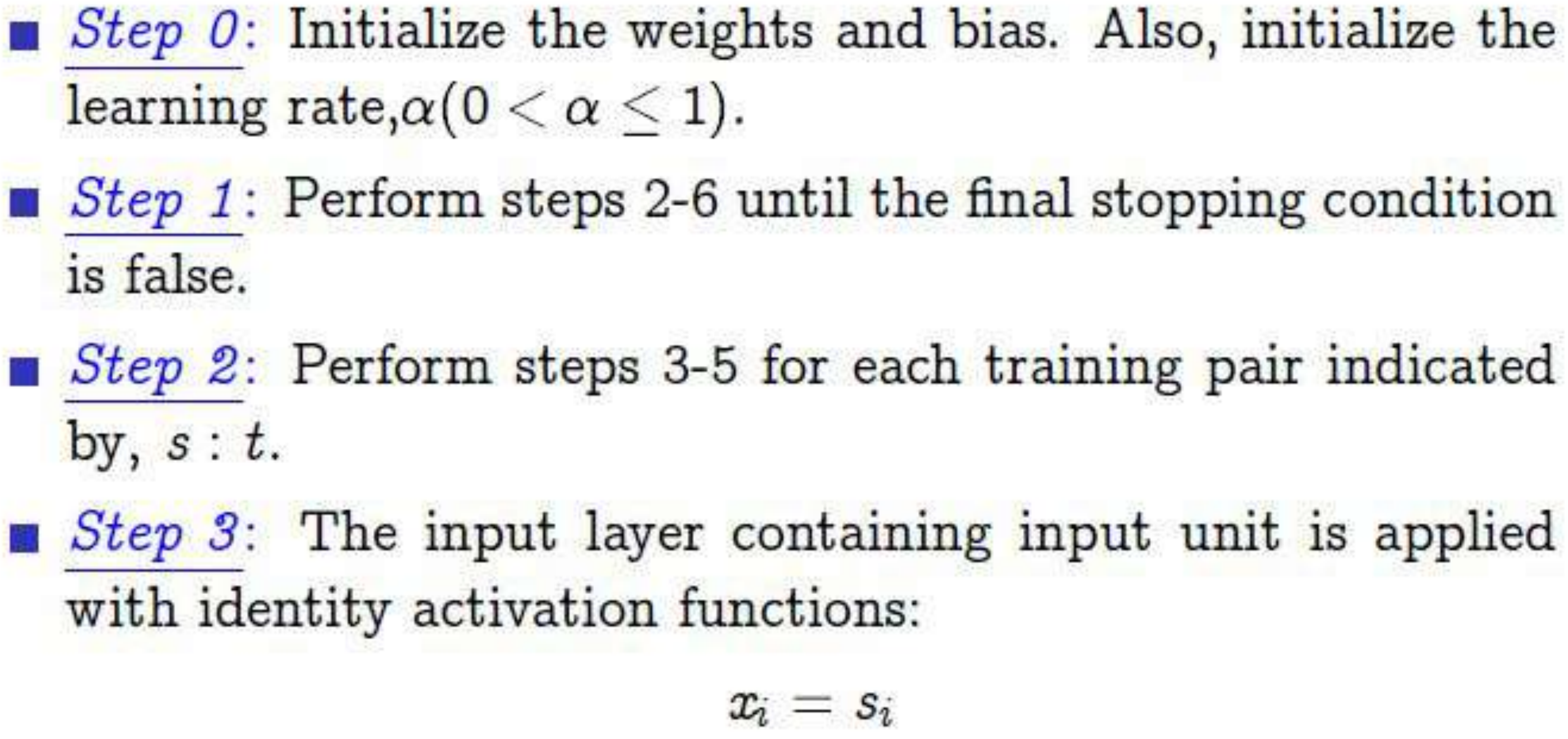


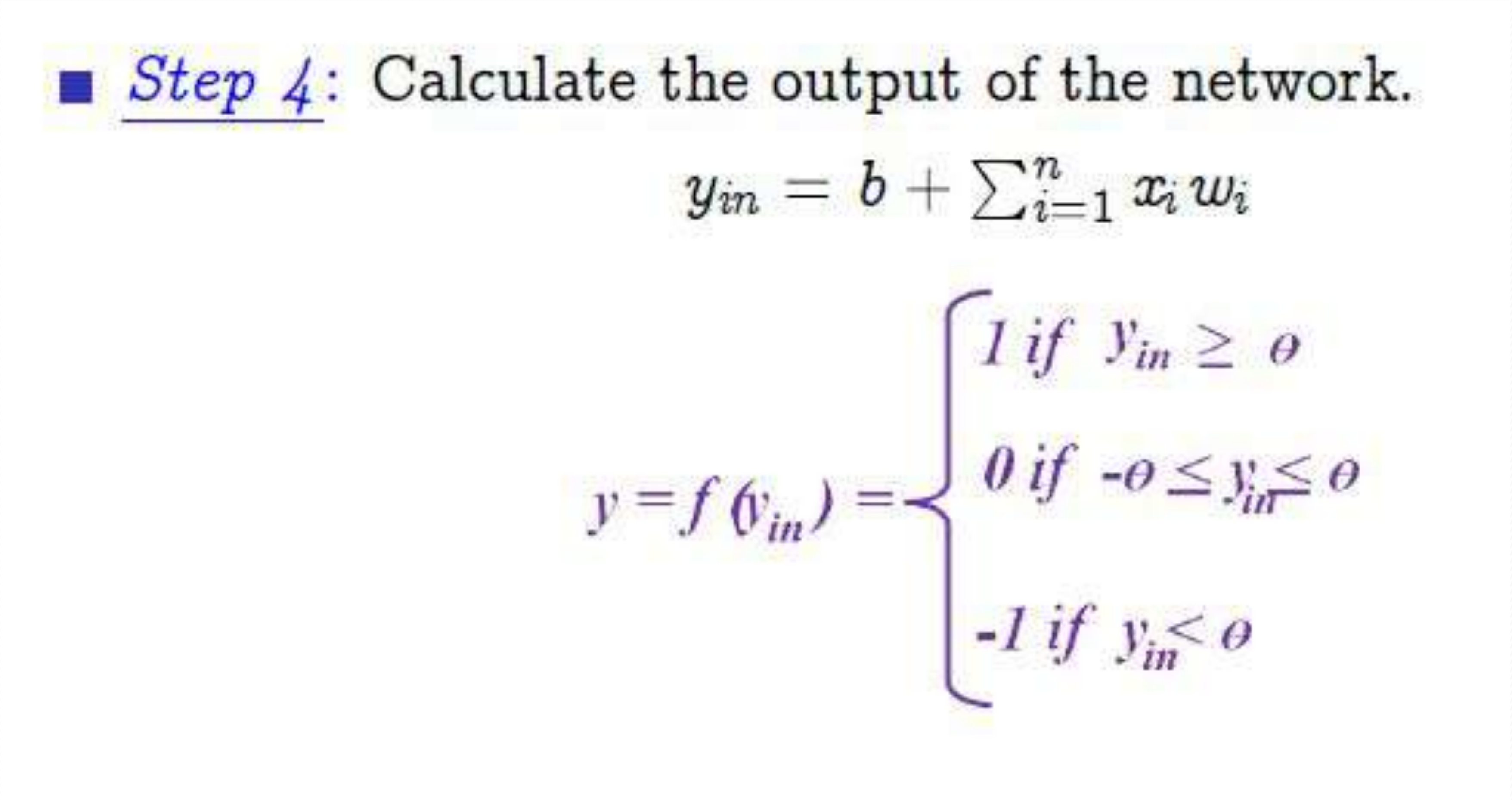
Figure: Flowchart for perceptron network with single output

Perceptron Training Algorithm for Single Output Class



- *Step 0*: Initialize the weights and bias. Also, initialize the learning rate, $\alpha (0 < \alpha \leq 1)$.
- *Step 1*: Perform steps 2-6 until the final stopping condition is false.
- *Step 2*: Perform steps 3-5 for each training pair indicated by, $s : t$.
- *Step 3*: The input layer containing input unit is applied with identity activation functions:

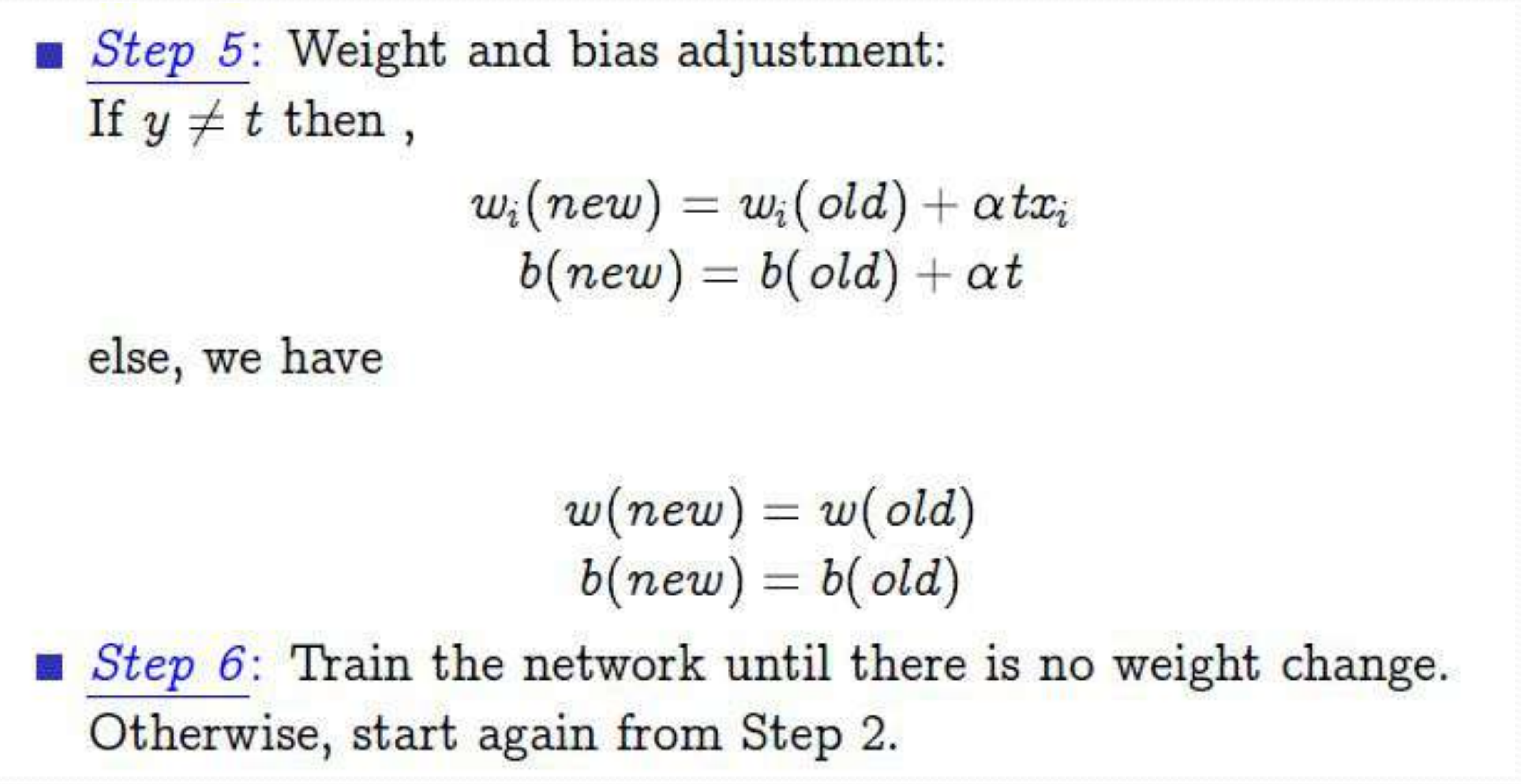
$$x_i = s_i$$



- *Step 4*: Calculate the output of the network.

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

$$y = f(y_{in}) = \begin{cases} 1 \text{ if } y_{in} \geq \theta \\ 0 \text{ if } -\theta \leq y_{in} \leq \theta \\ -1 \text{ if } y_{in} < \theta \end{cases}$$



- *Step 5*: Weight and bias adjustment:
  If $y \neq t$ then ,

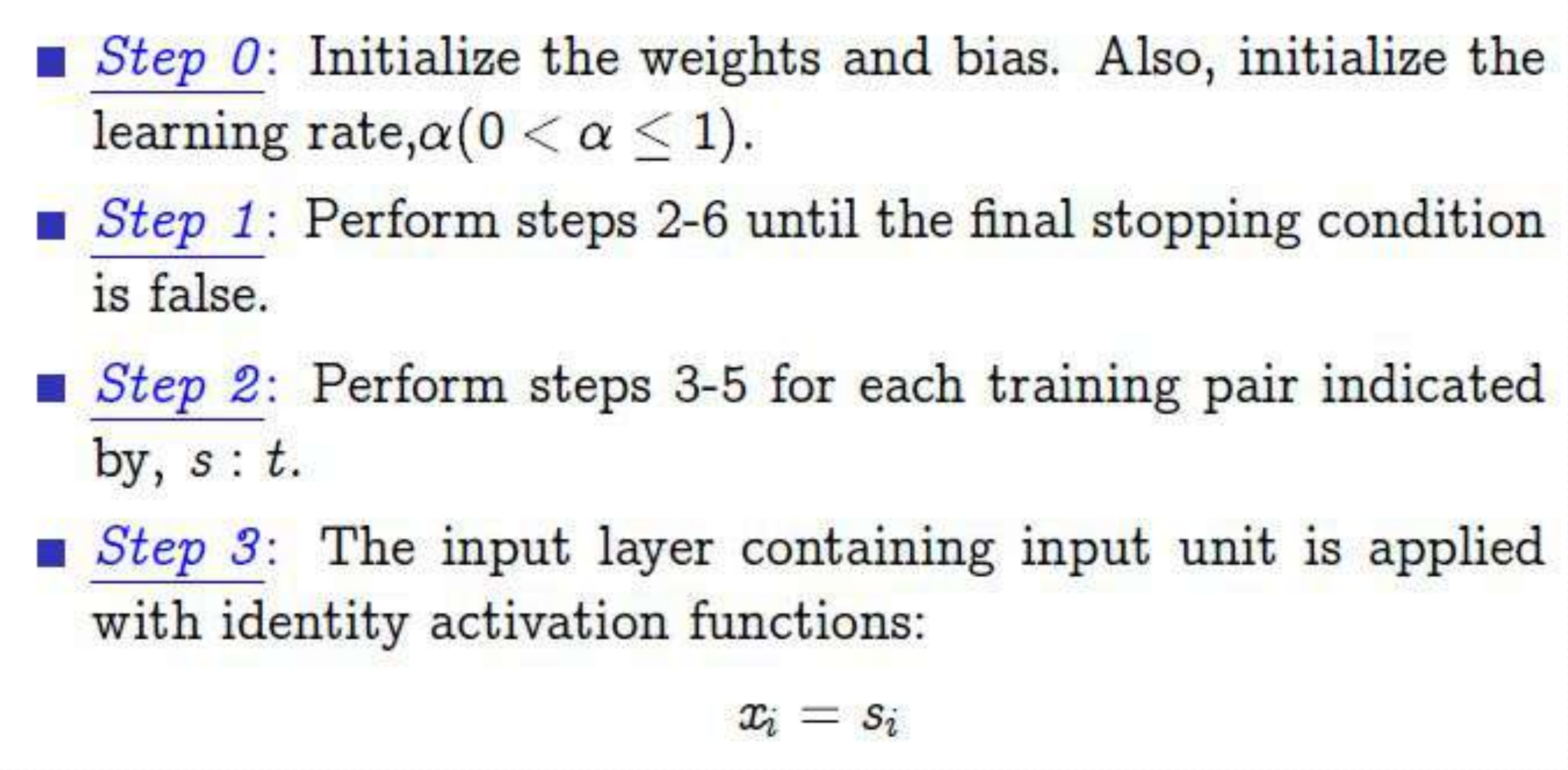
$$w_i(new) = w_i(old) + \alpha t x_i$$
$$b(new) = b(old) + \alpha t$$

  else, we have

$$w(new) = w(old)$$
$$b(new) = b(old)$$

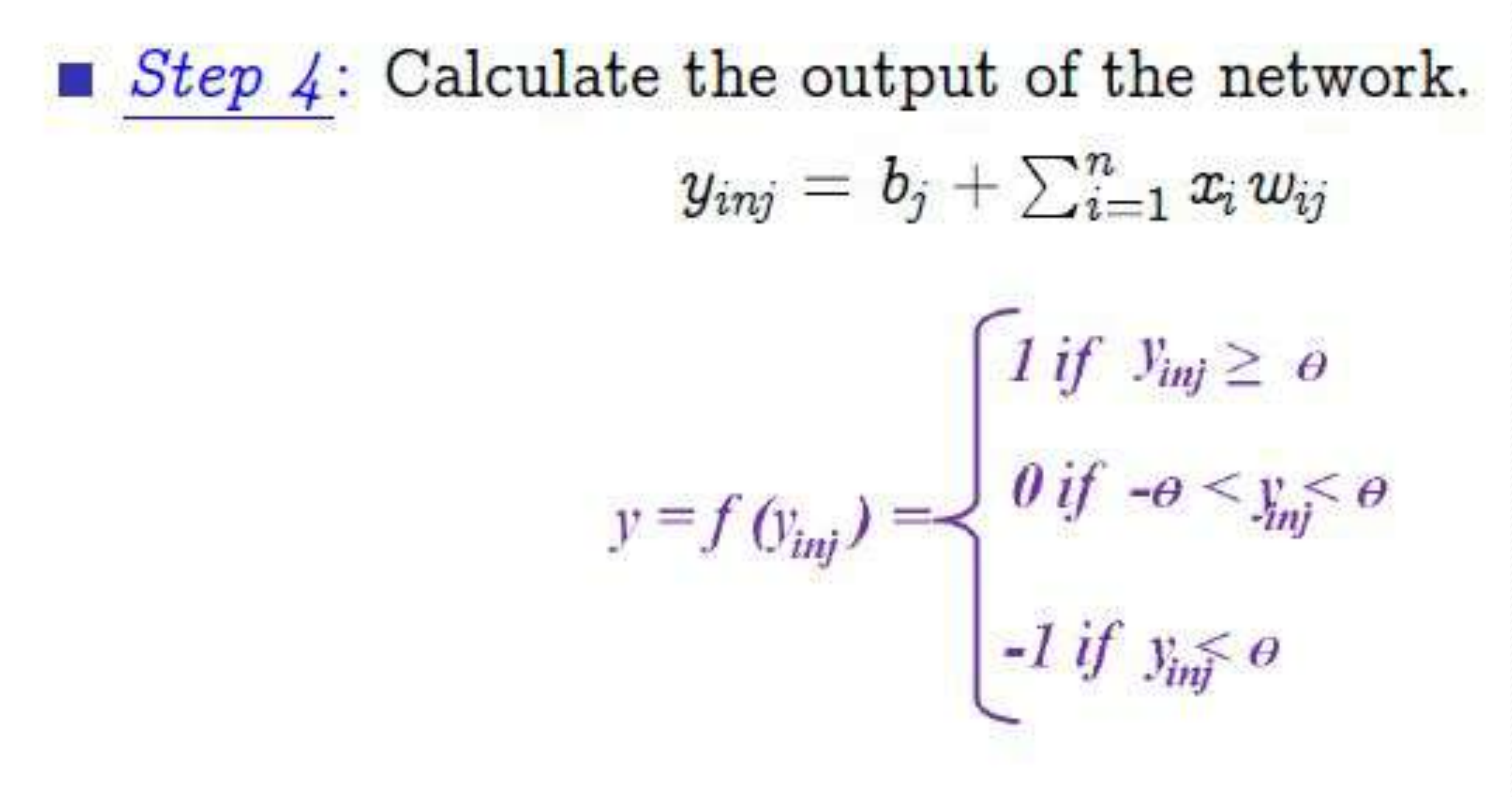
- *Step 6*: Train the network until there is no weight change.
  Otherwise, start again from Step 2.

Perceptron Training Algorithm for Multiple Output Class



- *Step 0*: Initialize the weights and bias. Also, initialize the learning rate,$\alpha (0 < \alpha \leq 1)$.
- *Step 1*: Perform steps 2-6 until the final stopping condition is false.
- *Step 2*: Perform steps 3-5 for each training pair indicated by, $s : t$.
- *Step 3*: The input layer containing input unit is applied with identity activation functions:

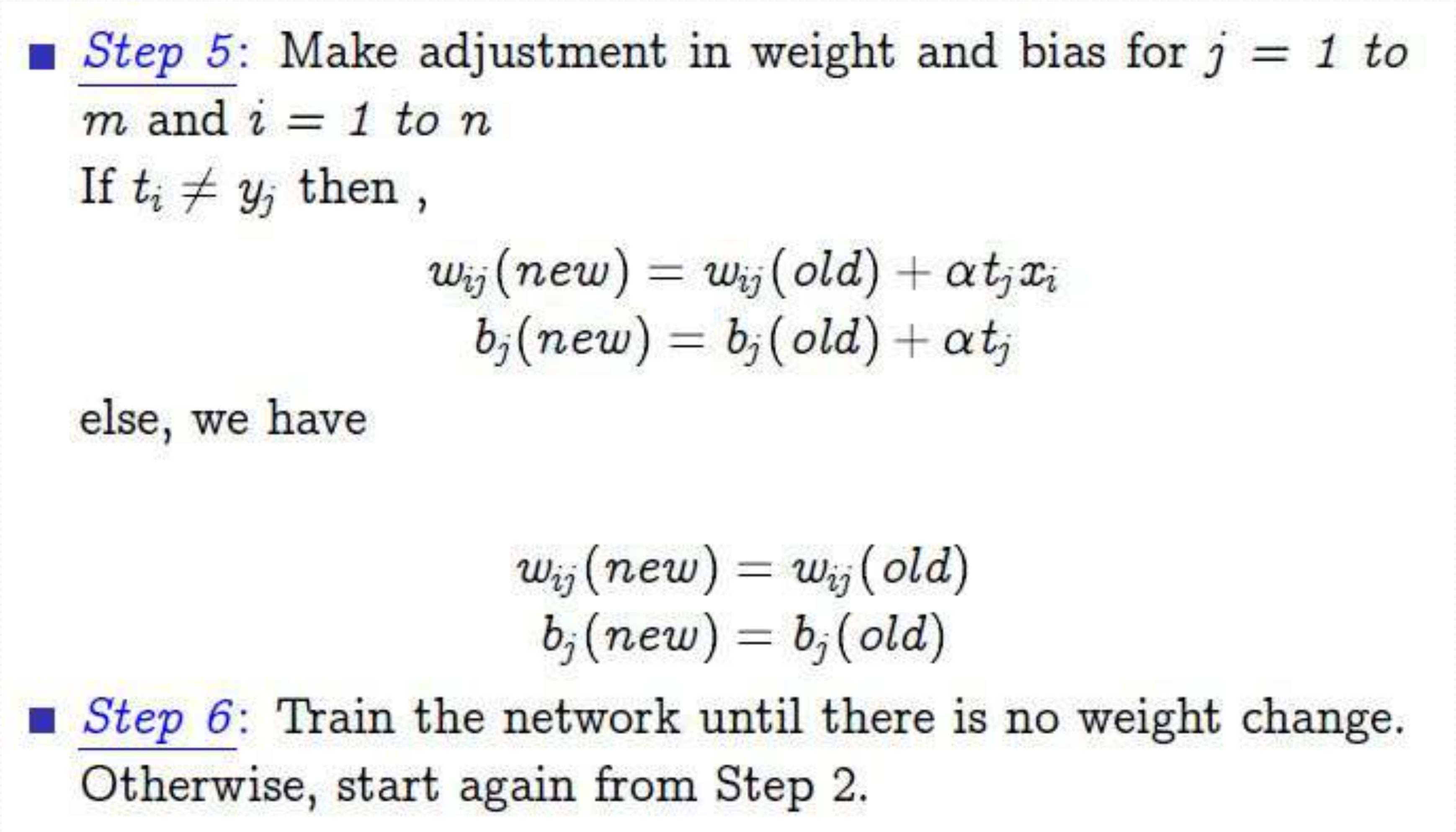
$$x_i = s_i$$



- *Step 4*: Calculate the output of the network.

$$y_{inj} = b_j + \sum_{i=1}^{n} x_i w_{ij}$$

$$y = f(y_{inj}) = \begin{cases} 1 \text{ if } y_{inj} \geq \theta \\ 0 \text{ if } -\theta < y_{inj} < \theta \\ -1 \text{ if } y_{inj} < \theta \end{cases}$$



- *Step 5*: Make adjustment in weight and bias for $j = 1$ *to* m and $i = 1$ *to* n

  If $t_i \neq y_j$ then ,

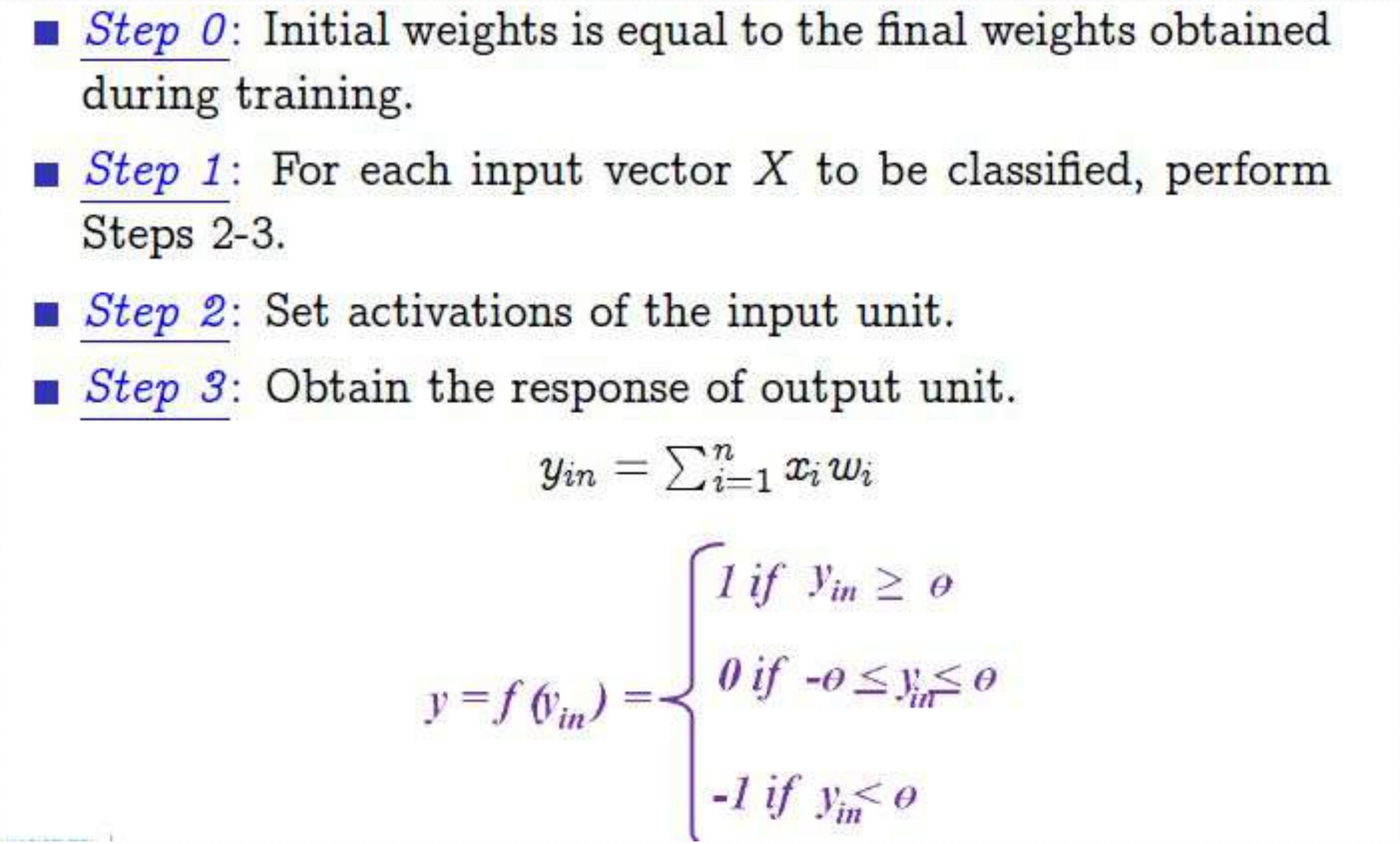
  $$w_{ij}(new) = w_{ij}(old) + \alpha t_j x_i$$
  $$b_j(new) = b_j(old) + \alpha t_j$$

  else, we have

  $$w_{ij}(new) = w_{ij}(old)$$
  $$b_j(new) = b_j(old)$$

- *Step 6*: Train the network until there is no weight change. Otherwise, start again from Step 2.

Perceptron Network Testing Algorithm



- *Step 0*: Initial weights is equal to the final weights obtained during training.
- *Step 1*: For each input vector X to be classified, perform Steps 2-3.
- *Step 2*: Set activations of the input unit.
- *Step 3*: Obtain the response of output unit.

$$y_{in} = \sum_{i=1}^{n} x_i w_i$$

$$y = f(y_{in}) = \begin{cases} 1 \text{ if } y_{in} \geq \theta \\ 0 \text{ if } -\theta \leq y_{in} \leq \theta \\ -1 \text{ if } y_{in} < \theta \end{cases}$$

Fuzzy Logic

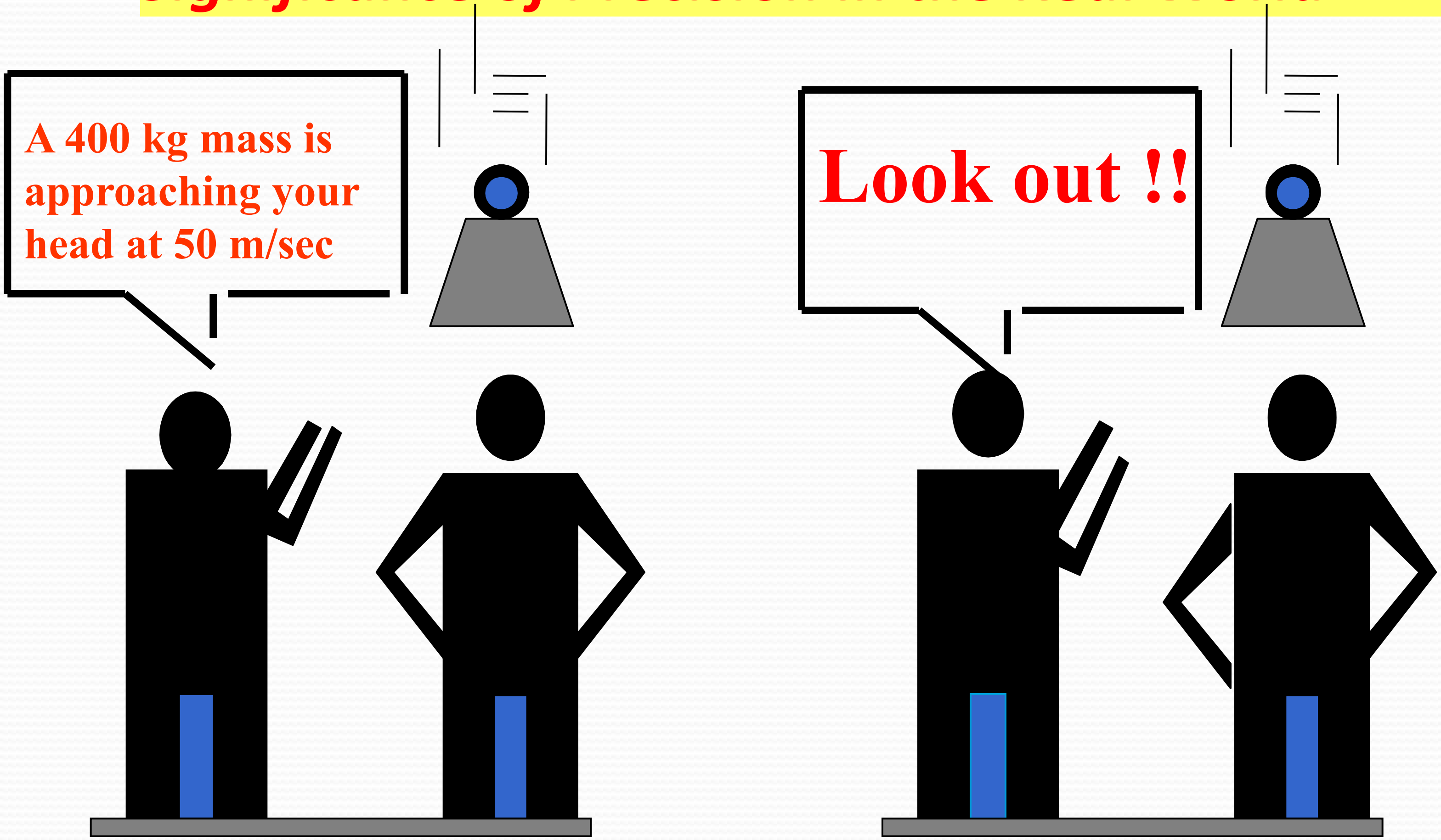
- ❑Humans are more efficient in dealing with fuzzy data than computers (e.g crossing a busy road).
- ❑Fuzzy logic is used to convey the human capability of handling fuzzy information to the computer
- ❑It is a mathematical tool to capture and handle the fuzzy data that is used in the natural language

Why Fuzzy Logic

- In the real word there exists much fuzzy knowledge, that is, knowledge which is vague, imprecise, uncertain, ambiguous, inexact, or probabilistic in nature.
- Human can use such information because the human thinking and reasoning frequently involve fuzzy information, possibly originating from inherently inexact human concepts and matching of similar rather than identical experience.
- The computing system, based upon classical set theory and two-valued logic, cannot give answers to some questions as a human does, because they do not have completely true answers.

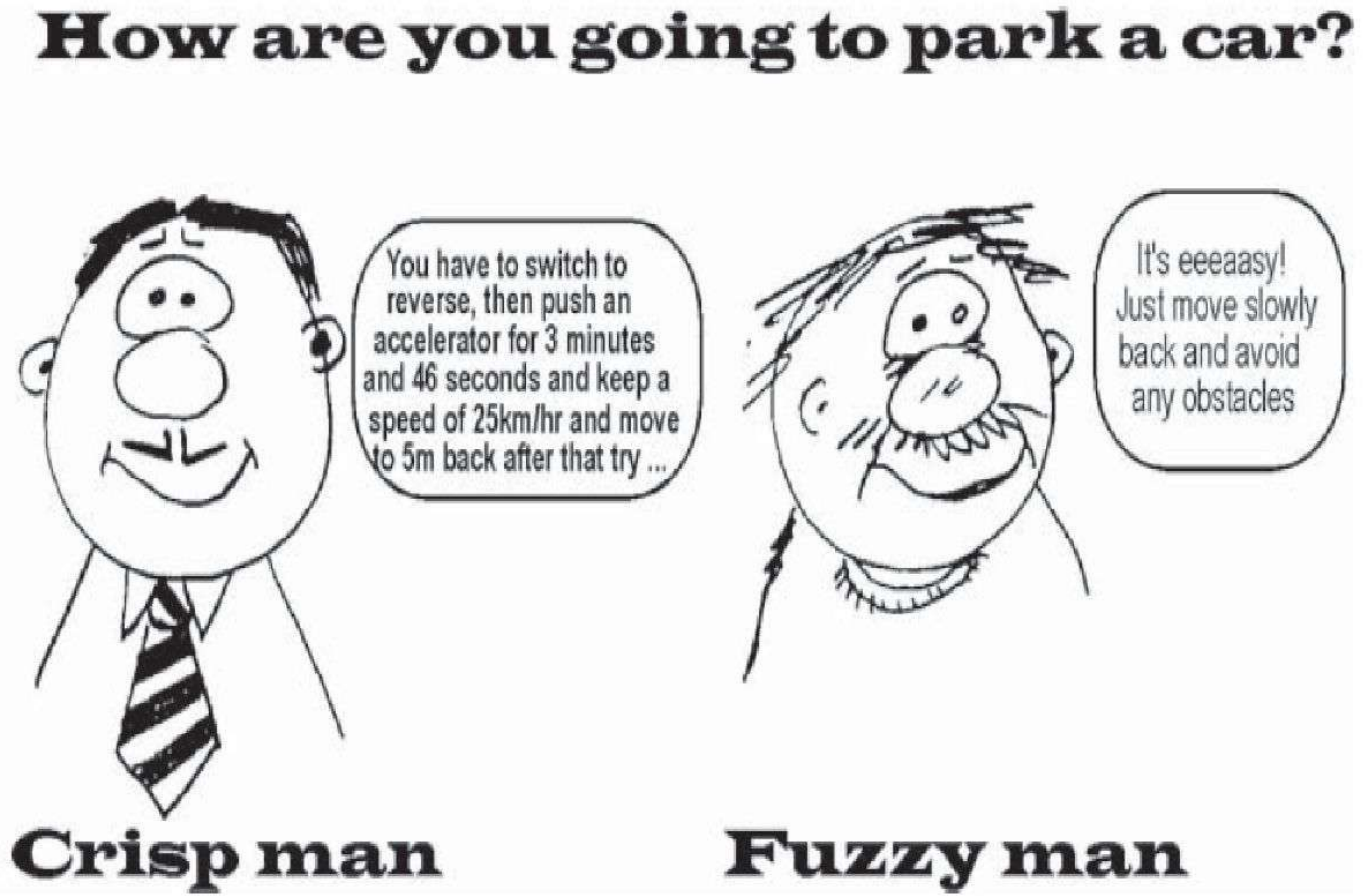
Fuzzy Logic?

Significance of Precision in the Real World



Precise man

Fuzzy man


How are you going to park a car?
You have to switch to reverse, then push an accelerator for 3 minutes and 46 seconds and keep a speed of 25km/hr and move to 5m back after that try ...
It's eeeaasy! Just move slowly back and avoid any obstacles
Crisp man
Fuzzy man

Types of uncertainties and its modeling

- Stochastic uncertainties:

(E.g the probability of hitting the target is 0.65)

- lexical uncertainties:

Examples include the expressions like

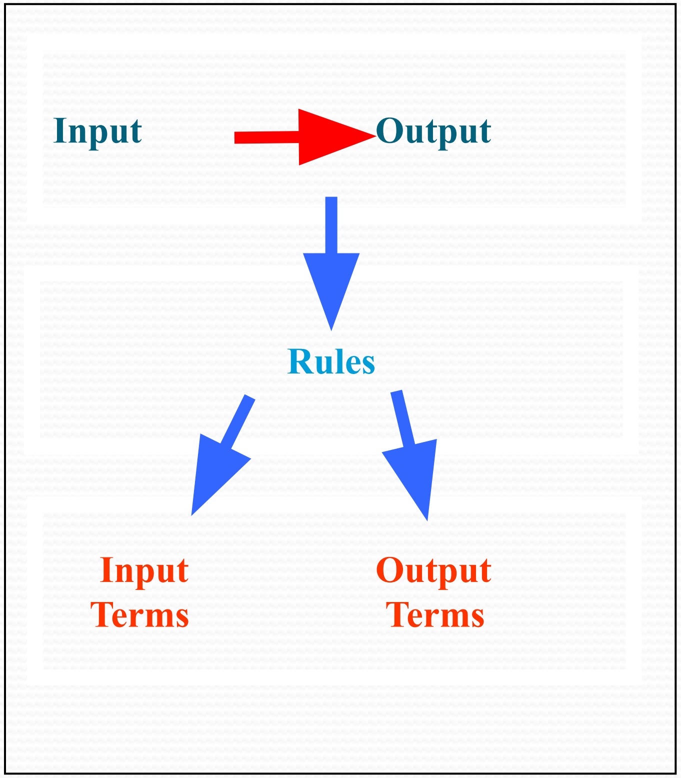
  - Healthy man.
  - Depressed patient.
  - Hot weather.

- Stochastic uncertainties are modeled by probabilities
- Lexical uncertainties are modeled by fuzzy sets

- Uncertainty can be caused by imprecision in measurement due to imprecision of tools or other factors. Uncertainty can also be caused by vagueness in the language objects and situations.

FL maps an input space to an output space using a list of if-then rules

The General Case



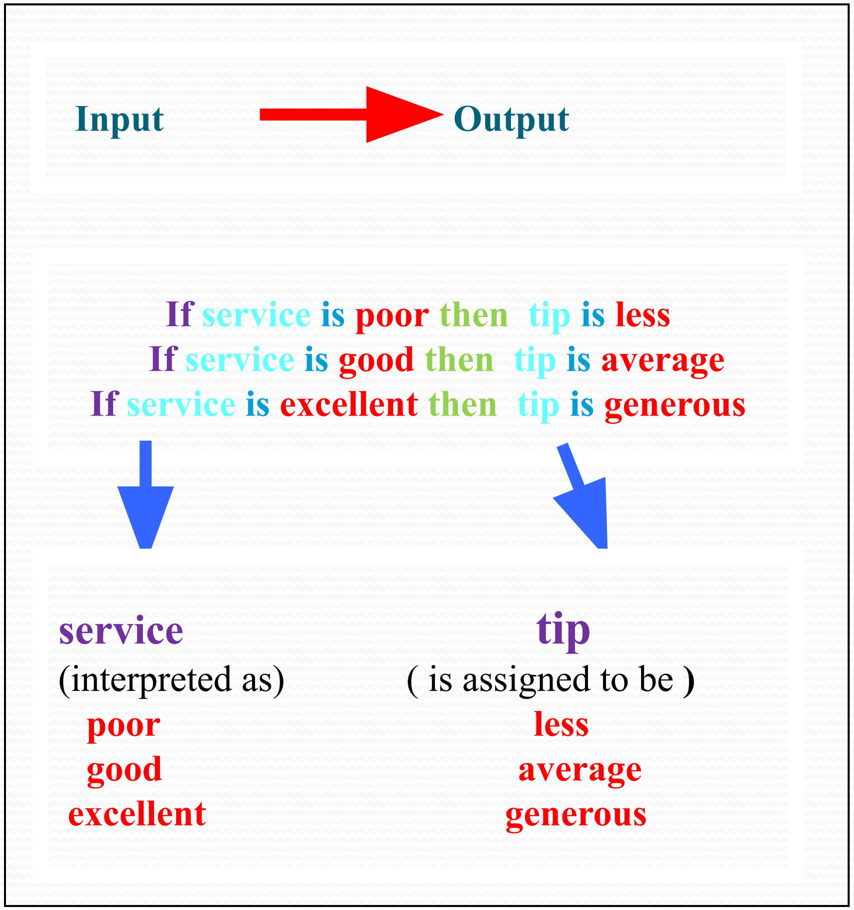
Input → Output

Rules

Input Terms

Output Terms

A Specific Example



Input → Output

If service is poor then tip is less

If service is good then tip is average

If service is excellent then tip is generous

service
(interpreted as)
poor
good
excellent

tip
(is assigned to be)
less
average
generous

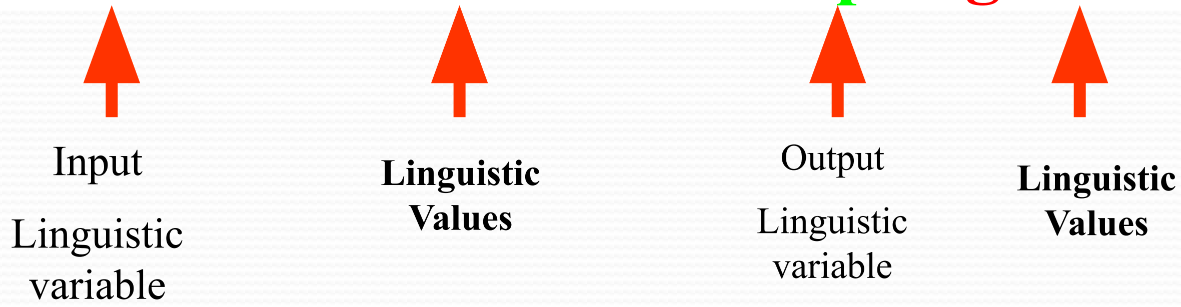
Linguistic Rules,Variables and values

Rules

If service is poor then tip is less

If service is good then tip is average

If service is excellent then tip is generous



Input

Linguistic variable

Linguistic Values

Output

Linguistic variable

Linguistic Values

Linguistic Rules of fuzzy systems are the laws it execute

Linguistic variables brings a concept from our everyday language

Linguistic values describe the characteristic of the *Linguistic variables*

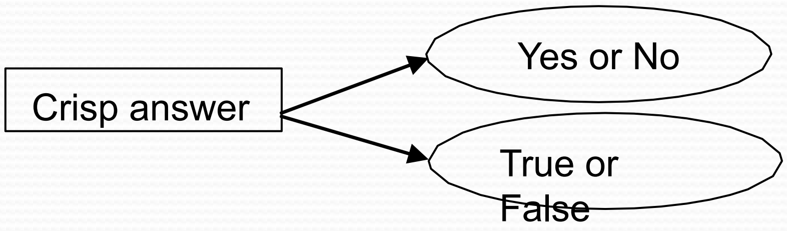
History of Fuzzy Logic

- Fuzzy set thory was introduced by Professor Lotfi Zadeh (USA) in 1965 as an extension of the classical set thory
- 1972 First working group on fuzzy systems in Japan by Toshiro Terano
- 1973 A paper on fuzzy algorithms by Zadeh (USA)
- 1974 Steam engine control by Ebrahim Mamdani (UK)
- Too many events, inventions and projects to mention till 1991
- After 1991 fuzzy technology came out of scientific laboratories and became an industrial tool.
- In the last two decades, the fuzzy sets theory has established itself as a new methodology for dealing with any sort of ambiguity and uncertainty.

Fuzzy Sets

- A classical *set X* is a collection of definite, distinguishable objects of our intuition that can be treated as a whole. The objects are the *members* of *X*
- A crisp (classical) set is a set for which each value is either included or not included in the set.
- For a fuzzy set, every value has a membership value, and so is a member to some extent.
- The ***membership value*** defines the extent to which a variable is a member of a fuzzy set.
- The membership value is from 0 (not at all a member of the set) to 1.

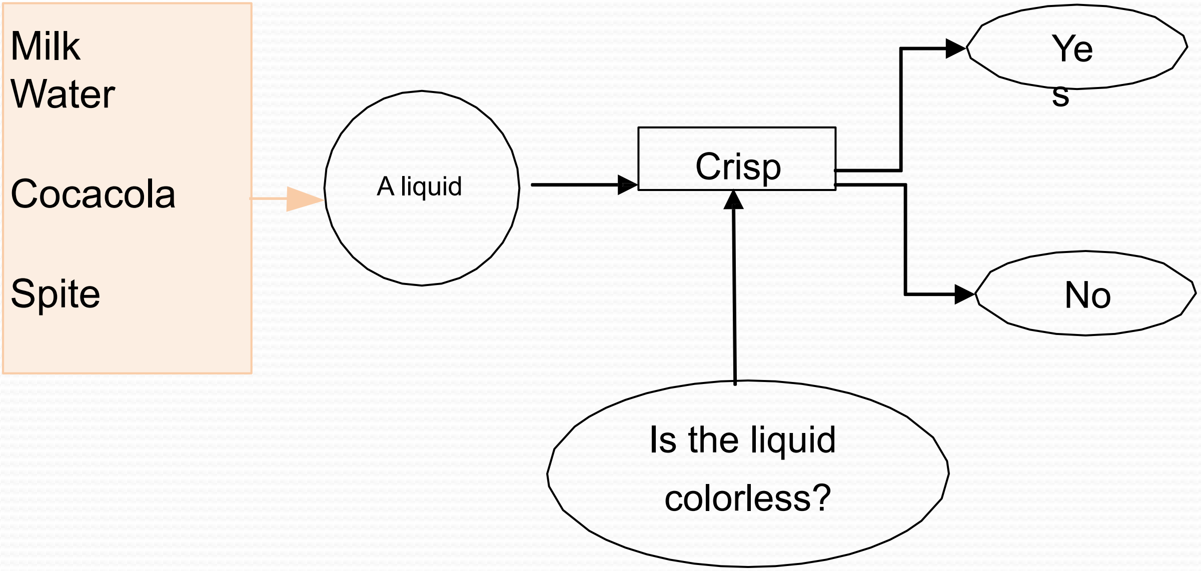
Fuzzy logic vs. Crisp logic



Crisp answer

Yes or No

True or False



Milk
Water

Cocacola

Spite

A liquid

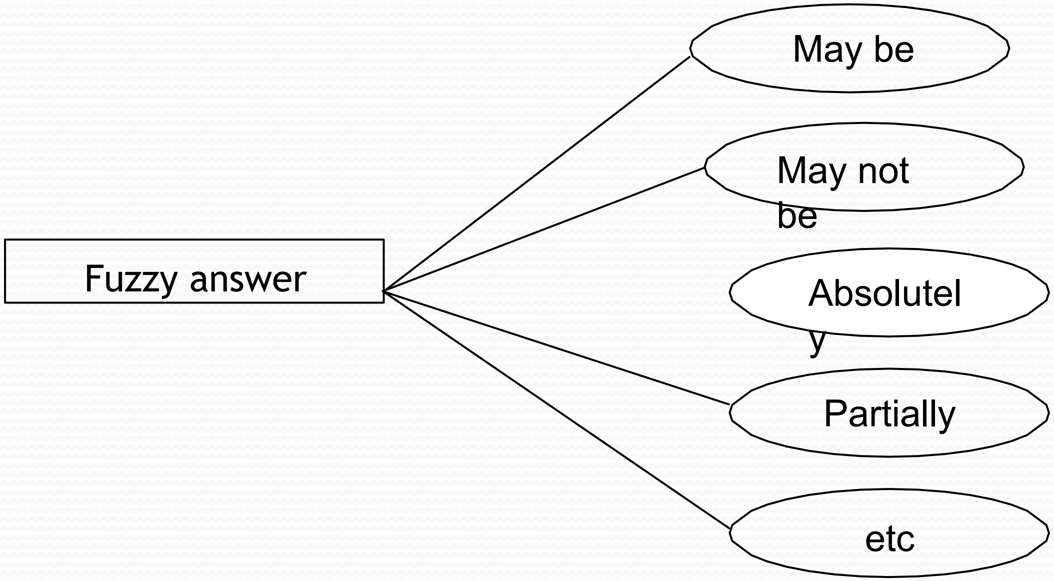
Crisp

Ye
s

No

Is the liquid colorless?

Fuzzy logic vs. Crisp logic



Fuzzy answer

May be

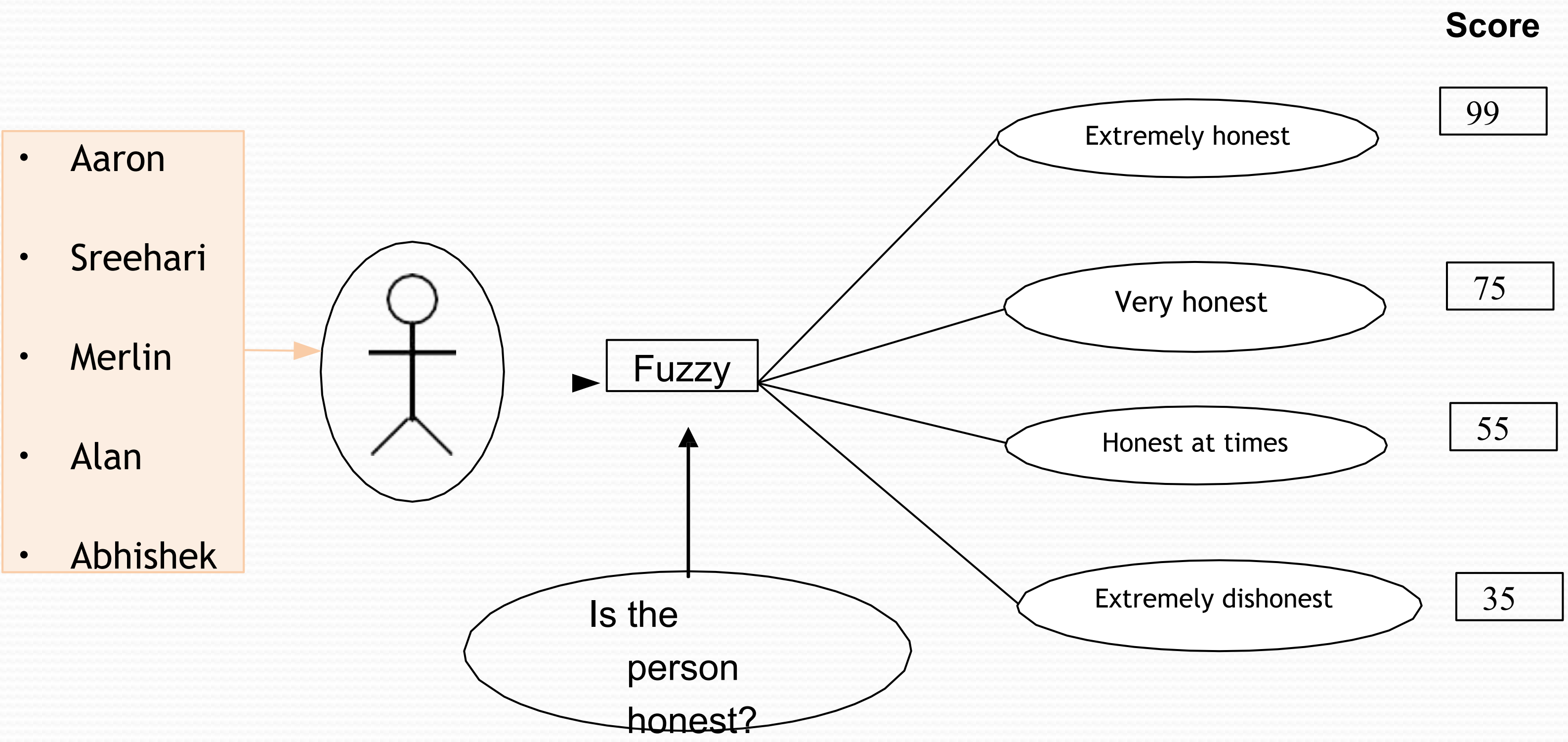
May not be

Absolutely

Partially

etc

Fuzzy logic vs. Crisp logic



Score

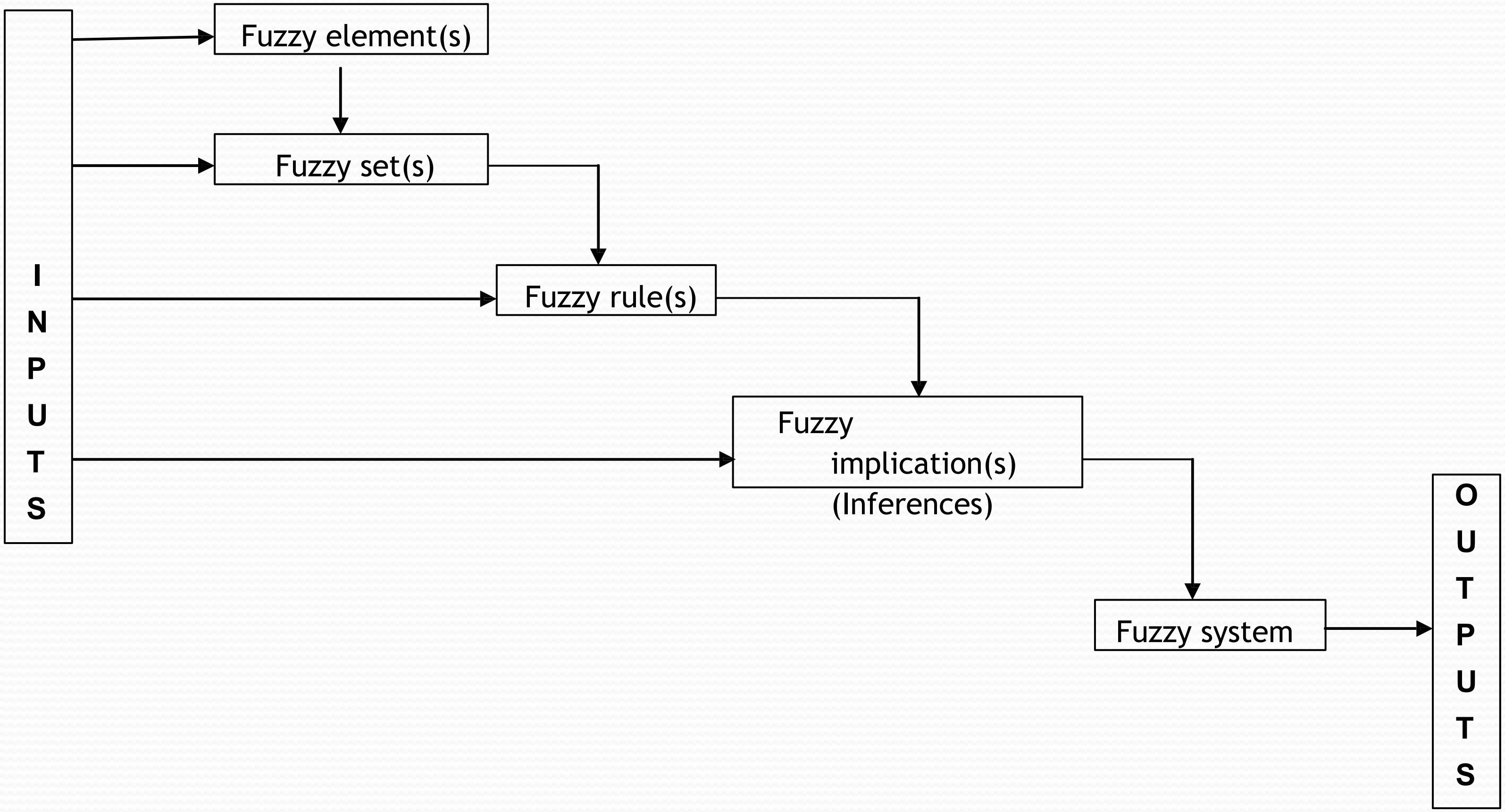
- Aaron
- Sreehari
- Merlin
- Alan
- Abhishek

Fuzzy

Is the person honest?

Extremely honest	99
Very honest	75
Honest at times	55
Extremely dishonest	35

Concept of fuzzy system



INPUTS

Fuzzy element(s)

Fuzzy set(s)

Fuzzy rule(s)

Fuzzy implication(s) (Inferences)

Fuzzy system

OUTPUTS

Concept of fuzzy set

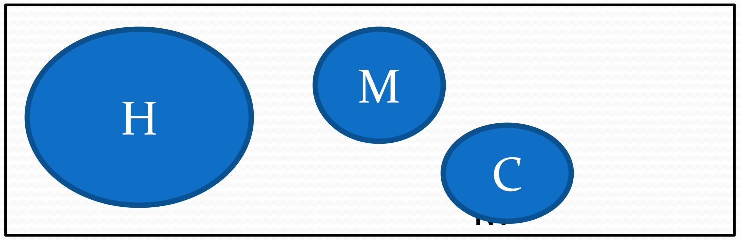
To understand the concept of **fuzzy set** it is better, if we first clear our idea of **crisp set**.

X = The entire population of India.

H = All Hindu population = $\{ h_1, h_2, h_3, \ldots , h_L \}$

M = All Muslim population = $\{ m_1, m_2, m_3, \ldots , m_N \}$

C = All Christian population = $\{ c_1, c_2, c_3, \ldots , c_N \}$



Here, All are the sets of finite numbers of individuals. Such a set is called crisp set.

Refreshing Crisp sets

Example of a fuzzy set

Let us discuss about fuzzy set.

X = All students in EC360 Soft computing

S = All **Good students**.

S = { (s, g) | s ∈ X } and g(s) is a measurement of goodness of the student *s*.

Example:

S = { (Aaron, 0.8), (Merlin, 0.7), (Sreehari, 0.4), (Sangeeth, 0.9) } etc.

Fuzzy Set vs Crisp Set

Crisp set	Fuzzy Set
1. $S = \{ s \mid s \in X \}$	1. $F = (s, \mu) \mid s \in X$ and $\mu(s)$ is the degree of s.
2. It is a collection of elements.	2. It is collection of ordered pairs.
3. Inclusion of an element $s \in X$ into S is crisp, that is, has strict boundary **yes** or **no**.	3. Inclusion of an element $s \in X$ into F is fuzzy, that is, if present, then with a degree of **membership**.

Fuzzy set Representation

- A fuzzy set can be expressed as a set of ordered pairs

$$A = \{(x, \mu_A(x)) | x \in X\}$$

Fuzzy set

Membership
function
(MF)

Universe or
universe of discourse

- ***A fuzzy set is totally characterized by a membership function (MF).***
- ***MF maps each element of X to a membership grade (or value) beteewn 0 and one***

Alternate Notation

- A fuzzy set A can be alternatively denoted as follows:

X is discrete →

$$A = \sum_{x_i \in X} \mu_A(x_i) / x_i$$

X is continuous →

$$A = \int_X \mu_A(x) / x$$

- **Note that Σ and integral signs stand for the union of membership grades; "/" stands for a marker and does not imply division.**
- Crisp Sets ≤ Fuzzy Sets or in other words, Crisp Sets are Special cases of Fuzzy Sets

Example of Fuzzy set Representation

- A= { (x1,0.8), (x2, 0.3), (x3, 0.1), (x4, 0.9) }
- Can be represented in another way as
- A= 0.8/x1 + 0.3/x2 + 0.1/x3 + 0.9/x4

Example (Discrete Universe)

$$U = \{1, 2, 3, 4, 5, 6, 7, 8\}$$ —— # courses a student may take in a semester.

$$A = \left\{ \begin{array}{cccc} (1, 0.1) & (2, 0.3) & (3, 0.8) & (4, 1) \\ (5, 0.9) & (6, 0.5) & (7, 0.2) & (8, 0.1) \end{array} \right\}$$ —— appropriate # courses taken

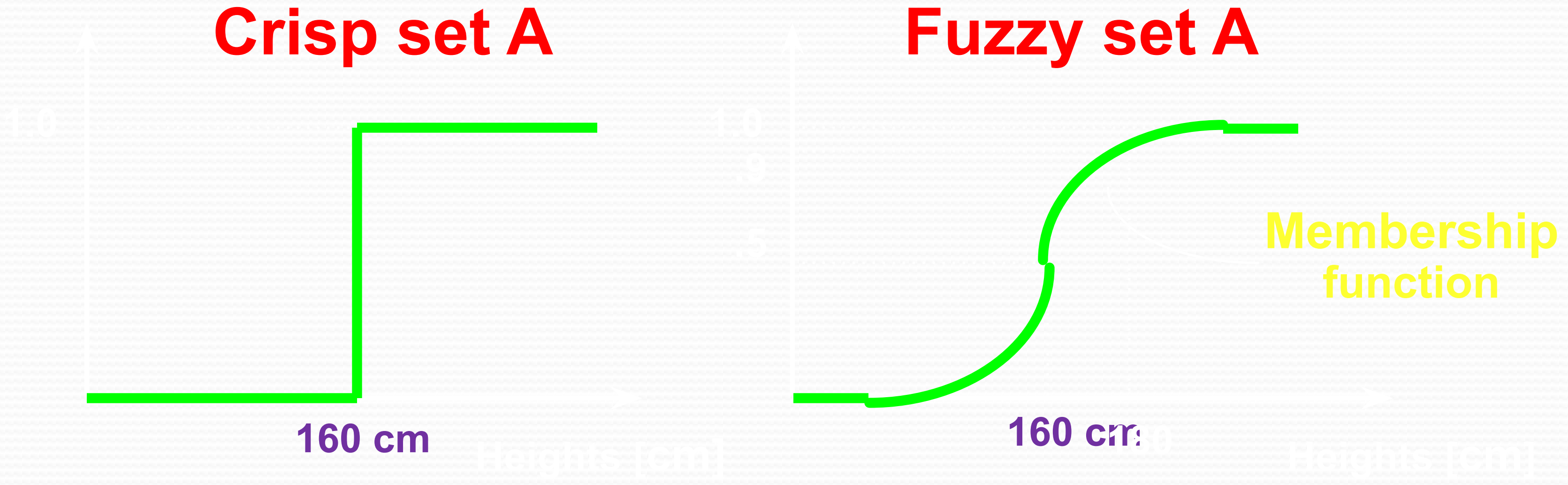
Alternative Representation:

$$A = 0.1/1 + 0.3/2 + 0.8/3 + 1.0/4 + 0.9/5 + 0.5/6 + 0.2/7 + 0.1/8$$

Fuzzy Sets

- Sets with fuzzy boundaries

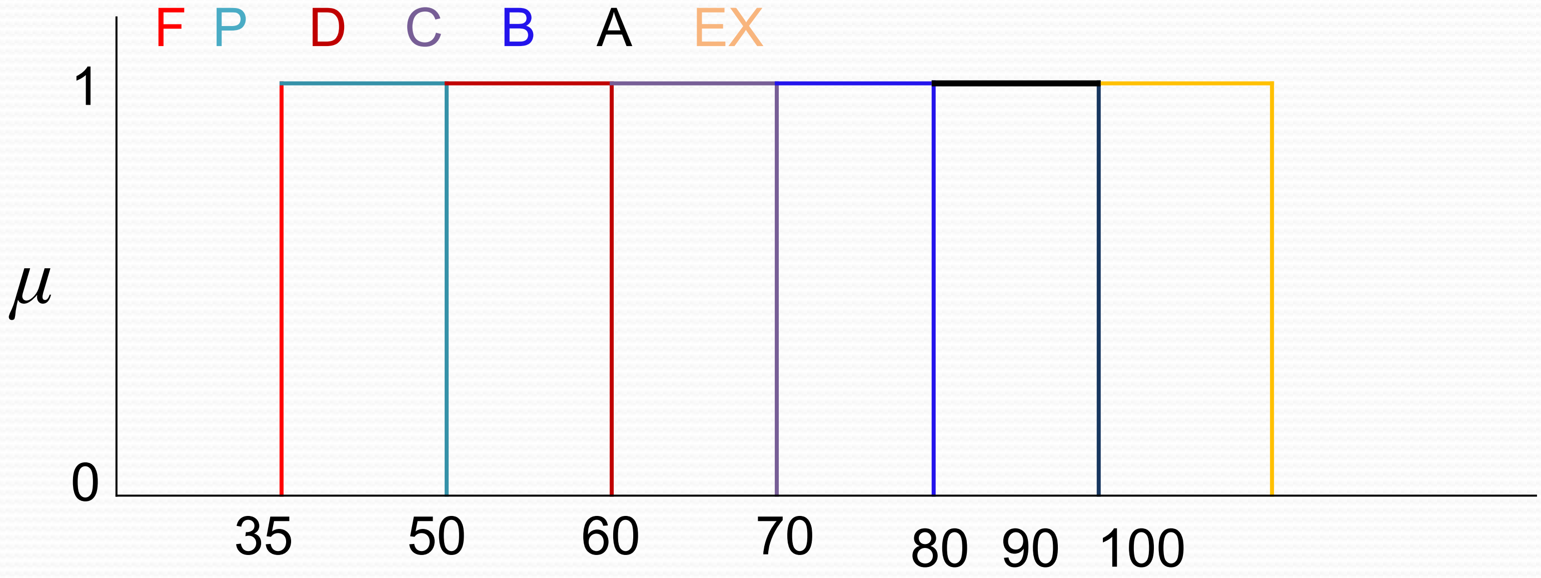
A = Set of tall people



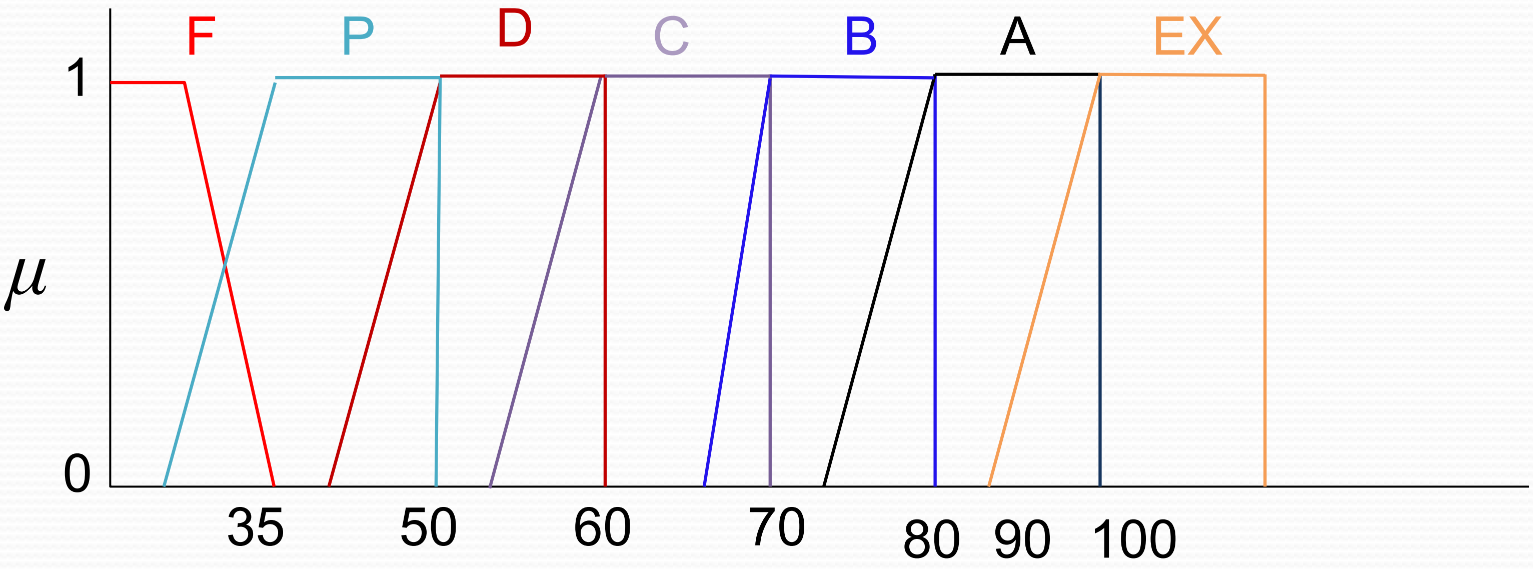
Example: Course evaluation in a crisp way

1 EX = Marks ≥ 90
2 A = 80 ≤ Marks <
3 90 B = 70 ≤ Marks
4 < 80 C = 60 ≤
5 Marks < 70 D = 50
6 ≤ Marks < 60 P =
7 35 ≤ Marks < 50 F
= Marks < 35

Example: Course evaluation in a crisp way



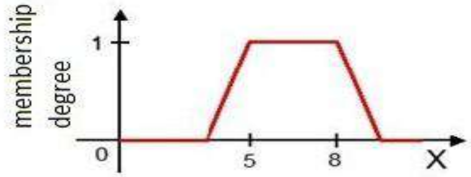
Example: Course evaluation in a fuzzy way



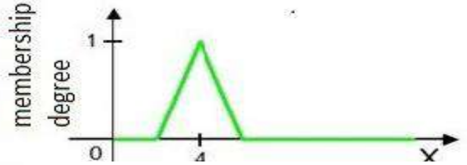
Graphical Representation

Union = Maximum, Intersection=Minimum

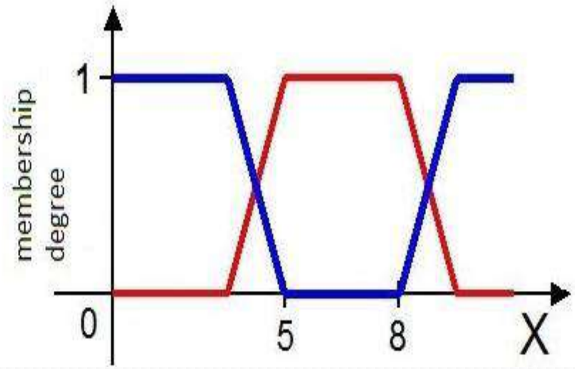
A



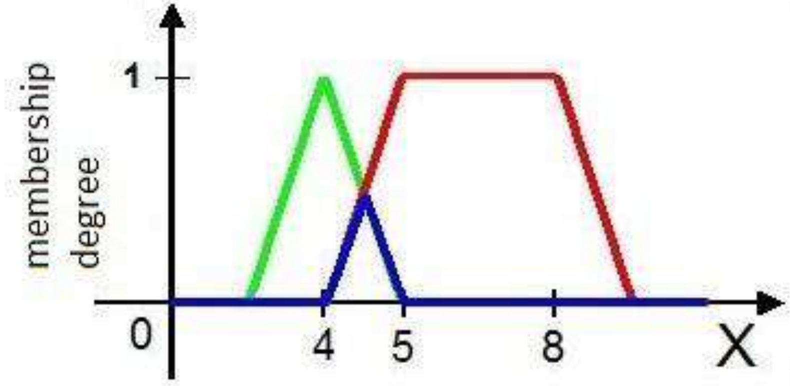
B



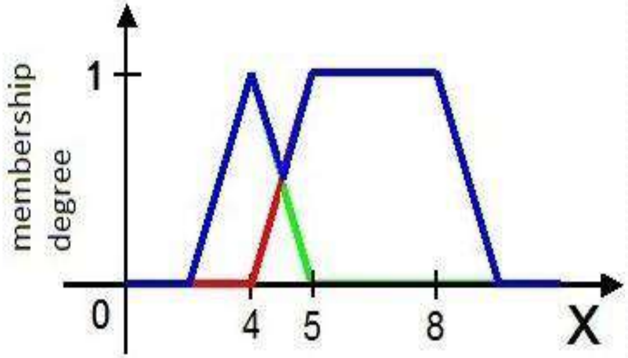
$\bar{A}$



A⋂B

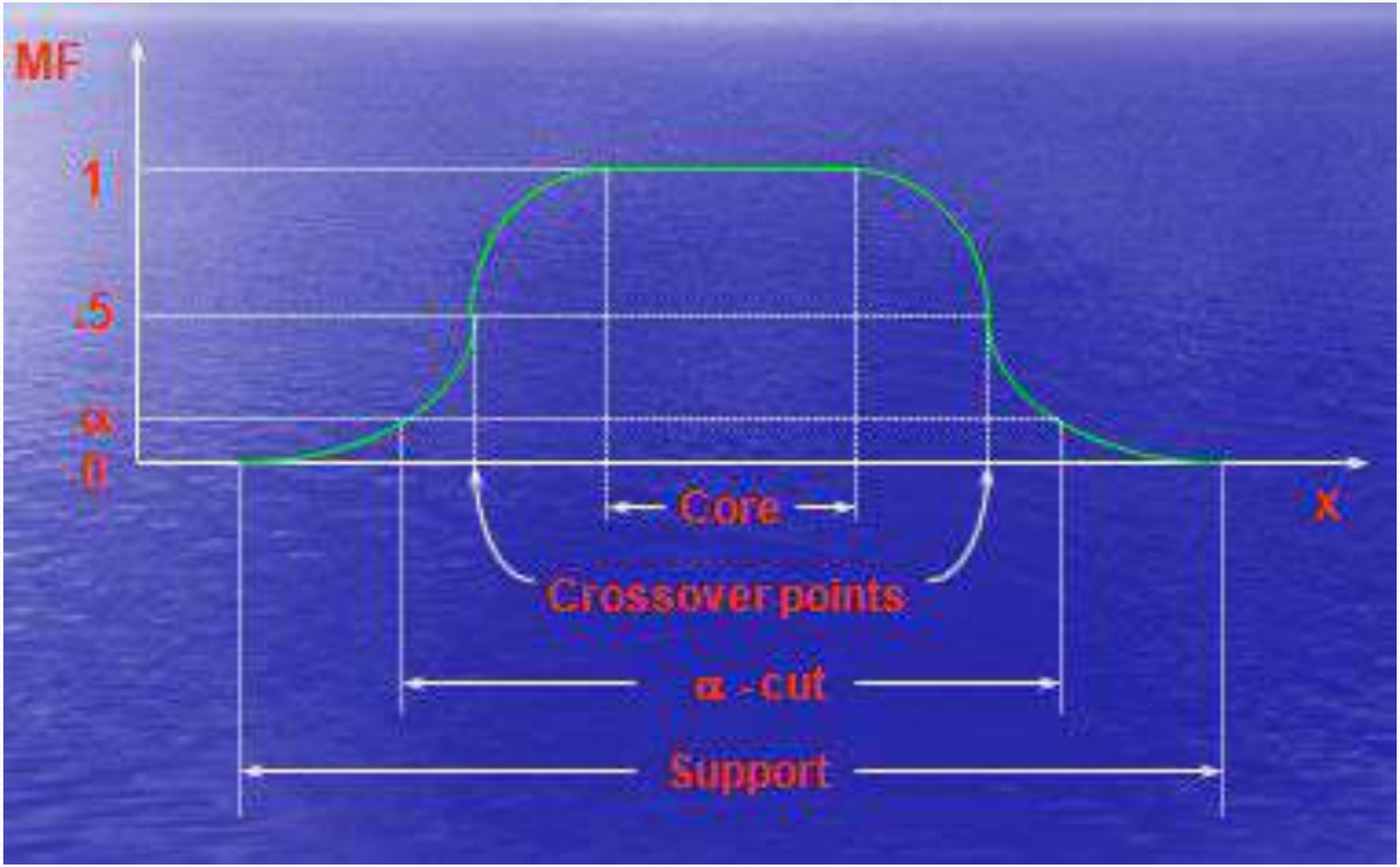


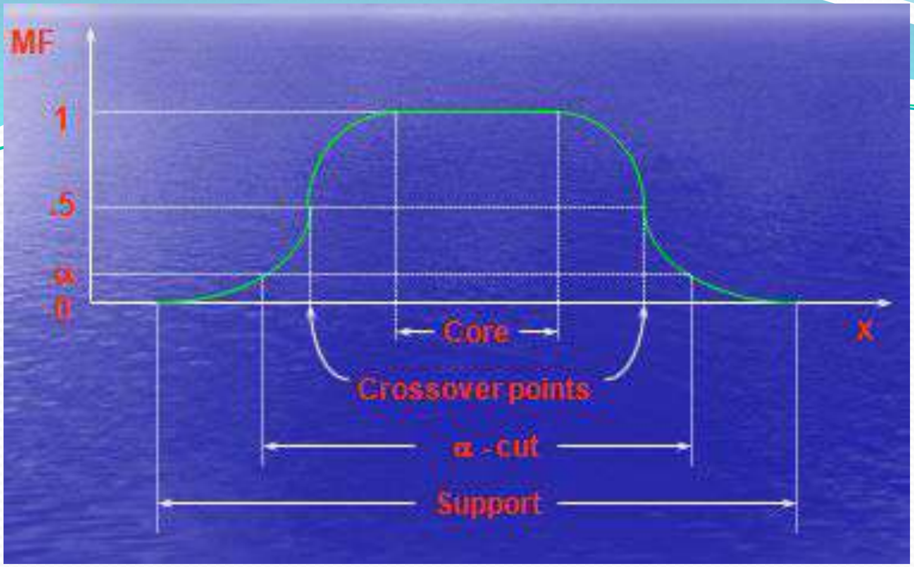
A⋃B



Some Definitions

- Support
- Core
- Normality
- Crossover points
- Fuzzy singleton
  - α-cut, strong α -cut
- Convexity
- Bandwidth
- Symetricity

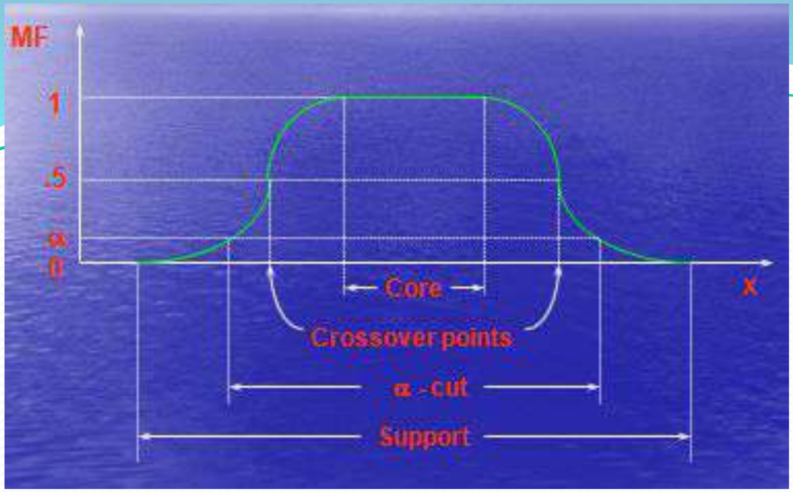

MF
1
.5
α
0
X
Core
Crossover points
α - cut
Support



- Support
- The support S(A) of a fuzzy set A is the crisp set of all the elements of the universal set (UOD) for which membership function has non-zero value

$$S\ (A) = \{ u \in U / \mu_A(u) > 0 \}$$

$$Support\ (A) = \left\{ x \middle| \mu_A(x) > 0 \right\}$$

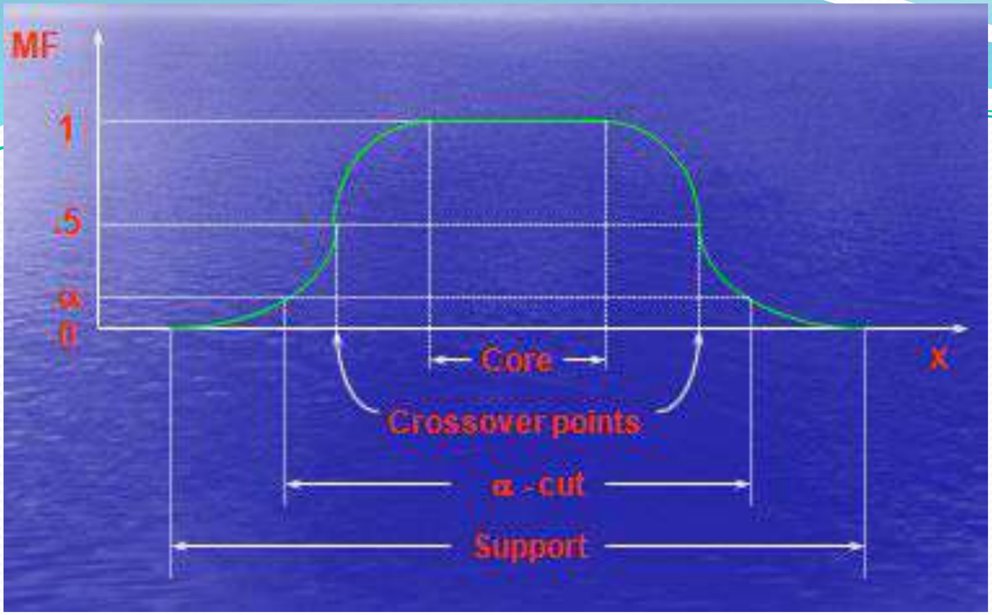


α – cut (or α level) set

- The set of elements that belong to the fuzzy set A at least to the degree α is called the α-level-set or α-cut-set $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$

- Strong α cut $A'_\alpha = \{x | \mu_A(x) > \alpha\}$

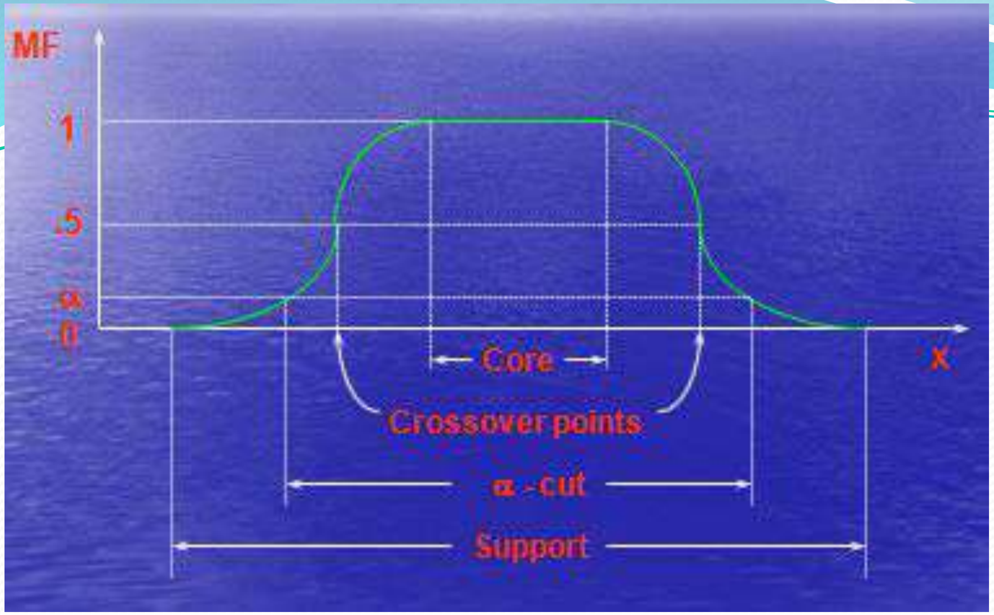
An α-cut set is crisp or fuzzy?



Crossover point

- The element of the universal set, for which the membership function has the value of 0.5, is called a crossover point.

$$Crossover(A) = \{x \mid \mu_A(x) = 0.5\}$$



Core:

- Is the set of all elements x in X that belong to the fuzzy set A such that $\mu_A(x)=1$:

$$Core(A)=\{x \mid \mu_A(x)=1\}$$

Height of a fuzzy set

The height of a fuzzy set A, hgt(A) is given by a supremum of the membership function over all u∈U

$$hgt(A) = sup_U\ \mu_A(\ u\ )$$

(Supremum in this definition means the highest possible (or almost possible) degree.)

Normality

A fuzzy set is normal if its core is nonempty. In other words, we can always find a point x ε X such that $\mu_A(x)=1$

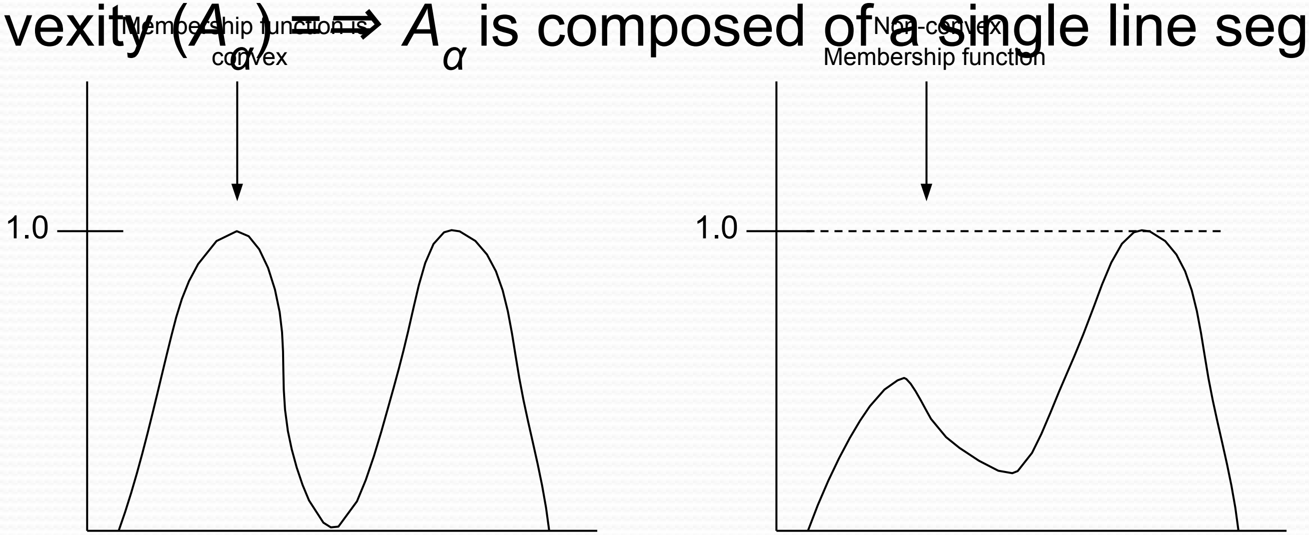
Convexity : A fuzzy set A is convex if and only if for any x_1 and $x_2 \in X$

and any $\lambda \in [0, 1]$

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2))$$

Note :

- A is convex if all its α- level sets are convex.
- Convexity $(A_\alpha) \Longrightarrow A_\alpha$ is composed of a single line segment only.



Bandwidth :

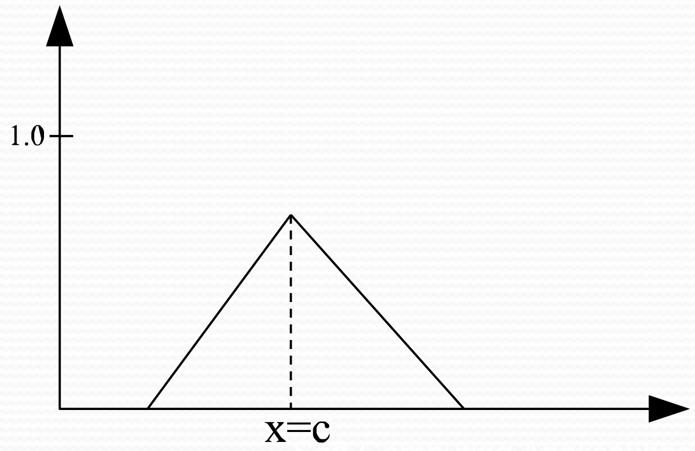
For a normal and convex fuzzy set, the bandwidth (or width) is defined as the distance between the two unique crossover points:

Bandwidth(A) = | x_1 - x_2 |

where $\mu_A(x_1) = \mu_A(x_2) = 0.5$

Symmetry :

A fuzzy set A is symmetric if its membership function around a certain point $x = c$, namely $\mu_A(x + c) = \mu_A(x - c)$ for all x $\in$ X.



Fuzzy terminologies: Open and Closed

A fuzzy set *A* is

Open left

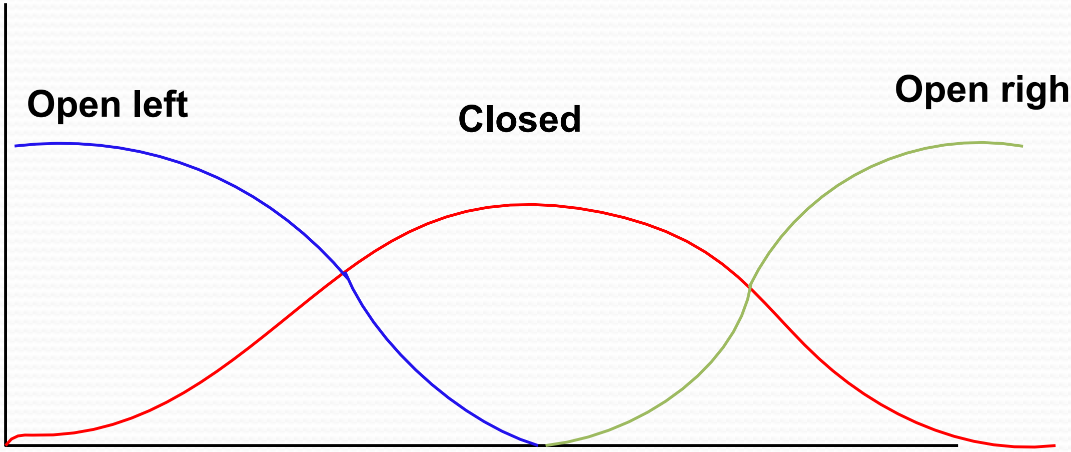
If $\lim_{x \to -\infty} \mu_A(x) = 1$ and $\lim_{x \to +\infty} \mu_A(x) = 0$

Open right

If $\lim_{x \to -\infty} \mu_A(x) = 0$ and $\lim_{x \to +\infty} \mu_A(x) = 1$

Closed

If : $\lim_{x \to -\infty} \mu_A(x) = \lim_{x \to +\infty} \mu_A(x) = 0$



Singleton

- It is a fuzzy set whose support is a single point in X with $\mu_A(x)=1$

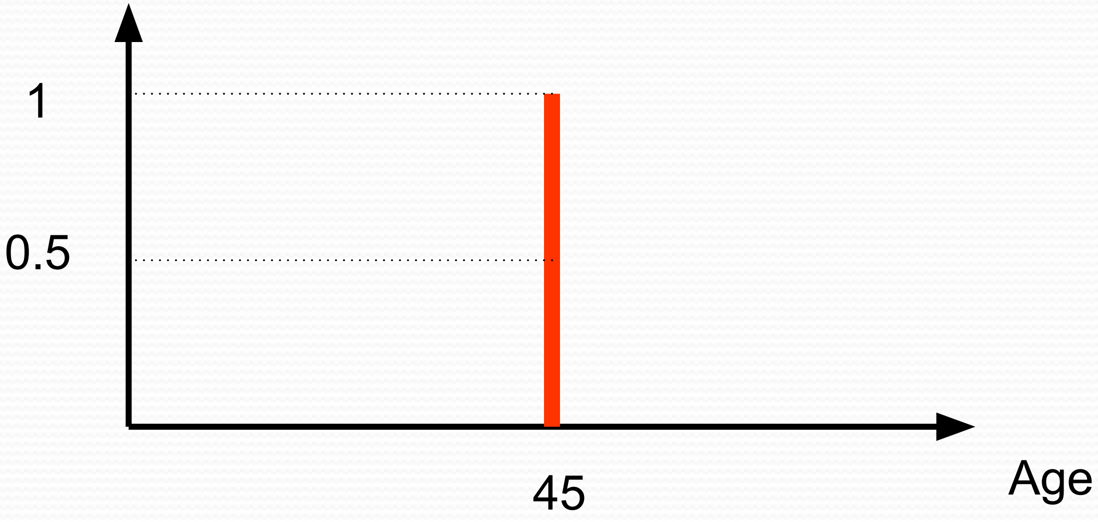
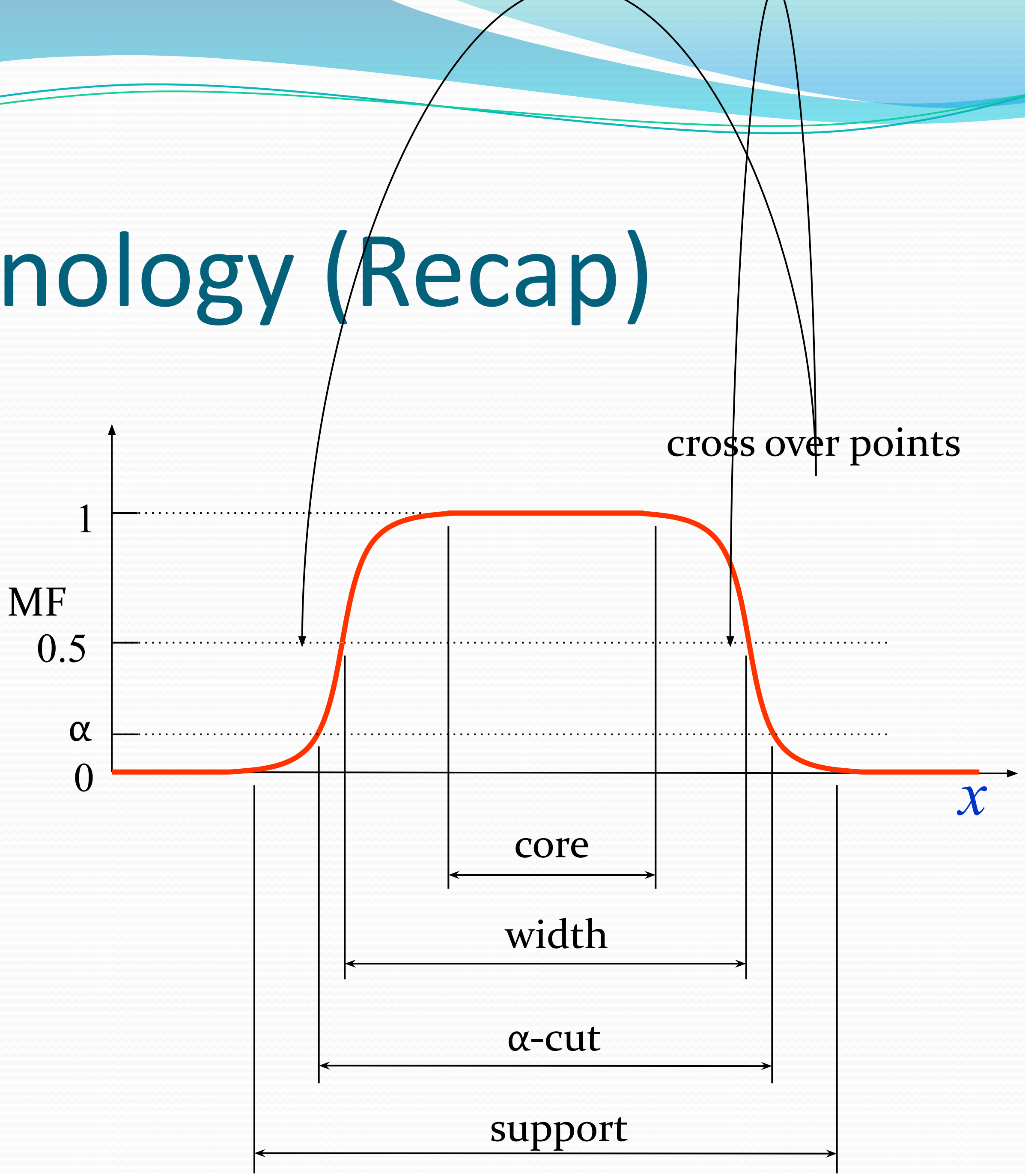


Fig: Fuzzy singleton 45 years old

Terminology (Recap)



Fuzzy vs. Probability

Fuzzy : When we say about certainty of a thing

Example: A patient come to the doctor and he has to diagnose so that medicine can be prescribed.
Doctor prescribed a medicine with certainty 60% that the patient is suffering from flue. So, the disease will be cured with certainty of 60% and uncertainty 40%. Here, in stead of flue, other diseases with some other certainties may be.

Probability: When we say about the chance of an event to occur

Example: India will win the T20 tournament with a chance 60% means that out of 100 matches, India own 60 matches.

Prediction vs. Forecasting

The Fuzzy vs. Probability is analogical to Prediction vs. Forecasting

Prediction : When you start guessing about things.

Forecasting : When you take the information from the past job and apply it to new job.

The main difference:

Prediction is based on thebest guess from experiences.
Forecasting is based ondata you have actually recorded and packed from previous job.

Set Operations

- Product of a two fuzzy sets
- Equality
- Product of a fuzzy set with a crisp number
- Power of a fuzzy set
- Difference
- Disjunctive sum

Fuzzy arithmetic

- Fuzzy Arithmetic uses arithmetic on closed intervals. The basic fuzzy arithmetic operations are defined as follows:
- **Addition:** [a, b] + [c, d] = [a+ c, b + d]
- **Subtraction:** [a, b] – [c, d] = [a – d, b – c]
- **Multiplication:** [a,b].[c,d] = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]
- **Division:** [a, b] / [c, d] = [a, b] . [1/d, 1/c] = [min(a/c, a/d, b/c, b/d),max(a/c, a/d, b/c, b/d)]

Product of a two fuzzy sets

The product of two fuzzy sets A and B is a new set A.B whose MF is defined as

Example

$$A = \{(x_1, 0.2), (x_2, 0.8), (x_3, 0.4)\}$$

$$B = \{(x_1, 0.4), (x_2, 0), (x_3, 0.1)\}$$

Find A.B

Solution

$$A \cdot B = \{(x_1, 0.08), (x_2, 0), (x_3, 0.04)\}$$

Equality

Two fuzzy sets A and B are said to be equal A=B if

$$\mu_A(x) = \mu_B(x)$$

Example

$$A = \{(x_1, 0.2), (x_2, 0.8)\}$$

$$B = \{(x_1, 0.6), (x_2, 0.8)\}$$

$$C = \{(x_1, 0.2), (x_2, 0.8)\}$$

$$\boldsymbol{A \neq B}$$

$$\boldsymbol{A = C}$$

Product of a fuzzy set with a number

Multiplying a fuzzy set A by a crisp number a results in a new fuzzy set a.A with the MF :

$$\mu_{a\cdot A}(x) = a.\mu_A(x)$$

Example

$$A = \{(x_1,0.4),(x_2,0.6),(x_3,0.8)\}$$

$$a = 0.3$$

$$a.A = \{(x_1,0.12),(x_2,0.18),(x_3,0.24)\}$$

Power of a fuzzy set

The α power of a fuzzy set A is a new set A$^{\alpha}$ with the MF

$$\mu_{A^{\alpha}}(x) = (\mu_A(x))^{\alpha}$$

Example

$$A = \{(x_1, 0.4), (x_2, 0.6), (x_3, 0.8)\}$$

$$\alpha = 2$$

$$A^{\alpha} = \{(x_1, 0.16), (x_2, 0.36), (x_3, 0.64)\}$$

Difference

Difference of two fuzzy sets A and B is a new set A-B defined as:

$$A - B = A \cap \overline{B}$$

Example

$$A = \{(x_1, 0.2), (x_2, 0.5), (x_3, 0.6)\}$$

$$B = \{(x_1, 0.1), (x_2, 0.4), (x_3, 0.5)\}$$

Find A-B

Solution

$$\overline{B} = \{(x_1, 0.9), (x_2, 0.6), (x_3, 0.5)\}$$

$$A - B = A \cap \overline{B} = \{(x_1, 0.2), (x_2, 0.5), (x_3, 0.5)\}$$

Fuzzy set operations contd.

Algebric product or Vector product (A•B):

$$\mu_{A\bullet B}(x) = \mu_A(x) \bullet \mu_B(x)$$

Scalar product ($a \times A$):

$$\mu_{\alpha A}(x) = a \cdot \mu_A(x)$$

Sum ($A + B$**):**

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

Difference ($A - B = A \cap B^C$**):**

$$\mu_{A-B}(x) = \mu_{A\cap B^{C}}(x)$$

Disjunctive sum: $A \oplus B = (A^C \cap B)\ U\ (A \cap B^C)$**)**

Bounded Sum: | A(x) ⊕ B(x) |

$$\mu_{|A(x)\oplus B(x)|} = \min\{1, \mu_A(x) + \mu_B(x)\}$$

Bounded Difference: | A(x) *g* B(x) |

$$\mu_{|A(x)gB(x)|} = \max\{0, \mu_A(x) + \mu_B(x) - 1\}$$

Fuzzy set operations contd.

Equality (*A* = *B*):

$$\mu_A(x) = \mu_B(x)$$

Power of a fuzzy set A^{α}:

$$\mu_A{}^{\alpha}(x) = \{\mu_A(x)\}^{\alpha}$$

If $a < 1$, then it is called *dilation*

If $a > 1$, then it is called *concentration*

Disjunctive sum

The disjunctive sum of two fuzzy sets A and B is a new set $A \oplus B$ defined as:

$$A \oplus B = (\overline{A} \cap B) \cup (A \cap \overline{B})$$

Example

$$A = \{(x_1,0.4),(x_2,0.8),(x_3,0.6)\} \quad B = \{(x_1,0.2),(x_2,0.6),(x_3,0.9)\}$$

Find $A \oplus B$

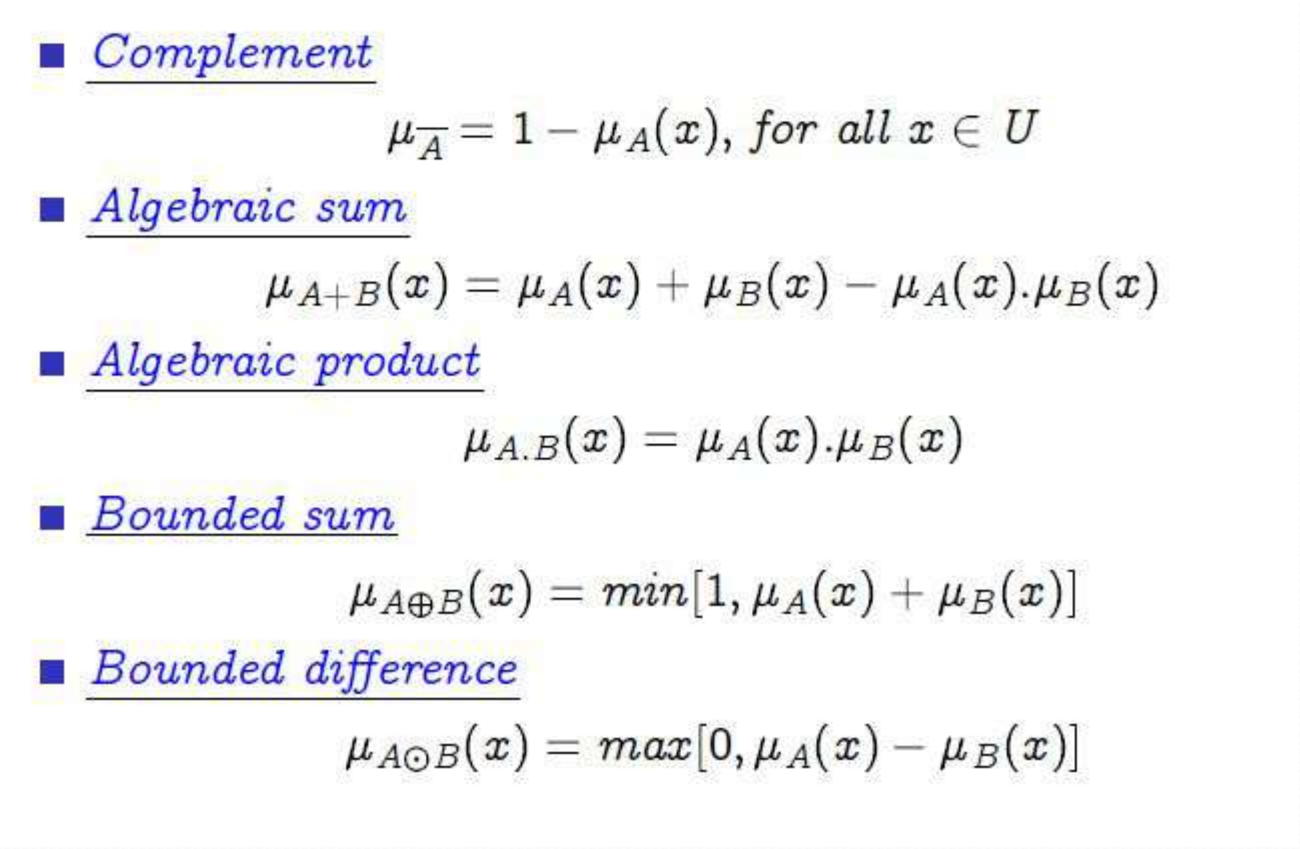
Solution

$$\overline{A} = \{(x_1,0.6),(x_2,0.2),(x_3,0.4)\} \quad \overline{B} = \{(x_1,0.8),(x_2,0.4),(x_3,0.1)\}$$

$$\overline{A} \cap B = \{(x_1,0.2),(x_2,0.2),(x_3,0.4)\} \qquad A \cap \overline{B} = \{(x_1,0.4),(x_2,0.4),(x_3,0.1)\}$$

$$A \oplus B = (\overline{A} \cap B) \cup (A \cap \overline{B}) = (x_1,0.4),(x_2,0.4),(x_3,0.4)$$

Fuzzy set operations (Recap)



- *Complement*

$$\mu_{\overline{A}} = 1 - \mu_A(x), \; for\ all\ x \in U$$

- *Algebraic sum*

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)$$

- *Algebraic product*

$$\mu_{A.B}(x) = \mu_A(x).\mu_B(x)$$

- *Bounded sum*

$$\mu_{A \oplus B}(x) = min[1, \mu_A(x) + \mu_B(x)]$$

- *Bounded difference*

$$\mu_{A \odot B}(x) = max[0, \mu_A(x) - \mu_B(x)]$$

Properties of the fuzzy sets

- The properties of the classical set also suits for the properties of the fuzzy sets. The important properties of fuzzy set includes:
- Commutativity
- $A \cup B = B \cup A, \; A \cap B = B \cap A$
- Associativity

$$A \cup (B \cup C) = (A \cup B) \cup C, \quad A \cap (B \cap C) = (A \cap B) \cap C$$

- Distributivity

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

Properties of the fuzzy sets

- Idem potency (the same power)

$$A \cup A = A, \qquad A \cap A = A.$$

- Identity

$$A \cup \varphi = A \quad \text{and } A \cap X = A \;, A \cap \varphi = \varphi \text{ and } A \cup X = X.$$

Where

Φ is the empty set (the degree of membership of all its elements is zero)

- Transitivity

$$\text{If } A \subset B \subset C \text{ then } A \subset C$$

- Involution (double complement)

.

$$\overline{\overline{A}} = A$$

Power of a fuzzy set A^{α}:

$$\mu_{A^{\alpha}}(x) = \{\mu_A(x)\}^{\alpha}$$

If $a < 1$, then it is called *dilation*

If $a > 1$, then it is called *concentration*

Properties of fuzzy sets (Recap)

Idempotence :

$$A \cup A = A$$

$$A \cap A = \varnothing$$

$$A \cup \varnothing = A$$

$$A \cap \varnothing = \varnothing$$

Transitivity :

If $A \subseteq B$, $B \subseteq C$ then $A \subseteq C$

Involution :

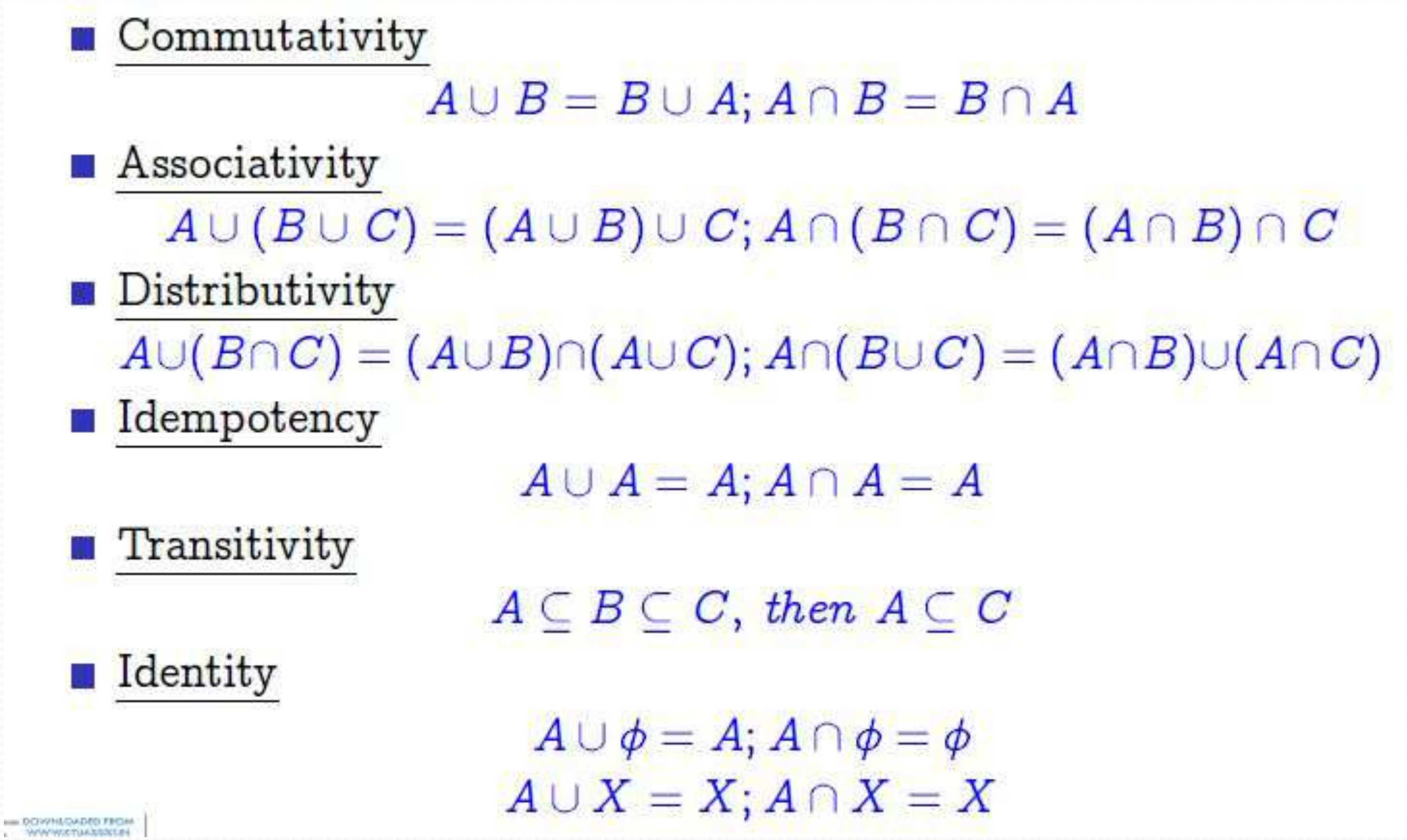
$$(A^c)^c = A$$

De Morgan’s law :

$$(A \cap B)^c = A^c \cup B^c$$

$$(A \cup B)^c = A^c \cap B^c$$

Properties of Fuzzy Set (Recap)



- Commutativity

$$A \cup B = B \cup A; A \cap B = B \cap A$$

- Associativity

$$A \cup (B \cup C) = (A \cup B) \cup C; A \cap (B \cap C) = (A \cap B) \cap C$$

- Distributivity

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C); A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

- Idempotency

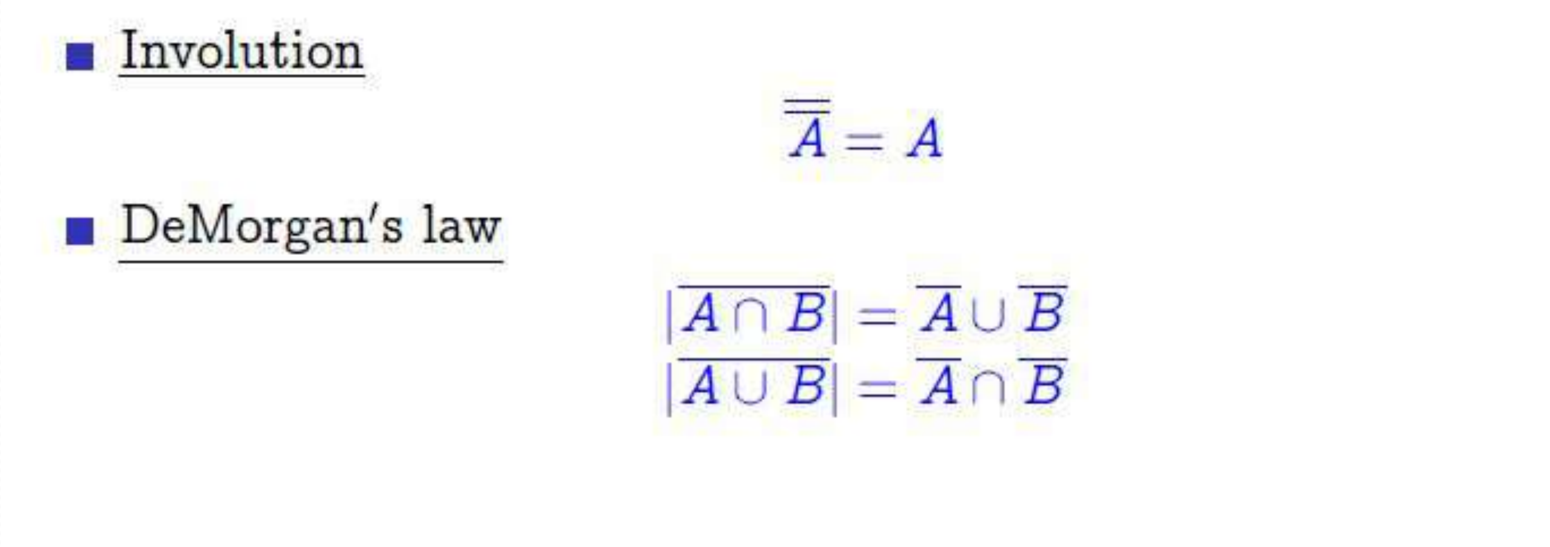
$$A \cup A = A; A \cap A = A$$

- Transitivity

$$A \subseteq B \subseteq C, \; then \; A \subseteq C$$

- Identity

$$A \cup \phi = A; A \cap \phi = \phi$$
$$A \cup X = X; A \cap X = X$$



- Involution

$$\overline{\overline{A}} = A$$

- DeMorgan's law

$$|\overline{A \cap B}| = \overline{A} \cup \overline{B}$$
$$|\overline{A \cup B}| = \overline{A} \cap \overline{B}$$

Example

Consider two fuzzy sets *A* and *B*. *F*ind Complement, Union, Intersection

$$\underset{\sim}{A} = \left\{ \frac{1}{2} + \frac{0.5}{3} + \frac{0.6}{4} + \frac{0.2}{5} + \frac{0.6}{6} \right\},$$

$$\underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.8}{3} + \frac{0.4}{4} + \frac{0.7}{5} + \frac{0.3}{6} \right\}.$$

Solution...

Complement

$$\bar{\underset{\sim}{A}} = \left\{ \frac{0}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.8}{5} + \frac{0.4}{6} \right\},$$

$$\bar{\underset{\sim}{B}} = \left\{ \frac{0.5}{2} + \frac{0.2}{3} + \frac{0.6}{4} + \frac{0.3}{5} + \frac{0.7}{6} \right\}.$$

Union

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \left\{ \frac{1}{2} + \frac{0.8}{3} + \frac{0.6}{4} + \frac{0.7}{5} + \frac{0.6}{6} \right\}.$$

Maximum is used

…….Solution

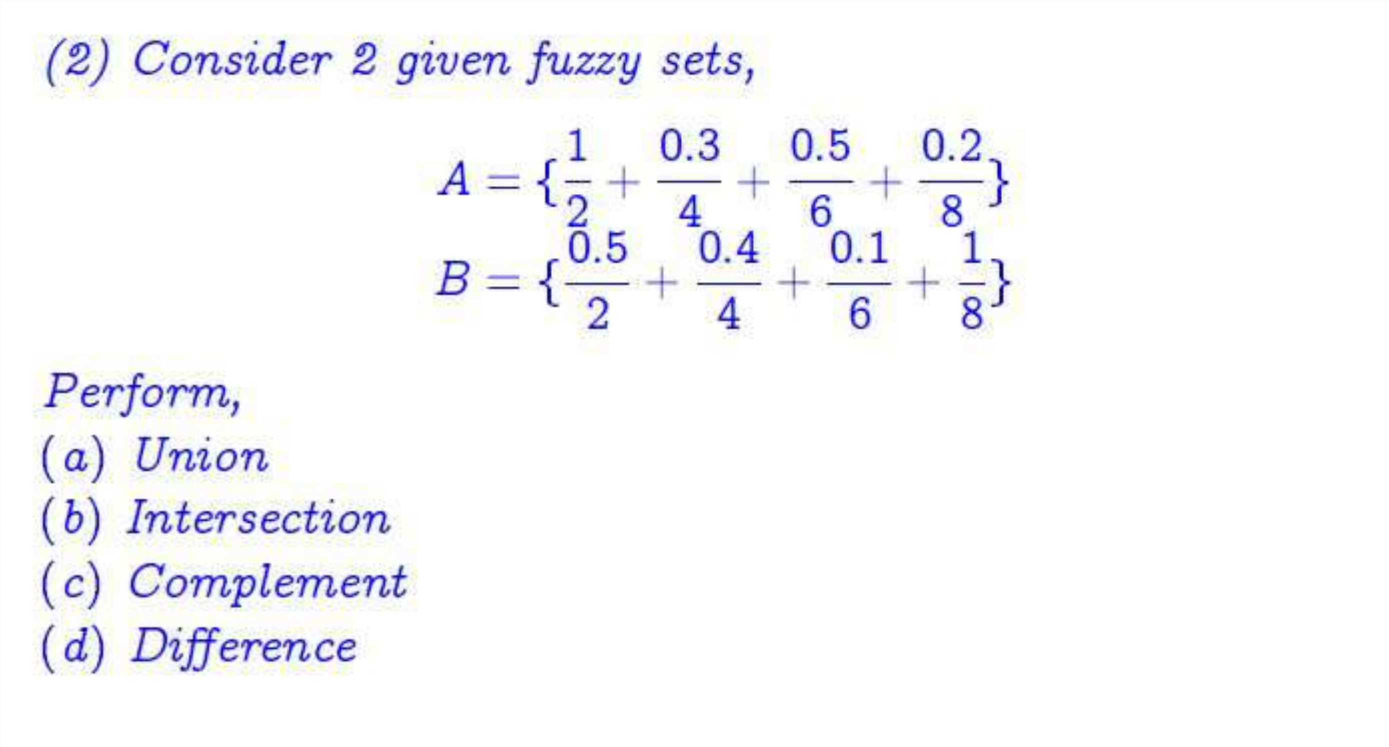
$$\underset{\sim}{A} = \left\{ \frac{1}{2} + \frac{0.5}{3} + \frac{0.6}{4} + \frac{0.2}{5} + \frac{0.6}{6} \right\},$$

$$\underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.8}{3} + \frac{0.4}{4} + \frac{0.7}{5} + \frac{0.3}{6} \right\}.$$

intersection

$$\underset{\sim}{A} \cap \underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.2}{5} + \frac{0.3}{6} \right\}.$$

Minimum is used



(2) Consider 2 given fuzzy sets,

$$A = \{\frac{1}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0.2}{8}\}$$
$$B = \{\frac{0.5}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{1}{8}\}$$

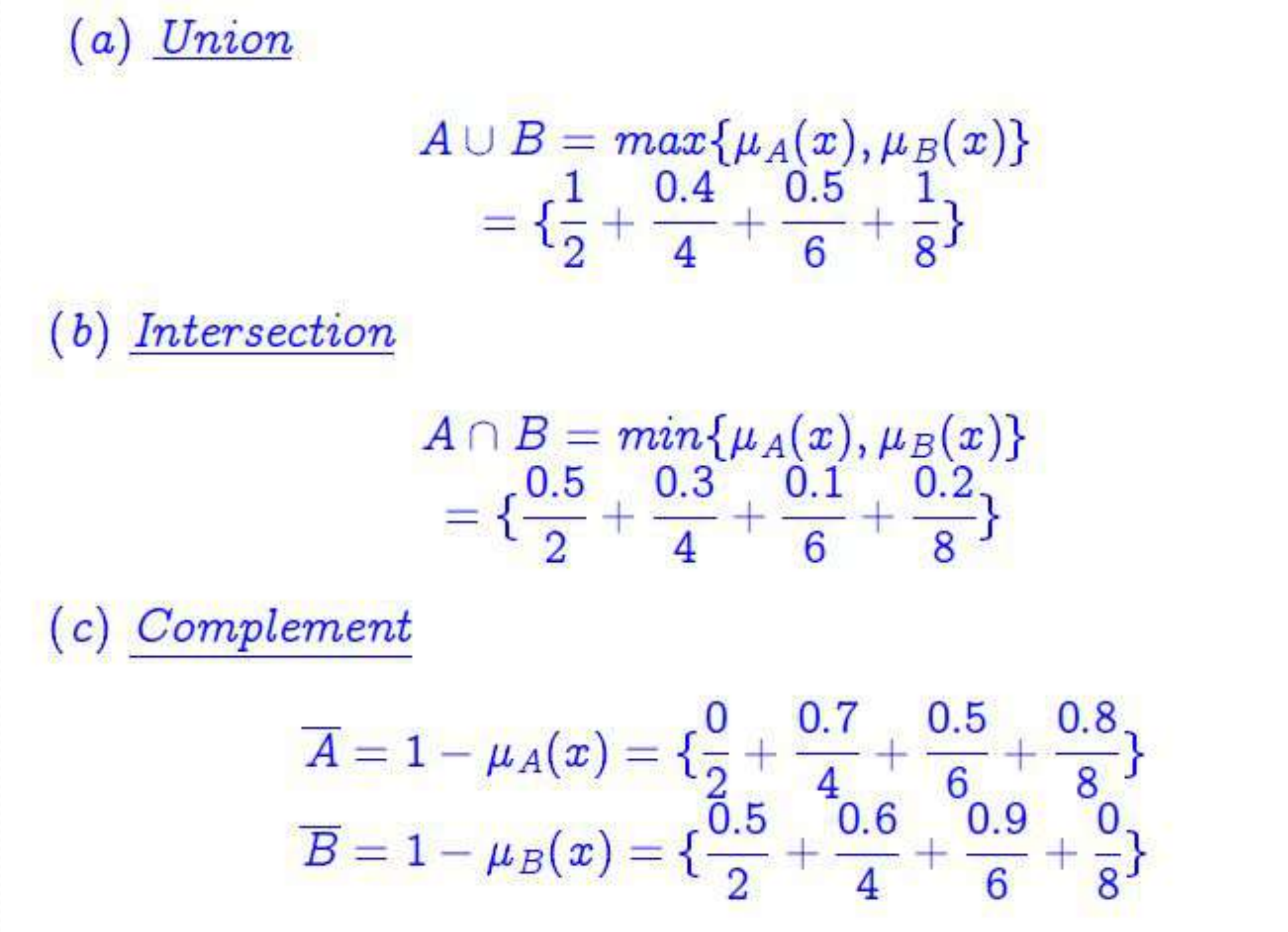
Perform,

(a) Union

(b) Intersection

(c) Complement

(d) Difference



(a) $\underline{Union}$

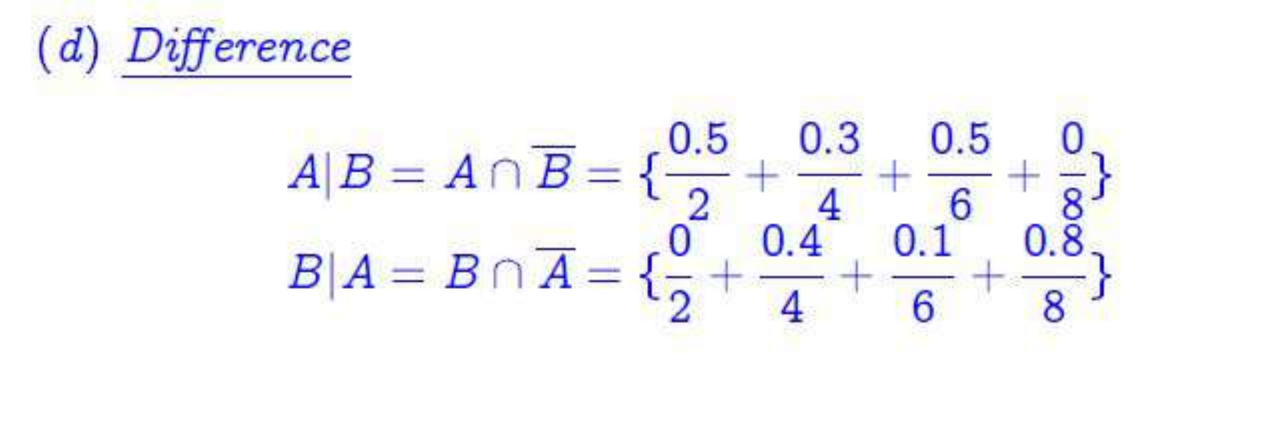
$$A \cup B = max\{\mu_A(x), \mu_B(x)\}$$
$$= \{\frac{1}{2} + \frac{0.4}{4} + \frac{0.5}{6} + \frac{1}{8}\}$$

(b) $\underline{Intersection}$

$$A \cap B = min\{\mu_A(x), \mu_B(x)\}$$
$$= \{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.1}{6} + \frac{0.2}{8}\}$$

(c) $\underline{Complement}$

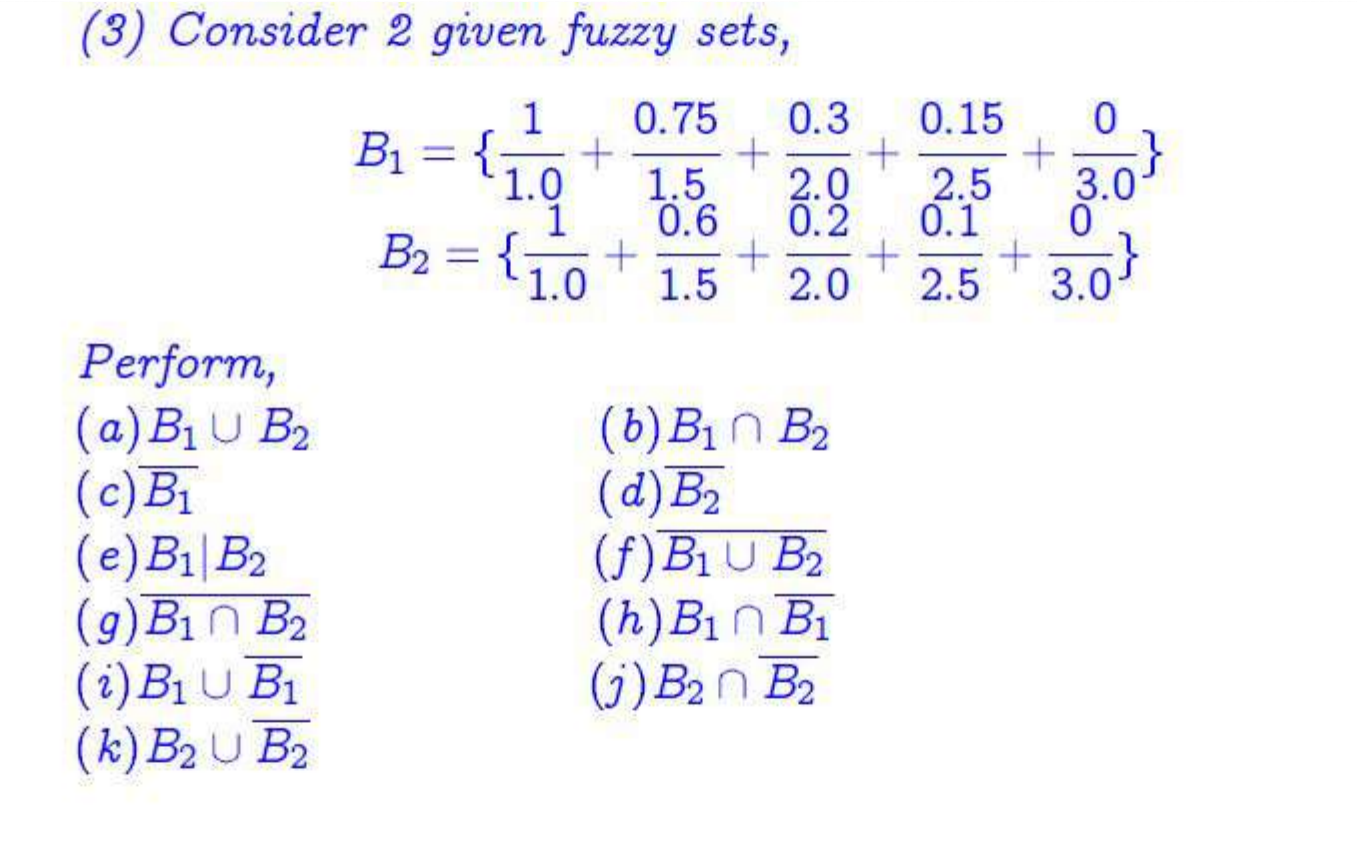
$$\overline{A} = 1 - \mu_A(x) = \{\frac{0}{2} + \frac{0.7}{4} + \frac{0.5}{6} + \frac{0.8}{8}\}$$
$$\overline{B} = 1 - \mu_B(x) = \{\frac{0.5}{2} + \frac{0.6}{4} + \frac{0.9}{6} + \frac{0}{8}\}$$



(d) *Difference*

$$A|B = A \cap \overline{B} = \{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0}{8}\}$$

$$B|A = B \cap \overline{A} = \{\frac{0}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{0.8}{8}\}$$



(3) Consider 2 given fuzzy sets,

$$B_1 = \{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\}$$
$$B_2 = \{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\}$$

Perform,

$(a) B_1 \cup B_2$

$(b) B_1 \cap B_2$

$(c) \overline{B_1}$

$(d) \overline{B_2}$

$(e) B_1 | B_2$

$(f) \overline{B_1 \cup B_2}$

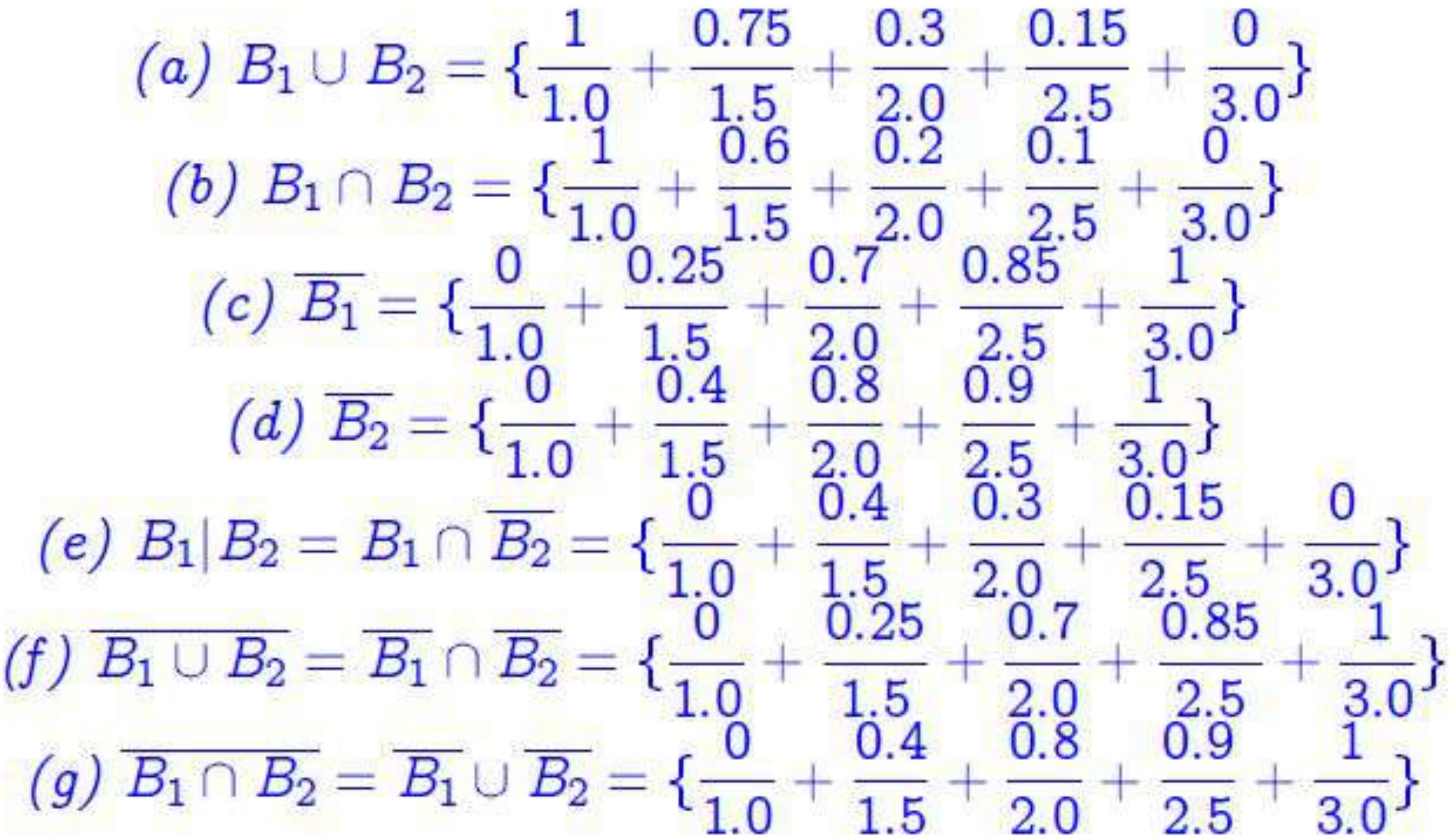
$(g) \overline{B_1 \cap B_2}$

$(h) B_1 \cap \overline{B_1}$

$(i) B_1 \cup \overline{B_1}$

$(j) B_2 \cap \overline{B_2}$

$(k) B_2 \cup \overline{B_2}$



$$(a)\ B_1 \cup B_2 = \{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\}$$

$$(b)\ B_1 \cap B_2 = \{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\}$$

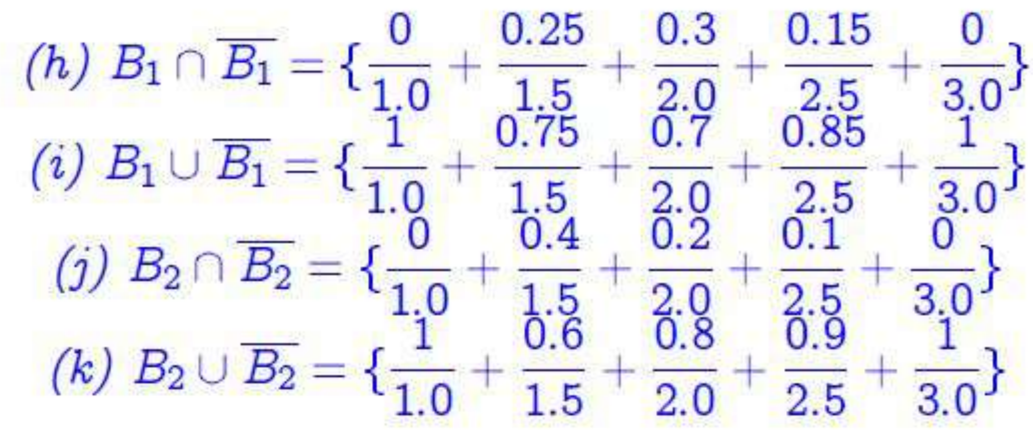
$$(c)\ \overline{B_1} = \{\frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0}\}$$

$$(d)\ \overline{B_2} = \{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0}\}$$

$$(e)\ B_1 | B_2 = B_1 \cap \overline{B_2} = \{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\}$$

$$(f)\ \overline{B_1 \cup B_2} = \overline{B_1} \cap \overline{B_2} = \{\frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0}\}$$

$$(g)\ \overline{B_1 \cap B_2} = \overline{B_1} \cup \overline{B_2} = \{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0}\}$$

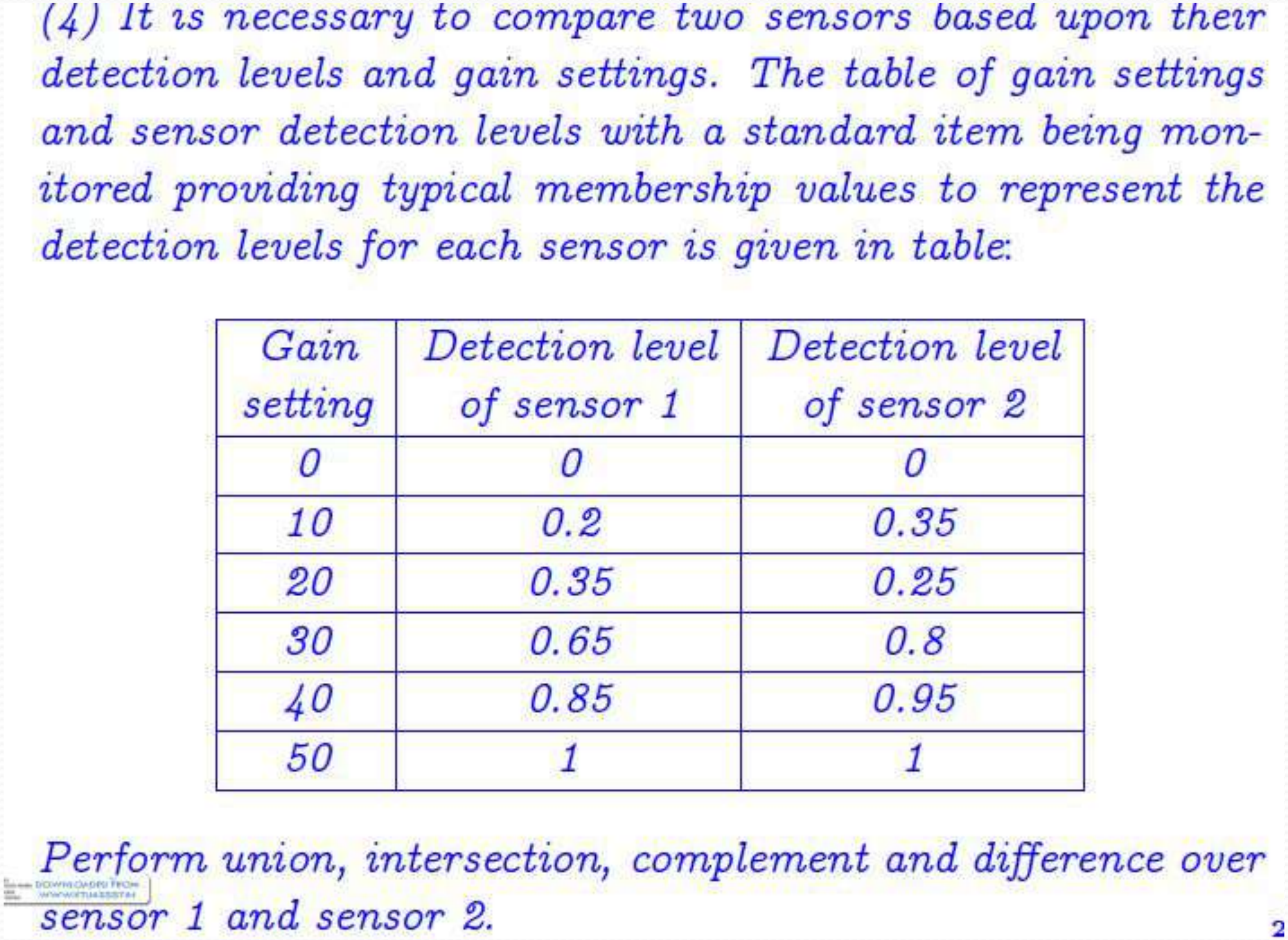


$$(h)\ B_1 \cap \overline{B_1} = \{\frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\}$$

$$(i)\ B_1 \cup \overline{B_1} = \{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0}\}$$

$$(j)\ B_2 \cap \overline{B_2} = \{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\}$$

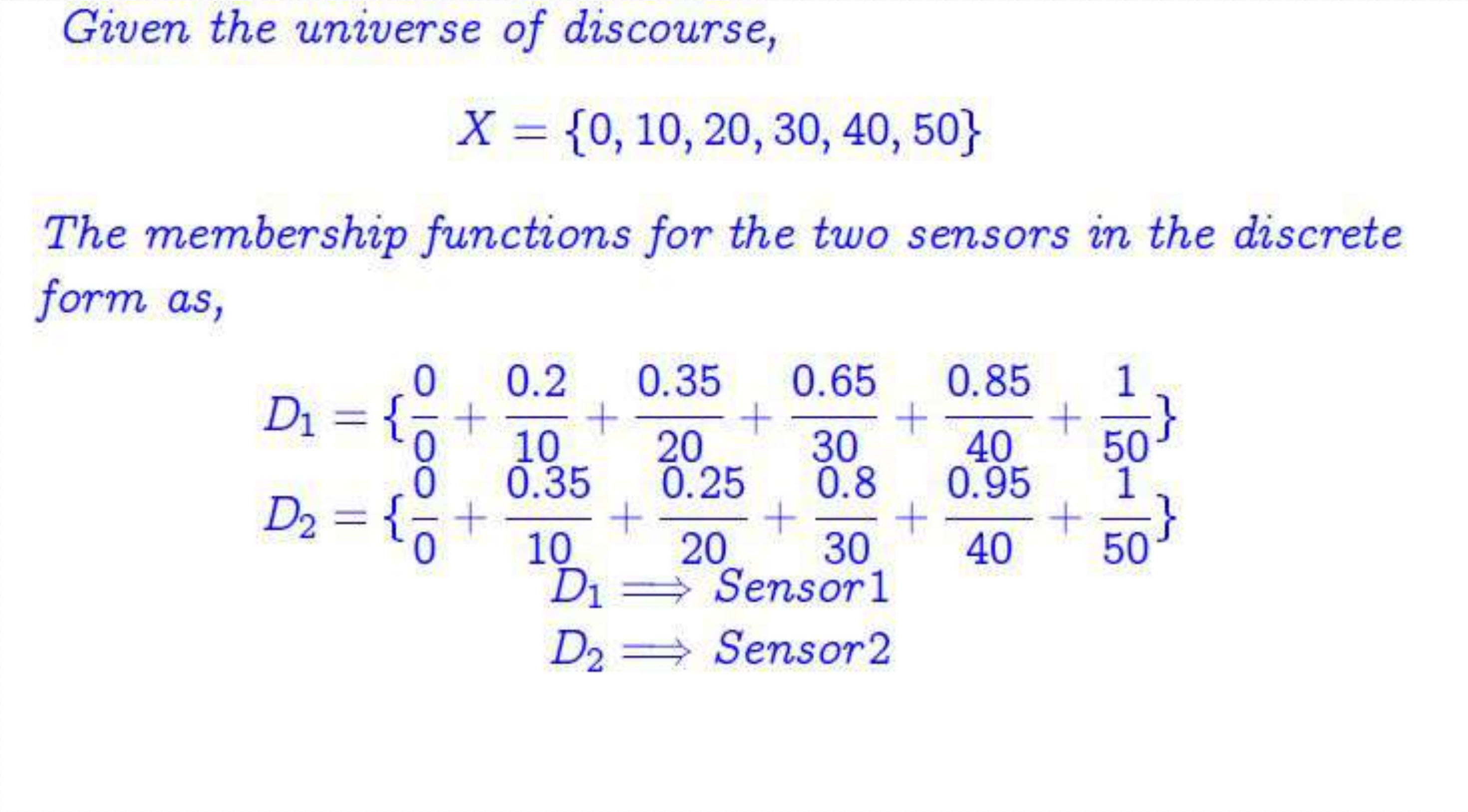
$$(k)\ B_2 \cup \overline{B_2} = \{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0}\}$$



(4) It is necessary to compare two sensors based upon their detection levels and gain settings. The table of gain settings and sensor detection levels with a standard item being monitored providing typical membership values to represent the detection levels for each sensor is given in table:

Gain setting	Detection level of sensor 1	Detection level of sensor 2
0	0	0
10	0.2	0.35
20	0.35	0.25
30	0.65	0.8
40	0.85	0.95
50	1	1

Perform union, intersection, complement and difference over sensor 1 and sensor 2.



Given the universe of discourse,

$$X = \{0, 10, 20, 30, 40, 50\}$$

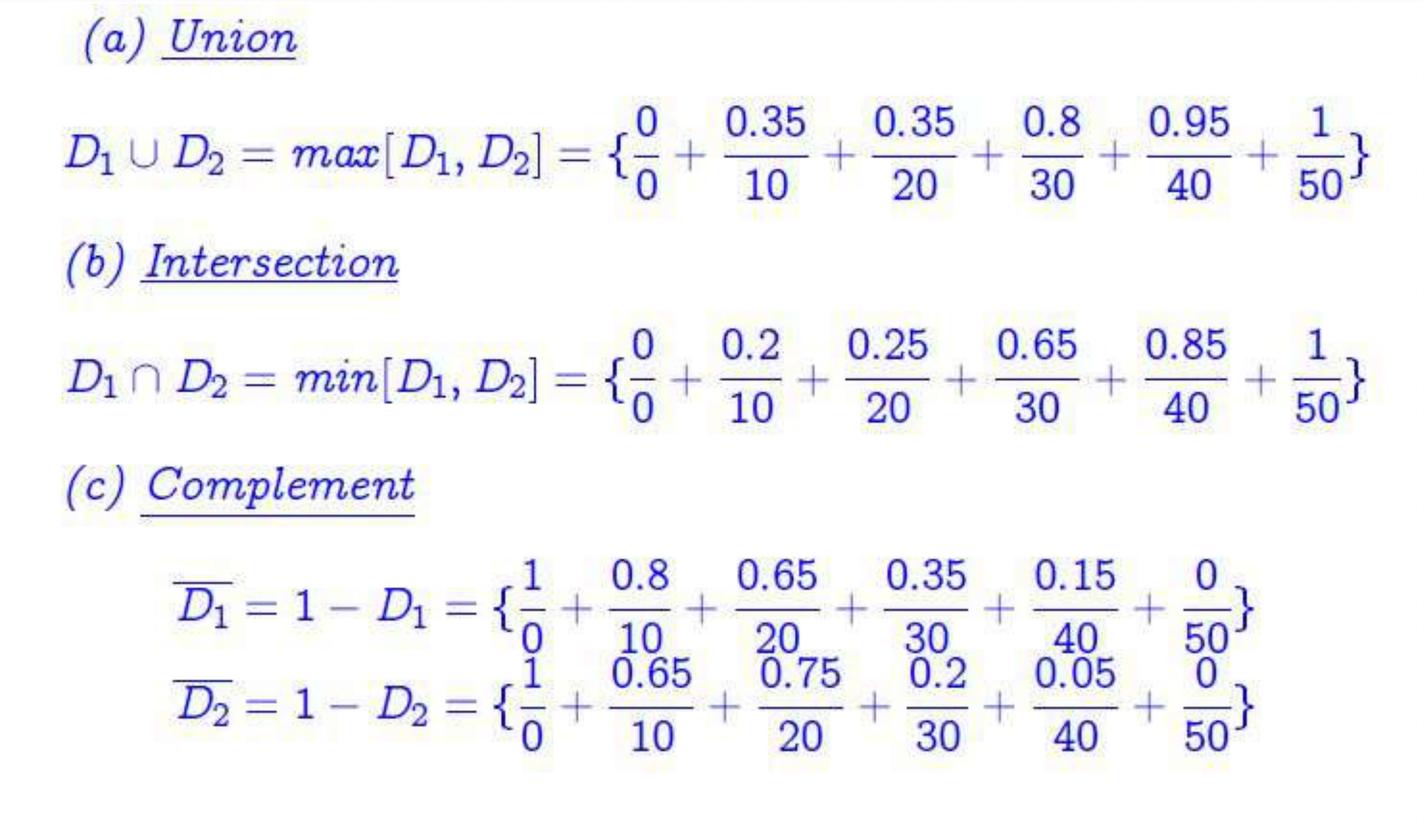
The membership functions for the two sensors in the discrete form as,

$$D_1 = \{\frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50}\}$$

$$D_2 = \{\frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50}\}$$

$$D_1 \Longrightarrow Sensor1$$

$$D_2 \Longrightarrow Sensor2$$



(a) Union

$$D_1 \cup D_2 = max[D_1, D_2] = \{\frac{0}{0} + \frac{0.35}{10} + \frac{0.35}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50}\}$$

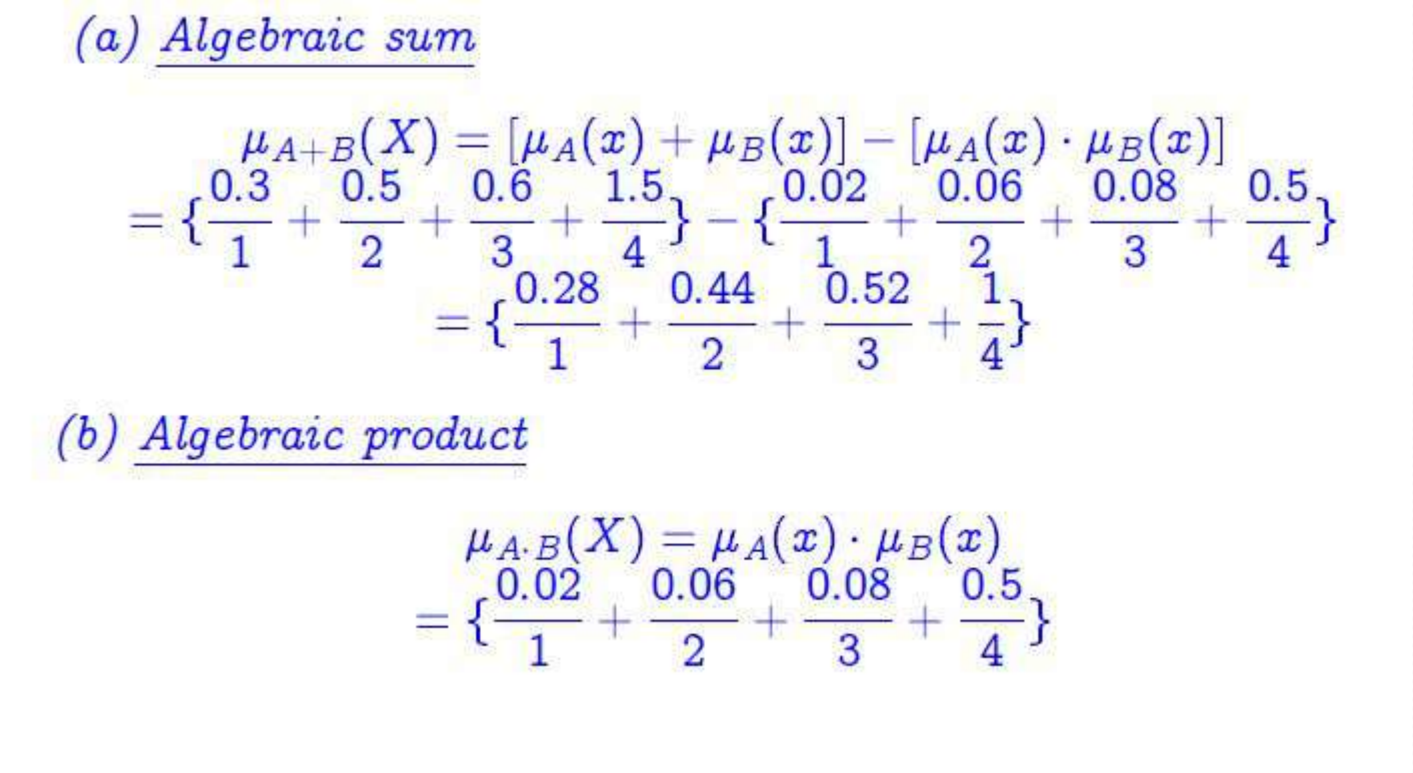
(b) Intersection

$$D_1 \cap D_2 = min[D_1, D_2] = \{\frac{0}{0} + \frac{0.2}{10} + \frac{0.25}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50}\}$$

(c) Complement

$$\overline{D_1} = 1 - D_1 = \{\frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50}\}$$

$$\overline{D_2} = 1 - D_2 = \{\frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50}\}$$



(a) Algebraic sum

$$\mu_{A+B}(X) = [\mu_A(x) + \mu_B(x)] - [\mu_A(x) \cdot \mu_B(x)]$$

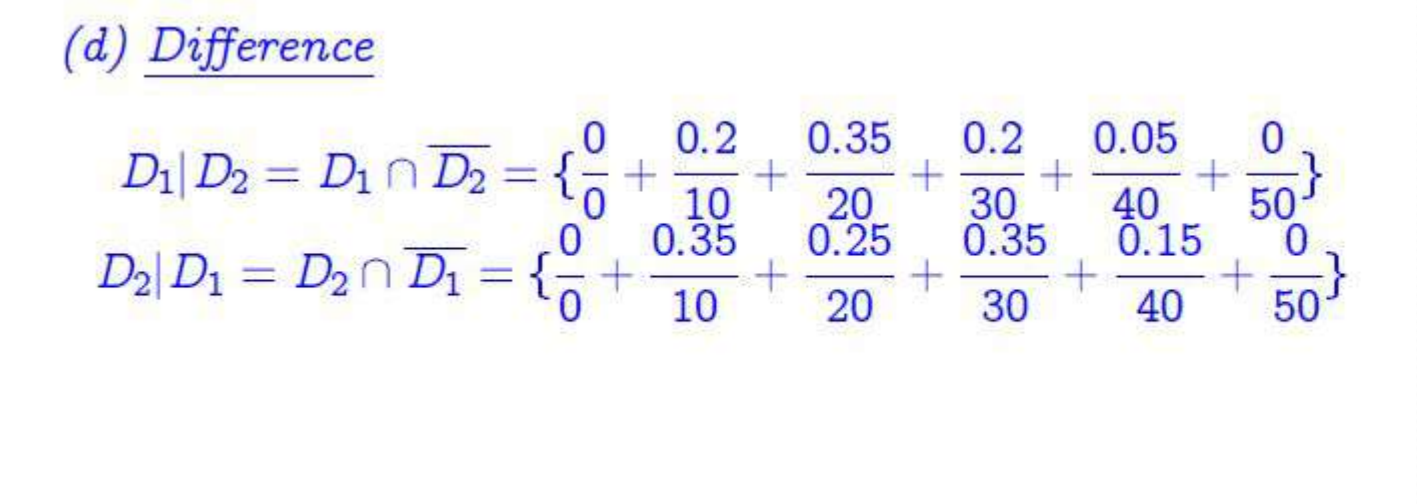
$$= \{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{1.5}{4}\} - \{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\}$$

$$= \{\frac{0.28}{1} + \frac{0.44}{2} + \frac{0.52}{3} + \frac{1}{4}\}$$

(b) Algebraic product

$$\mu_{A \cdot B}(X) = \mu_A(x) \cdot \mu_B(x)$$

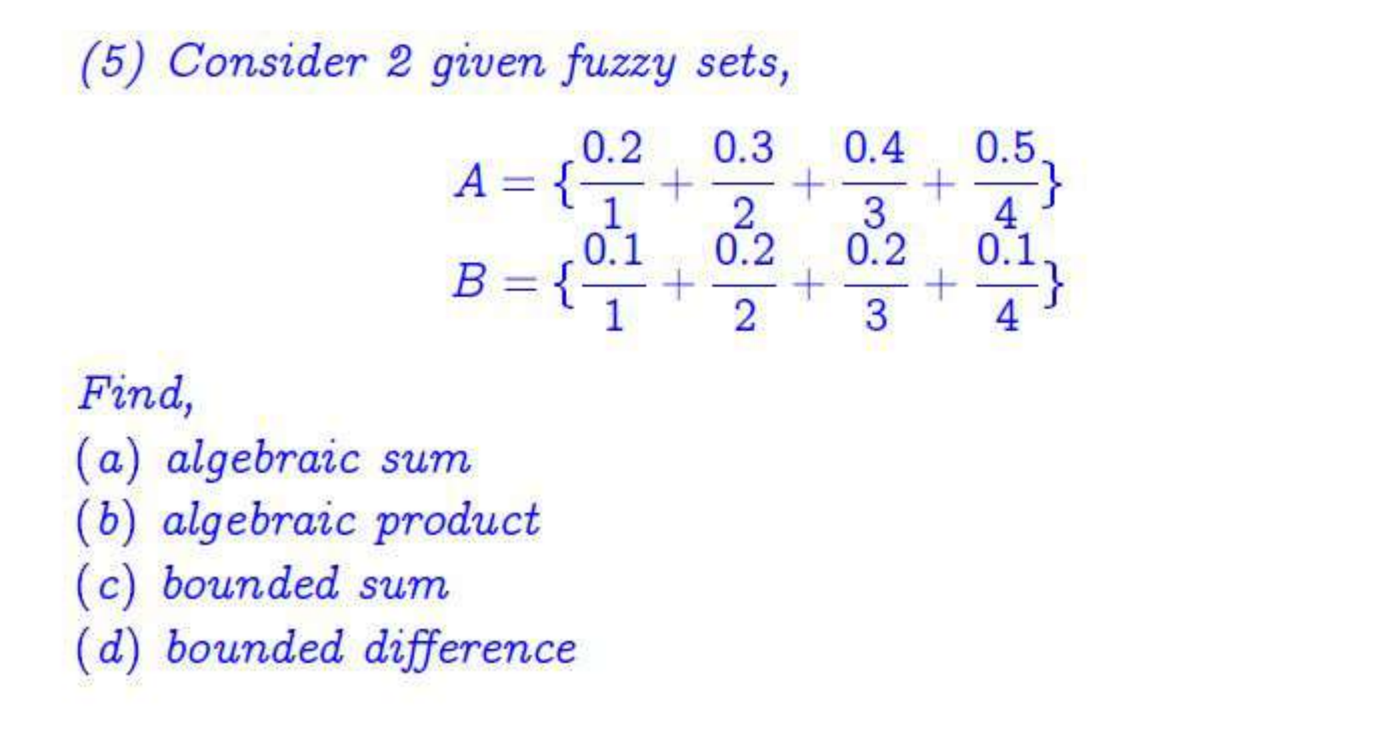
$$= \{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\}$$



(d) Difference

$$D_1 | D_2 = D_1 \cap \overline{D_2} = \{\frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50}\}$$

$$D_2 | D_1 = D_2 \cap \overline{D_1} = \{\frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50}\}$$



(5) Consider 2 given fuzzy sets,

$$A = \{\frac{0.2}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0.5}{4}\}$$
$$B = \{\frac{0.1}{1} + \frac{0.2}{2} + \frac{0.2}{3} + \frac{0.1}{4}\}$$

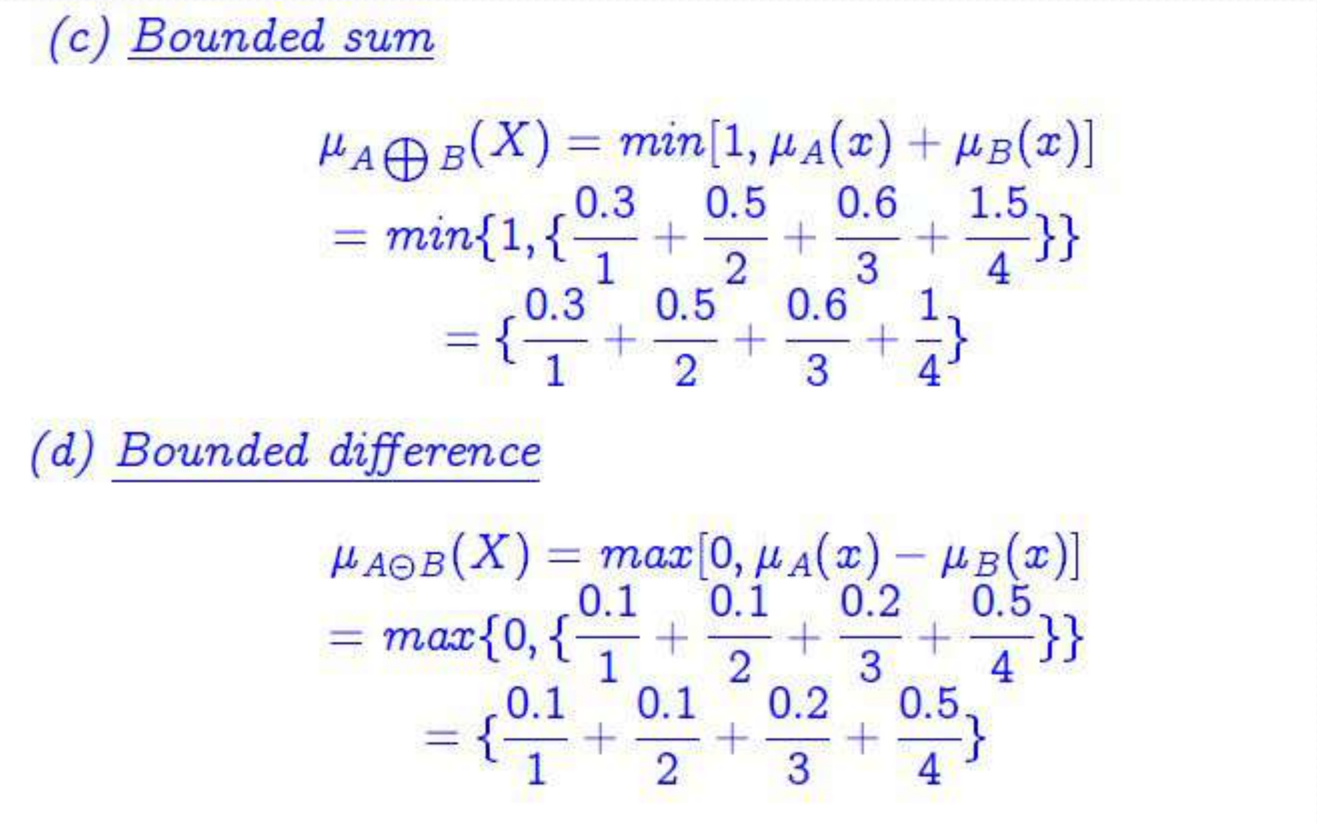
Find,

(*a*) *algebraic sum*

(*b*) *algebraic product*

(*c*) *bounded sum*

(*d*) *bounded difference*



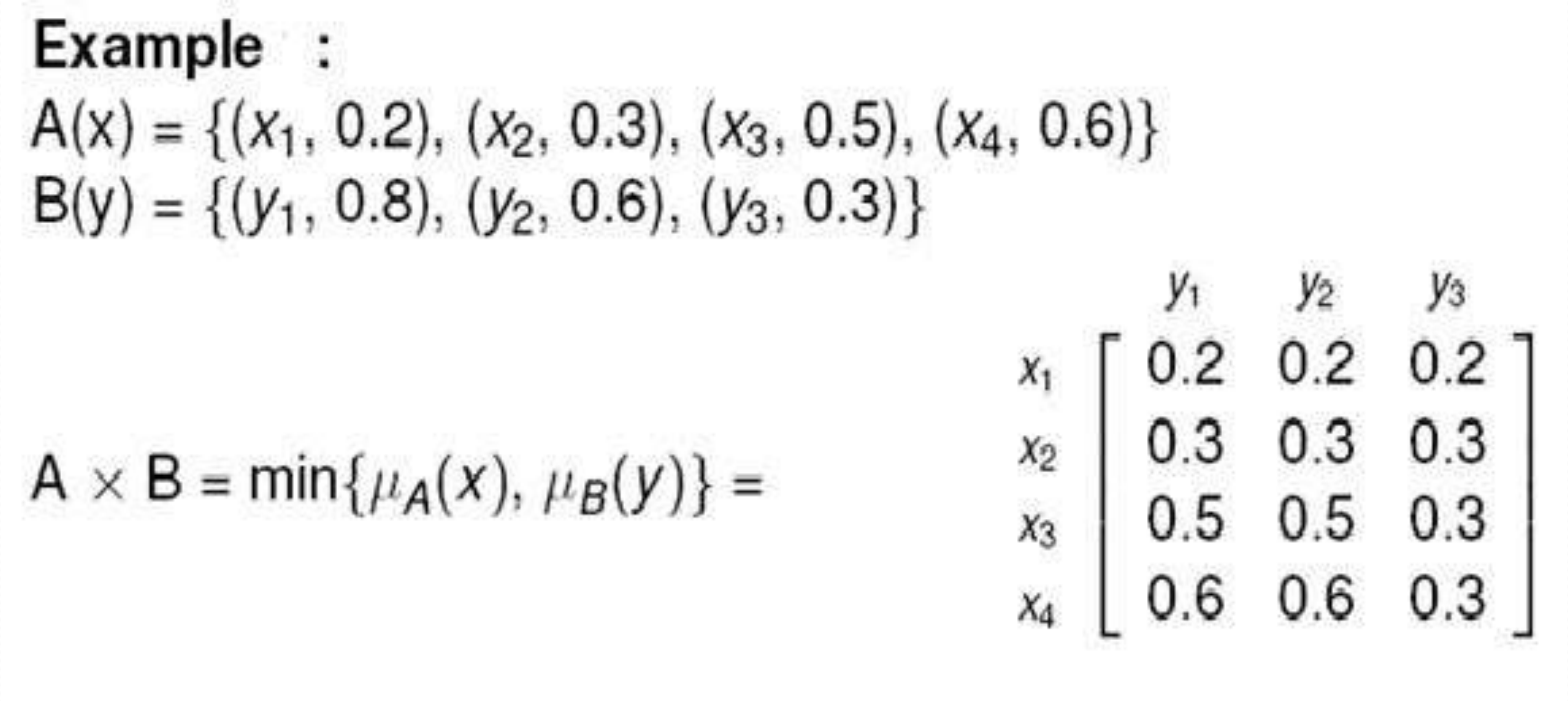
(c) *<u>Bounded sum</u>*

$$\begin{aligned}\mu_{A\bigoplus B}(X) &= min[1, \mu_A(x) + \mu_B(x)]\\ &= min\{1, \{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{1.5}{4}\}\}\\ &= \{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{1}{4}\}\end{aligned}$$

(d) *<u>Bounded difference</u>*

$$\begin{aligned}\mu_{A\ominus B}(X) &= max[0, \mu_A(x) - \mu_B(x)]\\ &= max\{0, \{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\}\}\\ &= \{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\}\end{aligned}$$

Problems (book)



Example :

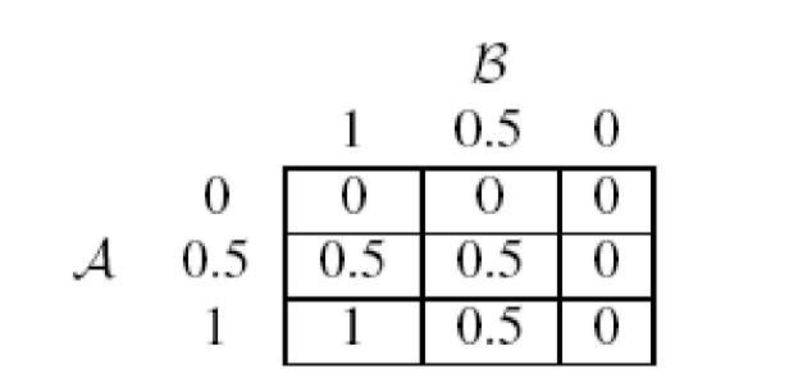
$A(x) = \{(x_1, 0.2), (x_2, 0.3), (x_3, 0.5), (x_4, 0.6)\}$

$B(y) = \{(y_1, 0.8), (y_2, 0.6), (y_3, 0.3)\}$

$$A \times B = \min\{\mu_A(x), \mu_B(y)\} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \\ 0.5 & 0.5 & 0.3 \\ 0.6 & 0.6 & 0.3 \end{bmatrix} \end{array}$$

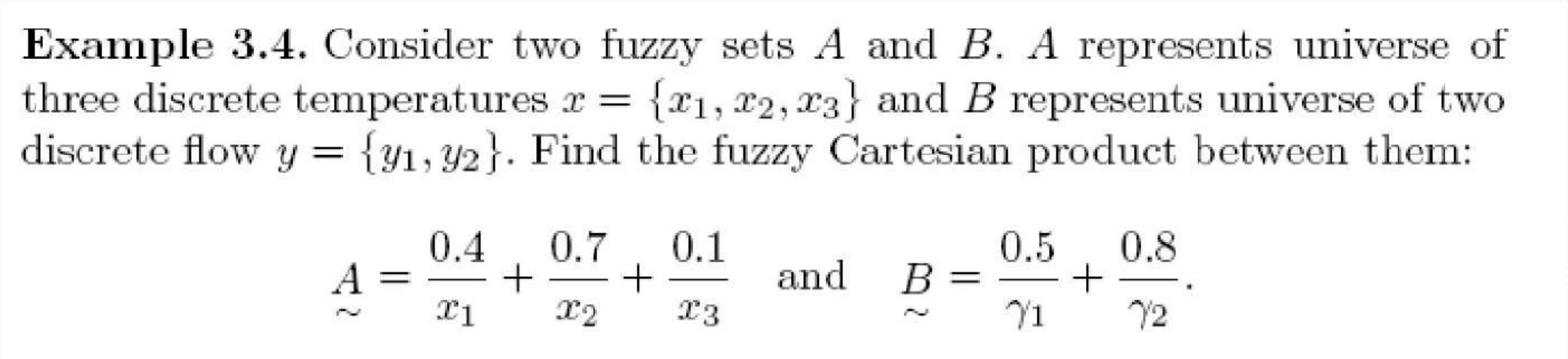
Assume two Fuzzy sets

- A={(1,0), (2,.5),(3,1)}
- B={(1,1), (2,.5),(3,0)}
- $A \times B$ can be arranged as a two-dimensional fuzzy set:



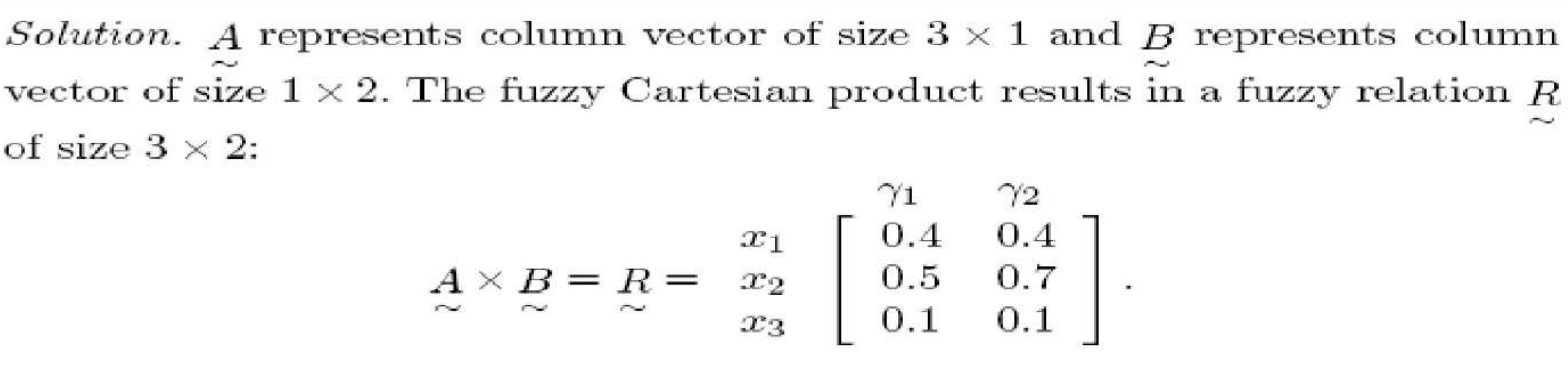
			B	
		1	0.5	0
A	0	0	0	0
	0.5	0.5	0.5	0
	1	1	0.5	0

*A × B={((1,1),0), ((1,2),0),((1,3),0) ((2,1),0.5),
((2,2),0.5), ((2,3),0),((3,1),1), ((3,2),0.5) ((3,3),0)}*



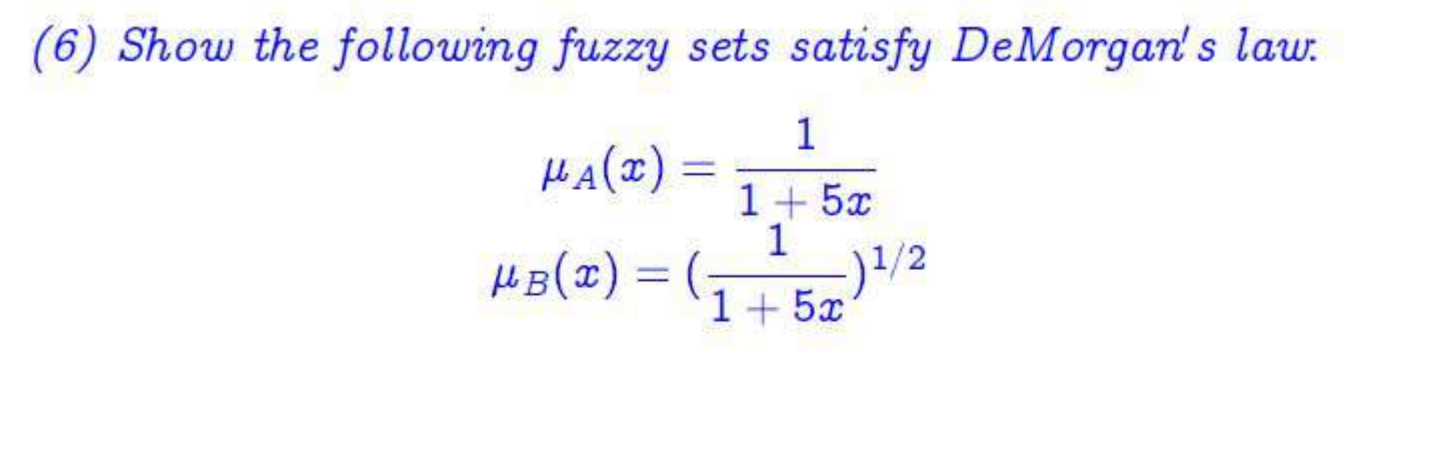
Example 3.4. Consider two fuzzy sets A and B. A represents universe of three discrete temperatures $x = \{x_1, x_2, x_3\}$ and B represents universe of two discrete flow $y = \{y_1, y_2\}$. Find the fuzzy Cartesian product between them:

$$\underset{\sim}{A} = \frac{0.4}{x_1} + \frac{0.7}{x_2} + \frac{0.1}{x_3} \quad \text{and} \quad \underset{\sim}{B} = \frac{0.5}{\gamma_1} + \frac{0.8}{\gamma_2}.$$



Solution. $\underset{\sim}{A}$ represents column vector of size 3×1 and $\underset{\sim}{B}$ represents column vector of size 1×2. The fuzzy Cartesian product results in a fuzzy relation $\underset{\sim}{R}$ of size 3×2:

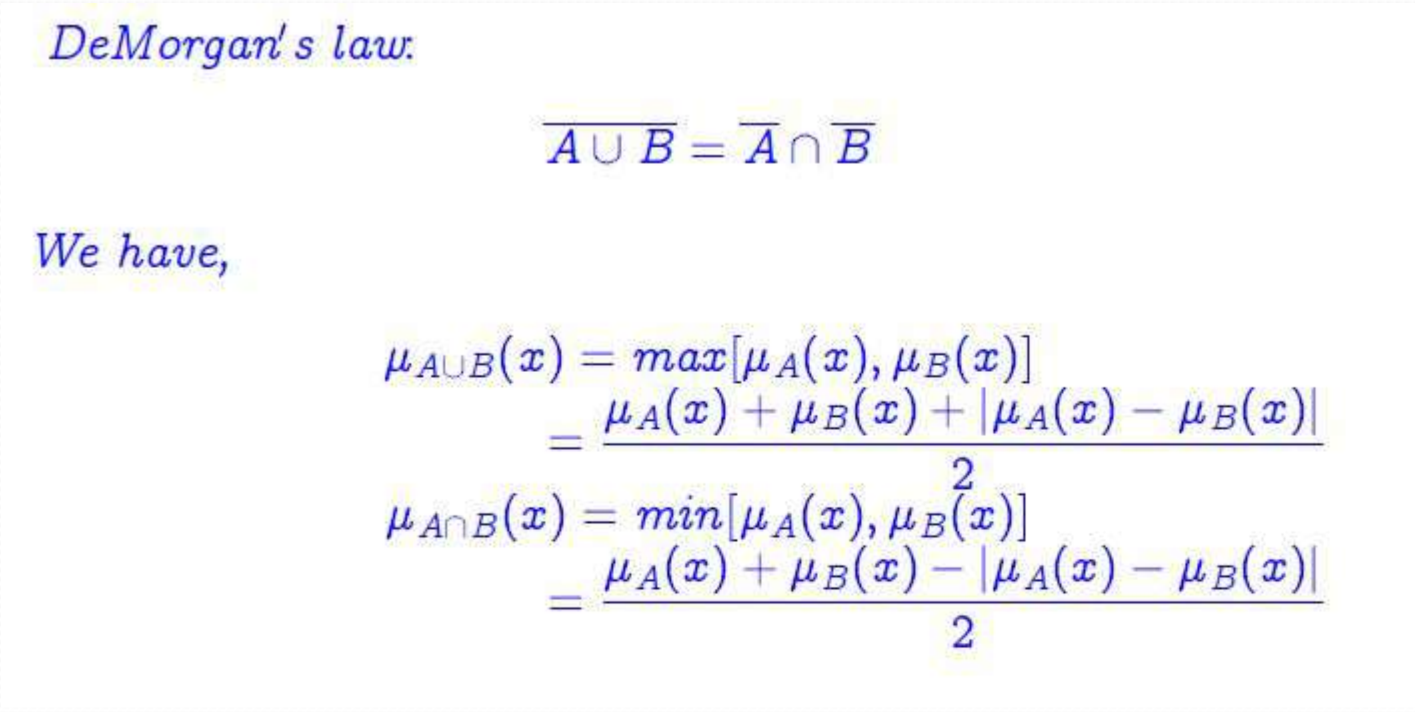
$$\underset{\sim}{A} \times \underset{\sim}{B} = \underset{\sim}{R} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} \gamma_1 & \gamma_2 \end{array} \\ \left[\begin{array}{cc} 0.4 & 0.4 \\ 0.5 & 0.7 \\ 0.1 & 0.1 \end{array} \right] \end{array}.$$



(6) Show the following fuzzy sets satisfy DeMorgan's law.

$$\mu_A(x) = \frac{1}{1+5x}$$

$$\mu_B(x) = (\frac{1}{1+5x})^{1/2}$$

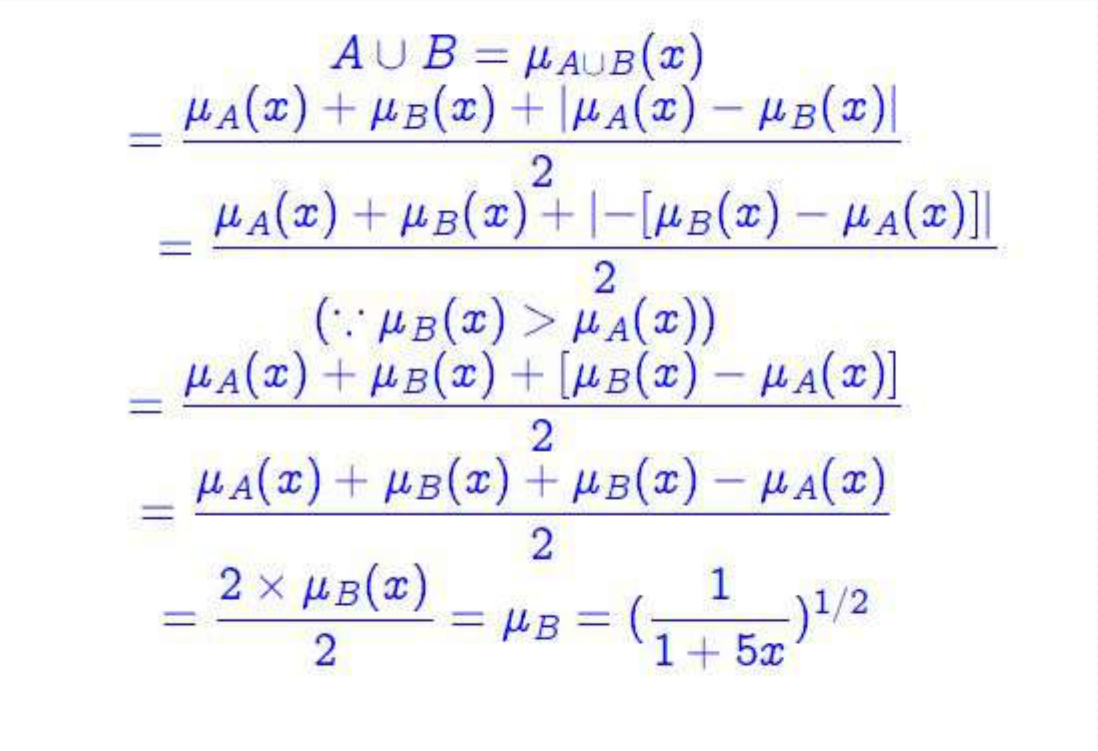


DeMorgan' s law.

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

We have,

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]$$
$$= \frac{\mu_A(x) + \mu_B(x) + |\mu_A(x) - \mu_B(x)|}{2}$$
$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$$
$$= \frac{\mu_A(x) + \mu_B(x) - |\mu_A(x) - \mu_B(x)|}{2}$$



$$A \cup B = \mu_{A \cup B}(x)$$

$$= \frac{\mu_A(x) + \mu_B(x) + |\mu_A(x) - \mu_B(x)|}{2}$$

$$= \frac{\mu_A(x) + \mu_B(x) + |-[\mu_B(x) - \mu_A(x)]|}{2}$$

$$(\because \mu_B(x) > \mu_A(x))$$

$$= \frac{\mu_A(x) + \mu_B(x) + [\mu_B(x) - \mu_A(x)]}{2}$$

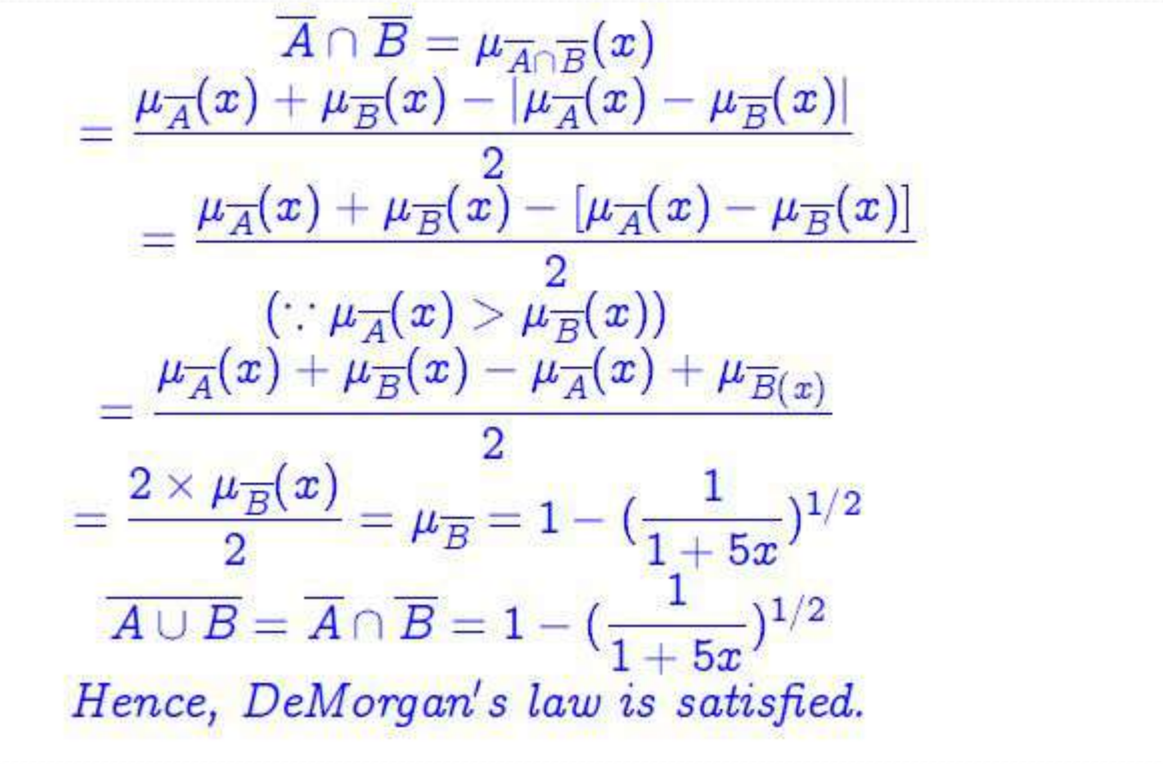
$$= \frac{\mu_A(x) + \mu_B(x) + \mu_B(x) - \mu_A(x)}{2}$$

$$= \frac{2 \times \mu_B(x)}{2} = \mu_B = (\frac{1}{1 + 5x})^{1/2}$$

$$\overline{A \cup B} = 1 - A \cup B = 1 - (\frac{1}{1+5x})^{1/2}$$

$$\overline{A} = 1 - \mu_A(x) = 1 - \frac{1}{1+5x}$$

$$\overline{B} = 1 - \mu_B(x) = 1 - (\frac{1}{1+5x})^{1/2}$$



$$\overline{A}\cap\overline{B} = \mu_{\overline{A}\cap\overline{B}}(x)$$

$$= \frac{\mu_{\overline{A}}(x) + \mu_{\overline{B}}(x) - |\mu_{\overline{A}}(x) - \mu_{\overline{B}}(x)|}{2}$$

$$= \frac{\mu_{\overline{A}}(x) + \mu_{\overline{B}}(x) - [\mu_{\overline{A}}(x) - \mu_{\overline{B}}(x)]}{2}$$

$$(\because \mu_{\overline{A}}(x) > \mu_{\overline{B}}(x))$$

$$= \frac{\mu_{\overline{A}}(x) + \mu_{\overline{B}}(x) - \mu_{\overline{A}}(x) + \mu_{\overline{B}(x)}}{2}$$

$$= \frac{2 \times \mu_{\overline{B}}(x)}{2} = \mu_{\overline{B}} = 1 - (\frac{1}{1+5x})^{1/2}$$

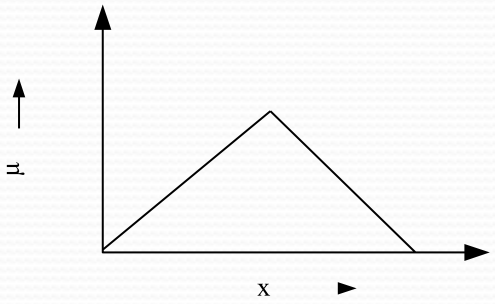
$$\overline{A \cup B} = \overline{A}\cap\overline{B} = 1 - (\frac{1}{1+5x})^{1/2}$$

Hence, DeMorgan's law is satisfied.

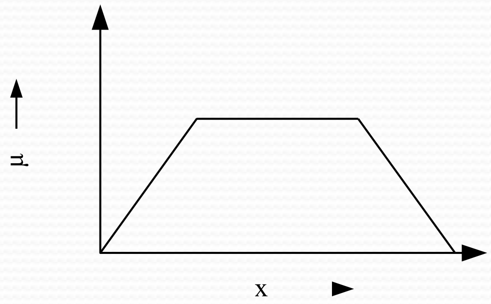
Fuzzy membership functions

Membership function on a discrete universe of course is trivial. However, a membership function on a continuous universe of discourse needs a special attention.

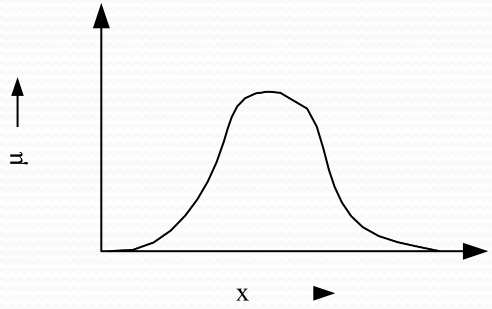
Following figures shows a typical examples of membership functions.



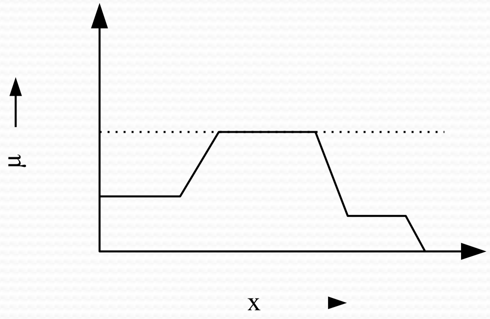
< triangular >



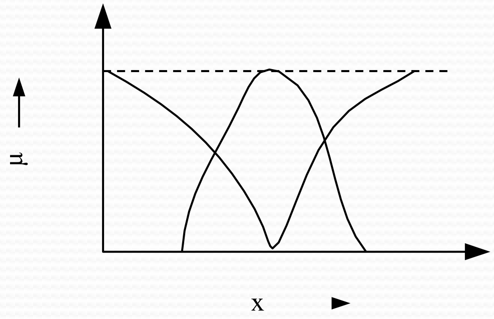
< trapezoidal >



< curve >



< non-uniform >



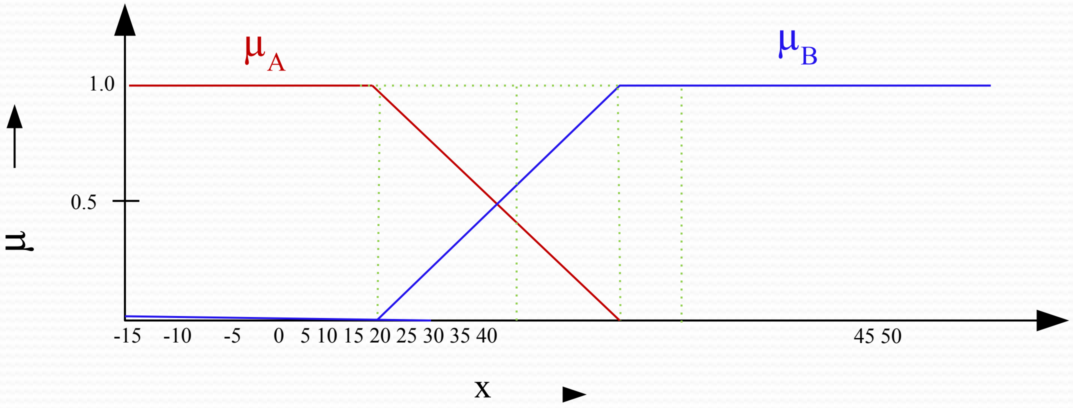
< non-uniform >

A real-life example

Two fuzzy sets *A* and *B* with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively defined as below.

A = **Cold climate** with $\mu_A(x)$ as the MF.

B = **Hot climate** with $\mu_B(x)$ as the M.F.



Here, *X* being the universe of discourse representing entire range of temperatures.

Example 2: A real-life example

What are the fuzzy sets representing the following?

1. Not cold climate
2. Not hold climate
3. Extreme climate
4. Pleasant climate

Note: Note that ”Not cold climate” ƒ= ”Hot climate” and vice-versa.

A real-life example

Answer would be the following

1. **Not cold climate**

   $\bar{A}$ with $1 - \mu_A(x)$ as the MF.

2. **Not hot climate**

   $\bar{B}$ with $1 - \mu_B(x)$ as the MF.

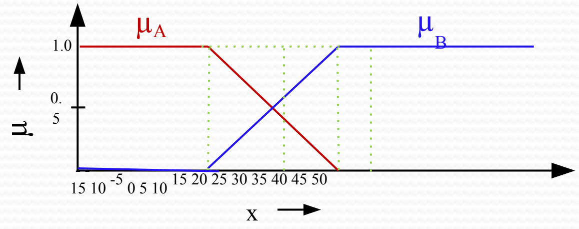
3. **Extreme climate**

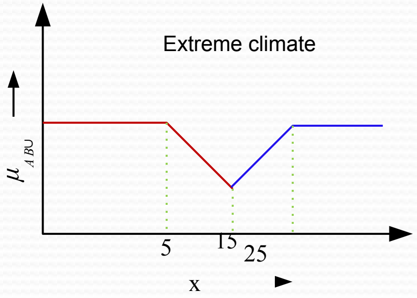
   $A \cup B$ with $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ as the MF.

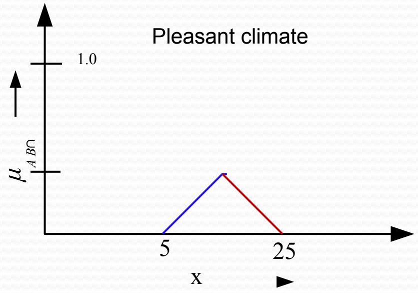
4. **Pleasant climate**

   $A \cap B$ with $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ as the MF.

The plot of the MFs of $A \cup B$ and $A \cap B$ are shown in the following.

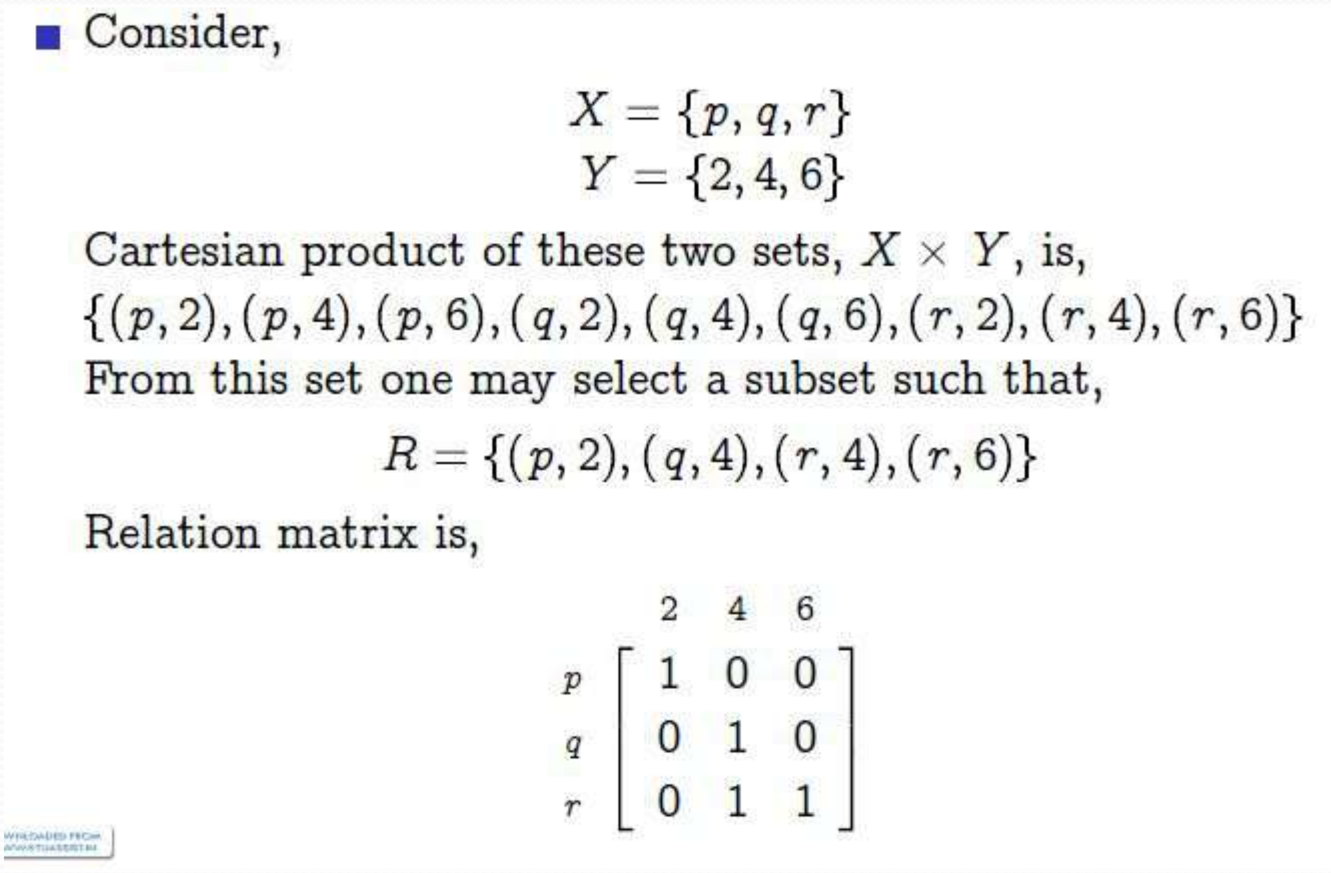



Extreme climate


Pleasant climate

Fuzzy Relations

Consider the Crisp Set Relation



- Consider,

$$X = \{p, q, r\}$$
$$Y = \{2, 4, 6\}$$

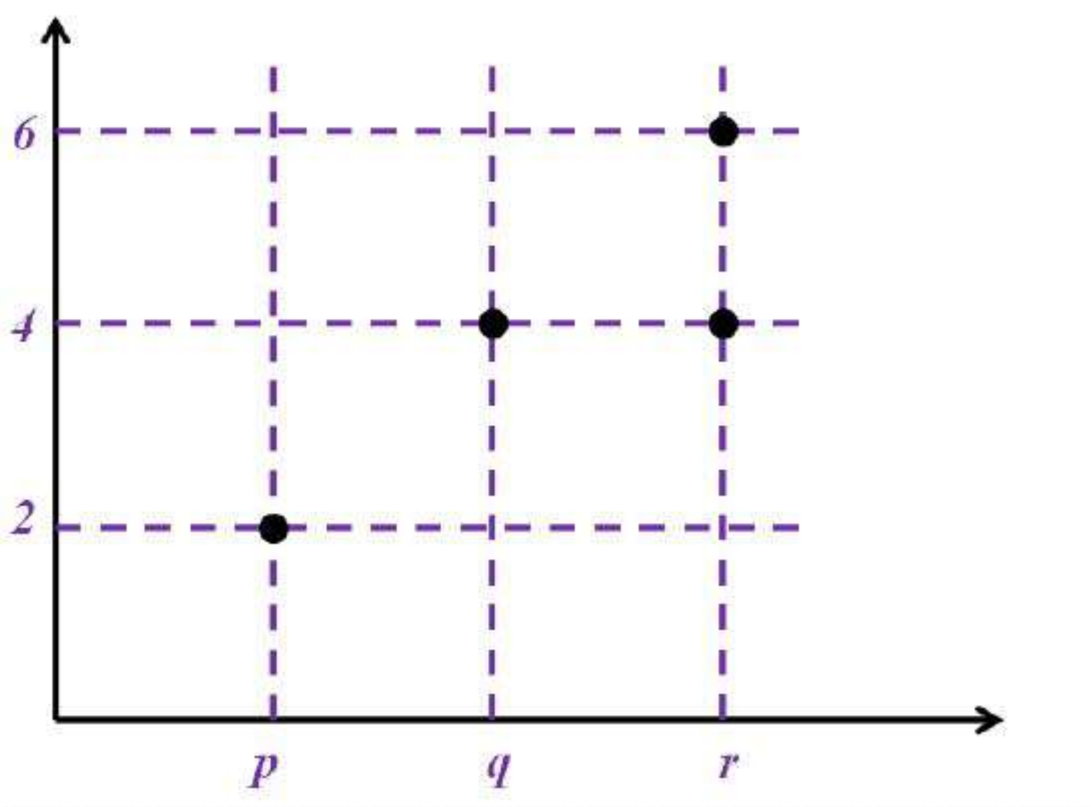
Cartesian product of these two sets, $X \times Y$, is,
$\{(p,2),(p,4),(p,6),(q,2),(q,4),(q,6),(r,2),(r,4),(r,6)\}$
From this set one may select a subset such that,

$$R = \{(p,2),(q,4),(r,4),(r,6)\}$$

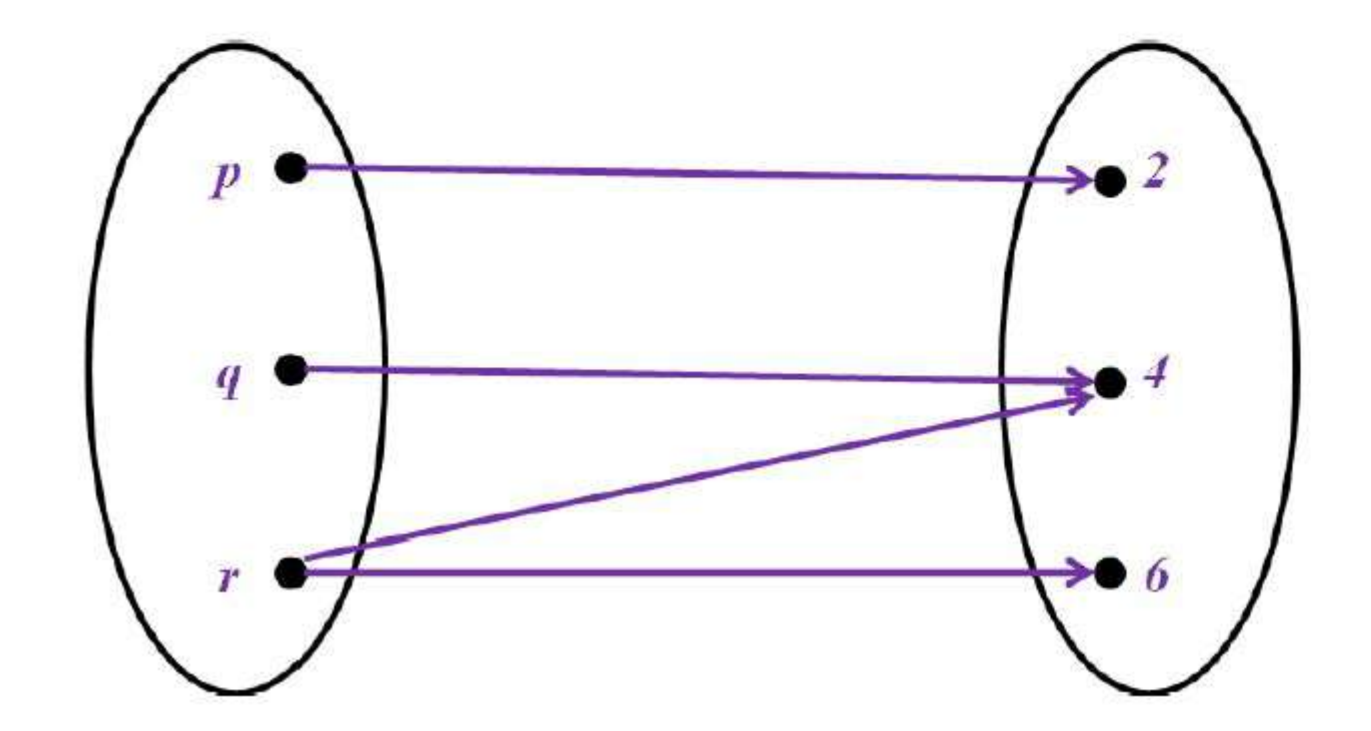
Relation matrix is,

$$\begin{array}{cc} & \begin{array}{ccc} 2 & 4 & 6 \end{array} \\ \begin{array}{c} p \\ q \\ r \end{array} & \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right] \end{array}$$

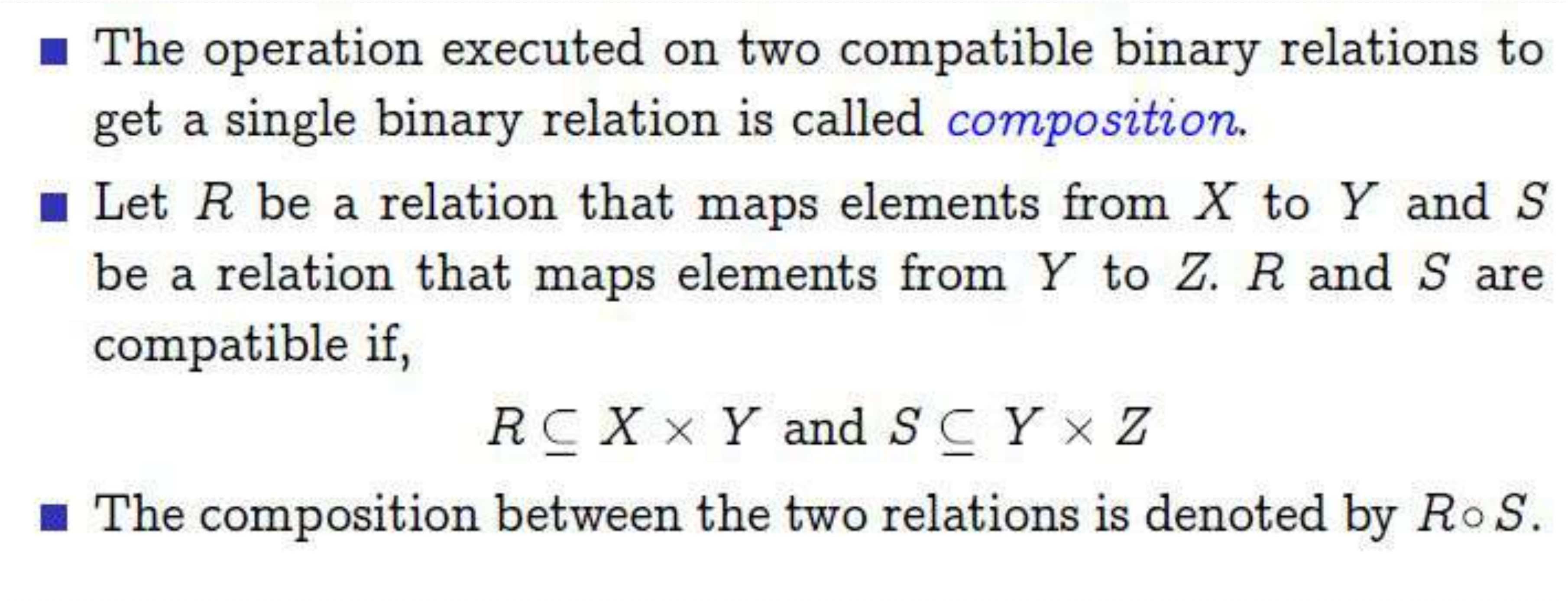
Coordinate diagram of the relation



Mapping of the relation



Composition on Classical Relations

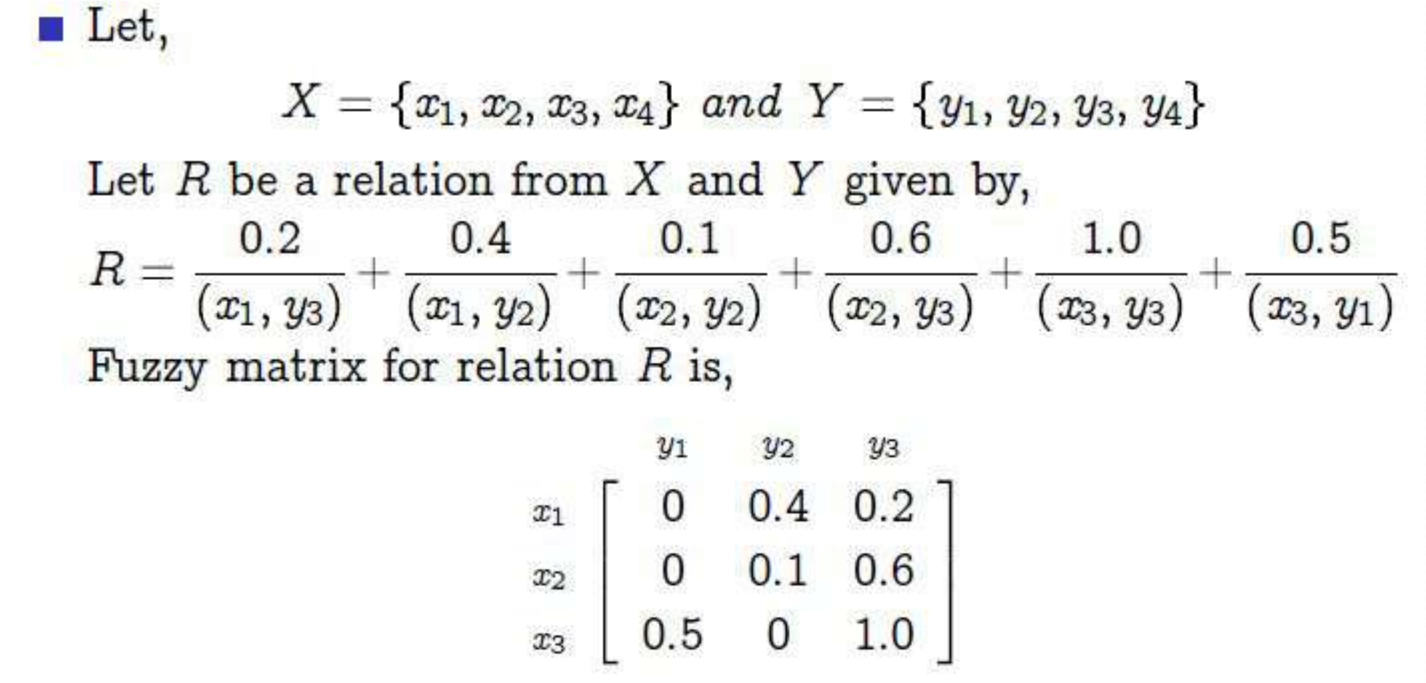


- The operation executed on two compatible binary relations to get a single binary relation is called *composition*.
- Let R be a relation that maps elements from X to Y and S be a relation that maps elements from Y to Z. R and S are compatible if,

$$R \subseteq X \times Y \text{ and } S \subseteq Y \times Z$$

- The composition between the two relations is denoted by $R \circ S$.

Fuzzy Relation



- Let,

$$X = \{x_1, x_2, x_3, x_4\}\ and\ Y = \{y_1, y_2, y_3, y_4\}$$

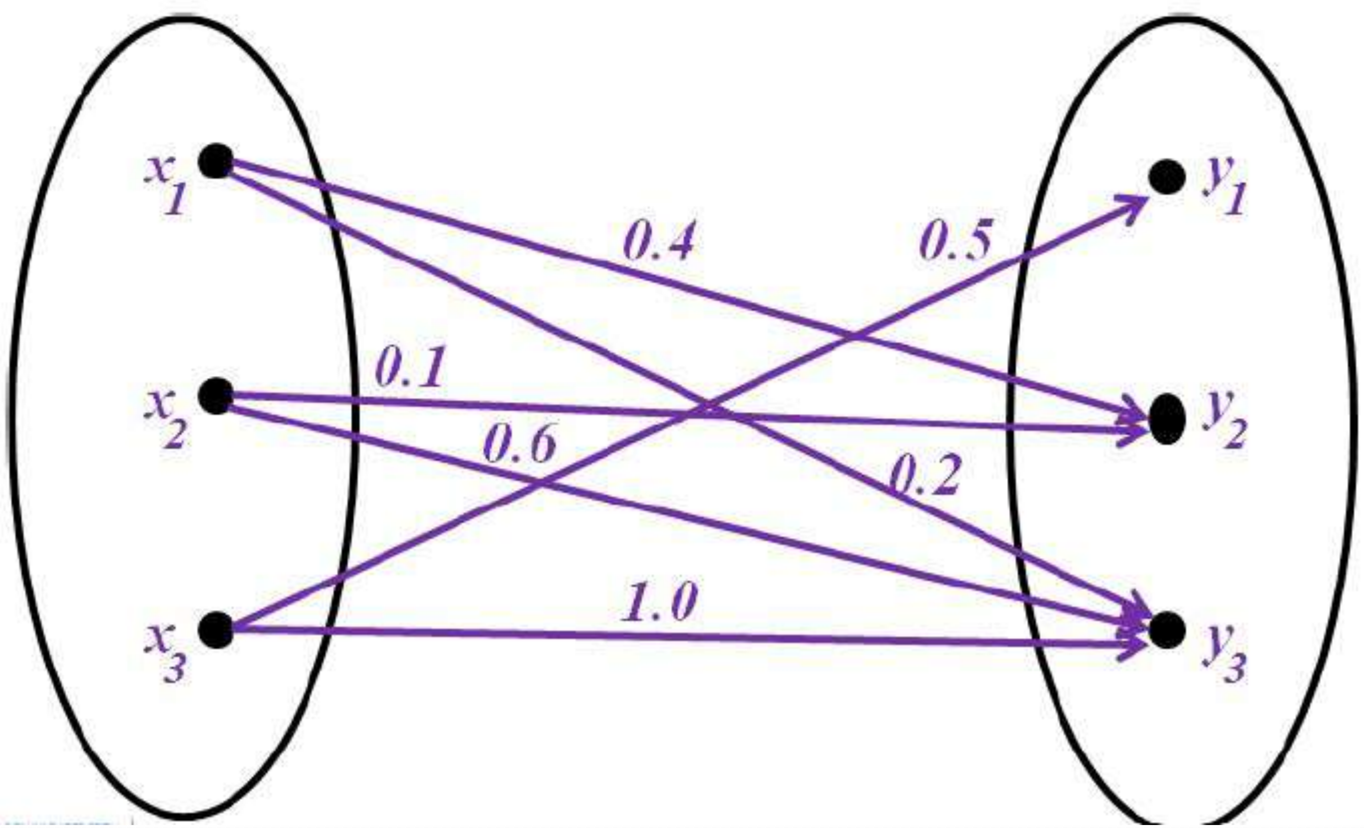
Let R be a relation from X and Y given by,

$$R = \frac{0.2}{(x_1, y_3)} + \frac{0.4}{(x_1, y_2)} + \frac{0.1}{(x_2, y_2)} + \frac{0.6}{(x_2, y_3)} + \frac{1.0}{(x_3, y_3)} + \frac{0.5}{(x_3, y_1)}$$

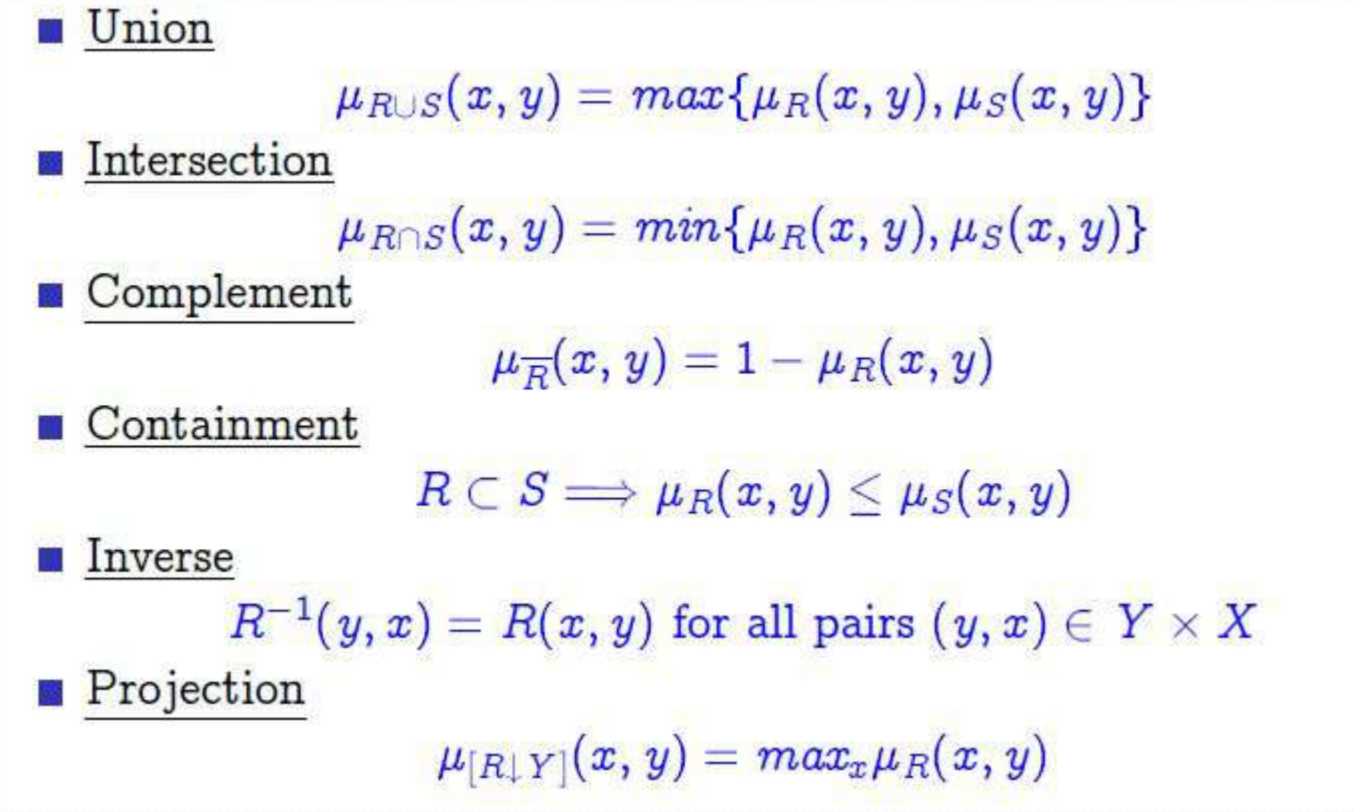
Fuzzy matrix for relation R is,

$$\begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0 & 0.4 & 0.2 \\ x_2 & 0 & 0.1 & 0.6 \\ x_3 & 0.5 & 0 & 1.0 \end{array}$$

Bipartite Graph



Operation on Fuzzy Relations



- Union

$$\mu_{R\cup S}(x,y) = max\{\mu_R(x,y),\mu_S(x,y)\}$$

- Intersection

$$\mu_{R\cap S}(x,y) = min\{\mu_R(x,y),\mu_S(x,y)\}$$

- Complement

$$\mu_{\overline{R}}(x,y) = 1 - \mu_R(x,y)$$

- Containment

$$R \subset S \Longrightarrow \mu_R(x,y) \leq \mu_S(x,y)$$

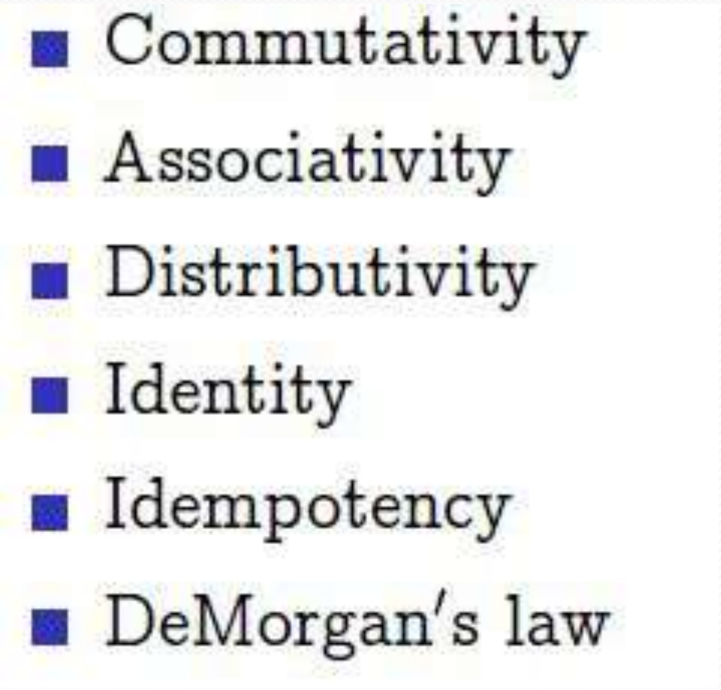
- Inverse

$$R^{-1}(y,x) = R(x,y) \text{ for all pairs } (y,x) \in Y \times X$$

- Projection

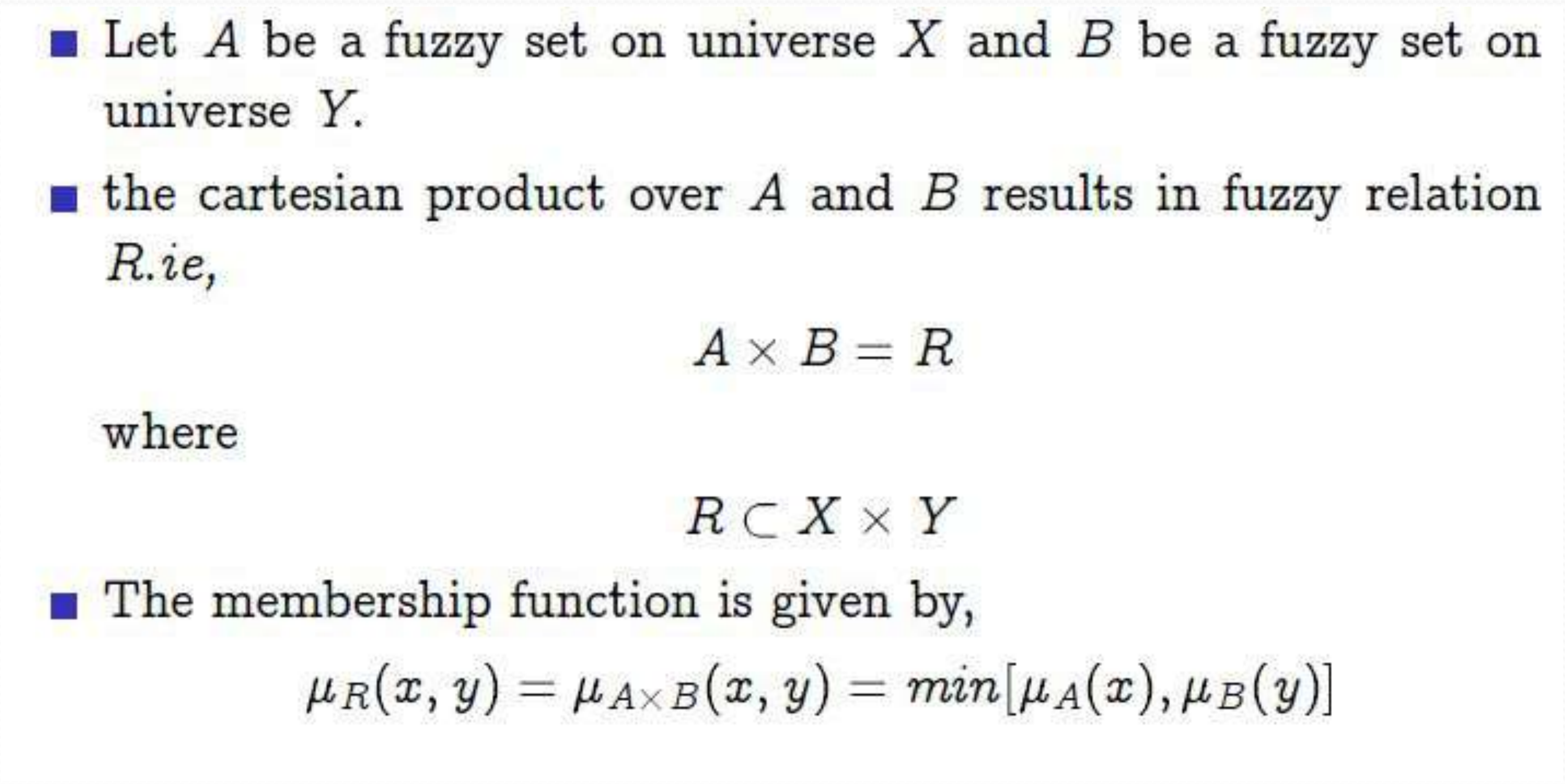
$$\mu_{[R\downarrow Y]}(x,y) = max_x \mu_R(x,y)$$

Properties of Fuzzy Relations



- Commutativity
- Associativity
- Distributivity
- Identity
- Idempotency
- DeMorgan's law

Fuzzy Compositions



- Let A be a fuzzy set on universe X and B be a fuzzy set on universe Y.
- the cartesian product over A and B results in fuzzy relation $R.ie,$

$$A \times B = R$$

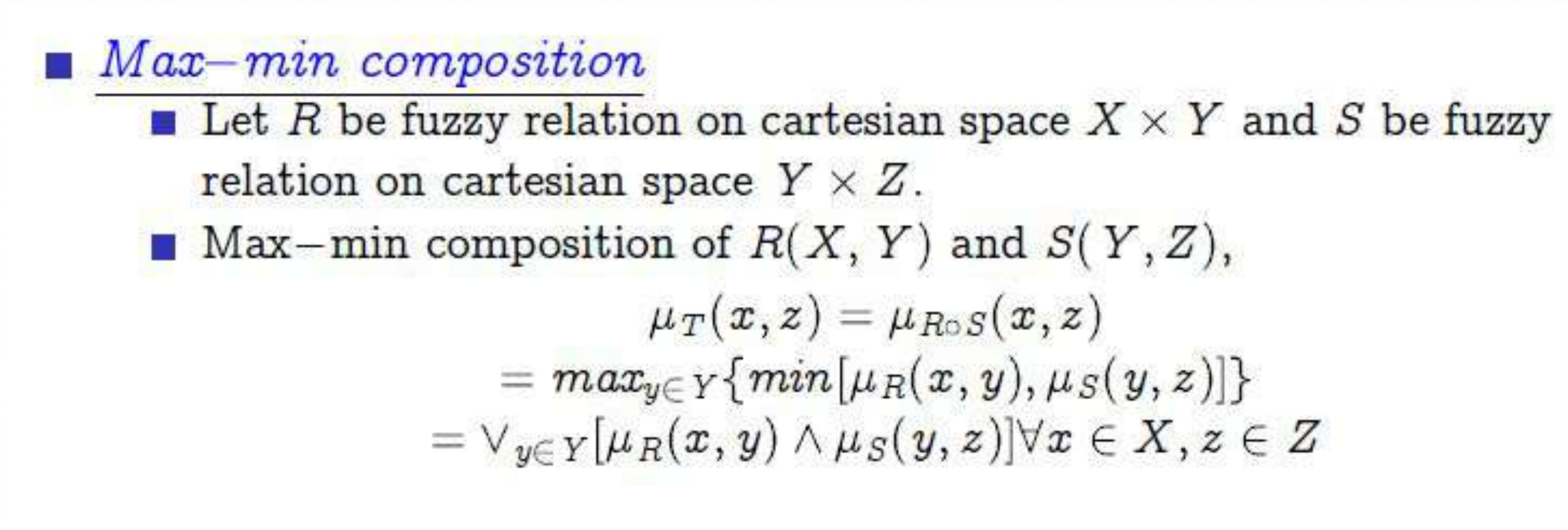
where

$$R \subset X \times Y$$

- The membership function is given by,

$$\mu_R(x, y) = \mu_{A \times B}(x, y) = min[\mu_A(x), \mu_B(y)]$$

Fuzzy Composition Techniques

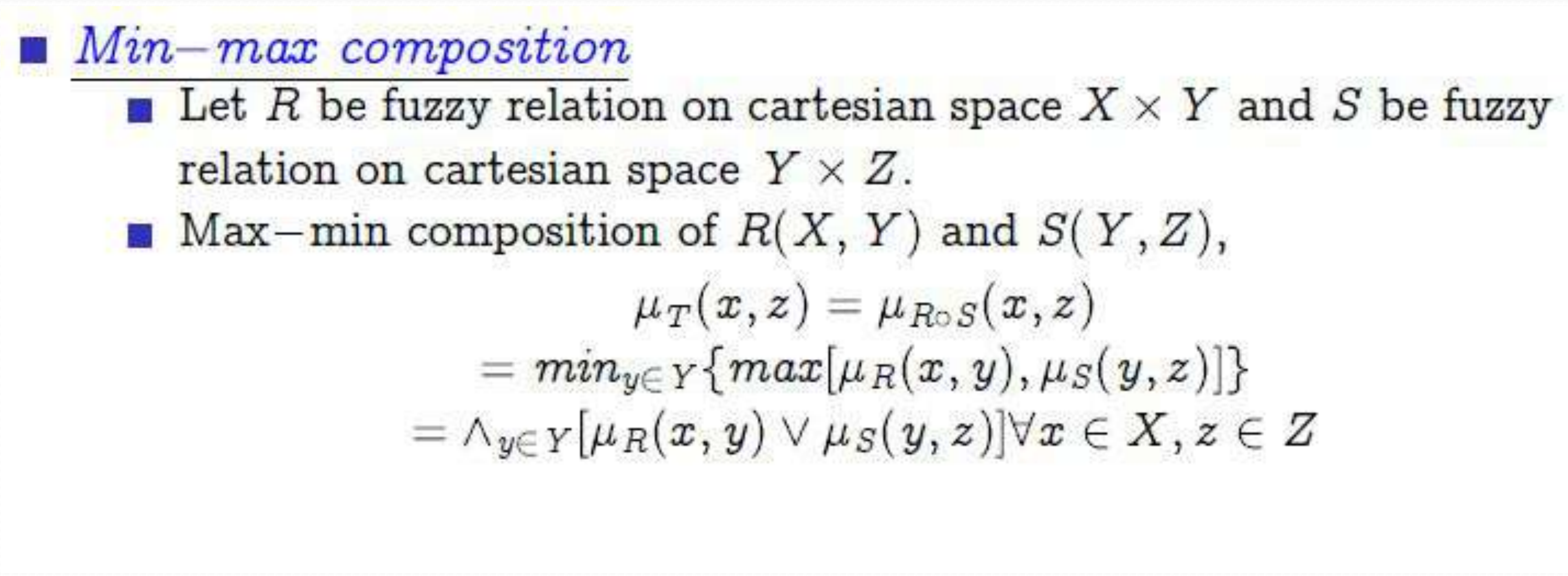


- *Max–min composition*
  - Let R be fuzzy relation on cartesian space $X \times Y$ and S be fuzzy relation on cartesian space $Y \times Z$.
  - Max–min composition of $R(X, Y)$ and $S(Y, Z)$,

$$\begin{aligned} \mu_T(x, z) &= \mu_{R \circ S}(x, z) \\ &= max_{y \in Y}\{min[\mu_R(x, y), \mu_S(y, z)]\} \\ &= \vee_{y \in Y}[\mu_R(x, y) \wedge \mu_S(y, z)] \forall x \in X, z \in Z \end{aligned}$$

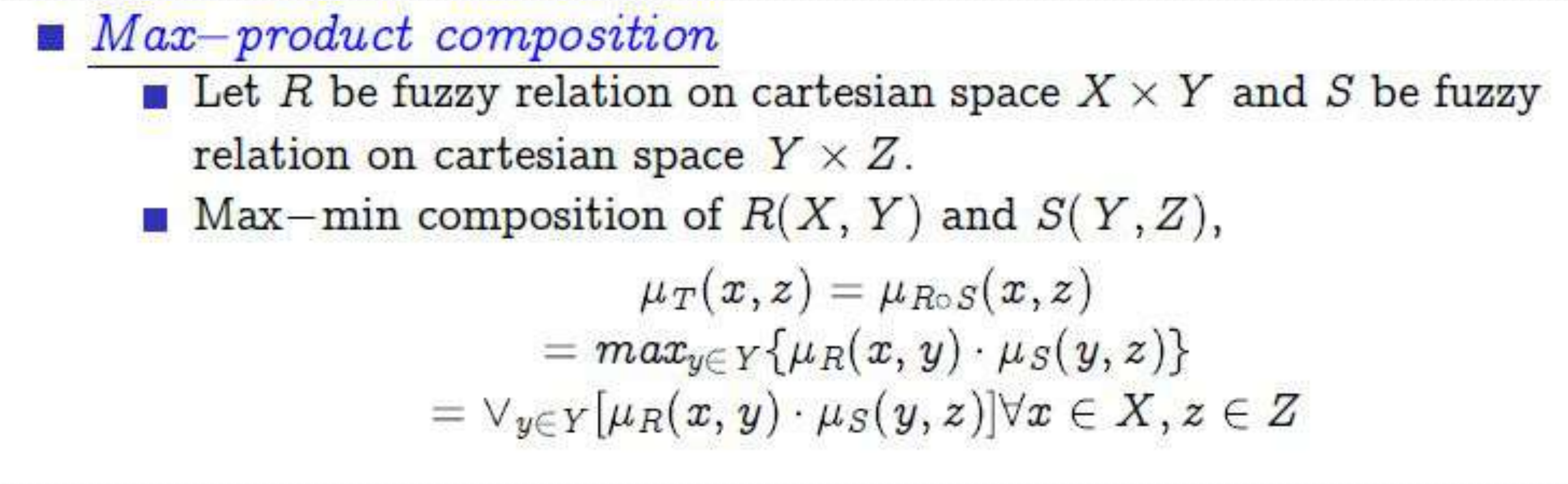
Other Compositions Techniques

- Min-max Composition
- Max-product Composition



- ***Min–max composition***
  - Let R be fuzzy relation on cartesian space $X \times Y$ and S be fuzzy relation on cartesian space $Y \times Z$.
  - Max–min composition of $R(X, Y)$ and $S(Y, Z)$,

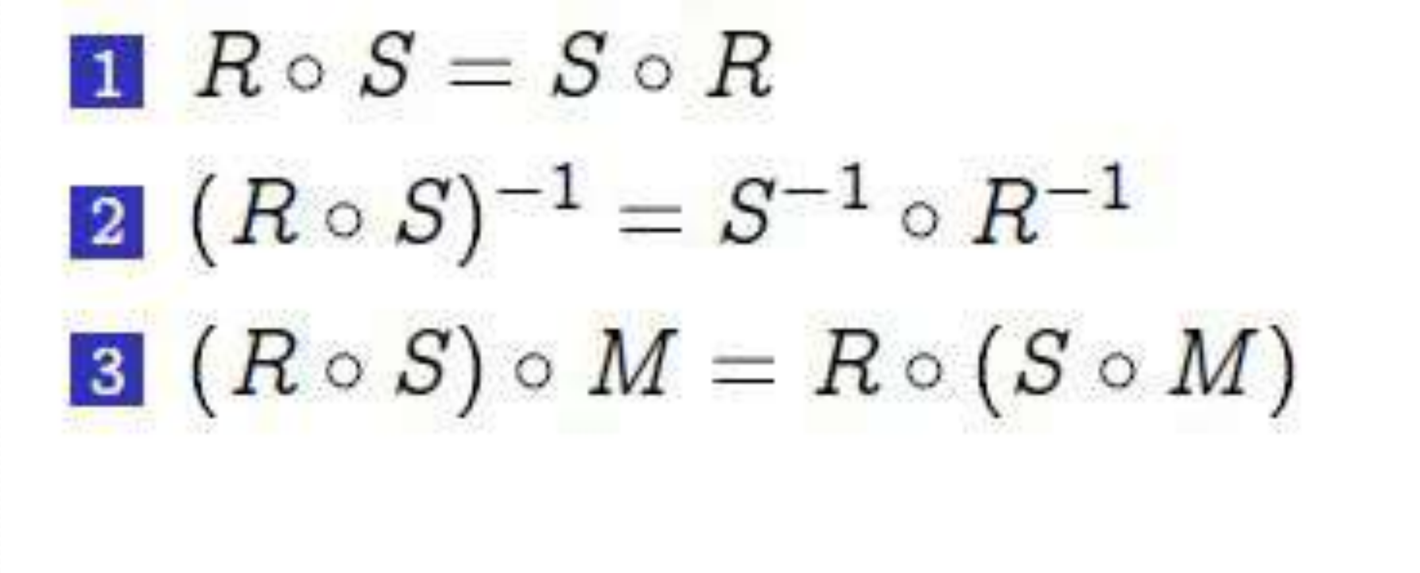
$$\begin{aligned}\mu_T(x, z) &= \mu_{R \circ S}(x, z)\\ &= min_{y \in Y}\{max[\mu_R(x, y), \mu_S(y, z)]\}\\ &= \wedge_{y \in Y}[\mu_R(x, y) \vee \mu_S(y, z)] \forall x \in X, z \in Z\end{aligned}$$



- *Max–product composition*
  - Let R be fuzzy relation on cartesian space $X \times Y$ and S be fuzzy relation on cartesian space $Y \times Z$.
  - Max−min composition of $R(X, Y)$ and $S(Y, Z)$,

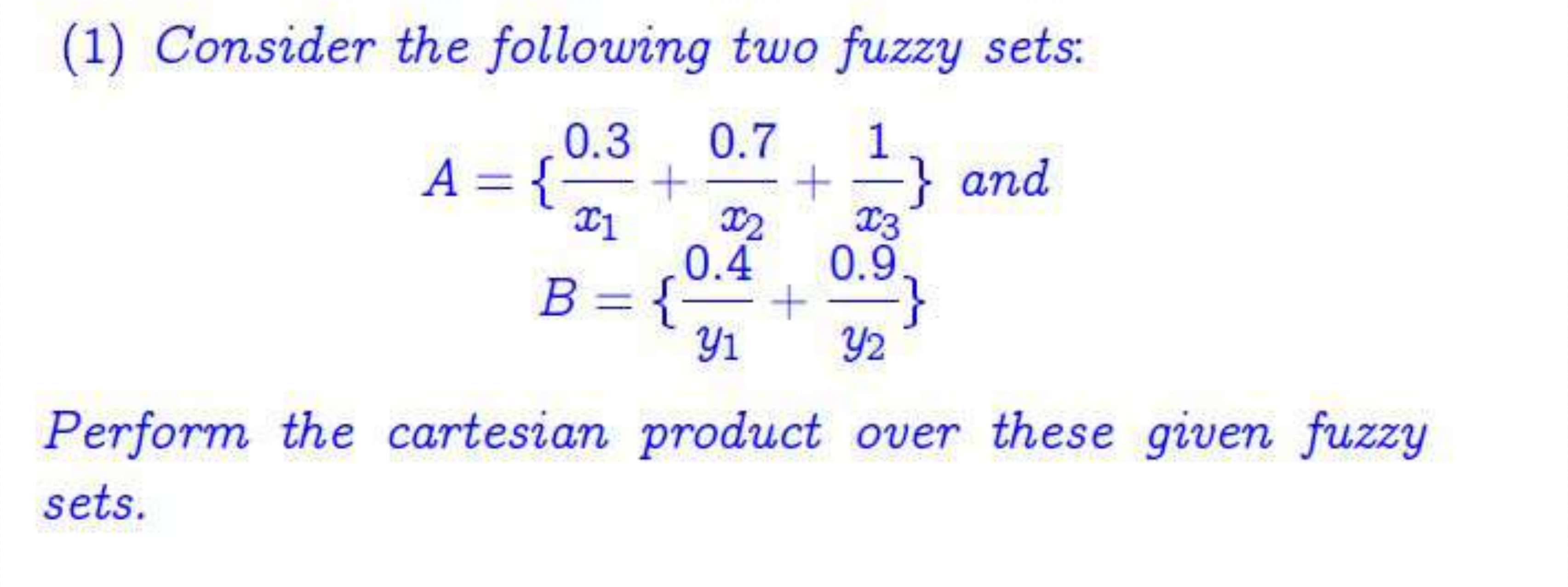
$$\begin{aligned}\mu_T(x, z) &= \mu_{R \circ S}(x, z) \\ &= max_{y \in Y}\{\mu_R(x, y) \cdot \mu_S(y, z)\} \\ &= \vee_{y \in Y}[\mu_R(x, y) \cdot \mu_S(y, z)] \forall x \in X, z \in Z\end{aligned}$$

Properties of Fuzzy Composition



1. $R \circ S = S \circ R$
2. $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$
3. $(R \circ S) \circ M = R \circ (S \circ M)$

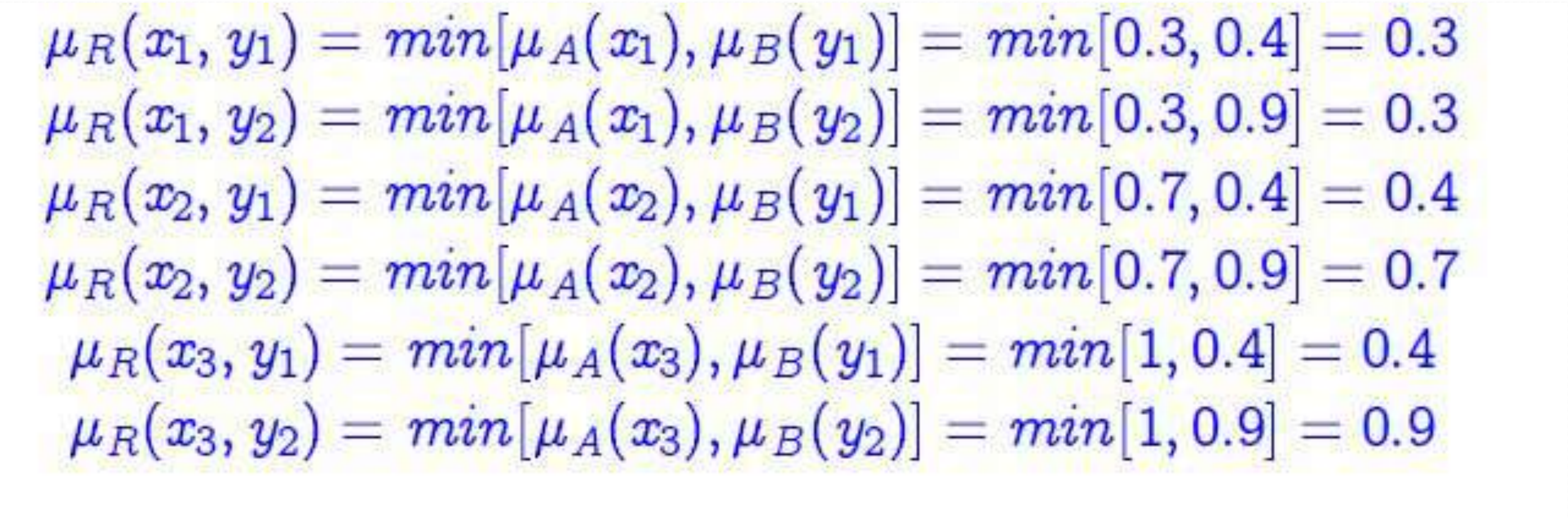
Problems



(1) *Consider the following two fuzzy sets:*

$$A = \{\frac{0.3}{x_1} + \frac{0.7}{x_2} + \frac{1}{x_3}\} \text{ and}$$
$$B = \{\frac{0.4}{y_1} + \frac{0.9}{y_2}\}$$

Perform the cartesian product over these given fuzzy sets.



$$\mu_R(x_1, y_1) = min[\mu_A(x_1), \mu_B(y_1)] = min[0.3, 0.4] = 0.3$$

$$\mu_R(x_1, y_2) = min[\mu_A(x_1), \mu_B(y_2)] = min[0.3, 0.9] = 0.3$$

$$\mu_R(x_2, y_1) = min[\mu_A(x_2), \mu_B(y_1)] = min[0.7, 0.4] = 0.4$$

$$\mu_R(x_2, y_2) = min[\mu_A(x_2), \mu_B(y_2)] = min[0.7, 0.9] = 0.7$$

$$\mu_R(x_3, y_1) = min[\mu_A(x_3), \mu_B(y_1)] = min[1, 0.4] = 0.4$$

$$\mu_R(x_3, y_2) = min[\mu_A(x_3), \mu_B(y_2)] = min[1, 0.9] = 0.9$$

$$\mu_R(x_1, y_1) = min[\mu_A(x_1), \mu_B(y_1)] = min[0.3, 0.4] = 0.3$$

$$\mu_R(x_1, y_2) = min[\mu_A(x_1), \mu_B(y_2)] = min[0.3, 0.9] = 0.3$$

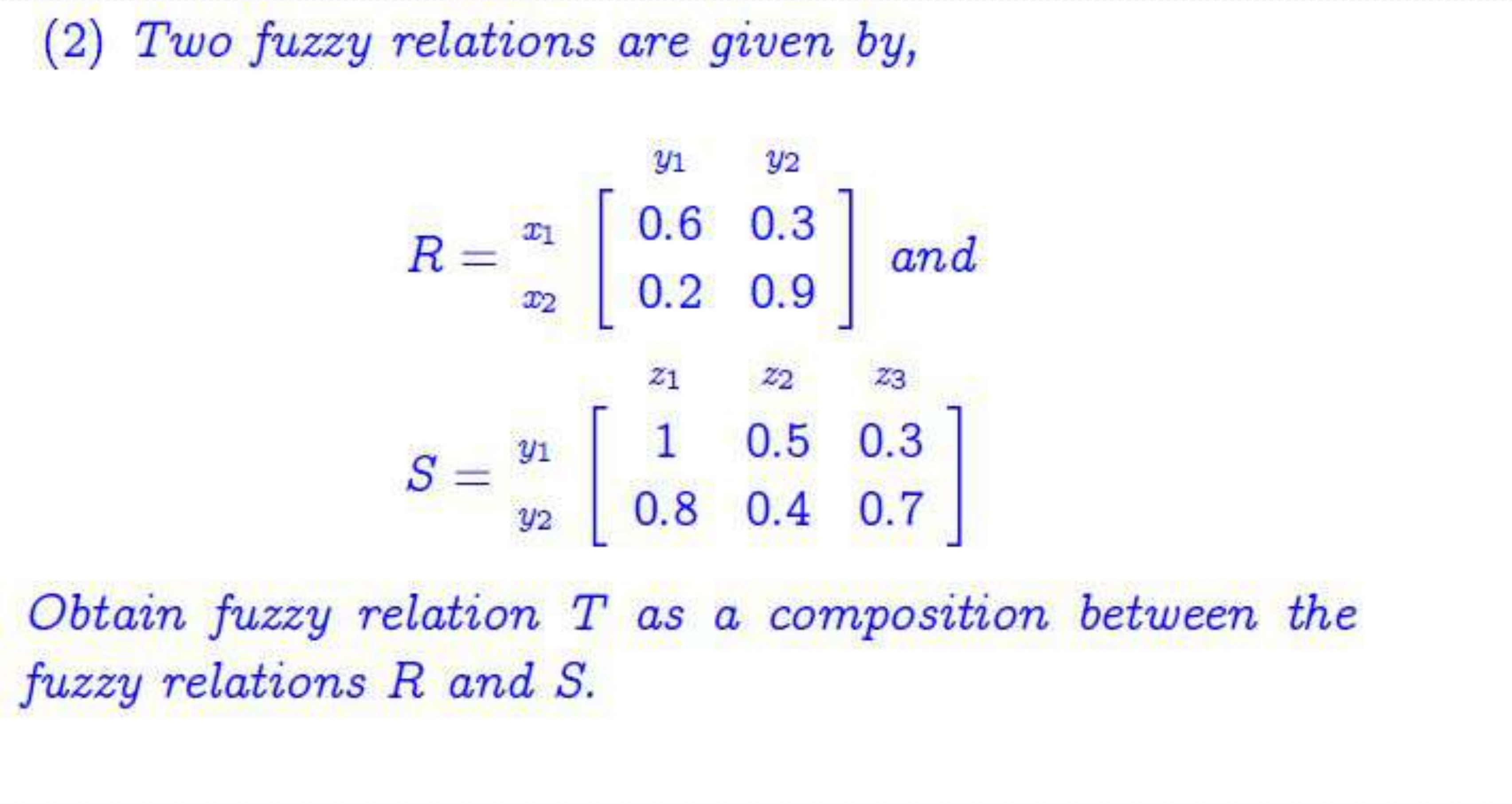
$$\mu_R(x_2, y_1) = min[\mu_A(x_2), \mu_B(y_1)] = min[0.7, 0.4] = 0.4$$

$$\mu_R(x_2, y_2) = min[\mu_A(x_2), \mu_B(y_2)] = min[0.7, 0.9] = 0.7$$

$$\mu_R(x_3, y_1) = min[\mu_A(x_3), \mu_B(y_1)] = min[1, 0.4] = 0.4$$

$$\mu_R(x_3, y_2) = min[\mu_A(x_3), \mu_B(y_2)] = min[1, 0.9] = 0.9$$

$$R = A \times B = \begin{array}{c c} & \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \left[\begin{array}{cc} 0.3 & 0.3 \\ 0.4 & 0.7 \\ 0.4 & 0.9 \end{array}\right] \end{array}$$

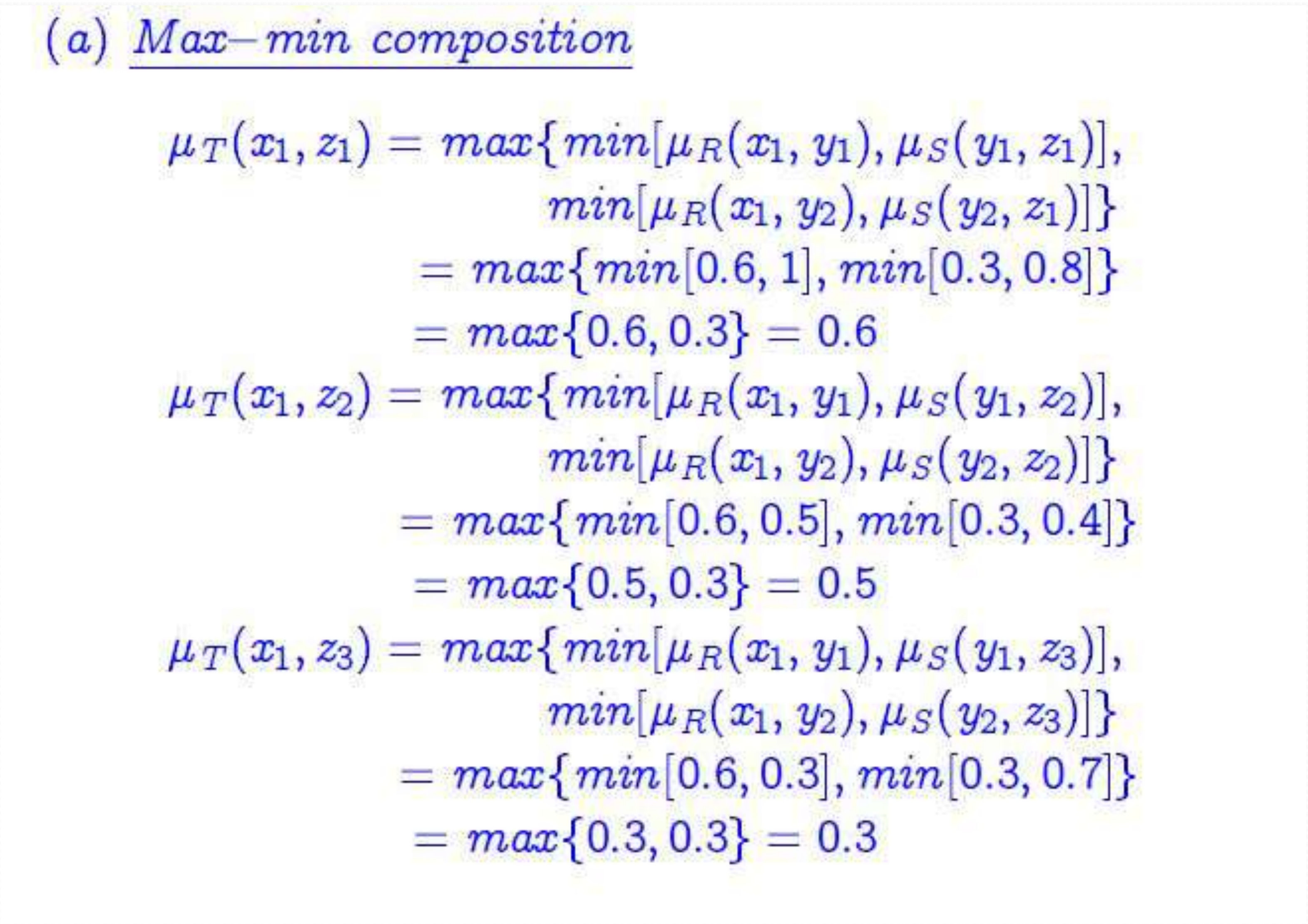


(2) *Two fuzzy relations are given by,*

$$R = \begin{array}{c|cc} & y_1 & y_2 \\ \hline x_1 & 0.6 & 0.3 \\ x_2 & 0.2 & 0.9 \end{array} \quad \textit{and}$$

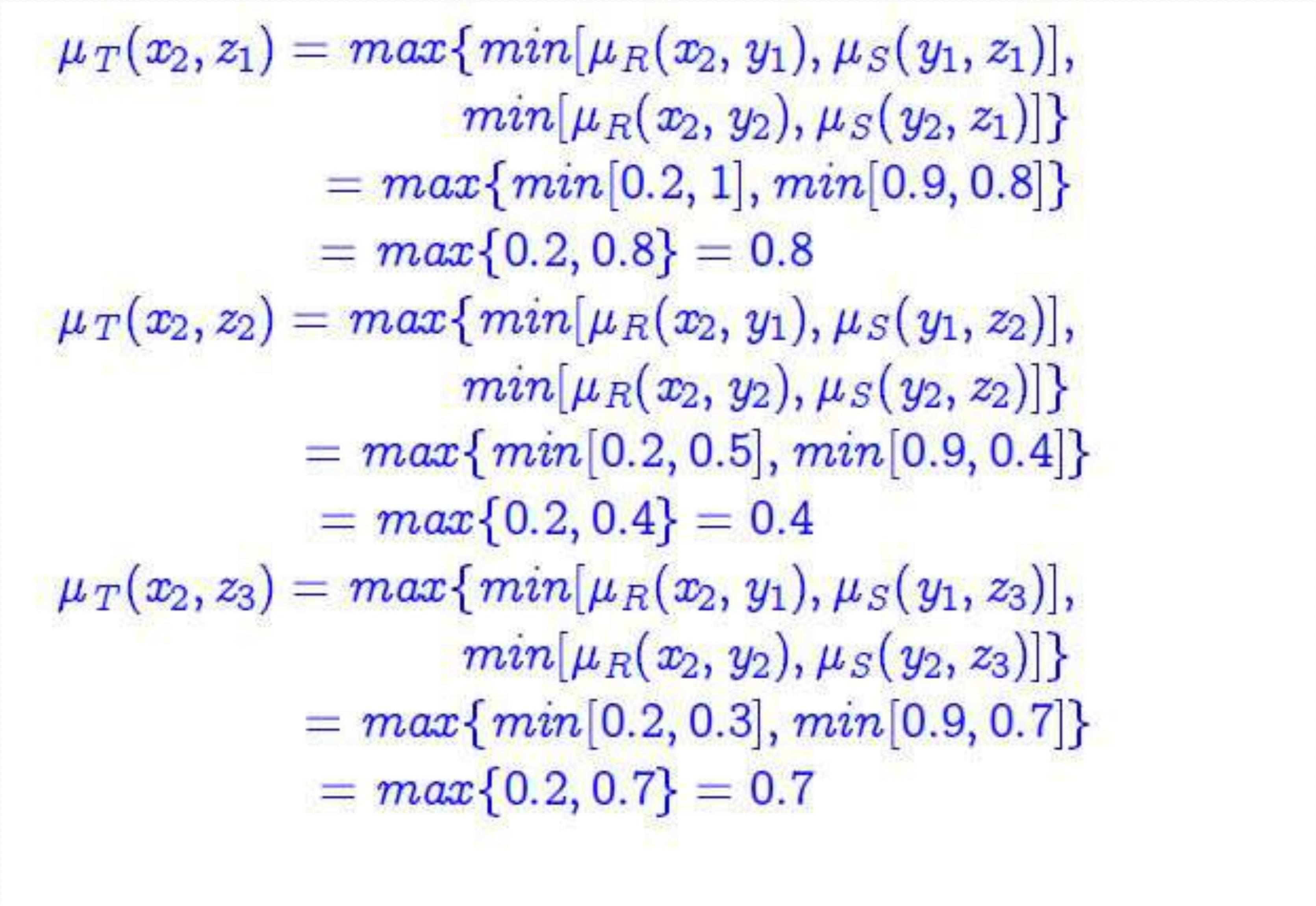
$$S = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ \hline y_1 & 1 & 0.5 & 0.3 \\ y_2 & 0.8 & 0.4 & 0.7 \end{array}$$

Obtain fuzzy relation T as a composition between the fuzzy relations R and S.

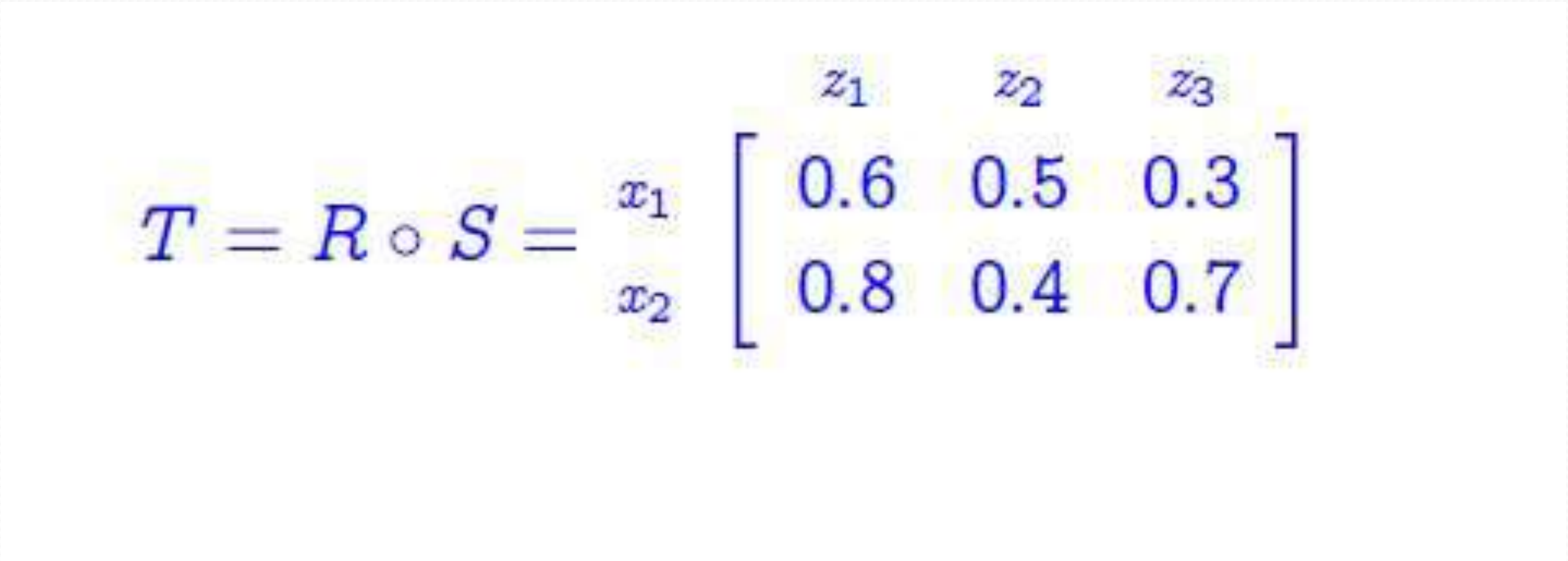


(a) *Max–min composition*

$$\begin{aligned}
\mu_T(x_1, z_1) &= max\{min[\mu_R(x_1, y_1), \mu_S(y_1, z_1)],\\
&\qquad min[\mu_R(x_1, y_2), \mu_S(y_2, z_1)]\}\\
&= max\{min[0.6, 1], min[0.3, 0.8]\}\\
&= max\{0.6, 0.3\} = 0.6\\
\mu_T(x_1, z_2) &= max\{min[\mu_R(x_1, y_1), \mu_S(y_1, z_2)],\\
&\qquad min[\mu_R(x_1, y_2), \mu_S(y_2, z_2)]\}\\
&= max\{min[0.6, 0.5], min[0.3, 0.4]\}\\
&= max\{0.5, 0.3\} = 0.5\\
\mu_T(x_1, z_3) &= max\{min[\mu_R(x_1, y_1), \mu_S(y_1, z_3)],\\
&\qquad min[\mu_R(x_1, y_2), \mu_S(y_2, z_3)]\}\\
&= max\{min[0.6, 0.3], min[0.3, 0.7]\}\\
&= max\{0.3, 0.3\} = 0.3
\end{aligned}$$

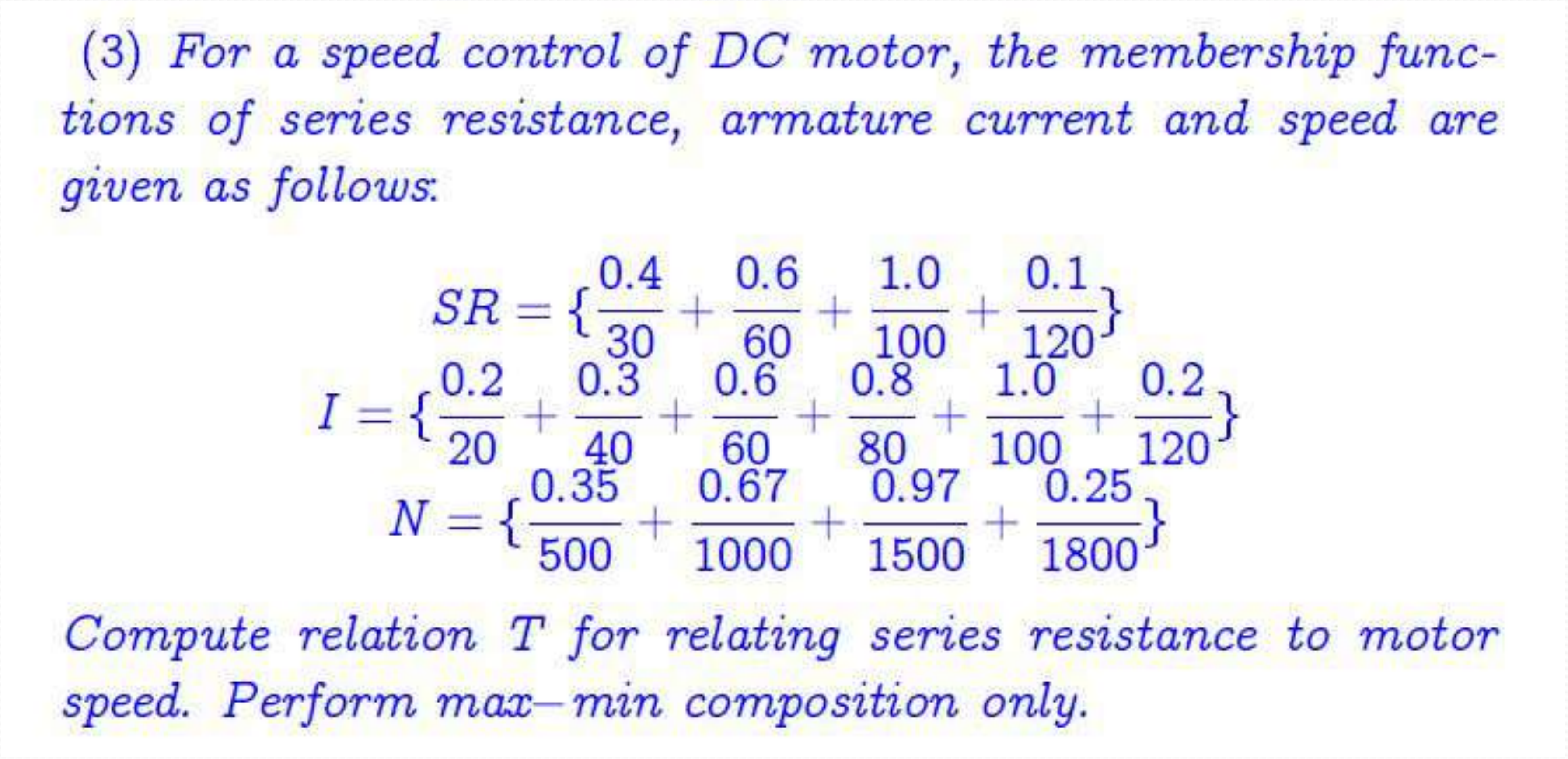


$$\begin{aligned}
\mu_T(x_2, z_1) &= max\{min[\mu_R(x_2, y_1), \mu_S(y_1, z_1)], \\
&\qquad min[\mu_R(x_2, y_2), \mu_S(y_2, z_1)]\} \\
&= max\{min[0.2, 1], min[0.9, 0.8]\} \\
&= max\{0.2, 0.8\} = 0.8 \\
\mu_T(x_2, z_2) &= max\{min[\mu_R(x_2, y_1), \mu_S(y_1, z_2)], \\
&\qquad min[\mu_R(x_2, y_2), \mu_S(y_2, z_2)]\} \\
&= max\{min[0.2, 0.5], min[0.9, 0.4]\} \\
&= max\{0.2, 0.4\} = 0.4 \\
\mu_T(x_2, z_3) &= max\{min[\mu_R(x_2, y_1), \mu_S(y_1, z_3)], \\
&\qquad min[\mu_R(x_2, y_2), \mu_S(y_2, z_3)]\} \\
&= max\{min[0.2, 0.3], min[0.9, 0.7]\} \\
&= max\{0.2, 0.7\} = 0.7
\end{aligned}$$



$$T = R \circ S = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{ccc} z_1 & z_2 & z_3 \\ \hline \end{array}$$

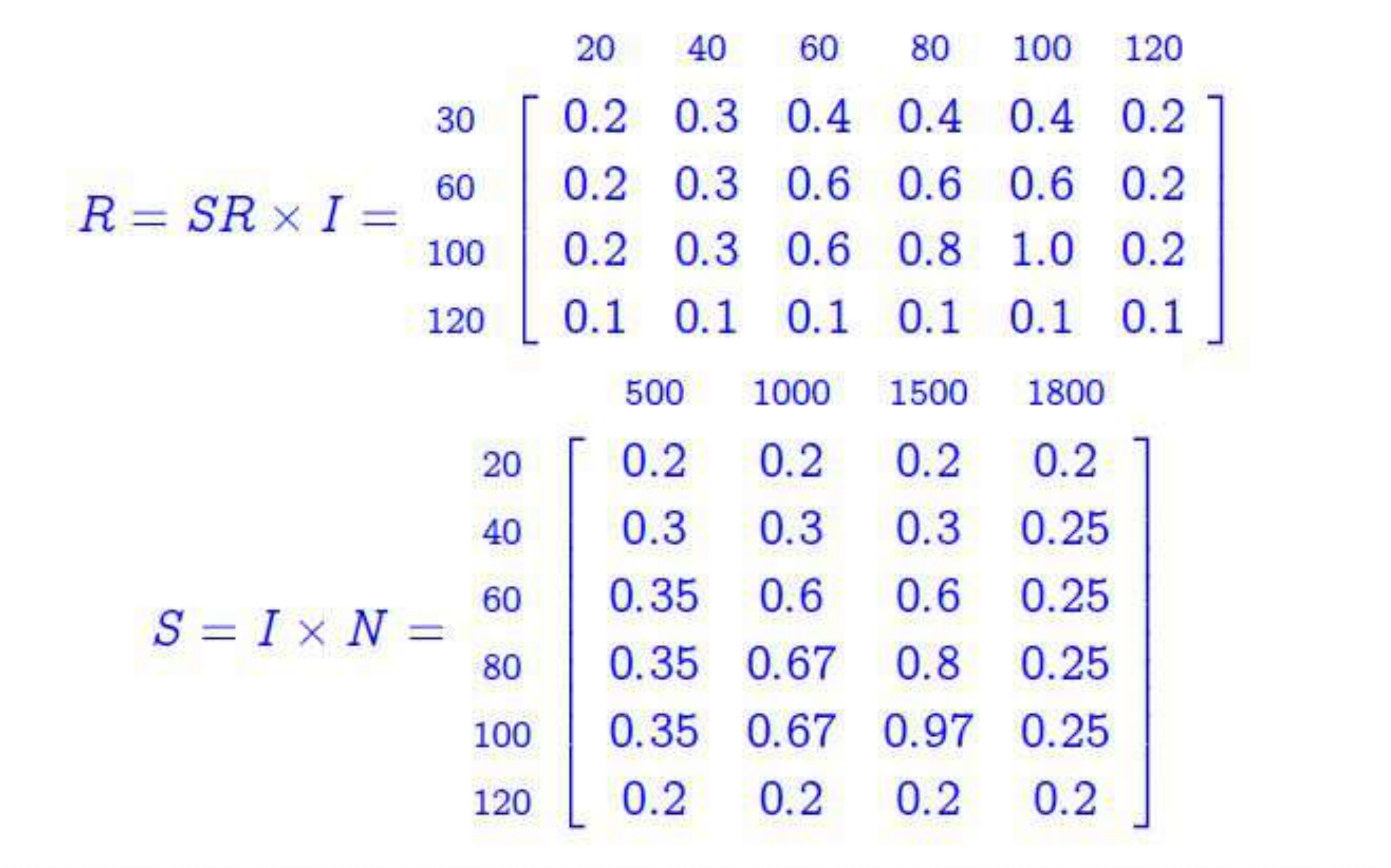
$$T = R \circ S = \begin{array}{cc} & \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{bmatrix} 0.6 & 0.5 & 0.3 \\ 0.8 & 0.4 & 0.7 \end{bmatrix} \end{array}$$



(3) *For a speed control of DC motor, the membership functions of series resistance, armature current and speed are given as follows:*

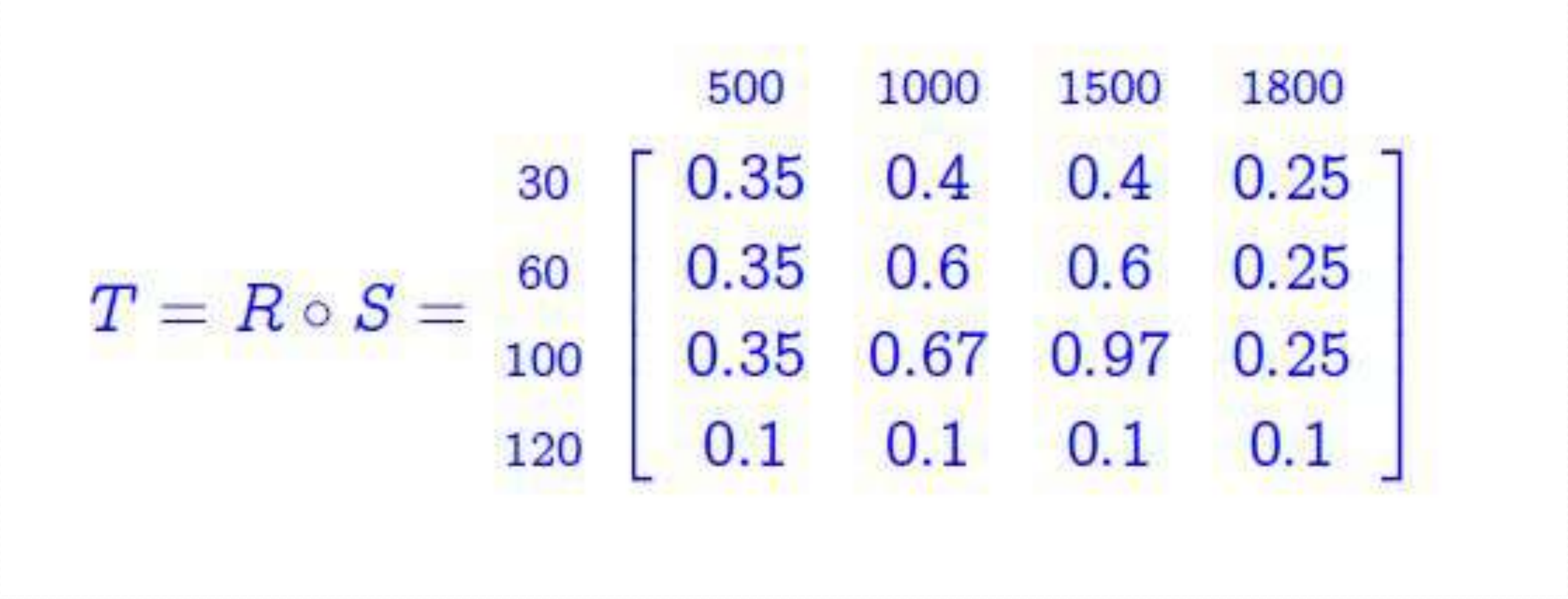
$$SR = \{\frac{0.4}{30} + \frac{0.6}{60} + \frac{1.0}{100} + \frac{0.1}{120}\}$$
$$I = \{\frac{0.2}{20} + \frac{0.3}{40} + \frac{0.6}{60} + \frac{0.8}{80} + \frac{1.0}{100} + \frac{0.2}{120}\}$$
$$N = \{\frac{0.35}{500} + \frac{0.67}{1000} + \frac{0.97}{1500} + \frac{0.25}{1800}\}$$

Compute relation T for relating series resistance to motor speed. Perform max–min composition only.



$$R = SR \times I = \begin{array}{c|cccccc} & 20 & 40 & 60 & 80 & 100 & 120 \\ \hline 30 & 0.2 & 0.3 & 0.4 & 0.4 & 0.4 & 0.2 \\ 60 & 0.2 & 0.3 & 0.6 & 0.6 & 0.6 & 0.2 \\ 100 & 0.2 & 0.3 & 0.6 & 0.8 & 1.0 & 0.2 \\ 120 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{array}$$

$$S = I \times N = \begin{array}{c|cccc} & 500 & 1000 & 1500 & 1800 \\ \hline 20 & 0.2 & 0.2 & 0.2 & 0.2 \\ 40 & 0.3 & 0.3 & 0.3 & 0.25 \\ 60 & 0.35 & 0.6 & 0.6 & 0.25 \\ 80 & 0.35 & 0.67 & 0.8 & 0.25 \\ 100 & 0.35 & 0.67 & 0.97 & 0.25 \\ 120 & 0.2 & 0.2 & 0.2 & 0.2 \end{array}$$



$$T = R \circ S = \begin{array}{c|cccc} & 500 & 1000 & 1500 & 1800 \\ \hline 30 & 0.35 & 0.4 & 0.4 & 0.25 \\ 60 & 0.35 & 0.6 & 0.6 & 0.25 \\ 100 & 0.35 & 0.67 & 0.97 & 0.25 \\ 120 & 0.1 & 0.1 & 0.1 & 0.1 \end{array}$$

$$\mu_{S\circ R}(x,y) = \max_v \min\left(\mu_R(x,v), \mu_S(v,y)\right)$$

Example

R	a	b	c	d
1	0.1	0.2	0.0	1.0
2	0.3	0.3	0.0	0.2
3	0.8	0.9	1.0	0.4

S	α	β	γ
a	0.9	0.0	0.3
b	0.2	1.0	0.8
c	0.8	0.0	0.7
d	0.4	0.2	0.3

	0.1	0.2	0.0	1.0
mi	0.9	0.2	0.8	0.4
n max	0.1	0.2	0.0	0.4

$R\circ S$	α	β	γ
1	$\boxed{0.4}$	0.2	0.3
2	0.3	0.3	0.3
3	0.8	0.9	0.8