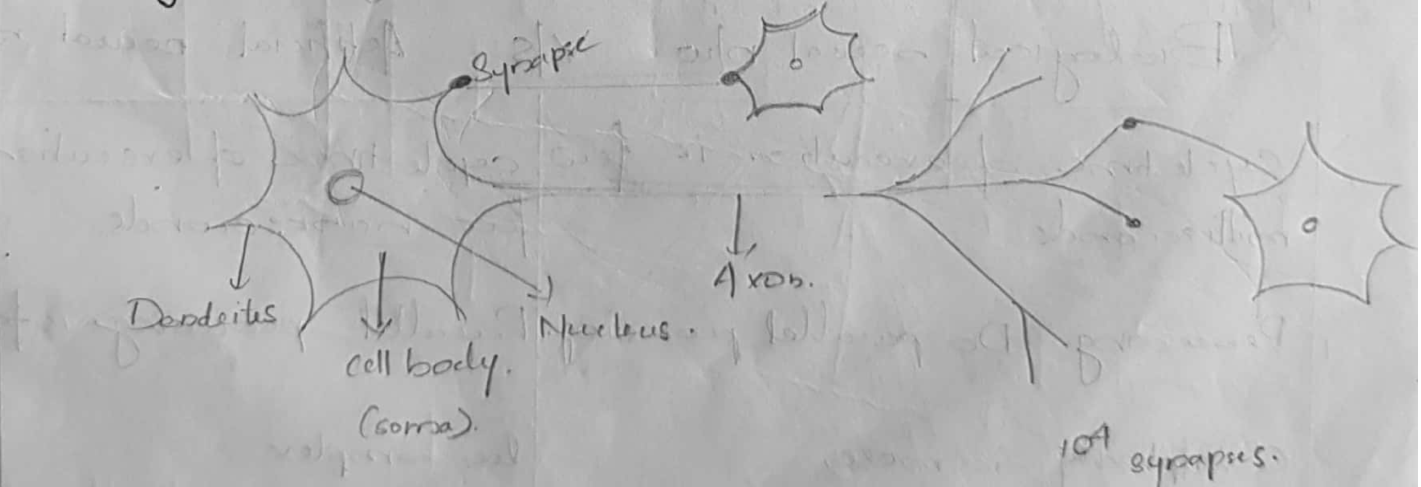
7/8/17

Module 1

Introduction to Soft Computing

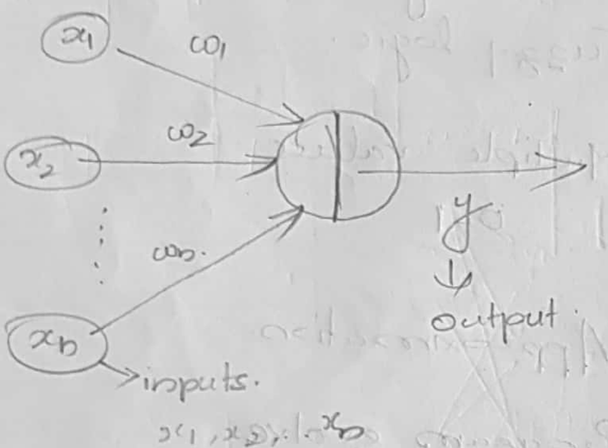
Hard computing	v/s Soft computing
Precise systems. precise calculations	Real world problems. like handwriting recognition, speaker recognition, weather forecasting.
Binary logic	Fuzzy logic.
Two states low, high 0,1	Multiple values 0–1
Precision	Approximation.
Requires program	Programs evolve
Maybe sequential	Parallel processing.
Deterministic	Randomness.
Can't tolerate errors.	Tolerate errors.

Biological Neural Network:



Biological neuron v/s Artificial neuron

Biological neuron	Artificial neuron
Cell	neuron
Dendrites	Weighted links
Soma	Net input
Axon	output



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 x_i is the i^{th} input

w_i is i^{th} weight.

Biological neural n/w V/S Artificial neural n/w

Biological neural n/w	Artificial neural n/w
Cycle time of execution is few milliseconds.	cycle time of execution is few nanoseconds.
Processing - Do parallel processing.	Parallel processing is faster
Complexity is more. 10^{11} neurons 10^{15} interconnections	less complex
Storage capacity - stores info in synapse or interconnection.	older memory is lost when overwritten

Scanned by CamScanner

Biological Neural N/W v/s Artificial Neural N/W.

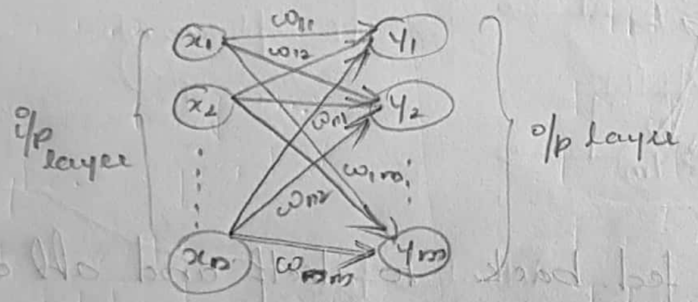
fails to recollect memory	Once stored information is permanently stored.
More fault tolerant	Less fault tolerant
control mechanisms done by chemicals.	CPU controls.

8/8/17 Tue.

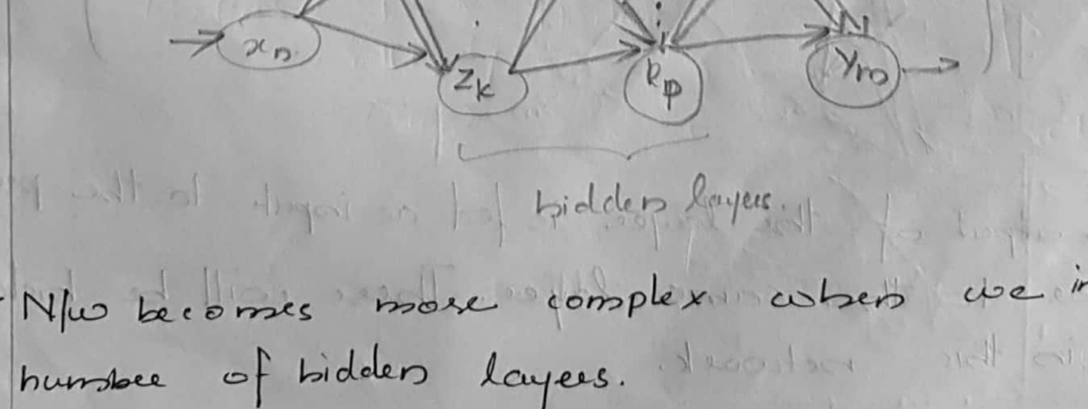
Basic Models of Artificial Neural Network

Interconnection

1. Single layer feed forward network.



2. Multi layer feed forward network

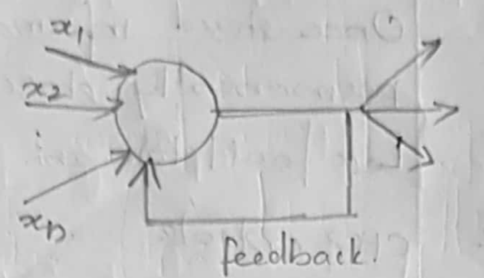


There are hidden layers present.

Features are given as i/p

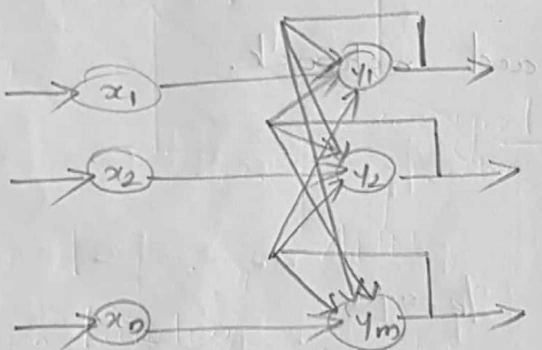
→ N/w becomes more complex when we increase the number of hidden layers.

3. Single node with own feedback



when the output is directed back as input to the same or preceding layer it will result in feedback networks.

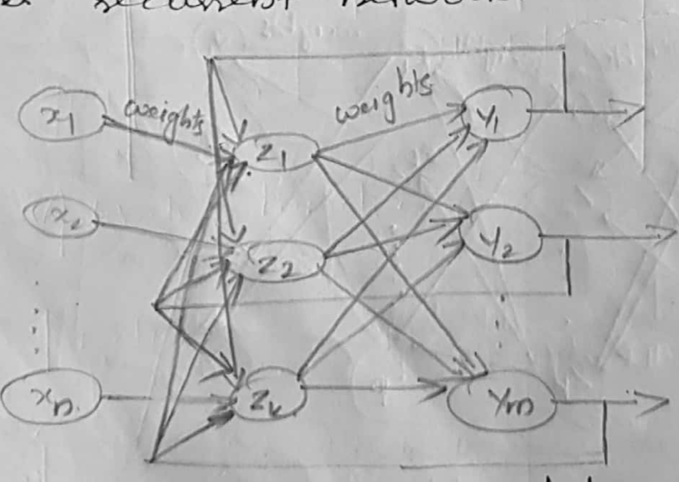
4. Single layer recurrent network.



single layer of i/p & o/p neurons

The output is fed back to itself and all other neurons

5. Multi layer recurrent network.



The output of the layer is fed as input to the previous layer or to the same layer. There will be hidden layers in this network.

6 Lateral inhibition network

Each processing neuron receives two different inputs. excitory inputs from nearby processing neuron and inhibitory input from more distantly located processing element.

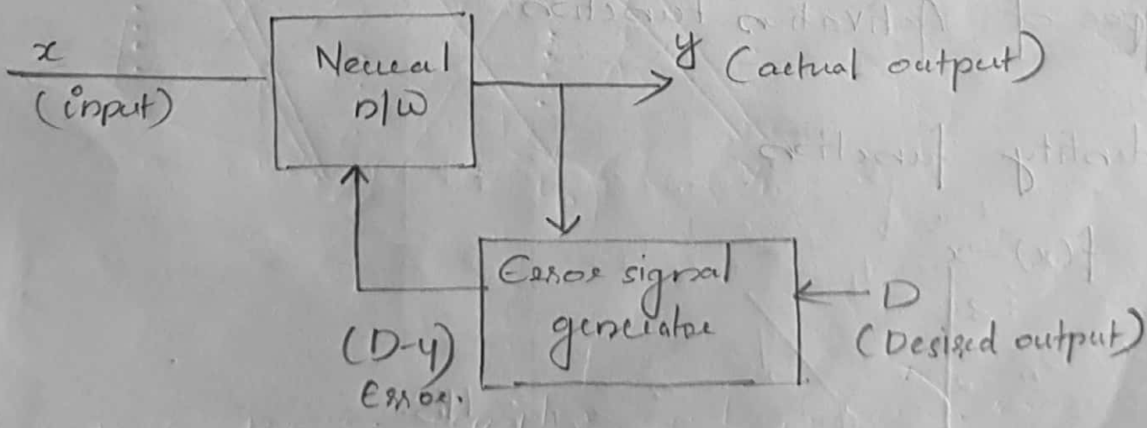
10/8/17 Thu.

Learning. - 3 types.

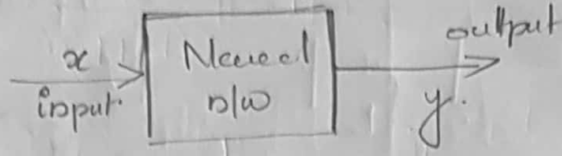
1. Supervised learning.
2. Un-supervised learning.
3. Reinforcement learning

1. Supervised learning:

* under the guidance of a supervisor
* Eg: student learns under the guidance of a teacher.
* correction is made

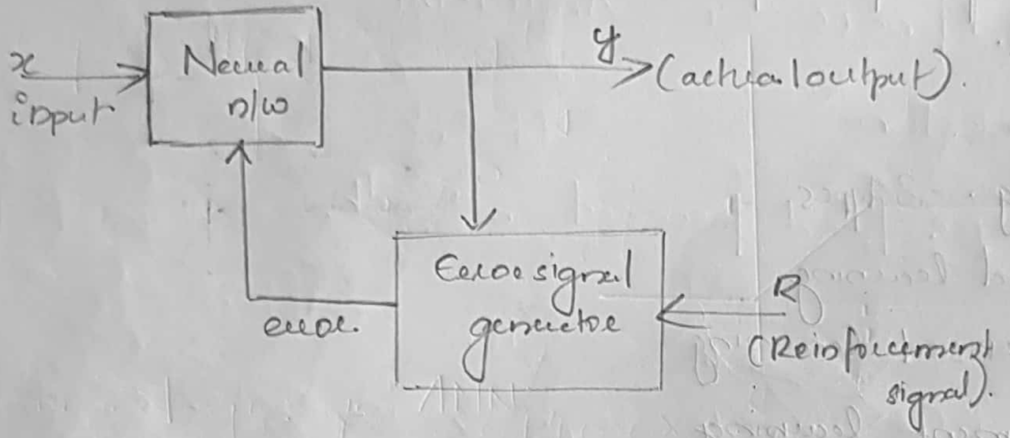


> Un-supervised learning:

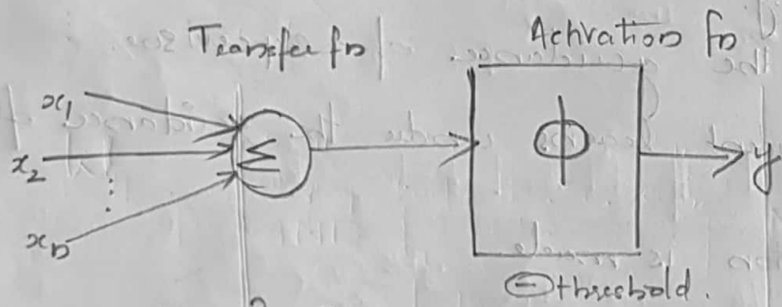


* Similar outputs are grouped together to form clusters.

> Reinforcement learning.



→ Activation Function.

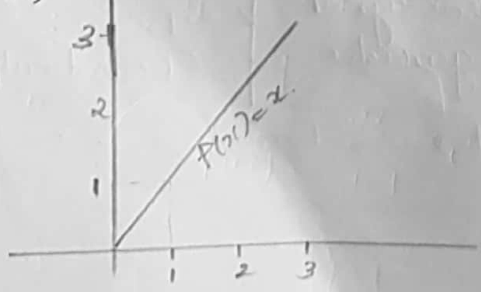


$y_{in} = \sum_{i=1}^{n} x_i w_i$

> Types of Activation Function:

1. Identity function.

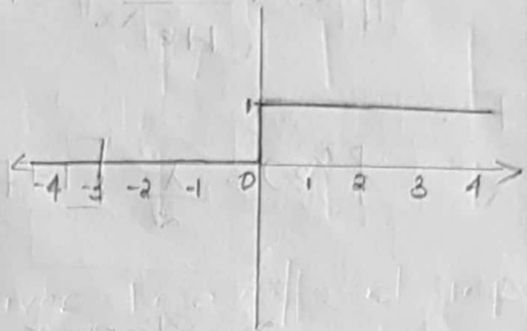
$f(x) = x$



2. Binary step function

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

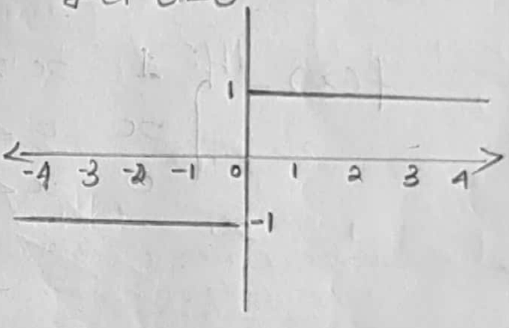
Let $\theta = 0$.



3. Bipolar Step function.

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

Let $\theta = 0$



4. Sigmoidal function.

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

where λ = steepness parameter

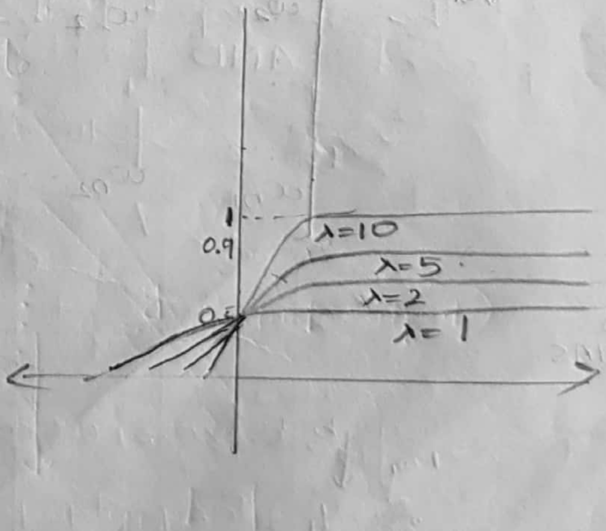
Let $\lambda = 10$.

$$f(0) = \frac{1}{1 + e^{0}} = 0.5$$

$\lambda = 5$

$$f(1) = \frac{1}{1 + e^{-10}} = 0.9$$

$$f(-1) = \frac{1}{1 + e^{10}} =$$



range: 0 to 1.

$$f'(x) = \lambda f(x) \left[1 - f(x)\right]$$

5. Bipolar Sigmoidal function.

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1.$$

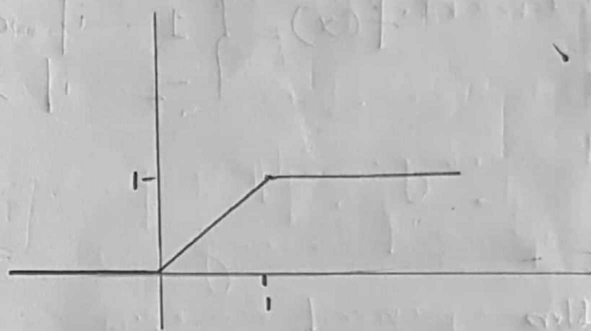
o/p -1 to $+1$

$$f'(x) = \frac{\lambda}{2}\left[1+f(x)\right]\left[1-f(x)\right].$$

λ – Steepness parameter.

6. Ramp function

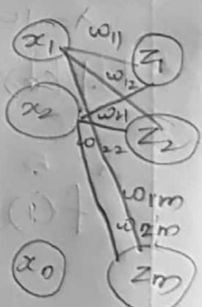
$$f(x) = \begin{cases} 1 & x > 1 \\ x & 0 \le x \le 1 \\ 0 & x < 0. \end{cases}$$



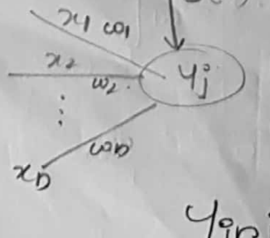
→ Terminologies in ANN:

1. Weight matrix / Connection matrix

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & & & \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix}$$



2. BIAS.



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

3. Threshold.

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

$\theta \rightarrow$ Threshold. (output depends on threshold value)

4. Learning rate. (α)

Control on weight adjustment.
The value of α ranges from 0 to 1.

Notations:

→ x_i - Activation of input unit i / input signal.

→ w_{ij} = Weight in the connection from unit i in one layer to unit j in the next layer.

→ b_j = Bias acting on unit j. [Bias is usually 1]

→ W = Weight matrix.

→ y_{inj} = Net input to unit y_j.

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

→ Norm of x. $\|x\|$

1-norm $\|x\| = |x_1| + |x_2| + \dots + |x_n|$

2-norm $\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2}$

⋮

p-norm $\|x\|_p = \left(|x_1|^p + |x_2|^p + \dots + |x_n|^p \right)^{1/p}$

→ θ_j = Threshold for activation of neuron y_j.

→ S = Training input vector (s_1 s_2 ... s_n)

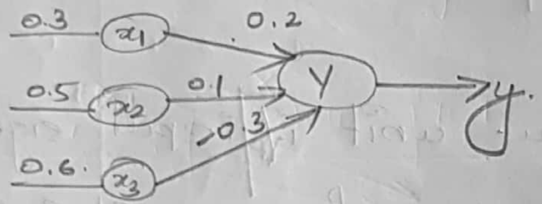
→ T = Training output vector (t_1 t_2 ... t_n).

→ X = Input vector $X = \{x_1\ x_2 \ldots x_n\}$ Can be training vector/testing vector

→ $\Delta w_{ij} = w_{ij}(\text{new}) - w_{ij}(\text{old})$.

→ α; learning rate.

1. Calculate the net input ~~neuron~~

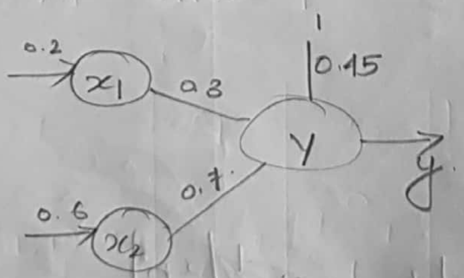


$$y_{in} = 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times -0.3$$
$$= 0.06 + 0.05 - 0.18$$
$$= -0.07$$

$$\begin{array}{r} 0.06 \\ 0.05 \\ \hline 0.11 \end{array} \quad \begin{array}{r} 0.18 \\ 0.11 \\ \hline .0.07 \end{array}$$

17/8/17 Thu

2. Calculate the net input to the neural n/w given to you.



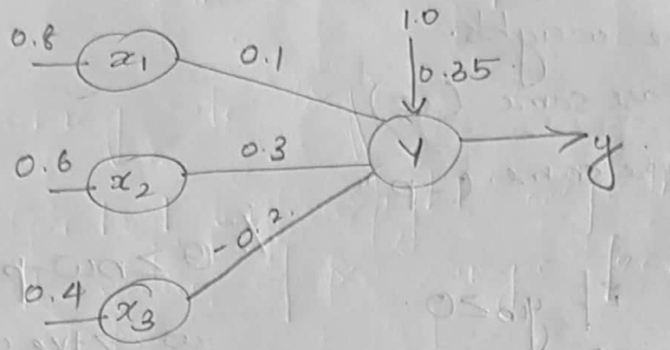
$$y_{in} = 0.2 \times 0.3 + 0.6 \times 0.7 + 0.45 \times 1$$
$$= 0.06 + 0.42 + 0.45$$
$$= 0.93$$

$$\begin{array}{r} 0.42 \\ 0.45 \\ 0.06 \\ \hline 93 \end{array}$$

3. Obtain the output of the neuron Y for the nlw shown in the figure using activation function as

(i) Binary sigmoidal.

(ii) Bipolar sigmoidal.



Ans: (i)

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

Let $\lambda = 1$

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

$$= 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times -0.2 + 1 \times 0.35$$

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

$$= \underline{\underline{0.53}}$$

$$\begin{array}{r} 0.08 \\ 0.18 \\ 0.35 \\ \hline 0.61 - \\ 0.08 \\ \hline 0.53 \end{array}$$

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

$$= 0.629$$

Ans (ii) Bipolar sigmoidal

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1$$

$$= \frac{2}{1+e^{-0.53}} - 1$$

$$\approx \cancel{-0.3705} = \underline{\underline{0.2589}}$$

Scanned by CamScanner

→ McCulloch-pitts Neuron. (M-P Neuron).

→ Binary activation [0,1] Either 0 or 1 will be output

→ weights can be:

* Excitatory +ve weights
* Inhibitory -ve weights.

All positive weights are same (ω)

All negative weights are same. ($-p$).

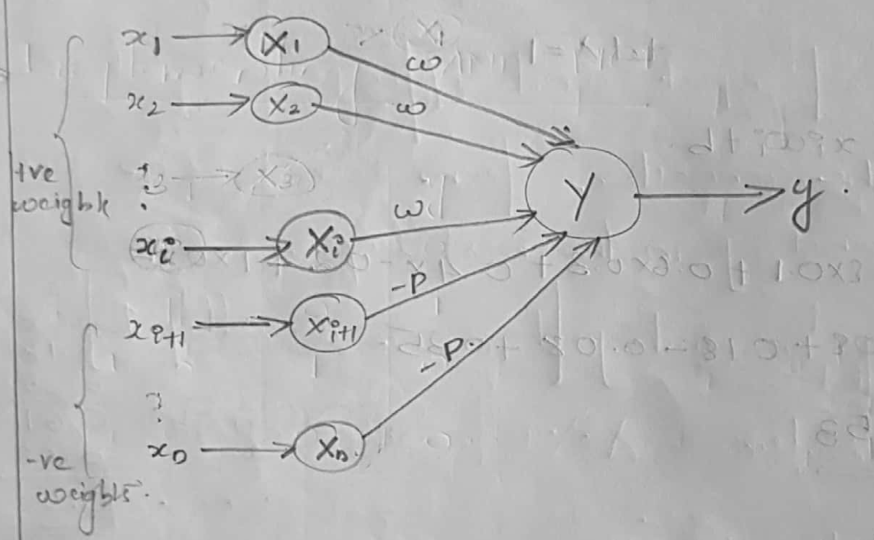
$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

$\theta \geq n\omega - p$.

$\omega \rightarrow$ +ve weight

$p \rightarrow$ -ve weight (only magnitude)

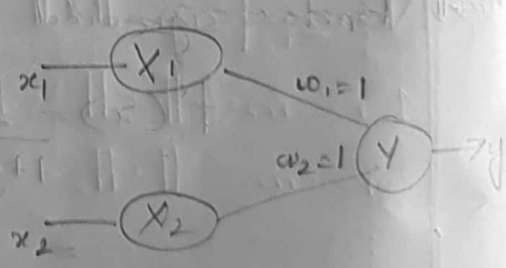
$n \rightarrow$ no of inputs



→ Implement AND function using M-P neuron.

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

2 i/ps.
1 o/p.



$y_{in} = x_1 \omega_1 + x_2 \omega_2$. $\omega_1, \omega_2 = 1$

(0,0)
$y_{in} = 0 \times \omega_1 + 0 \times \omega_2$
$= 0 \times 1 + 0 \times 1$
$= \underline{0}$

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

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

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

Implementing AND function using M-P neuron

$\theta \geq nw - p$.
$\theta \geq 2 \times 1 - 0$
$\underline{\theta \geq 2}$

θ = threshold = 2.

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} \geq 2 \\ 0 & y_{in} < 2. \end{cases}$$

C PROGRAM.

```
f(x)
{ return x
}
Calculateyin(x1, w1, x2, w2).
{
 yin = x1 x w1 + x2 x w2
 return yin
}
findyin (yin)
{ if yin >= 2
        return 1
  else
        return 0 }
```

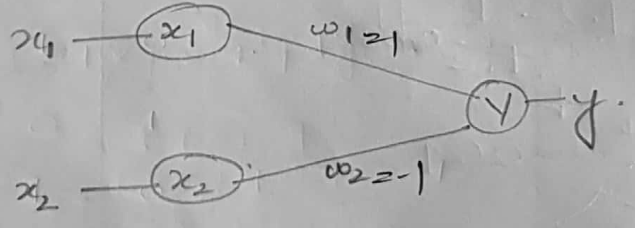
Void main.

```
{ w1=1
  w2=1
  read x1,x2
  x1=f(x1)
  x2=f(x2)
  yin = calculateyin (x1, w1, x2, w2).
  y = findy (yin).
  print y
}.
```

22/08/17 Tue

Implementing AND NOT.

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



$y_{in}(1,1)$ ∗ $w_1=1$ $w_2=1$

$= 1\times1+1\times1$

$=2$

$y_{in}(1,0)$

$=1\times1+0\times1$

$=1+0=1$

$(0,1)$

$=0\times1+1\times1$

$=1$

$(0,0)$
$= 0\times1+0\times1$
$= 0.$

$w_1 = 1 \quad w_2 = -1$

$(1,1)$
$= 1\times1 + 1\times-1$
$= 1-1=0.$

$(1,0)$ $= 1\times1+0\times-1$
$= 1$

$(0,1)$ $= 0\times1 + 1\times-1$
$= -1$

$(0,0)$ $= 0\times1 + 0\times-1$
$= 0.$

$\theta \geq nw - p.$

$\theta \geq 2\times1-1$

$\theta \geq 1$

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

XOR using M-P neuron

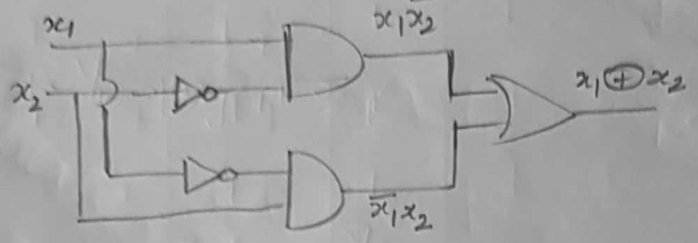
x_1	x_2	y
1	1	0
1	0	1
0	1	1
0	0	0

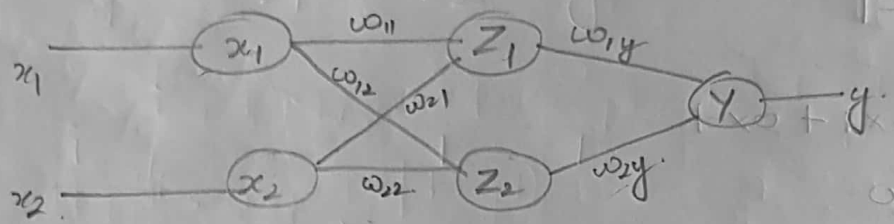
$x_1 \oplus x_2 = x_1\overline{x_2} + \overline{x_1}x_2.$

$Z_1 = x_1\overline{x_2}$

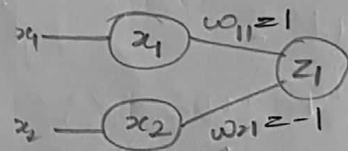
$Z_2 = \overline{x_1}x_2.$

$Z_1 + Z_2$





→ $Z_1 = x_1\overline{x_2}$



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

Assume $\omega_{11} = \omega_{21} = 1.$

Net input.

$z_{1in}(0,0)$

$= 0\times1 + 0\times1$

$= 0$

$z_{1in}(0,1)$

$= 0\times1 + 1\times1$

$= 1$

$(1,0)$

$= 1\times1 + 0\times1 = 1$

Scanned by CamScanner

(1,1)

$= 1 \times 1 + 1 \times 1$

$= 2.$

Assume $\omega_{11} = 1$, $\omega_{21} = -1$

$z_{1in}(0,0)$

$= 0 \times 1 + 0 \times -1$

$= 0$

$z_{1in}(0,1)$

$= 0 \times 1 + 1 \times -1$

$= -1$

$z_{1in}(1,0)$

$= 1 \times 1 + 0 \times -1$

$= 1$

$z_{1in}(1,1)$

$= 1 \times 1 + 1 \times -1$

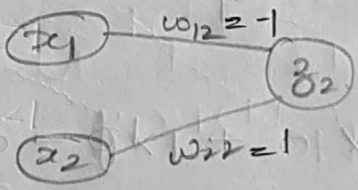
$= 0$

$\theta \geq n\omega - p$

$\geq 2 \times 1 - 1$

$\theta \geq 1$

→ $Z_2 = \bar{x}_1 x_2$



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

Assume ~~(-1,1)~~ . $\omega_{12} = -1$, $\omega_{22} = 1$

$z_{2in}(0,0)$

$= 0 \times -1 + 0 \times 1$

$= 0$

$z_{2in}(0,1)$

$= 0 \times -1 + 1 \times 1$

$= 1$

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

$= -1$

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

$= 0$

$\theta \geq n\omega - p$

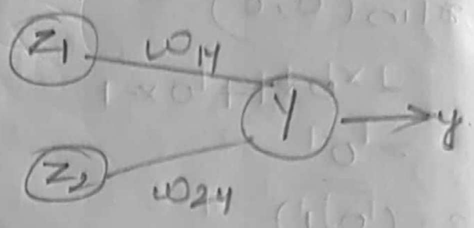
$\geq 2 \times 1 - 1$

$\theta \geq 1$

9/8/17 Tue

$z_1 + z_2$.

net $y_{in} = \omega_{1y} z_1 + \omega_{2y} z_2$



z_1 z_2 Y

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

Calculate net inputs

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

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

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

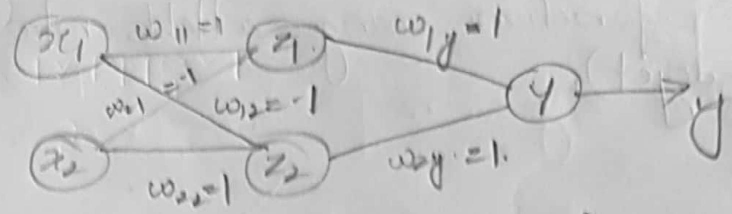
$y_{in\,(0,0)} = 1 \times 0 + 1 \times 0 = 0$.

$\theta \geq n\omega - p$.

$\geq 2 \times 1 - 1$

$\theta \geq 1$

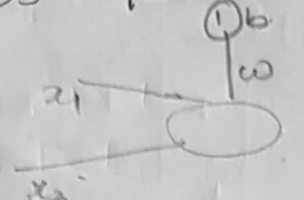
$f(y_{in}) = \begin{cases} 1 & y_{in} \geq 1 \\ 0 & y_{in} < 1 \end{cases}$



M-P Neuron for XOR

→ HEBB's NETWORK:

Hebb's n/w includes bias.



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

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

~~Algorithm:~~

Training Algorithm

Step 0: Initialise base $w_i = 0$ for all $i = 1$ to n, where n is the total number of input neurons.

Step 1: Steps 2 to 4 is performed for each input training vector and target output pair $(s:t)$

Step 2: The input units activations are set using identity function.

$x_i = s_i$ for all $i = 1$ to n.

Step 3: Output units activations are set.

ie $y = t$.

Step 4: Adjust weights

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

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

1) Implement AND function using a Hebb n/w (use bipolar input & target).

AND.

x_1	x_2	b	Target.
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

Training Algo:

$w_1 = w_2 = b = 0$

* Take first i/p.

$[x_1 \;\; x_2 \;\; b] = [1 \;\; 1 \;\; 1]$ $\quad y = 1$

Apply hebb rule.

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

$w_1(new) = w_1(old) + x_1 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

$w_2(new) = w_2(old) + x_2 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

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

$= 0 + 1 = \underline{\underline{1}}$

Take 2^{nd} input

$[x_1 \quad x_2 \quad b] = [1 \quad -1 \quad 1]$

$y = -1$

Apply Hebb rule:

$w_1(new) = w_1(old) + x_1 y$

$= 1 + 1 \times -1 = \underline{\underline{0}}$

$w_2(new) = w_2(old) + x_2 y$

$= 1 + -1 \times -1 = \underline{\underline{2}}$

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

$= 1 + -1 = \underline{\underline{0}}$

Take 3^{rd} input

$[x_1 \quad x_2 \quad b] = [-1 \quad 1 \quad 1]$

$y = -1$

Apply Hebb Rule:

$w_1(new) = w_1(old) + x_1 y$

$= 0 + -1 \times -1 = \underline{\underline{1}}$

$w_2(new) = 2 + 1 \times -1 = \underline{\underline{1}}$

$b(new) = 0 + -1 = \underline{\underline{-1}}$

Take 4^{th} input

$[x_1 \quad x_2 \quad b] = [-1 \quad -1 \quad 1]$

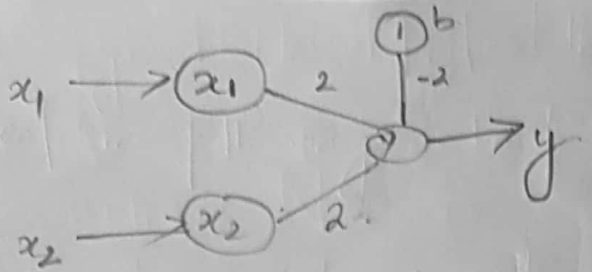
$y = -1$

$w_1(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$w_2(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$b(new) = -1 + -1 = \underline{\underline{-2}}$

Scanned by CamScanner

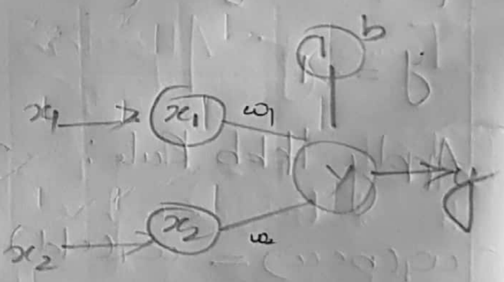


x_1	x_2	y
1	1	2 → +ve excitation
1	-1	-2
-1	1	-2
-1	-1	-6

} inhibitions

2) Design a hebb n/w to implement OR function (Use bipolar input & target).

x_1	x_2	b	Target
1	1	1	1
1	-1	1	1
-1	1	1	1
-1	-1	1	-1



$w_1 = w_2 = b = 0$

* First i/p

$[x_1 \;\; x_2 \;\; b] = [1 \;\; 1 \;\; 1]$ $\quad y = 1$

Apply hebb rule.

$w_1(\text{new}) = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_2(\text{new}) = 0 + 1 \times 1 = \underline{\underline{1}}$

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

* Second input

$[x_1 \ x_2 \ b] = [1 \ -1 \ 1] \quad y = 1$

$w_1(new) = 1 + 1 \times 1 = \underline{\underline{2}}$

$w_2(new) = 1 + -1 \times 1 = \underline{\underline{0}}$

$b(new) = 1 + 1 = \underline{\underline{2}}$

* Third input

$[x_1 \ x_2 \ b] = [-1 \ 1 \ 1] \quad y = 1$

$w_1(new) = 2 + -1 \times 1 = \underline{\underline{1}}$

$w_2(new) = 0 + 1 \times 1 = \underline{\underline{1}}$

$b(new) = 2 + 1 \times 1 = \underline{\underline{3}}$

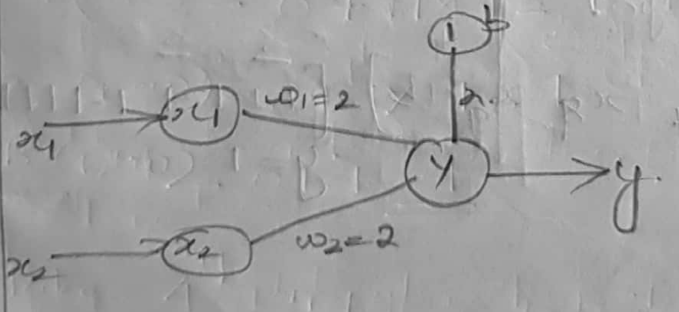
* Fourth i/p

$[x_1 \ x_2 \ b] = [-1 \ -1 \ 1] \quad y = -1$

$w_1(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$w_2(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

$b(new) = 3 - 1 = \underline{\underline{2}}$



x_1	x_2	y
1	1	6
1	-1	2
-1	1	2
-1	-1	-2 → inhibition

+ve excitation

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} \geq 2 \\ -1 & y_{in} < 2 \end{cases}$$

Scanned by CamScanner

11/9/23 Mon.

Q) Using hebb rule find the weights required to perform following classification for the given input pattern.

The '+' symbol represents value 1 and the empty squares represent '-1'. Consider 'I' belongs to the the target class 1 and 'O' belongs to target class -1

x_1	x_2	x_3
+	+	+
x_4	x_5 +	x_6
x_7 +	x_8 +	x_9 +

'I'

1

+	+	+
+		+
+	+	+

'O'

-1

Ans:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	b	y
1	1	1	-1	1	-1	1	1	1	1	1
1	1	1	1	-1	1	1	1	1	1	-1

$w_1 = w_2 = \dots w_9 = b = 0.$

$[x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ b] = [1\ 1\ 1\ -1\ 1\ -1\ 1\ 1\ 1\ 1]$

$y = 1$

$w_1 new = w_1(old) + x_1 y$

$= 0 + 1 \times 1 = \underline{\underline{1}}$

$w_2 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_3 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_4 new = 0 + -1 \times 1 = \underline{\underline{-1}}$

$w_5 new = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_6 new = 0 + -1 \times 1 = \underline{-1}$

$w_7(new) = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_8(new) = 0 + 1 \times 1 = \underline{\underline{1}}$

$w_9(new) = 0 + 1 \times 1 = \underline{\underline{1}}$

$b(new) = 0 + 1 = \underline{\underline{1}}$

x. $[x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8 \; x_9 \; b] = [1 \; 1 \; 1 \; 1 \; -1 \; 1 \; 1 \; 1 \; 1 \; 1] \; t_j = -1$

$w_1(new) = 1 + 1 \times -1$
$= \underline{\underline{0}}$

$w_2(new) = 1 + 1 \times -1 = \underline{\underline{0}}$

$w_3(new) = 1 + 1 \times -1 = \underline{\underline{0}}$

$w_4(new) = -1 + 1 \times -1 = \underline{\underline{-2}}$

$w_5(new) = 1 + -1 \times -1 = \underline{\underline{2}}$

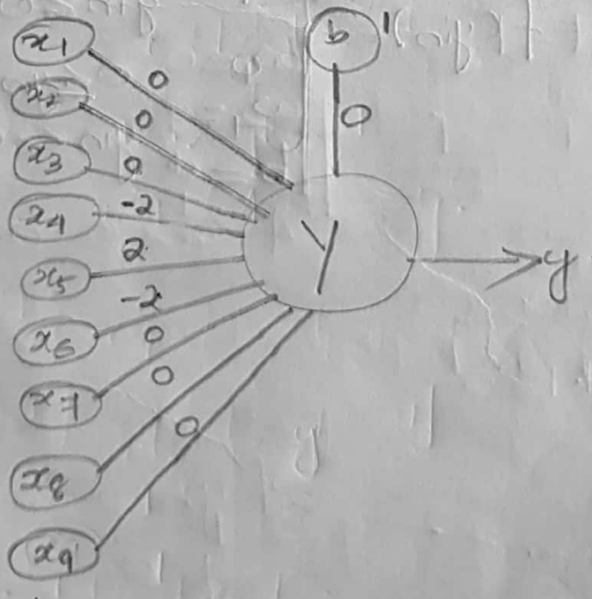
$w_6(new) = -1 + 1 \times -1 = \underline{\underline{-2}}$

$w_7(new) = 1 + 1 \times -1 = \underline{\underline{0}}$

$w_8(new) = 1 + 1 \times -1 = \underline{\underline{0}}$

$w_9(new) = 1 + 1 \times -1 = \underline{\underline{0}}$

$b(new) = 1 + -1 = \underline{\underline{0}}$



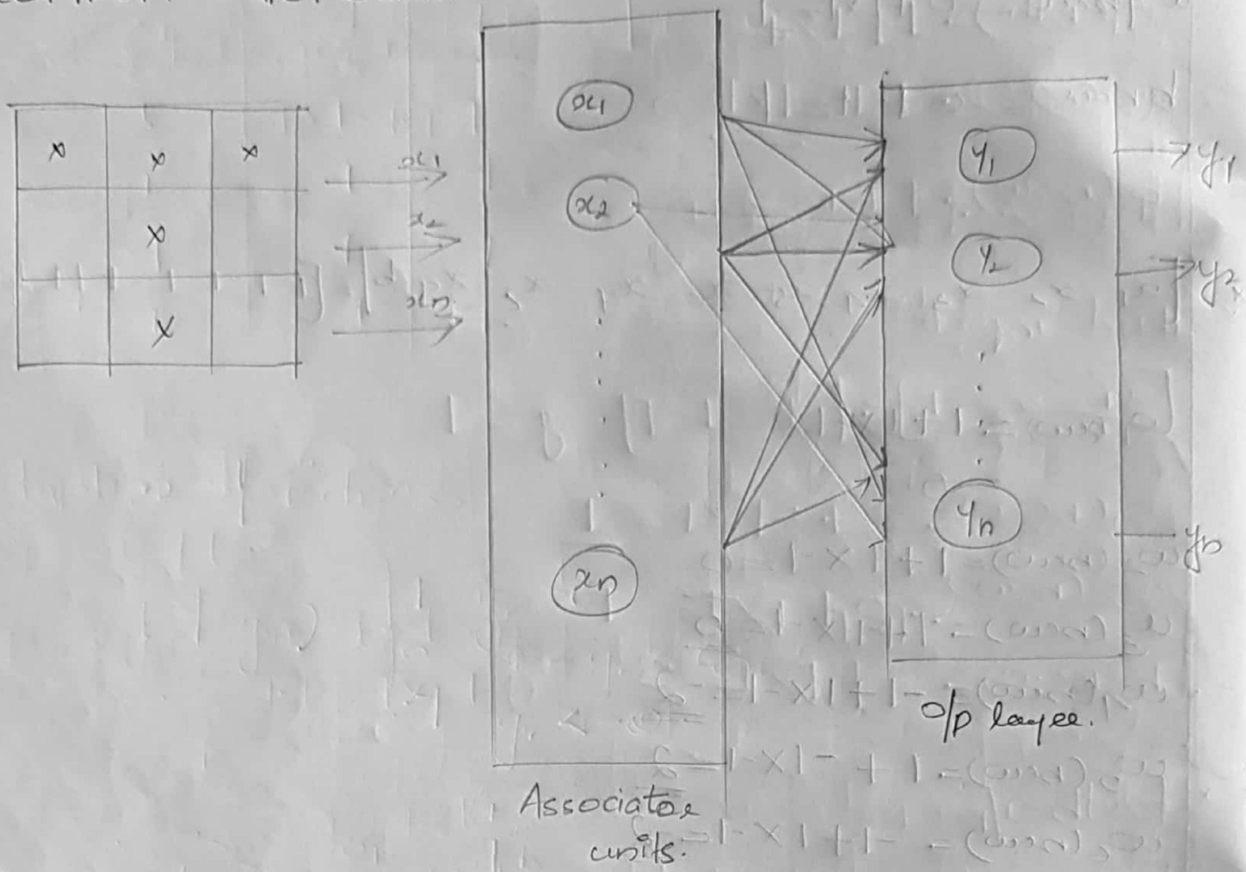
First input $y_{in} = 6 \; (0+0+0+2+2+2+0+0+0+0)$ Excitation

Second input $y_{in} = -6 \; (0+0+0-2-2-2+0+0+0+0)$ Inhibition

7/9/17 Thu

MODULE 2

PERCEPTRON NETWORK.



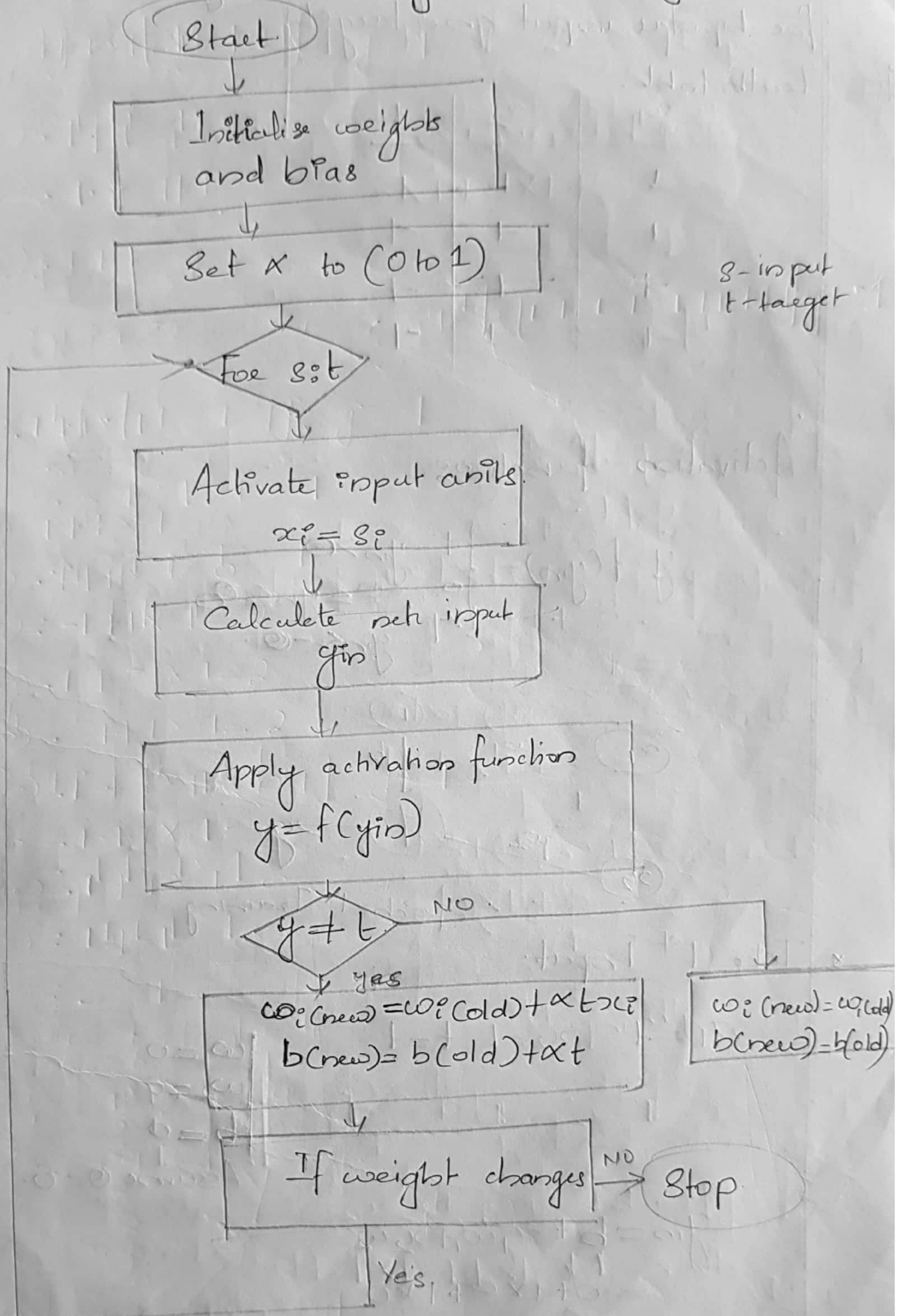
If $y \neq t$

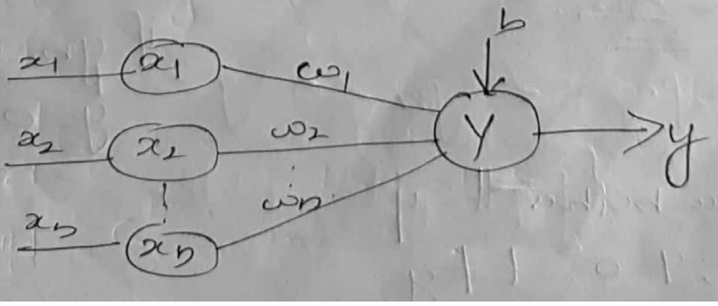
$w_i(new) = w_i(old) + \alpha\, t x_i$

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

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

Flow Chart: Perceptron Training Method for single output.





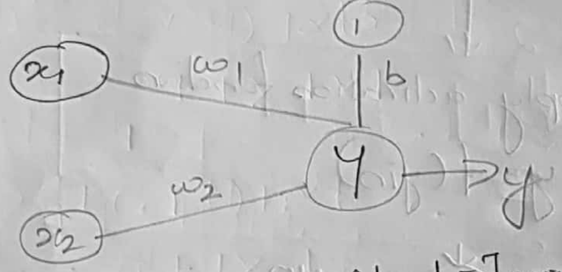
1. Implement AND function using perceptron networks for bipolar input and target

Ans. Truth table

x_1	x_2	t
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Activation function:

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > 0 \\ 0 & y_{in} = 0 \\ -1 & y_{in} < 0 \end{cases}$$



epoch - 1 - Training

* Take 1st Input

$x_1 = 1 \quad x_2 = 1 \quad t = 1$

$w_1 = 0$
$w_2 = 0$
$b = 0$
Assume $\theta = 0$.

$$y_{in} = b + x_1 w_1 + x_2 w_2$$
$$= 0 + 1 \times 0 + 1 \times 0$$
$$= \underline{\underline{0}}$$

if $y_{in} = 0$.
$y = 0$.

Check whether $t = y$

$t = 1 ; y = 0 \quad t \neq y$.

$w_1(new) = w_1(old) + \alpha t x_1$ | Assume $\alpha = 1$

$= 0 + 1 \times 1 \times 1$

$= \underline{\underline{1}}$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 0 + 1 \times 1 \times 1$

$= \underline{\underline{1}}$

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

$= 0 + 1 \times 1$

$= \underline{\underline{1}}$.

* Take 2nd input

$x_1 = 1 \quad x_2 = -1 \quad t = -1$

| $w_1 = 1$
| $w_2 = 1$
| $b = 1$

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

$= 1 + 1 \times 1 + -1 \times 1$

$= \underline{\underline{1}}$

if $y_{in} = 1$

$y = 1 = f(y_{in})$.

$y = 1 ; t = -1 \quad y \neq t$.

$w_1(new) = w_1(old) + \alpha t x_1$ | Assume $\alpha = 1$.

$= 1 + 1 \times -1 \times 1$

$= \underline{\underline{0}}$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 1 + 1 \times -1 \times -1$

$= \underline{\underline{2}}$.

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

$= 1 + 1 \times -1$

$= 0$//

* Take 3^{rd} input

$x_1 = -1 \quad x_2 = 1 \quad t = -1$

$w_1 = 0$
$w_2 = 2$
$b = 0$

$y_{in} = b + x_1 w_1 + x_2 w_2$
$= 0 + -1 \times 0 + 1 \times 2$
$= 2$

If $y_{in} = 2$.

$y_{in} > 0$.
$y = 1$
$y \neq t$.

$w_1(new) = w_1(old) + \alpha t x_1$
$= 0 + 1 \times -1 \times -1$
$= 1$

$w_2(new) = w_2(old) + \alpha t x_2$
$= 2 + 1 \times -1 \times 1$
$= 1$

$b(new) = b(old) + \alpha t$
$= 0 + 1 \times -1$
$= -1$

* Take 4^{th} input

$x_1 = -1 \quad x_2 = -1 \quad t = -1$

$w_1 = 1$
$w_2 = 1$
$b = -1$

$y_{in} = b + x_1 w_1 + x_2 w_2$
$= -1 + -1 \times 1 + -1 \times 1$
$= -3$

if $y_{in} = -3 \quad y_{in} < 0 \quad y = -1$

$y = t$. No updation

$w_1 = 1$

$w_2 = 1$

$b = -1$

16/9/17 Sat.

epoch-2 - Testing

$w_1 = 1 \quad w_2 = 1 \quad b = -1$

- Take 1st input.

$x_1 = 1 \quad x_2 = 1 \quad t = 1.$

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

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

$= -1 + 1 \times 1 + 1 \times 1$

$= -1 + 2 = \underline{1}.$

$\therefore y = 1$

$t = y.$

- Take 2nd input

$x_1 = 1 \quad x_2 = -1 \quad t = -1$

$y_{in} = -1 + 1 \times 1 + -1 \times 1$

$= \underline{-1}$

$\therefore y = -1$

$y = t = -1.$

- Take 3rd input

$x_1 = -1 \quad x_2 = 1 \quad t = -1$

$y_{in} = -1 + -1 \times 1 + 1 \times 1$

$= \underline{-1}$

$\therefore y_{in} = -1$

$y = t = -1$

* Take 4^{th} input

$x_1 = -1 \quad x_2 = -1 \quad t = -1$

$y_{in} = -1 + -1 \times 1 + -1 \times 1$

$= -3$

$y = -1$

$y = t = -1$

→ Perceptron Training Algorithm for Single Output

Step 0: Initialise $w_i = b = 0$, $\alpha = 1$

Step 1: Perform steps 2 to 6 until final stopping condition is false

Step 2: Perform steps 3 to 5 for each training pair $s:t$

Step 3: Apply identity activation function in the input pair

ie $x_i = s_i$

Step 4: Calculate net input $y_{in} = b + \sum_{i=1}^{n} x_i w_i$ where n is the number of input neurons in the input pair

Calculate $y = f(y_{in}) = \begin{cases} 1 & y_{in} > \theta \\ 0 & -\theta \le y_{in} \le \theta \\ -1 & y_{in} < -\theta \end{cases}$

Step 5: Weight & bias adjustments. Compare actual output and decide target output.

if $y \neq t$

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

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

else

$w_i(new) = w_i(old)$

$b(new) = b(old)$

Step6: Train the network until there is no weight change. This is the stopping condition for the network. This is not met, then start again from step(2).

→ Perceptron Testing Algorithms for Single Output

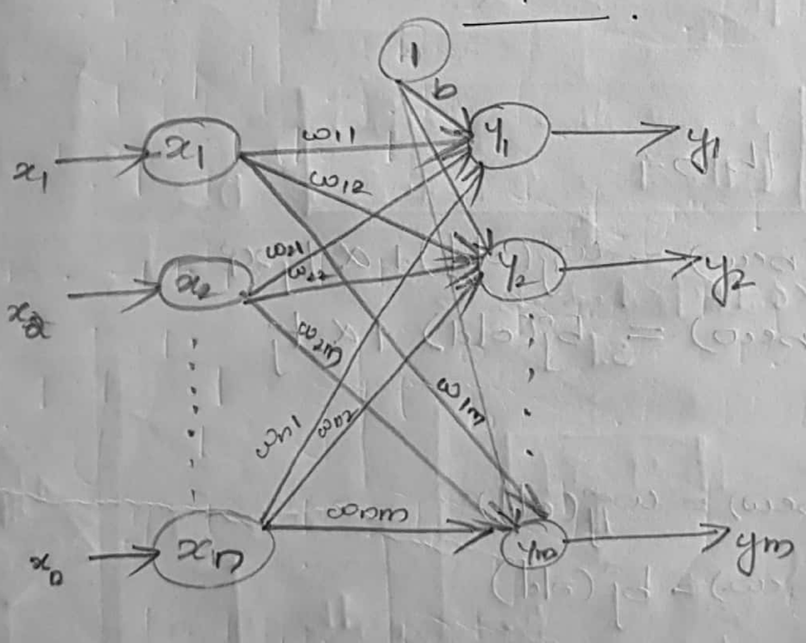
Step0: Assign weights that are obtained during training

Step1: For each input vector ~~x~~, to be classified perform steps 2 to 3

Step2: Apply input

Step3: Calculate y_{in} and y.

→ Perceptron network for multiple output.



Step0: $w_i = b = 0$. $\alpha = 1$

Step1: Perform steps 2 to 6 until final stopping condition is false.

Step2: Perform steps 3 to 5 for each train

Step3:

St

Step4:

~~$y = f(y_{inj})$~~

for $j = 1$ to m

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

$$y = f(y_{inj}) = \begin{cases} 1 & y_{inj} > \theta \\ 0 & -\theta \le y_{inj} \le \theta \\ -1 & y_{inj} < -\theta \end{cases}$$

Step5: for $j = 1$ to m

if $t_j \neq y_j$ then

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

else

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

1. Implement OR using binary input & bipolar targets.

Scanned by CamScanner

Binary inputs of Bi

x_1	x_2	t
1	1	1
1	0	1
0	1	1
0	0	-1

$$y = f(y_{in}) = \begin{cases} 1 & y_{in} > 0.2 \\ 0 & \cancel{y_{in} = 0.2} \quad -0.2 \le y_{in} \le 0.2 \\ -1 & y_{in} < 0.2 \end{cases}$$

$\alpha = 0.2$

epoch - 1

1st input:

$w_1 = w_2 = b = 0$. $\alpha = 1$.

$x_1 = 1 \quad x_2 = 1 \quad t = 1$

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

$= 0 + 1 \times 0 + 1 \times 0$.

$= \underline{0}$

$\therefore y = -1$

$y \neq t$.

$w_1(new) = w_1(old) + \alpha t x_1$

$= 0 + 0.2 \times 1 \times 1$

$= \underline{0.2}$

$w_2(new) = 0 + 0.2 \times 1 \times 1 = \underline{0.2}$

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

$= 0 + 0.2 \times 1 = \underline{0.2}$

* 2nd input

$x_1 = 1 \quad x_2 = 0 \quad t = 1 \qquad w_1 = w_2 = b = 0.2$

$y_{in} = 1 + 1 \times 0.2 + 0 \times 0.2$

$= \underline{\underline{1.2}}$

$\therefore y = 1 \quad t = 1$

$y = t.$

* 3rd input.

$x_1 = 0 \quad x_2 = 1 \quad t = 1 \qquad w_1 = w_2 = b = 0.2$

$y_{in} = 1 + 0 \times 0.2 + 1 \times 0.2$

$= \underline{\underline{1.2}}$

$y = 1 \quad t = 1$

$y = t.$

* 4th input.

$x_1 = 0 \quad x_2 = 0 \quad t = -1 \qquad w_1 = w_2 = b = 0.2$

$y_{in} = 1 + 0 \times 0.2 + 0 \times 0.2$

$= \underline{\underline{1}}$

$y = 1$

$y \neq t$

$w_1(new) = 0.2 + 0.2 \times -1 \times 0$

$= \underline{\underline{0.2}}$

$w_2(new) = 0.2 + 0.2 \times -1 \times 0$

$= \underline{\underline{0.2}}$

$b(new) = 1 + 0.2 \times -1$

$= \underline{\underline{0.8}}$

~~epoch 2~~

~~1st input~~

26/9/17 Tue.

Adaptive Neural Network:

Weights & bias are initialised random values but not zero.

$$w_{i(new)} = w_{i(old)} + \alpha(t - y_{in}) x_i$$

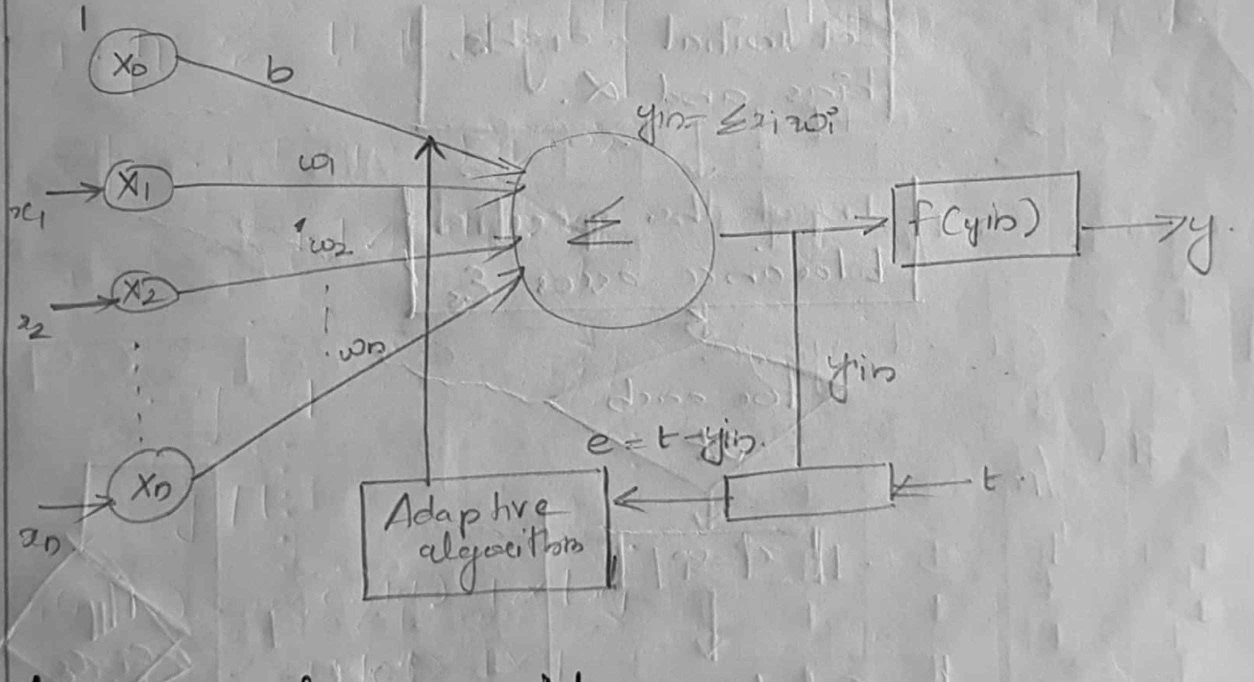
$$b(new) = b(old) + \alpha(t - y_{in})$$

$$y_{in} = \sum x_i w_i + b.$$

Single output network

→ Single neuron with many inputs.

→ Bipolar input/output.



→ Adaptive linear Neuron

→ Relationship b/w i/p and o/p is linear.

→ Adaptive uses bipolar activation.

→ The network is trained using delta rule also called widrow - Hoff rule (least mean square error)

~~→ Network is trained using delta rule.~~

→ Delta rule for adjusting the weight is-

$$\Delta w_i = \alpha (t - y_{in}) x_i$$

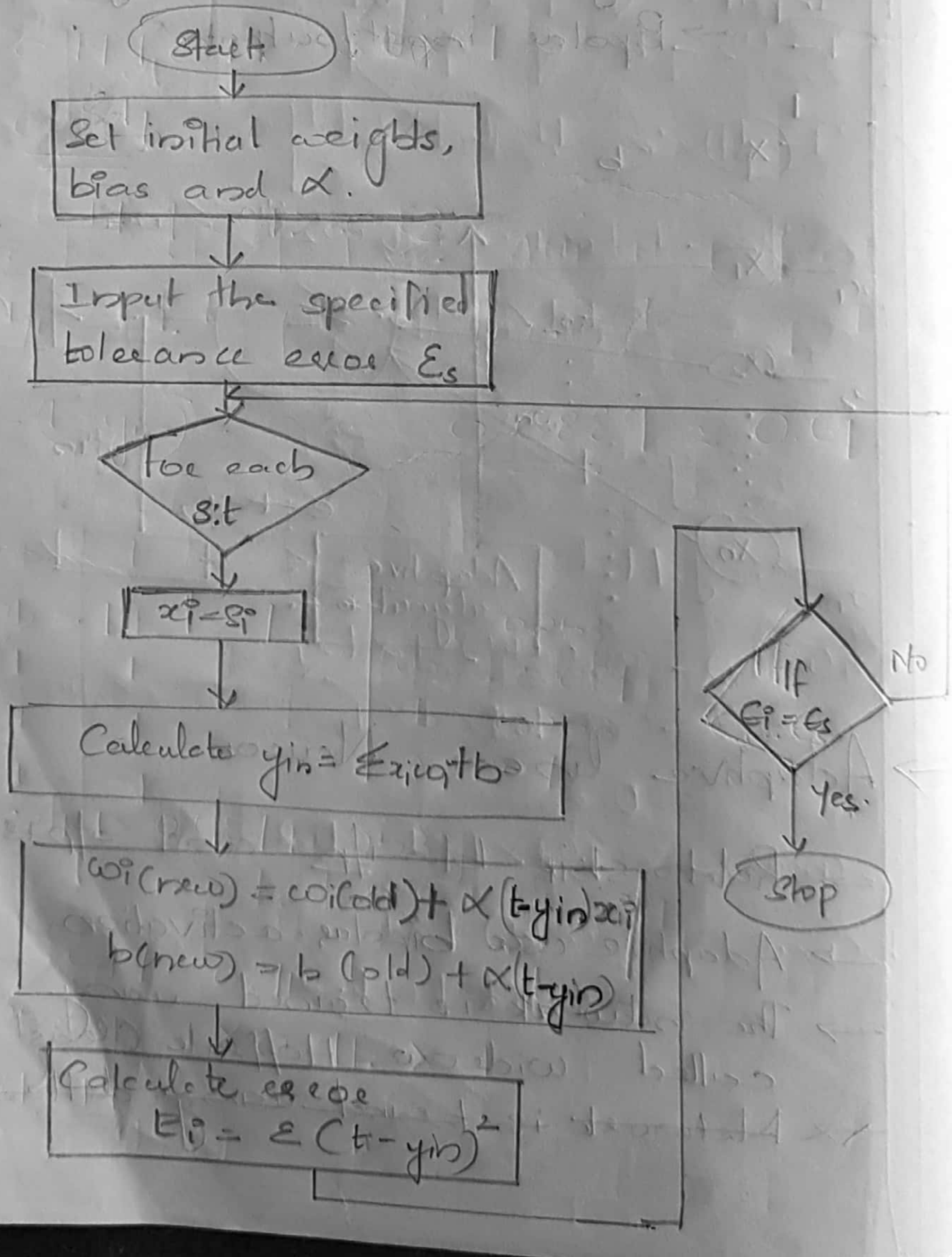
α = learning rate [0.1 to 1.0]

x_i = input

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

Inputs & outputs are either +1 or −1.

Algorithm



1. Implement OR function with bipolar inputs and targets using adaptive linear network.

Ans:

x_1	x_2	b	t
1	1	1	1
1	-1	1	1
-1	1	1	1
-1	-1	1	-1

Assume initial weights be 0.1, $b=0.1$

$\alpha = 0.1$

Epoch1

* Take 1^{st} i/p.

$x_1=1 \quad x_2=1 \; ; \; t=1$

$y_{in} = 1\times0.1 + 1\times0.1 + 1\times0.1$

$= \underline{0.3}$

$t - y_{in} = 1 - 0.3 = \underline{0.7}$

$w_1(new) = w_1(old) + \alpha(t - y_{in})x_1$

$= 0.1 + 0.1\times0.7\times1$

$= 0.1 + 0.07 = \underline{0.17}$

$w_2(new) = w_2(old) + \alpha(t - y_{in})x_2$

$= 0.1 + 0.1\times0.7\times1$

$= \underline{0.17}$

$b(new) = b(old) + \alpha(t - y_{in})$

$= 0.1 + 0.1\times0.7$

$= \underline{0.17}$

Calculate error $E_i = (t-y_{in})^2 = 0.7^2 = 0.49$

* Take 2^{nd} i/p.

$x_1 = 1 \quad x_2 = -1 ; \; t = 1$

$w_1 = 0.17$
$w_2 = 0.17$
$b = 0.17$
$\alpha = 0.1$

$y_{in} = 1 \times 0.17 + -1 \times 0.17 + 0.17$

$= 0.17$

$(t-y_{in}) = 1 - 0.17 = 0.83$

$w_{1(new)} = 0.17 + 0.1 \times 0.83 \times 1$

$= 0.253$

$w_{2(new)} = 0.17 + 0.1 \times 0.83 \times -1$

$= 0.087$

$b(new) = 0.17 + 0.1 \times 0.83$

$= 0.253$

Error $E_i = (t-y_{in})^2 = 0.83^2 = 0.6889 \approx 0.69$

* Take 3rd i/p

$x_1 = -1 \quad x_2 = 1 ; \; t = 1$

$w_1 = 0.253$
$w_2 = 0.087$
$b = 0.253$
$\alpha = 0.1$

$y_{in} = -1 \times 0.253 + 1 \times 0.087 + 0.253$

$= 0.087$

$(t-y_{in}) = 1 - 0.087 = 0.913$

$w_1(new) = 0.253 + 0.1 \times 0.913 \times -1$

$= 0.1617$

$\omega_2(new) = 0.087 + 0.1 \times 0.913 \times 1$

$= \underline{0.1783}$

$b(new) = 0.253 + 0.1 \times 0.913 = 0.\underline{3443}$

Error $E_i = 0.913^2 = 0.8335 \approx \underline{0.83}$

* Take 4^{th} i/p.

$x_1 = -1 \quad x_2 = -1 \quad t = -1$

$\omega_1 = 0.1617$

$\omega_2 = 0.1783$

$b = 0.3443$

$y_{in} = -1 \times 0.1617 + -1 \times 0.1783 + 0.3443$

$= \underline{0.0043}$

$t - y_{in} = -1 - 0.0043$

$= \underline{-1.0043}$

$\omega_1(new) = 0.1617 + 0.1 \times -1.0043 \times -1$

$= 0.26213$

$\omega_2(new) = 0.1783 + 0.1 \times -1.0043 \times -1$

$= \underline{0.27873}$

$b(new) = 0.3443 + 0.1 \times -1.0043$

$= \underline{0.24387}$

Error $E_i = -1.0043^2 = 1.0086 \approx 1.01$

Total mean square error:

$$\begin{array}{r} 0.49 \\ 0.69 \\ 0.83 \\ 1.01 \\ \hline 3.02 \end{array}$$

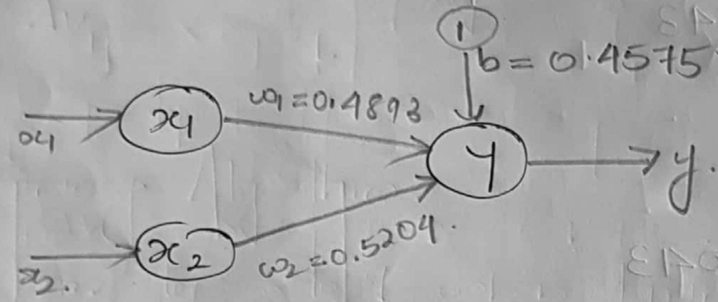
Error is high

Epoch	Total mean square
epoch 1	3.02
epoch 2	1.938
epoch 3	1.5506
epoch 4	1.417
epoch 5	1.377.

↓ reduced

Continue the Epochs.

Final n/w



h.w 2. Use adaline network to learn AND NOT function using bipolar i/p and targets. Perform two epochs of training.

AND NOT

x_1	x_2	b	t
1	1	1	-1
1	-1	1	1
-1	1	1	-1
-1	-1	1	-1

Assume initial weights be $w_1 = w_2 = 0.1$, $b = 0.1$

$\alpha = 0.1$

Epoch 1.

* Take 1st input

$x_1 = 1 \quad x_2 = 1 \quad t = -1$

$y_{in} = 1 \times 0.1 + 1 \times 0.1 + 1 \times 0.1$

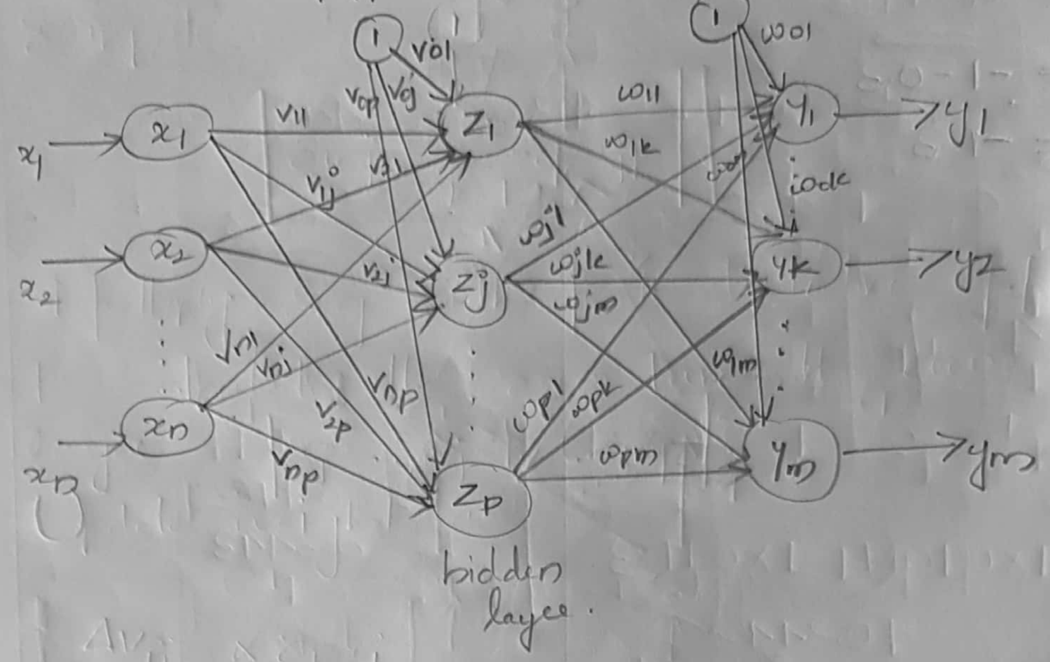
$= 0.3.$

$t - y_{in} = -1 - 0.3$

$= -1.3$

3/10/2017 Tue

BACK PROPAGATION NETWORK:



Training Algorithm.

Step 1: Perform steps 2 to 9 when stopping condition is false

Step 2: Perform steps 3 to 8 for each training pair.

Phase 1:

Feed forward.

Step 3: Each input unit receives input signal x_i and it is send to the hidden units.

Step 4: Each hidden unit z_j (~~where~~ for $j = 1$ to p) the net input is calculated as.

$$z_{inj} = v_{0j} + \sum_{j=1}^{n} x_i v_{ij}$$

Calculate the output of the hidden unit by applying binary sigmoidal function or bipolar sigmoidal function

$$z_j = f(z_{inj})$$

Scanned by CamScanner

Step5: For each output unit y_k $(k=1 \text{ to } m)$ calculate the net input.

$$y_{ink} = w_{ok} + \sum_{j=1}^{P} z_j w_{jk}$$

Apply activation function.

$$y_k = f(y_{ink})$$

Phase 2: Back propagation of error:

Step6: Each output unit y_k $(k=1 \text{ to } m)$ receives the target output t_1 to t_m. Calculate the error as

$$\boxed{\delta_k = (t_k - y_k)\, f'(y_{ink})}$$

where:

$f'(y_{ink})$ is the derivative of $f(y_{ink})$

t_k = target

y_k = output obtained.

For binary sigmoidal function $f(x)$

$$\boxed{f'(x) = \lambda f(x)\,[1 - f(x)]}$$ λ = steepness parameter.

Update weight as

$$\boxed{\Delta w_{jk} = \alpha\, \delta_k\, z_j}$$

where α = learning rate.

Update bias as

$$\boxed{\Delta w_{ok} = \alpha\, \delta_k}$$

Step 7: For each hidden unit z_j ($j = 1$ to p) sums its δ inputs.

$$\boxed{\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}}$$

Multiply δ_{inj} with first derivative of $f(z_{inj})$ to calculate the error term. The error in hidden layer $\boxed{\delta_j = \delta_{inj} f'(z_{inj})}$

Update weight as

$$\boxed{\Delta v_{ij} = \alpha \delta_j x_i}$$

Update bias as

$$\boxed{\Delta v_{oj} = \alpha \delta_j}$$

Phase 3: Weight & Bias Updation

Step 8: For each output unit y_k ($k = 1$ to m) update bias & weight as.

$$\boxed{w_{jk} = w_{jk}(old) + \Delta w_{jk}}$$

For each hidden unit z_j (for $j = 1$ to p) update weight and bias as

$$\boxed{\begin{aligned} v_{ij}(new) &= v_{ij}(old) + \Delta v_{ij} \\ v_{oj}(new) &= v_{oj}(old) + \Delta v_{oj} \end{aligned}}$$

Step 9: Check for stopping condition it may be certain number of epochs reached or actual output equals target output.

Testing Algorithm:

Step0: Initialise weights. The weights are taken from training algorithm.

Step1: Perform steps 2 to 4 for each input vector

Step2: Set the activation of input units for x_i ($i = 1$ to n)

Step3: Calculate net input for each hidden unit

$$\boxed{\begin{aligned} z_{inj} &= v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \\ z_j &= f(z_{inj}) \end{aligned}}$$

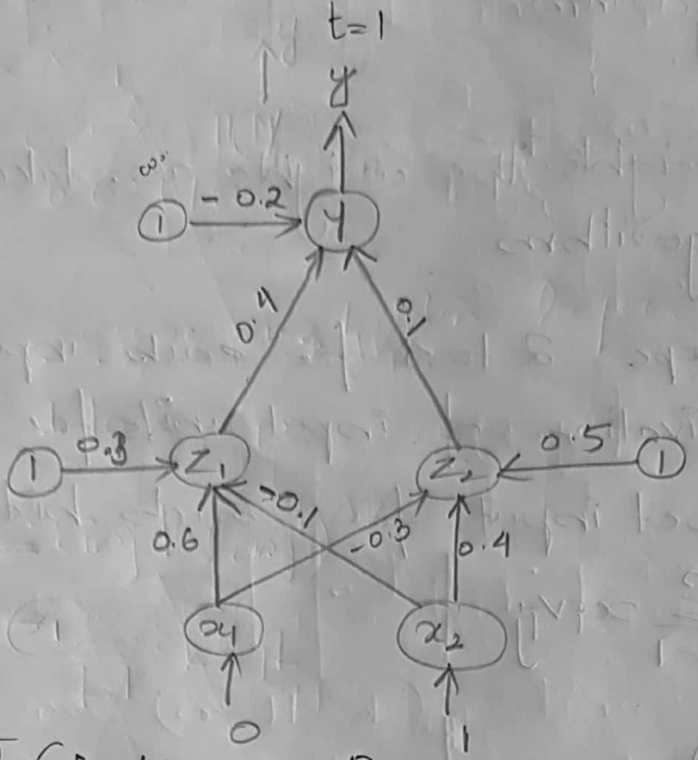
Step4: Compute the output of the units in the output layer [for $k = 1$ to m]

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_{in} = f(y_{ink}).$$

Use sigmoidal function for calculating the output.

1) Using back propagation network find the new weights & bias. It is presented with the input [0, 1] and target o/p is 1. Use $\alpha = 0.25$ and binary sigmoidal activation function



Ans: Phase I (feed forward) $\begin{bmatrix} x_1 & x_2 \\ 0 & 1 \end{bmatrix}$ $\begin{bmatrix} t \\ 1 \end{bmatrix}$

$$z_{in1} = 0\times 0.6 + 1\times -0.1 + 0.3$$

$$= 0 - 0.1 + 0.3$$

$$= \underline{0.2}$$

$$z_{in2} = 0\times -0.3 + 1\times 0.4 + 1\times 0.5$$

$$= 0 + 0.4 + 0.5 = \underline{0.9}$$

Binary Sigmoidal fn:

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

$$= \frac{1}{1+e^{-x}}$$

$$z_1 = f(z_{in1}) = \frac{1}{1+e^{-0.2}} = \underline{0.5498}$$

$$z_2 = f(z_{in2}) = \frac{1}{1+e^{-0.9}} = \underline{0.7109}$$

$$y_{in} = 0.5498 \times 0.4 + 0.7109 \times 0.1 + \overset{w_0}{1 \times -0.2}$$

$$= \underline{0.09101}$$

$$y = f(y_{in}) = \frac{1}{1+e^{-0.09101}}$$

$$= \underline{0.5227}$$

$$y \neq t$$

5/10/2017 Thu.

Phase $\underline{\text{II}}$ (Back propagation of error)

$$\delta_k = (t_k - y_k)\, f'(y_{ink})$$

$$\delta_1 = (t - y)\, f'(y_{in})$$

$$f'(y_{in}) = f(y_{in})\,[1 - f(y_{in})]$$

$$= 0.5227\,[1 - 0.5227]$$

$$= \underline{0.2495}$$

error $\delta_1 = (1 - 0.5227)\, 0.2495$

$$= \underline{0.1191}$$

Update weight:

$$\Delta w_1 = \alpha\, \delta_1\, z_1$$

$$= 0.25 \times 0.1191 \times 0.5498$$

$$= 0.01637 \approx \underline{0.01634}$$

$$\Delta w_2 = \alpha\, \delta_1\, z_2$$

$$= 0.25 \times 0.1191 \times 0.7109$$

$$= \underline{0.02117}$$

Update bias

$\Delta w_0 = \alpha \delta_1$

$= 0.25 \times 0.1191$

$= 0.02978$

Calculate error in hidden layer.

$j = 1$ to 2.

$\delta_j = \delta_{inj} f'(z_{inj})$

$\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}$

$\delta_{inj} = \delta_1 w_{j1}$ (only one o/p neuron).

$\delta_{in1} = \delta_1 w_{11} = 0.1191 \times 0.4 = 0.047$

$\delta_{in2} = \delta_2 w_{21} = 0.1191 \times 0.1 = 0.01191$

Error:

$\delta_1 = \delta_{in1} f'(z_{in1})$

$f'(z_{in1}) = f(z_{in1}) [1 - f(z_{in1})]$

$= 0.5498 [1 - 0.5498]$

$= 0.2475$ ~~0.450~~

$\delta_1 = \delta_{in1} \times f'(z_{in1})$

$= 0.047 \times 0.2475$

$= 0.01163$

Error $\delta_2 = \delta in_2 \, f'(z in_2)$

$f'(z in_2) = f(z in_2)\,[1 - f(z in_2)]$

$= 0.7109\,[1 - 0.7109]$

$= 0.2055$

$\delta_2 = \delta in_2 \times f'(z in_2)$

$= 0.01191 \times 0.2055$

$= 0.00245$

find changes.

$\Delta V_{21} = \alpha\, \delta_1\, x_2 = 0.25 \times 0.01163 \times 1$

$= 0.0029075$

$\Delta V_{12} = \alpha\, \delta_2\, x_1 = 0.21 \times 0.00245 \times 0$

$= 0$

$\Delta V_{01} = \alpha\, \delta_1 = 0.25 \times 0.01163$

$= 0.0029075$

$\Delta V_{22} = \alpha\, \delta_2\, x_2 = 0.25 \times 0.00245 \times 1$

$= 0.0006125$

$\Delta V_{02} = \alpha\, \delta_2 = 0.25 \times 0.00245$

$= 0.0006125$

Compute final weight

$V_{11}(new) = V_{11}(old) + \Delta V_{11}$

$= 0.6 + 0 = \underline{0.6}$

$V_{12}(new) = V_{12}(old) + \Delta V_{12}$

$= -0.3 + 0 = \underline{-0.3}$

$V_{21}(new) = V_{21}(old) + \Delta V_{21}$

$= -0.1 + 0.00295$

$= -0.09705$

$V_{22}(new) = V_{22}(old) + \Delta V_{22}$

$= 0.4 + 0.0006125$

$= \underline{0.4006125}$

$w_1(new) = w_1(old) + \Delta w_1$

$= 0.4 + 0.0164$

$= \underline{0.4164}$

$w_2(new) = w_2(old) + \Delta w_2 = \underline{0.12117}$

$V_{01}(new) = 0.30295 \quad (V_{01}(old) + \Delta V_{01}) \; 0.3 + 0.00295$

$V_{02}(new) = 0.5066125 \quad (V_{02}(old) + \Delta V_{02}) \; 0.5 + 0.0006125$

$w_0(new) = -0.17022$

7/10/2017 Sat

MODULE 3.

Fuzzy logic

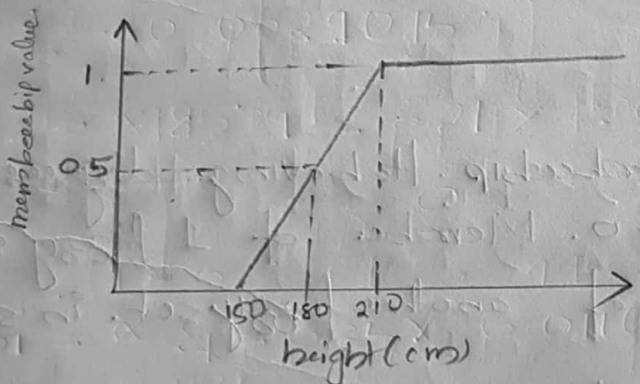
- multivalued logic
- many values b/w 0 & 1 [0,1]
- e.g. length : short, tall, tall, medium, .. tall.
- fuzzy descriptors / liguistic variables / linguistic descriptors.
- membership value of x_1 in fuzzy set A.

$$= \mu_A(x_1).$$

eg: μ_{short} (John) = 0.7

μ_{short} (John) = 0.
↳ membership value of john in fuzzy set

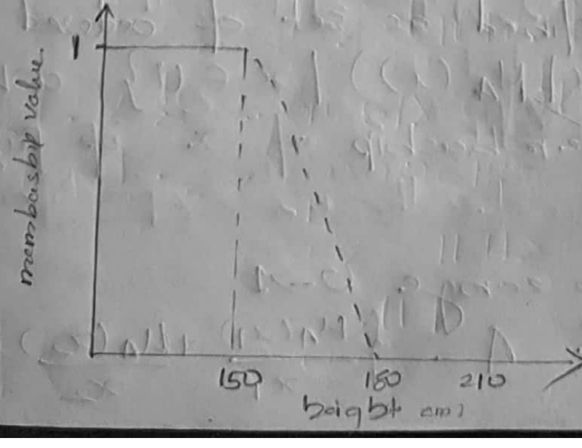
→ **Tall**. ~~short = 0~~.



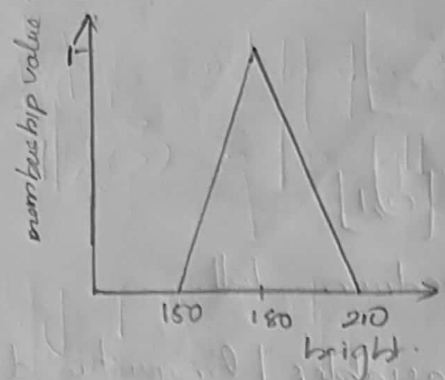
$\mu(tall)$(John) = 0.70

This means there is 70% chance that John is tall

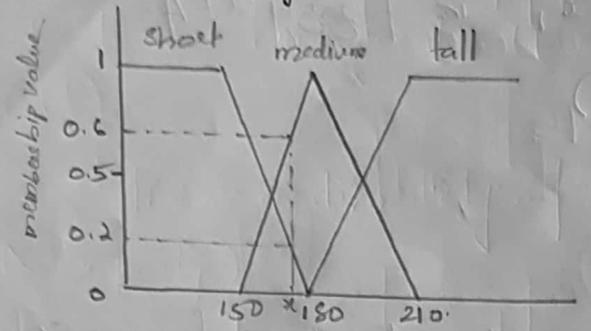
→ Fuzzy set short



Fuzzy set medium



Graph containing all tree



$\mu_{short}(x) = 0.2$

$\mu_{medium}(x) = 0.6$

$\mu_{tall}(x) = 0$

→ Fuzzy sets:

Allows partial membership. Its having degrees of membership b/w 1 & 0. Member of 1 fuzzy set can also be a member of another fuzzy set, in same universe

$$\underset{\sim}{A} = \{(x, \mu_A(x)) \mid x \in U\}, \mu_A(x) \in [0,1]$$

Universe of discourse

~~Fuzzy set A~~

U can be defined as set of ordered pairs and is given by $A = \{(x \mid \mu_A(x)) \mid x \in U\}$ where $\mu_A(x)$ is a degree of membership of x ~~in~~ of fuzzy set A

$\mu_A(x) \in [0,1]$

⇒ value is in range $0 \to 1$

Another way of rep^n. $A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \ldots \right\}$

Scanned by CamScanner

eg: $\underset{\sim}{A} = \left\{ \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$ membership value of $1 = 0.2$
$2 = 0.4$.

9/10/2017 Mon.

Representation of fuzzy set.

→ $A = \{ (x, \mu_A(x)) / x \in U \}$

x ↓ member. $\mu_A(x)$ ↓ membership fn.

$\mu_A(x) \in [0, 1]$

→ $A = \left\{ \frac{0.2}{1} + \frac{1.0}{2} + \frac{0.8}{3} \right\}$ (0.2 → value, 1 → member)

1st member → value is 0.2

→ Tabular repn.

A	2	4	6	8
	0.1	1.0	0.3	0.4

→ If for all the members the membership value is 1 ⇒ known as universal fuzzy set / whole fuzzy set.

→ Equal fuzzy set:

For all members the membership value is same.

Eg: $A = \left\{ \frac{0.8}{1} + \frac{0.9}{2} + \frac{0.7}{3} \right\}$

$B = \left\{ \frac{0.8}{1} + \frac{0.9}{2} + \frac{0.7}{3} \right\}$

$\boxed{\mu_A(x) = \mu_B(x)}$ for $x \in U$

→ Empty fuzzy set

A is said to be empty if

$\boxed{\mu_{\tilde{A}}(x) = 0}$ for all $x \in U$.

→ fuzzy power set.

The collection of all fuzzy sets and fuzzy subsets on universal U is called fuzzy power set P(U).

The cardinality of fuzzy power set P(U) is α (infinity).

$$\boxed{n\,P(U) = \alpha}$$

1> Let $X = \{2,4,6\}$. Find the cardinality of power set.

Ans: $P(X) = \{\phi, \{2\}, \{4\}, \{6\}, \{2,4\}, \{2,6\}, \{4,6\}, \{2,4,6\}\}$

→ If $A \subseteq U \Rightarrow \mu_{\tilde{A}}(x) \leq \mu_{\tilde{U}}(x)$ for all $x \in U$.

Fuzzy Set Operation:

1> Union:

$\tilde{A}$ $\tilde{B}$

$\mu_{\tilde{A} \cup \tilde{B}}(x) = \max[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)]$ for all $x \in U$

$= \mu_{\tilde{A}}(x) \boxed{\vee} \mu_{\tilde{B}}(x)$ where

$\vee$ indicates the max operation.

27) $U = \{a, b, c, d\}$

	a	b	c	d
$\tilde{A}$	0.5	0.8	0.0	0.3
$\tilde{B}$	0.2	1.0	0.1	0.7

Ans: $\mu_{A \cup B}(a) = \max\left[\mu_{\tilde{A}}(a), \mu_{\tilde{B}}(a)\right]$

$= \underline{\underline{0.5}}$

$\mu_{A \cup B}(b) = \max\left[\mu_{\tilde{A}}(b), \mu_{\tilde{B}}(b)\right]$

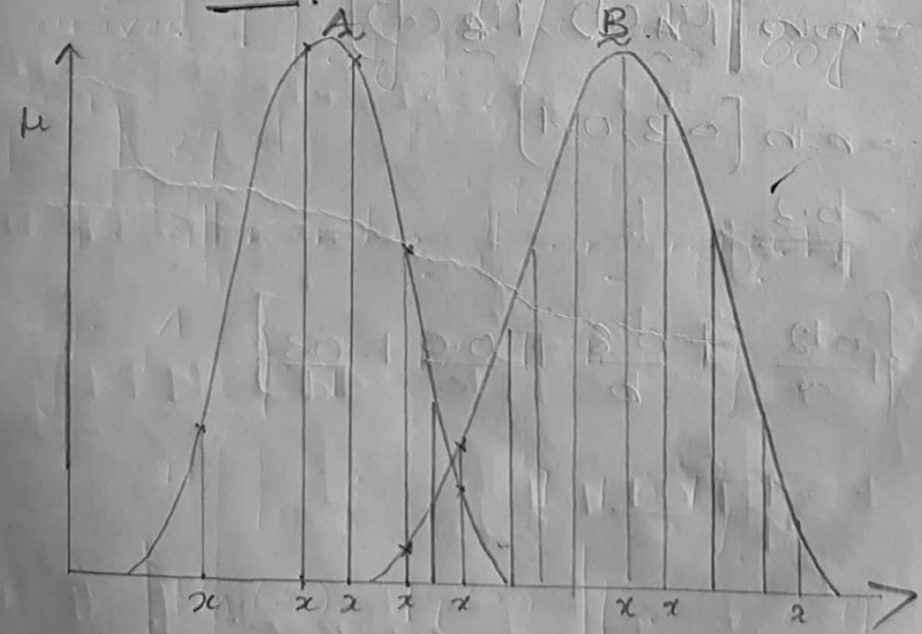
$= \underline{\underline{1.0}}$

$\mu_{A \cup B}(c) = \underline{\underline{0.1}}$

$\mu_{A \cup B}(d) = \underline{\underline{0.7}}$.

$$A \cup B = \left\{\frac{0.5}{a} + \frac{1.0}{b} + \frac{0.1}{c} + \frac{0.7}{d}\right\}$$

Graphical rep^n of $A \cup B$.



$\tilde{A} \cup \tilde{B}$

2) Intersection:

$$\underset{\sim}{A} \cap \underset{\sim}{B} = \min\left[\mu_A(x), \mu_B(x)\right]$$

$= \wedge$ indicates minimum operation.

$\vee$ - supremum
$\wedge$ - infimum

for the above eg:

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}} = \min\left[\mu_{\underset{\sim}{A}}(a), \mu_{\underset{\sim}{B}}(a)\right]$$

$$= \min[0.5, 0.2]$$

$$= \underline{\underline{0.2}}$$

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}} = \min\left[\mu_{\underset{\sim}{A}}(b), \mu_{\underset{\sim}{B}}(b)\right]$$

$$= \min[0.8, 1.0]$$

$$= \underline{\underline{0.8}}$$

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}} = \min\left[\mu_{\underset{\sim}{A}}(c), \mu_{\underset{\sim}{B}}(c)\right]$$

$$= \min[0.0, 0.1]$$

$$= \underline{\underline{0.0}}$$

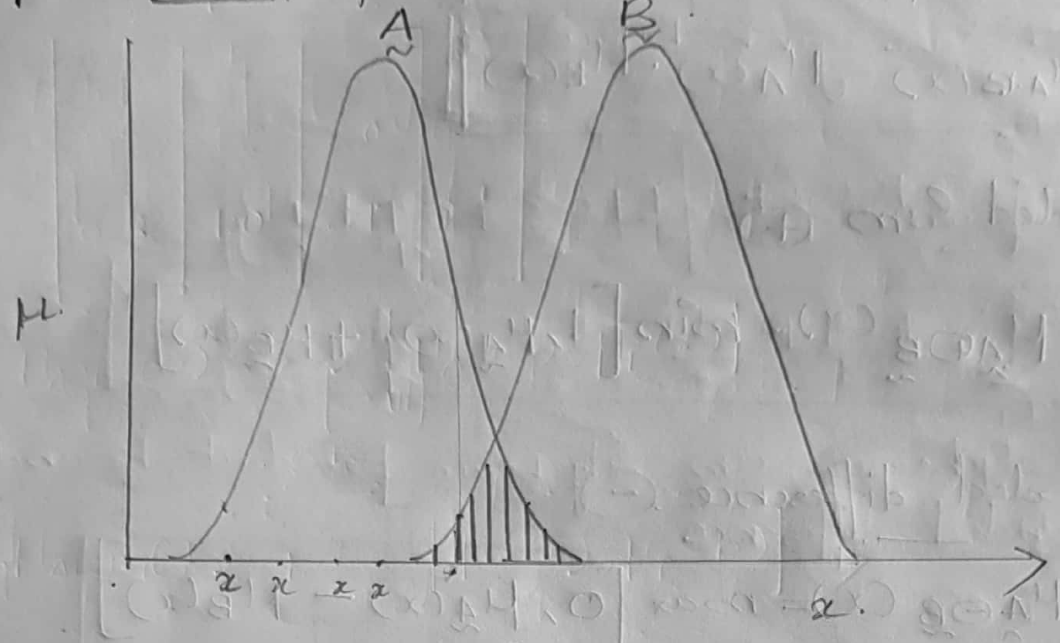
$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}} = \min\left[\mu_{\underset{\sim}{A}}(d), \mu_{\underset{\sim}{B}}(d)\right]$$

$$= \min[0.3, 0.7]$$

$$= \underline{\underline{0.3}}$$

$$A \cap B = \left\{\frac{0.2}{a} + \frac{0.8}{b} + \frac{0.0}{c} + \frac{0.3}{d}\right\}$$

Graphical rep^n of $A \cap B$

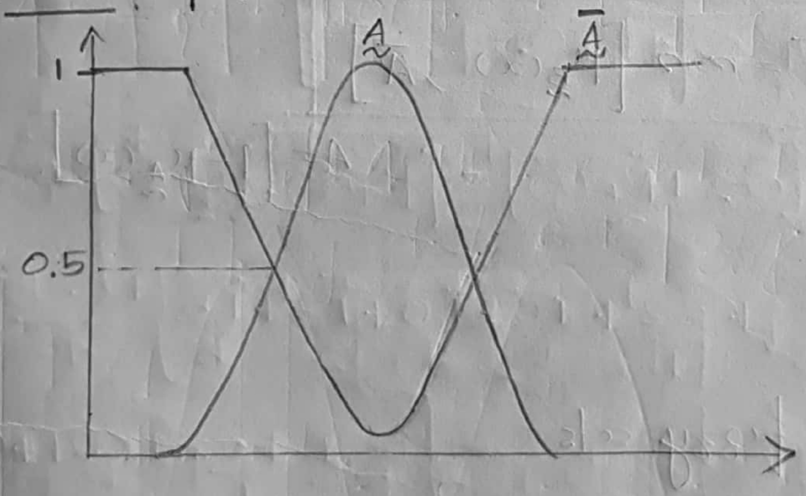


$\underset{\sim}{A} \cap \underset{\sim}{B}$

3> Complement

$$\bar{\underset{\sim}{A}} = \bar{\mu}_{\underset{\sim}{A}}(x) = 1 - \mu_{\underset{\sim}{A}}(x) \quad \text{for all } x \in U$$

Graphical rep^n:



$\bar{\underset{\sim}{A}}$

4> Algebraic sum

$\underset{\sim}{A} + \underset{\sim}{B}$

$$\boxed{\mu_{\underset{\sim}{A} + \underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) + \mu_{B}(x) - \mu_{\underset{\sim}{A}}(x) \cdot \mu_{\underset{\sim}{B}}(x)}$$

5> Algebraic product

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

6> Bounded Sum $\oplus$

$$\boxed{\mu_{\underset{\sim}{A} \oplus \underset{\sim}{B}}(x) = \min\left[1, \mu_{\underset{\sim}{A}}(x) + \mu_{\underset{\sim}{B}}(x)\right]}$$

7> Bounded difference $\ominus$

$$\boxed{\mu_{\underset{\sim}{A} \ominus \underset{\sim}{B}}(x) = \max\left[0, \mu_{\underset{\sim}{A}}(x) - \mu_B(x)\right]}.$$

10/10/2017 Tue.

8> Difference [A/B]

$$\boxed{A/B = A \cap \overline{B} = \min\left[\mu_{\underset{\sim}{A}}(x), \overline{B}\right]}$$

$$\overline{B} = \left[1 - \mu_{\underset{\sim}{B}}(x)\right]$$

$$\boxed{B/A = B \cap \overline{A} = \min\left[\mu_{\underset{\sim}{B}}(x), \overline{A}\right]}$$

$$\overline{A} = \left[1 - \mu_{\underset{\sim}{A}}(x)\right].$$

1) Consider two fuzzy sets

$$\underset{\sim}{A} = \left\{\frac{1}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0.2}{8}\right\}$$

$$\underset{\sim}{B} = \left\{\frac{0.5}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{1}{8}\right\}$$

Perform union, intersection, & complement of the above sets

Ans: $\underset{\sim}{A} \cup \underset{\sim}{B}$

→ $\mu_{\underset{\sim}{A} \cup \underset{\sim}{B}}(x) = \max\left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right]$

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \left\{\frac{1}{2} + \frac{0.4}{4} + \frac{0.5}{6} + \frac{1}{8}\right\}$$

→ $\underset{\sim}{A} \cap \underset{\sim}{B}$

$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}} = \min\left[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right]$

$$\underset{\sim}{A} \cap \underset{\sim}{B} = \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.1}{6} + \frac{0.2}{8}\right\}$$

→ $\overline{\underset{\sim}{A}} = \overline{\mu}_{\underset{\sim}{A}}(x) = 1 - \mu_{\underset{\sim}{A}}(x)$

$$\overline{\underset{\sim}{A}} = \left\{\frac{0}{2} + \frac{0.7}{4} + \frac{0.5}{6} + \frac{0.8}{8}\right\}$$

$$\overline{\underset{\sim}{B}} = \left\{\frac{0.5}{2} + \frac{0.6}{4} + \frac{0.9}{6} + \frac{0}{8}\right\}$$

→ $A / B = A \cap \overline{B} = \min\left[\mu_{\underset{\sim}{A}}(x), \overline{\underset{\sim}{B}}\right]$

$\underset{\sim}{B} = \left\{\frac{0.5}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{1}{8}\right\}$ | $\underset{\sim}{A} = \left\{\frac{1}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0.2}{8}\right\}$

$\overline{\underset{\sim}{A}} = \left\{\frac{0}{2} + \frac{0.7}{4} + \frac{0.5}{6} + \frac{0.8}{8}\right\}$ | $\overline{\underset{\sim}{B}} = \left\{\frac{0.5}{2} + \frac{0.6}{4} + \frac{0.9}{6} + \frac{0}{8}\right\}$

$$A / B = \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0}{8}\right\} \quad \min\left[\mu_{\underset{\sim}{A}}(x), \overline{B}\right]$$

$$B / A = \left\{\frac{0}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{0.8}{8}\right\} \quad \min\left[\mu_{\underset{\sim}{B}}(x), \overline{\underset{\sim}{A}}\right]$$

2) The membership function for a transistor and a resistor are given below

$$\underset{\sim}{\mu_T} = \left\{ \frac{0}{0} + \frac{0.2}{1} + \frac{0.7}{2} + \frac{0.8}{3} + \frac{0.9}{4} + \frac{1}{5} \right\}$$

$$\underset{\sim}{\mu_R} = \left\{ \frac{0}{0} + \frac{0.1}{1} + \frac{0.3}{2} + \frac{0.2}{3} + \frac{0.4}{4} + \frac{0.5}{5} \right\}$$

Find algebraic sum, algebraic product, bounded sum, bounded difference

Ans:→ Algebraic sum

$$\mu_{\underset{\sim}{T}+\underset{\sim}{R}}(x) = \mu_{\underset{\sim}{T}}(x) + \mu_{\underset{\sim}{R}}(x) - \mu_{\underset{\sim}{T}}(x) \cdot \mu_{\underset{\sim}{R}}(x)$$

$$= \cancel{\left\{ \frac{0}{0} + \frac{0.1}{1} + \frac{-1.1}{2} + \frac{-0.6}{3} + \frac{-2.3}{4} + \frac{1}{5} \right\}}$$

$$= \left[\mu_{\underset{\sim}{T}}(x) + \mu_{\underset{\sim}{R}}(x) \right] - \left[\mu_{\underset{\sim}{T}}(x) \cdot \mu_{\underset{\sim}{R}}(x) \right]$$

$$= \left\{ \frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5} \right\}$$

$$- \left\{ \frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5} \right\}$$

$$\mu_{\underset{\sim}{T}+\underset{\sim}{R}}(x) = \left\{ \frac{0}{0} + \frac{0.28}{1} + \frac{0.79}{2} + \frac{0.84}{3} + \frac{0.94}{4} + \frac{1}{5} \right\}$$

$\rightarrow \mu_{\underset{\sim}{T}.\underset{\sim}{R}}(x) = \mu_{\underset{\sim}{T}}(x) \cdot \mu_{\underset{\sim}{R}}(x)$

$$= \left\{ \frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5} \right\}$$

$\rightarrow \mu_{\underset{\sim}{T} \oplus \underset{\sim}{R}} = \min\left[1, \mu_{\underset{\sim}{T}}(x) + \mu_{\underset{\sim}{R}}(x)\right]$

$$= \min\left[1, \left\{ \frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5} \right\}\right]$$

$$= \left\{ \frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1}{4} + \frac{1}{5} \right\}$$

$\rightarrow \mu_{\underset{\sim}{T} \ominus \underset{\sim}{R}} = \max\left[0, \mu_{\underset{\sim}{T}}(x) - \mu_{\underset{\sim}{R}}(x)\right]$

$$= \max\left[0, \left\{ \frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5} \right\}\right]$$

$$= \left\{ \frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5} \right\}$$

3) It is necessary to compare two sensors based upon their detection level and gain settings. The membership values are given in the following table.

Gain Settings	Detection level (D_1) of Sensor I	Detection level of (D_2) Sensor II
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

Find

i) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}(x)$

ii) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x)$

iii) $\mu_{\overline{\underset{\sim}{D_1}}}(x)$

iv) $\mu_{\overline{\underset{\sim}{D_2}}}(x)$

v) $\mu_{\underset{\sim}{D_1} \cup \overline{\underset{\sim}{D_1}}}(x)$

vi) $\mu_{\underset{\sim}{D_1} \cap \overline{\underset{\sim}{D_1}}}(x)$

vii) $\mu_{\underset{\sim}{D_2} \cup \overline{\underset{\sim}{D_2}}}(x)$

viii) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x)$

ix) $\mu_{D_1/D_2}(x)$

x) $\mu_{D_2/D_1}(x)$

Ans:

i) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}(x) = \max\left[\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\right]$

$$\mu_{\underset{\sim}{D_1}}(x) = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

$$\mu_{\underset{\sim}{D_2}}(x) = \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

$$\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}(x) = \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.35}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

ii) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x) = \min\left[\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\right]$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.25}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

iii) $\mu_{\bar{\underset{\sim}{D_1}}}(x) = 1 - \mu_{\underset{\sim}{D_1}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

(iv) $\mu_{\bar{\underset{\sim}{D_2}}}(x) = 1 - \mu_{\underset{\sim}{D_2}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(v) $\mu_{\underset{\sim}{D_1} \cup \bar{D_1}}(x) = \max\left[\mu_{\underset{\sim}{D_1}}(x), \mu_{\bar{\underset{\sim}{D_1}}}(x)\right]$

$$\mu_{D_1}(x) = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

$$\mu_{\bar{\underset{\sim}{D_1}}}(x) = 1 - \mu_{\underset{\sim}{D_1}}(x)$$

$$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

$$\mu_{\underset{\sim}{D_1} \cup \bar{\underset{\sim}{D_1}}}(x) = \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

(vi) $\mu_{\underset{\sim}{D_1} \cap \bar{\underset{\sim}{D_1}}}(x) = \min\left[\mu_{\underset{\sim}{D_1}}(x), \mu_{\bar{\underset{\sim}{D_1}}}(x)\right]$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

(vii). $\mu_{\underset{\sim}{D_2} \cup \bar{\underset{\sim}{D_2}}}(x) = \max\left[\mu_{\underset{\sim}{D_2}}(x), \mu_{\bar{\underset{\sim}{D_2}}}(x)\right]$

$$\mu_{\underset{\sim}{D_2}}(x) = \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

$$\mu_{\bar{\underset{\sim}{D_2}}}(x) = 1 - \mu_{\underset{\sim}{D_2}}(x)$$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

$$\mu_{\underset{\sim}{D_2} \cup \overline{\underset{\sim}{D_2}}}(x) = \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

(viii) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x) = \min\left[\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\right]$

$$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(ix) $\mu_{D_1/D_2}(x) = D_1 \cap \overline{D_2}$

$$= \min\left[\mu_{\underset{\sim}{D_1}}(x), \overline{D_2}\right]$$

$$\mu_{\underset{\sim}{D_1}}(x) = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

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

$$\mu_{D_1/D_2}(x) = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(x) $\mu_{D_2/D_1}(x) = \min\left[\mu_{D_2}(x), \overline{D_1}\right]$

$$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

Scanned by CamScanner

14/10/2017 Sat.

Properties of fuzzy sets:

<1> Commutativity.

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup \underset{\sim}{A}$$

$$\underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{B} \cap \underset{\sim}{A}$$

<2> Associativity.

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cup C$$

$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cap C$$

<3> Distributivity:

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cap (\underset{\sim}{A} \cup \underset{\sim}{C})$$

$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cup (\underset{\sim}{A} \cap \underset{\sim}{C})$$

<4> Idempotency.

$$\underset{\sim}{A} \cup \underset{\sim}{A} = \underset{\sim}{A}$$

$$\underset{\sim}{A} \cap \underset{\sim}{A} = \underset{\sim}{A}$$

<5> Identity.

$$\underset{\sim}{A} \cup \phi = \underset{\sim}{A}$$

$$\underset{\sim}{A} \cup U = U$$

$$\underset{\sim}{A} \cap \phi = \phi$$

$$\underset{\sim}{A} \cap U = \underset{\sim}{A}$$

14/10/2017
Sat

<6> Transitivity

$$\underset{\sim}{A} \subseteq B \subseteq C \text{ , } A \subseteq C$$

<7> Involution. (Double negation)

$$\overline{\overline{\underset{\sim}{A}}} = \underset{\sim}{A}$$

<8> De Morgan's Law

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

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

Scanned by CamScanner

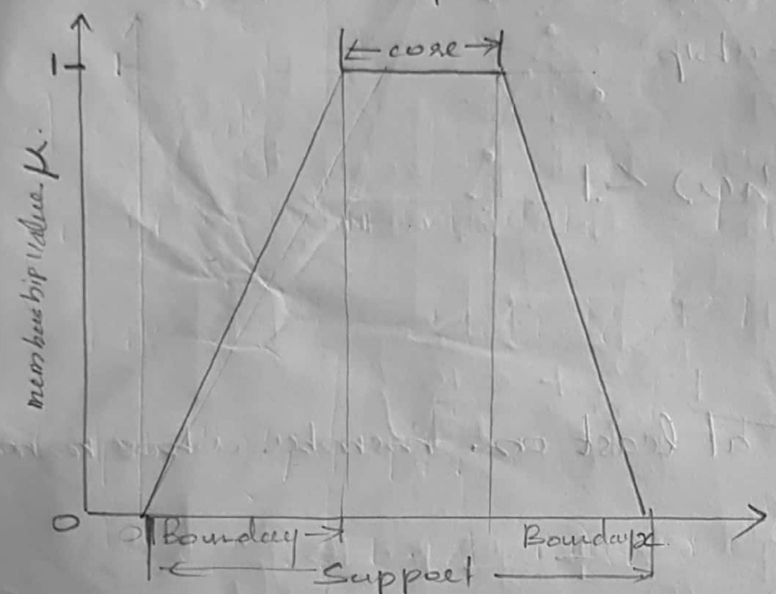
17/10/2017

Module 4

Fuzzy Membership Functions

$$\underset{\sim}{A} = \{(x, \mu_A(x)) / x \in U\}$$

→ features of membership.



Elements for which membership value is exactly 1 - core

Boundary values - Boundary members

→ **Core:**

Core of a membership function for a fuzzy set A is defined as the region of universe that has complete membership in set A

$$\mu_A(x) = 1$$

→ **Support:**

Support of a membership function for a fuzzy set A is defined as the region of universe that has elements with non-zero membership in set A.

→ Fuzzy singleton set.

→ only one element
$\mu_A(x)=1$

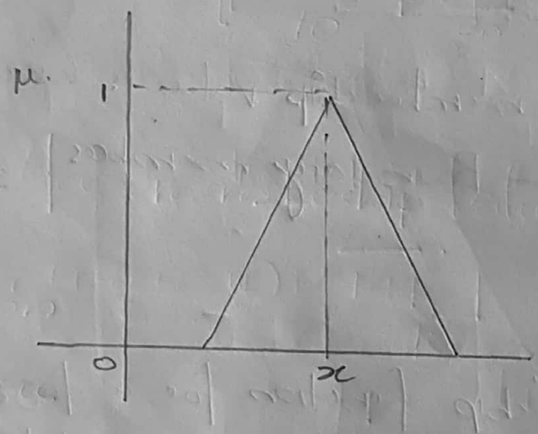
→ Boundary:

Boundary is defined as the region of universe that contain the elements which are non-zero but not complete membership.

$$0<\mu_A(x)<1$$

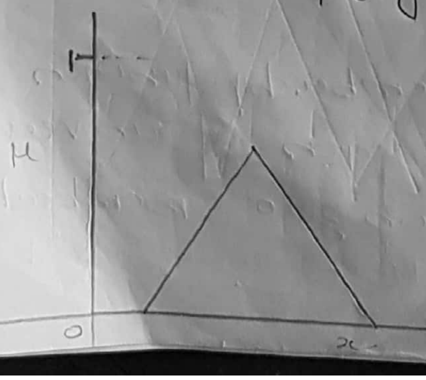
→ Normal fuzzy set

There will be at least one member whose membership value is 1.



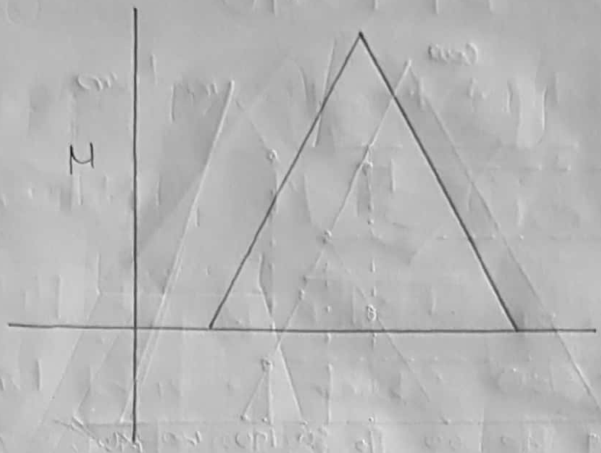
→ Sub normal fuzzy set:

If there is no member with membership value 1 → subnormal fuzzy set

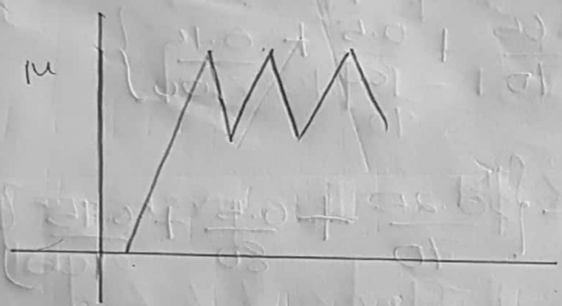


→ Convex fuzzy set:

The membership value first increases and then decreases

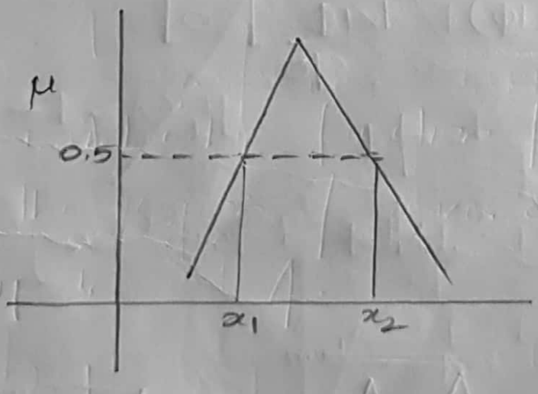


→ Non-Convex fuzzy sets:



→ Cross over points:

The elements whose membership value is 0.5.

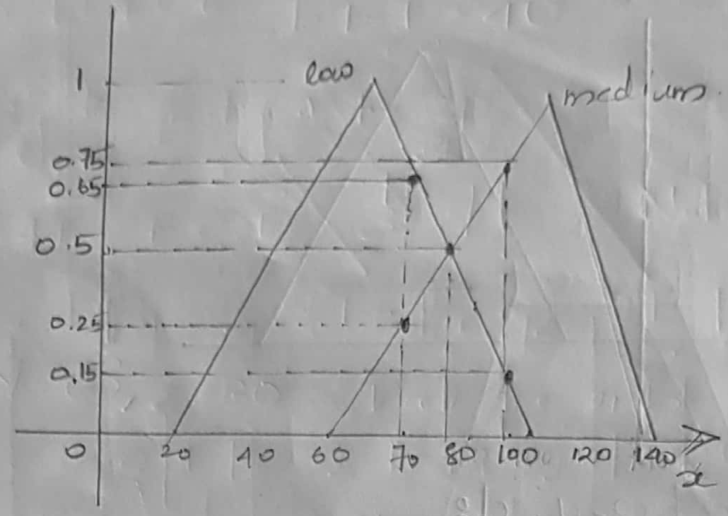


Fuzzification:

Converting crispset into a fuzzy set.

MODULE 4

Eg: $S = \{70, 80, 100\} \rightarrow$ Speed.



$$low = \left\{ \frac{0.65}{70} + \frac{0.5}{80} + \frac{0.15}{100} \right\}$$

$$medium = \left\{ \frac{0.25}{70} + \frac{0.5}{80} + \frac{0.75}{100} \right\}$$

→ Methods for membership value assignments.

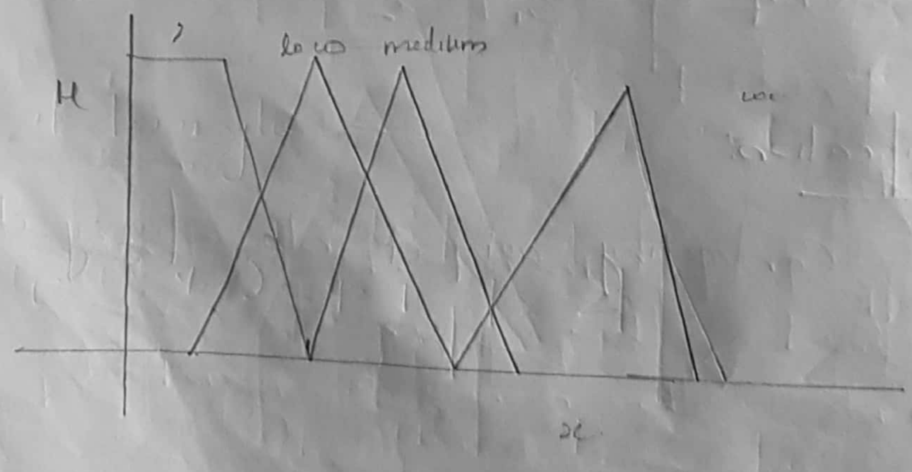
<1> Intuition

<2> Inference

<3> Rank ordering.

<1> Intuition

<1>



<2> Inference.

Eg: U = Universe of triangles.

$X \geq Y \geq Z \geq 0$.

$X + Y + Z = 180°$

Isosceles - Two angle are equal

Right - One angle 90

equilateral - All angle 60

Isosceles triangle

membership fn:

$$\boxed{\mu_I (X, Y, Z) = 1 - \frac{1}{60} \min (X - Y, Y - Z)}$$

$X \geq Y \geq Z,\ X + Y + Z = 180$

$X = 60 \quad Y = 60 \quad Z = 60$

$$= 1 - \frac{1}{60} \min (0, 0)$$

$$= \underline{\underline{1}}$$

Isosceles Δle returns the value 1.

Right angled triangle

$$\boxed{\mu_R (x, y, z) = 1 - \frac{1}{90} |X - 90|}$$

$X = 90 \quad Y = 60 \quad Z = 30$

$$\mu_R = 1 - \frac{1}{90} |0|$$

$$= \underline{\underline{1}}$$

17/10/2017 Tue

1) Check whether $X = 120$ $Y = 30$ $Z = 30$ forms a isosceles triangle

$$\mu_I (x, y, z) = 1 - \frac{1}{60} \min ((X - Y, Y - Z)$$

$$= 1 - \frac{1}{60} \min (90, 0)$$

$$= \underline{\underline{1}}.$$

$\therefore$ Triangle is isosceles

2) $X = 120$ $Y = 50$, $Z = 10$; Check for isosceles.

$x \geq y \geq z$, $x+y+z = 180°$

$$\mu_I(120, 50, 10) = 1 - \frac{1}{60} \min(70, 40)$$

$$= 1 - \frac{1}{60} \times 40$$

$$= 1 - \frac{2}{3}$$

$$= \frac{1}{3} \neq 1$$ ∴ Not an isosceles triangle

3) $x = 90°$, $y = 60°$, $z = 30$; Check for right triangle

$$\mu_R(x, y, z) = 1 - \frac{1}{90} |x - 90|$$

$$= 1 - \frac{1}{90} |90 - 90|$$

$$\cancel{\neq} = 1$$ ∴ The given angles form a right triangle

4) $x = 120$ $y = 50$ $z = 10$; Check for right angled triangle

$x \geq y \geq z$ $x+y+z = 180$

$$\mu_R(120, 50, 10) = 1 - \frac{1}{90} |30|$$

$$\neq 1$$

∴ Not a right angled triangle.

→ Equilateral triangle

$$\boxed{\mu_E(x, y, z) = 1 - \frac{1}{180} |x - z|}$$

1) $X=60 \quad Y=60 \quad Z=60$; Check for equilateral triangle.

$$\mu_E(60,60,60) = 1-\frac{1}{180}|60-60|$$

$$= \underline{\underline{1}}$$

$\therefore$ The given angles form ~~a~~ equilateral triangle

→ Isosceles and right angled triangle.

$$\boxed{\underset{\sim}{I} \cap \underset{\sim}{R} = \min\left[\mu_{\underset{\sim}{I}}(x), \mu_{\underset{\sim}{R}}(x)\right]}$$

1) $X=90$; $Y=45$, $Z=45$

$$\mu_{\underset{\sim}{I}} = 1-\frac{1}{60}\min(X-Y, Y-Z)$$

$$= 1-\frac{1}{60}\min(45, 0)$$

$$= \underline{\underline{1}}$$

$$\mu_{\underset{\sim}{R}} = 1-\frac{1}{90}|X-90|$$

$$= 1-\frac{1}{90}|0|$$

$$= \underline{\underline{1}}$$

$$\underset{\sim}{I} \cap \underset{\sim}{R} = \min\left[\mu_{\underset{\sim}{I}}(x), \mu_{\underset{\sim}{R}}(x)\right)$$

$$= \min(1, 1)$$

$$= \underline{\underline{1}}.$$

→ T = other triangles

$$\boxed{T = \overline{I \cup R \cup E}}$$

Apply de morgan's law.

$T = \bar{I} \cap \bar{R} \cap \bar{E}$

$= (1 - \mu_{\underset{\sim}{I}}(x)) \cap (1 - \mu_{\underset{\sim}{R}}(x)) \cap (1 - \mu_{\underset{\sim}{E}}(x))$

$= \min [1 - \mu_{\underset{\sim}{I}}(x), 1 - \mu_{\underset{\sim}{R}}(x), 1 - \mu_{\underset{\sim}{E}}(x)]$

$$\boxed{= \min [\bar{\mu}_{\underset{\sim}{I}}(x), \bar{\mu}_{\underset{\sim}{R}}(x), \bar{\mu}_{\underset{\sim}{E}}(x)]}$$

1) $X = 120, Y = 50, Z = 10$;

$\mu_I(x) = 1 - \frac{1}{60} \min\{X - Y, Y - Z)$

$= 1 - \frac{1}{60} \min(70, 40)$

$= 1 - \frac{1}{60} \times 40$

$= \frac{1}{3}$ $\qquad \bar{\mu}_I(x) = 1 - \frac{1}{3} = \frac{2}{3}$

$\mu_R(x) = 1 - \frac{1}{90} |X - 90|$

$= 1 - \frac{1}{90} \times 30$

$= \frac{2}{3}$ $\qquad \mu_{\bar{R}}(x) = 1 - \frac{2}{3} = \frac{1}{3}$

$\mu_E(x) = 1 - \frac{1}{180} |X - Z|$

$= 1 - \frac{1}{180} |120 - 10|$

$= 1 - \frac{11}{18} = \frac{7}{18}$ $\qquad \bar{\mu}_E(x) = 1 - \frac{7}{18} = \frac{11}{18}$

$$T = \min\left[\frac{2}{3}, \frac{1}{3}, \frac{11}{18}\right]$$

$$= \frac{1}{3}.$$

⟨3⟩ Rank Ordering

100 votes

C1 – 50 (1st)   C2 = 30 (2nd)   C3 = 20 (3rd)

$A = \{50, 30, 20\}$

$$\underset{\sim}{A} = \left\{\frac{0.5}{C1} + \frac{0.3}{C2} + \frac{0.2}{C3}\right\}.$$

1) Suppose 10000 people respond to a questionnaire about differences among 5 cars

Cars	Total
Maruti	1651
Scorpio	1955
Matiz	1860
Santro	1912
Octavia	2622

We Build fuzzy set. Write in the rank order.

Ans: Preference = $\left\{\frac{0.1651}{\text{Maruti}} + \frac{0.1955}{\text{Scorpio}} + \frac{0.186}{\text{Matiz}} + \frac{0.1912}{\text{Santro}} + \frac{0.2622}{\text{Octavia}}\right\}$

23/10/2017 Mon.

Lamda Cut (Alpha-cut).

$$\boxed{A_\lambda = \{x / \mu_A(x) \geq \lambda\}}$$ Weak Lamda set

Eg: $A = \left\{\frac{0.5}{1} + \frac{0.2}{2} + \frac{0.3}{3} + \frac{0.4}{4}\right\}$

$\lambda = 0.4$.

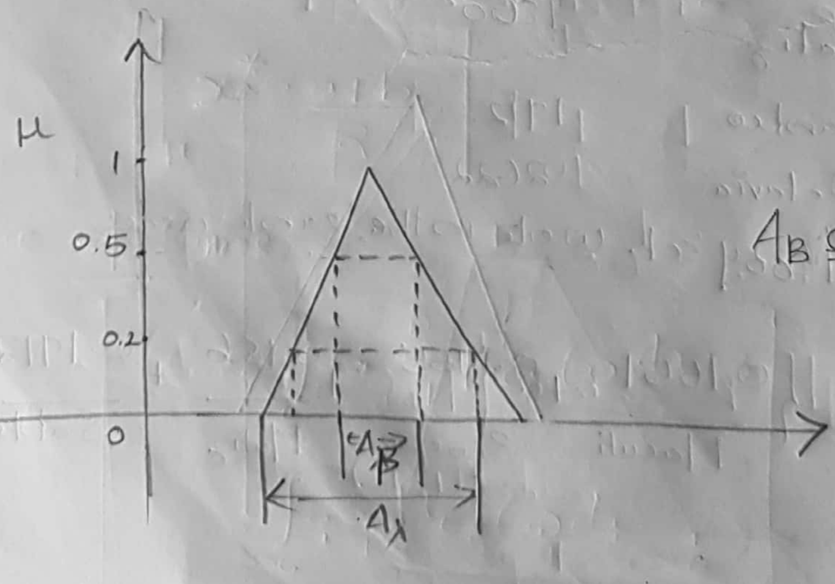
$A_{0.4} = \left\{\frac{0.5}{1} + \frac{0.4}{4}\right\} \Rightarrow$ Weak Lamda set.

$\Rightarrow$ Strong lamda set $\boxed{A_\lambda = \{x / \mu_A(x) > \lambda\}}$

$A_{0.4} = \left\{\frac{0.5}{1}\right\}$

Properties

1. $(A \cup B)_\lambda = A_\lambda \cup B_\lambda$
2. $(A \cap B)_\lambda = A_\lambda \cap B_\lambda$
3. $(\bar{A})_\lambda \neq (\overline{A_\lambda})$
4. $A_\beta \subseteq A_\lambda \qquad \lambda \leq \beta$.



$\lambda = 0.2$
$\beta = 0.5$

$A_\beta \subseteq A_\lambda$.

Scanned by CamScanner

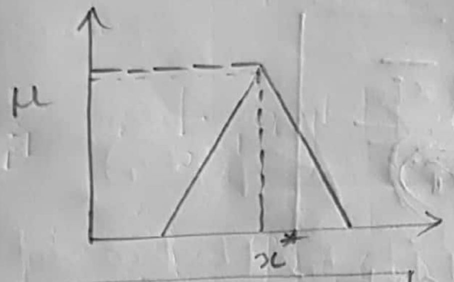
Defuzzification

converting a fuzzy set into a crispset

* find out a representative value representing the entire set

methods to defuzzify

1) max-membership height method



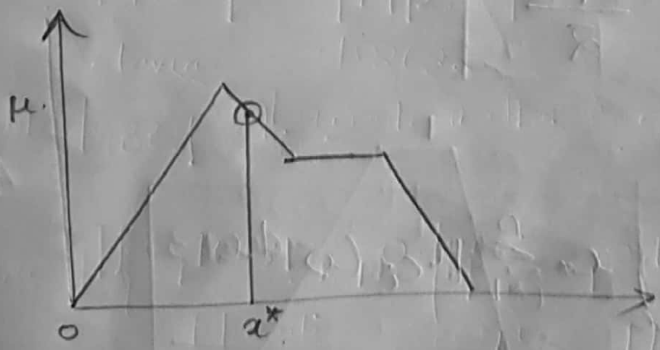
$$\boxed{\mu_c(x^*) \geq \mu_c(x)} \text{ for all } x \in X$$

2) Centroid method

It is also known as centre of mass / centre of area / centre of gravity method.

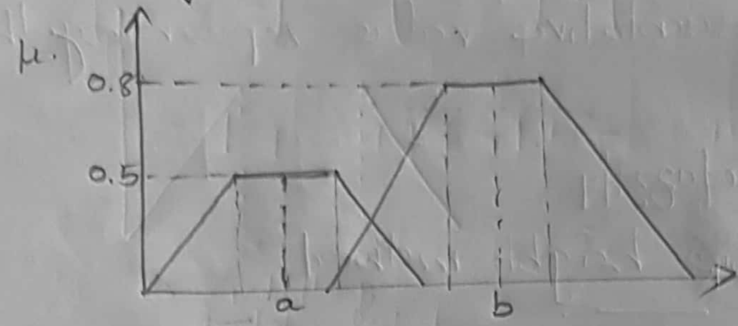
$$\boxed{x^* = \frac{\int \mu_c(x) \cdot x\,dx}{\int \mu_c(x)\,dx}}$$

where $\int$ is the algebraic integration



3> Weighted Average method

→ Used for symmetrical membership functions.

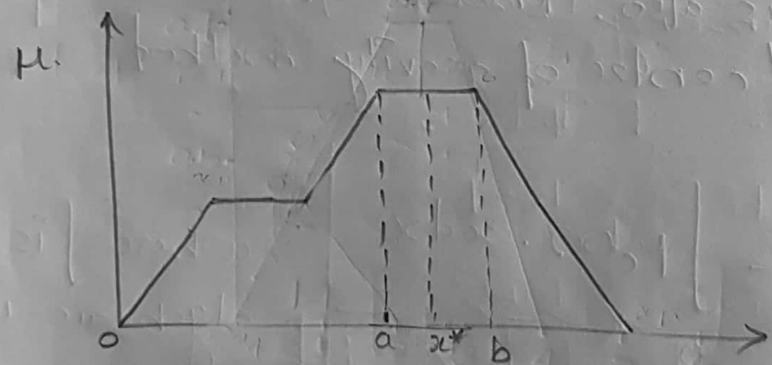


$\overline{x_i} \rightarrow \max$

$$\boxed{x^* = \frac{\sum \mu_c(\overline{x_i}) \cdot \overline{x_i}}{\sum \mu_c(\overline{x_i})}}$$

$$= \frac{0.5 * a + 0.8 * b}{0.5 + 0.8}$$

4> Middle of maxima (Mean max method).



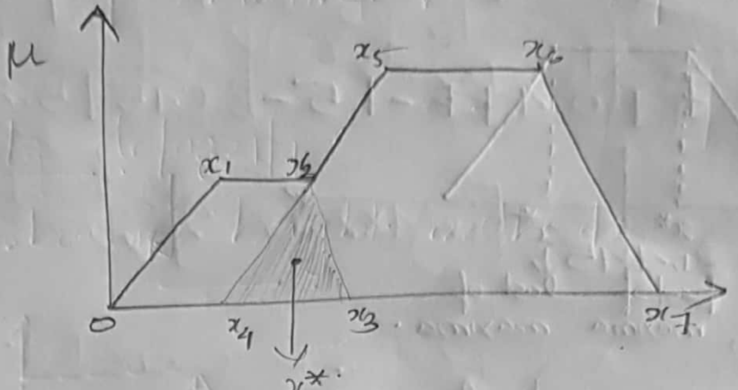
We take mean of the maximum value.

$$\boxed{x^* = \frac{a+b}{2}}$$

5> Centre of Sums

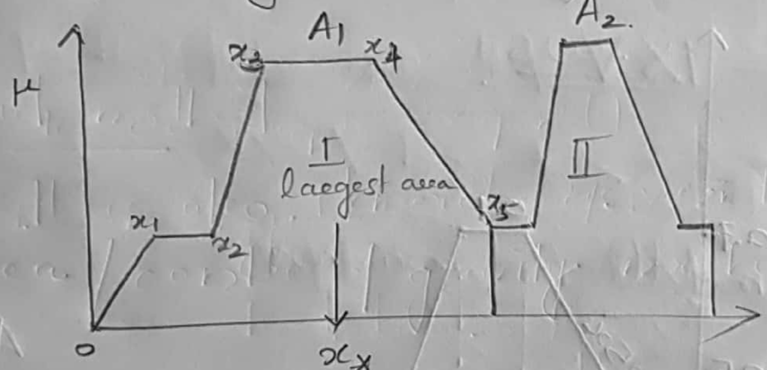
$$\boxed{x^* = \frac{\int_x x * \sum_{i=1}^{n} \mu_{c_i}(x)\, dx}{\int_x \sum_{i=1}^{n} \mu_{c_i}(x)\, dx}}$$

This uses algebraic sum of individual fuzzy subsets instead of their union. The drawback is that the intersecting areas are added twice



$$\int_0^{x_1} + \int_{x_1}^{x_2} + \int_{x_2}^{x_3} + \int_{x_4}^{x_5} + \int_{x_5}^{x_6} + \int_{x_6}^{x_7}$$

6) Center of largest area



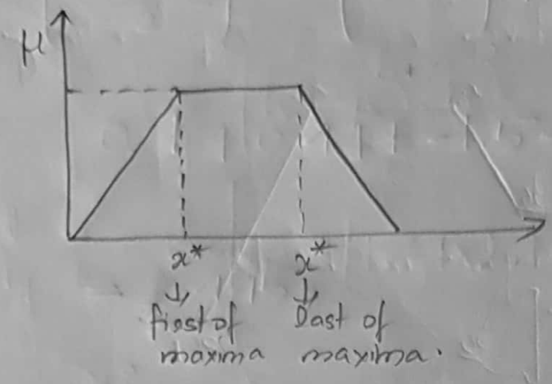
$$\boxed{x^* = \frac{\int \mu_{C_j}(x) \cdot x \, dx}{\int \mu_{C_j}(x) \, dx}}$$

C_j = region which has largest area.

This method can be adopted when the output consists of atleast two convex fuzzy subsets which are not overlapping.

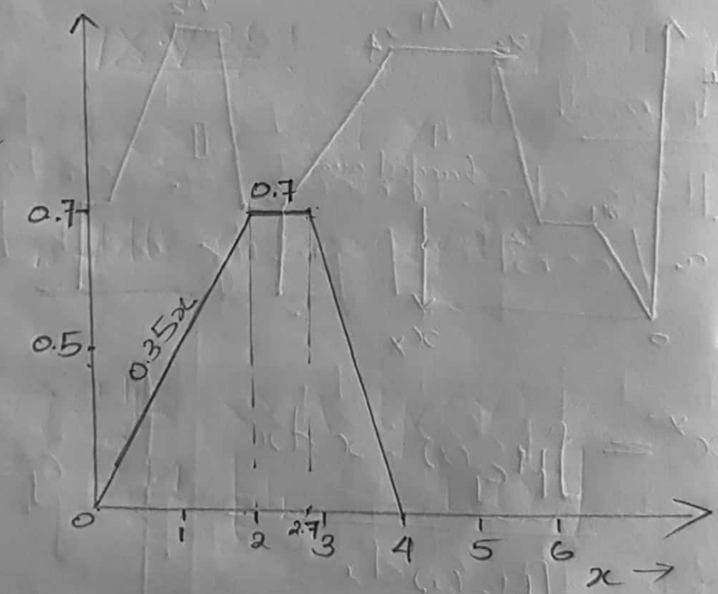
When there are two regions the centre of gravity of the region having the largest area is used to obtain the defuzzified value x^*.

7) First of maxima and last of maxima.

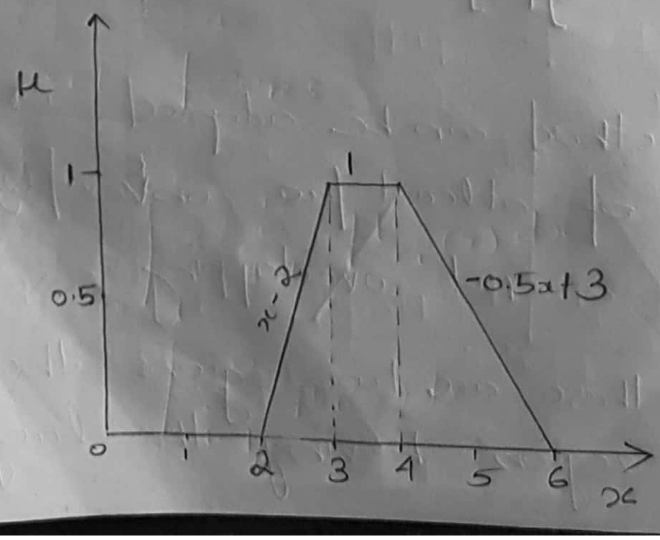


24/10/2017
Tue.

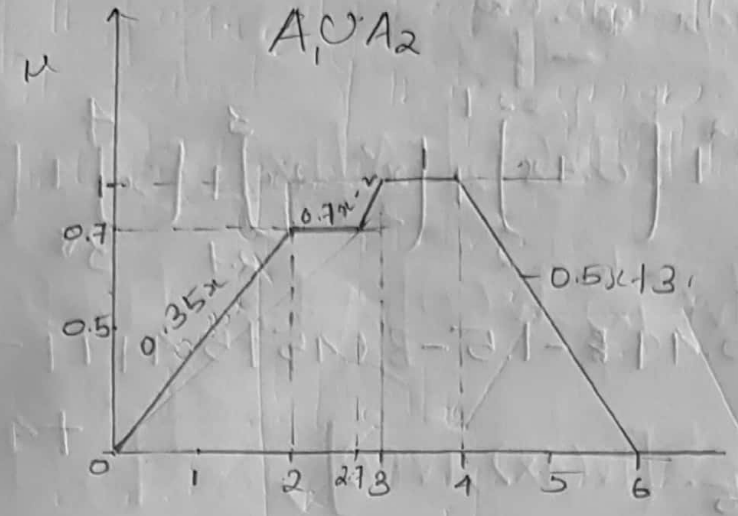
i) The membership function shown below determine the defuzzified output by all the seven methods



A_1



A_2.



method - 1 - max membership height method

$$x^* \mid \mu(x^*) > \mu(x) \text{ for } x \in X$$

$$x^* = 3$$

method 2 - centroid method.

$$x^* = \frac{\int \mu_c(x) \cdot x\,dx}{\int \mu_c(x)\,dx}$$

$\int \mu_c(x) \cdot x\,dx$

$\Downarrow$

$$= \int_0^2 0.35x \cdot x\,dx + \int_2^{2.7} 0.7x\,dx + \int_{2.7}^{3} (x-2)x\,dx + \int_3^4 1x\,dx + \int_4^6 (-0.5x+3)x\,dx$$

$$= \left[\frac{0.35x^3}{3}\right]_0^2 + \frac{0.7}{2}\left[x^2\right]_2^{2.7} + \left[\frac{x^3}{3} - \frac{\cancel{2}x^2}{\cancel{2}}\right]_{2.7}^{3} + \left[\frac{x^2}{2}\right]_3^4 + \left[\frac{-0.5x^3}{3} + \frac{3x^2}{2}\right]_4^6$$

$$= \frac{0.35}{3} \times 8 + \frac{0.7}{2} \times 3.29 + \frac{27}{3} - 9 - 6.561 + 7.29 + 8 - \frac{9}{2} - 36 + 54 + \frac{32}{3} - 24$$

$$= \underline{\underline{7.31383}}$$

$$\int \mu_c(x)\,dx = \int_0^2 0.35x\,dx + \int_2^{2.7} 0.7\,dx + \int_{2.7}^{3} (x-2)\,dx + \int_3^4 dx$$

Scanned by CamScanner

$$+\int_4^6 -0.5x+3\,dx$$

$$= \left[\frac{0.35x^2}{2}\right]_0^2 + \left[0.7x\right]_2^{2.7} + \left[\frac{x^2}{2}-2x\right]_{2.7}^{3} + \left[x\right]_3^4 + \left[\frac{-0.5x^2}{2}+3x\right]_4^6$$

$$= 0.7 + 0.49 - 1.5 - 3.645 + 5.4 + 1 + -9 + 18 + 4 - 12$$

$$= 3.445$$

method 3 - Weighted Average

$$x^* = \sum \mu_c(\bar{x_i}) \cdot \bar{x_i} \Big/ \sum \mu_c(\bar{x_i})$$

$$x^* = \frac{2.35 \times 0.7 + 3.5 * 1}{2.1 + 0.7}$$

$$= \underline{\underline{3.026}}$$

method 4 - middle of maxima

$$x^* = \frac{a+b}{2}$$

$$= \frac{3+4}{2} = \underline{\underline{3.5}}$$

method 5: Center of sums

$$x^* = \frac{\int_x x \cdot \sum_{i=1}^{n} \mu_{ci}(x)\,dx}{\int_x \sum_{i=1}^{n} \mu_{ci}(x)\,dx}$$

coordinates (4, 0) (2.7, 0.7)

$A_1 =$ Eqn of line.

$y_1 - y_1 = m(x - x_1)$

$m = \frac{y_2 - y_1}{x_2 - x_1}$

$$\frac{(y-0)(0-0.7)}{}$$

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \Rightarrow \frac{y - 0}{0.7 - 0} = \frac{x - 4}{2.7 - 4}.$$

$$\Rightarrow -1.3y = 0.7x - 2.8$$

$$y = \frac{1}{1.3}\left[2.8 - 0.7x\right]$$

$$\frac{\int_0^6 x * \sum_{i=1}^{n} \mu_{ci}(x)\,dx}{\int_0^6 \sum_{i=1}^{n} \mu_{ci}(x)\,dx}$$

=

26/10/17 Thu.

module 5

Fuzzy Rules

Proposition in any sentence.

8 is P
⇓ ⇓
subject | predicate

⇒ Two propositions X and Y can be connected by:

1) X AND Y conjunction ($\wedge$)
2) X OR Y disjunction ($\vee$)
3) $\neg X$
4) $\neg Y$
5) $X \Rightarrow Y = (\neg X \vee Y)$ implies if X then Y.
6) $X \Leftrightarrow Y = X \Rightarrow Y$ and $Y \Rightarrow X$

$$((\neg X) \vee Y) \wedge ((\neg Y) \vee X)$$

⇒ $tv(X \text{ AND } Y) = tv(X) \wedge tv(Y)$
$= \min[tv(X), tv(Y)]$

X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1

$\Rightarrow$ X OR Y

$tv(X \text{ OR } Y) = tv(X) \text{ OR } tv(Y)$

$= \max[tv(X), tv(Y)]$

X	Y	OR
0	0	0
0	1	1
1	0	1
1	1	1

$\Rightarrow \neg X$

$tv(\text{NOT } X) = 1 - tv(X)$

X	$\neg X$
0	1
1	0

$\Rightarrow X \Rightarrow Y = \neg X \vee Y$

$tv(\neg X) \text{ OR } tv(Y)$

$= \max[tv(\neg X), tv(Y)]$

$= \max[1 - tv(X), tv(Y)]$

X	$\neg X$	Y	$X \Rightarrow Y$
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

$\Rightarrow X \Leftrightarrow Y = ((\neg X) \vee Y) \wedge ((\neg Y) \vee X)$

$= [tv(\neg X) \text{ OR } tv(Y)] \wedge [tv(\neg Y) \text{ OR } tv(X)]$

$= \max[tv(\neg x), tv(Y)] \wedge \max[tv(\neg Y), tv(X)]$

$= \min[\max[tv(\neg x), tv(Y)], \max[tv(\neg Y), tv(X)]]$

X	¬X	Y	¬Y	X ⇔ Y
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1

$\Rightarrow$ Fuzzy Propositions:

- fuzzy predicates
- fuzzy predicate modifiers
- fuzzy quantifiers
- fuzzy qualifiers.

- fuzzy predicates:

tall, short, cold, hot, ... are fuzzy predicates.

Eg: climate is hot

- fuzzy predicate modifiers.

Eg: climate is very hot.

Climate is $\underbrace{\text{moderately}}_{\text{predicate modifier}}$ cold

- fuzzy quantifiers

many, most, .. are fuzzy quantifiers.

Eg: Most people are educated.

- fuzzy qualifiers

These are 4 types.

1> fuzzy truth qualifier.

x is T where T is a fuzzy truth value.

Eg: (PAUL is YOUNG) IS NOT TRUE

2> fuzzy probability qualification

x is λ where x is fuzzy probability.

Eg. $\underbrace{\text{PAUL IS YOUNG}}_{x}$ $\underbrace{\text{is}}_{\text{is}}$ $\underbrace{\text{likely}}_{\lambda}$.

3> fuzzy possibility qualification

x is π where π is fuzzy possibility.

Eg. $\underbrace{\text{PAUL is YOUNG}}_{x}$ $\underbrace{\text{is}}_{\text{is}}$ $\underbrace{\text{almost possible / impossible / possible}}_{\pi}$

4> fuzzy usually qualification

$U(x) = f$

Usually X is f.

⟹ Formation of Rules.

if anticedant then consequent.

This consists of

1> Assignment statements.
2> conditional statements.
3> Unconditional statements.

→ Assignment statements.

Eg: y = small
: color = red.
: Paul is not ~~very~~ tall and not very short

These statements usually use '=' operator.

→ Conditional statements

If A then B.

If A then B
else C.

If A_1 then
If A_2 then B
else C. } nested

Unconditional statements.

Eg: goto S
: stop
: Divide a by b.

* Rules can be either simple or compound.

28/10/2017 Sat.

Compound rules contain contain connectives like AND, OR...

⇒ Decomposition of compound rules.

* Multiple conjunctive antecedents.

If x is $\underset{\sim}{A_1}$ AND $\underset{\sim}{A_2}$ AND ... $\underset{\sim}{A_n}$ then y is $\underset{\sim}{B_n}$

Let $\underset{\sim}{A_m} = \underset{\sim}{A_1} \cap \underset{\sim}{A_2} \cap \ldots \cap \underset{\sim}{A_n}$

$$\mu_{\underset{\sim}{A_m}}(x) = \min \left[\mu_{\underset{\sim}{A_1}}(x), \mu_{A_2}(x) \ldots - \mu_{\underset{\sim}{A_n}} \right]$$

⇒ If x in A_m then y in B_n.

$$A_m \Rightarrow B_n.$$

$$\neg A_m \cup B_n$$

* Multiple disjunctive antecedants.

If x is $\underset{\sim}{A_1}$ OR x is $\underset{\sim}{A_2}$ OR x is $\underset{\sim}{A_n}$ then y is $\underset{\sim}{B_n}$

Let $\underset{\sim}{A_m} = \underset{\sim}{A_1} \cup \underset{\sim}{A_2} \cup \ldots \underset{\sim}{A_n}$

$$\mu_{\underset{\sim}{A_m}}(x) = \max\left[\mu_{\underset{\sim}{A_1}}(x), \mu_{\underset{\sim}{A_2}}(x) \ldots \mu_{\underset{\sim}{A_n}}\right]$$

If x is $\underset{\sim}{A_m}$ then y is $\underset{\sim}{B_n}$

$$A_m \Rightarrow B_n.$$

$$\neg A_m \cup B_n$$

* ELSE and UNLESS

If $\underset{\sim}{A_1}$ then B_1, ELSE B_2

$\Downarrow$

If $\underset{\sim}{A_1}$, then $\underset{\sim}{B_1}$

If not $\underset{\sim}{A_1}$ then $\underset{\sim}{B_2}$

UNLESS $\underset{\sim}{A_1}$ then

$\underset{\sim}{B_1}$

ELSE

$\underset{\sim}{B_2}$

$\Downarrow$

If not $\underset{\sim}{A_1}$ then $\underset{\sim}{B_1}$

If $\underset{\sim}{A_1}$, then $\underset{\sim}{B_2}$

Scanned by CamScanner

Nested IF

IF $\underset{\sim}{A_1}$ THEN IF $\underset{\sim}{A_2}$ then $\underset{\sim}{B_1}$

$$\Downarrow$$

If $\underset{\sim}{A_1}$ AND $\underset{\sim}{A_2}$ THEN $\underset{\sim}{B_1}$

Let $\underset{\sim}{A_n} = A_1 \wedge A_2$

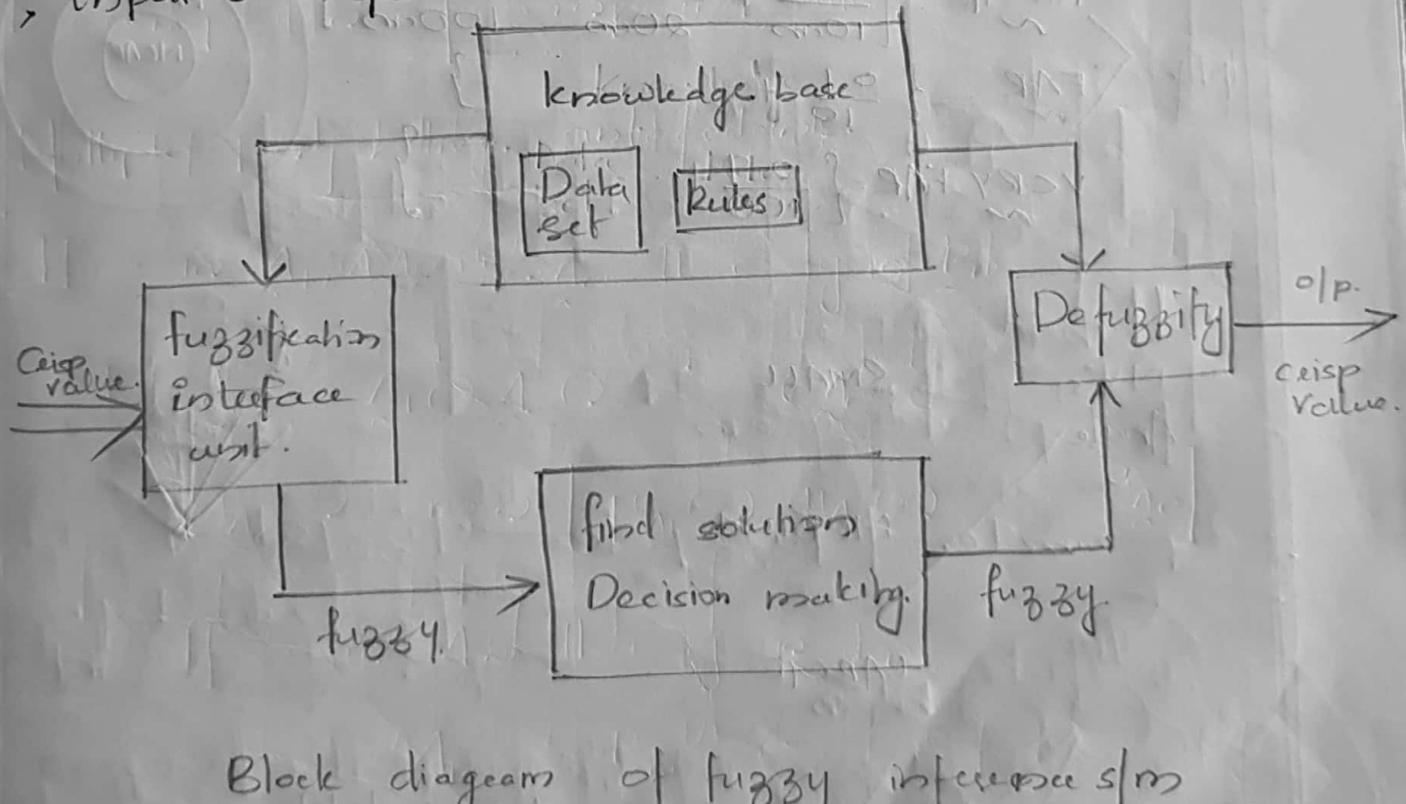
$$\Downarrow$$

If $\underset{\sim}{A_n}$ then $\underset{\sim}{B_1}$

⟹ Fuzzy Inference System: (FIS)

eg. Mamdani (1975)
Sugeno (1985)

- Using fuzzy rules fuzzy inference system finds the solution.
- input is crispset.

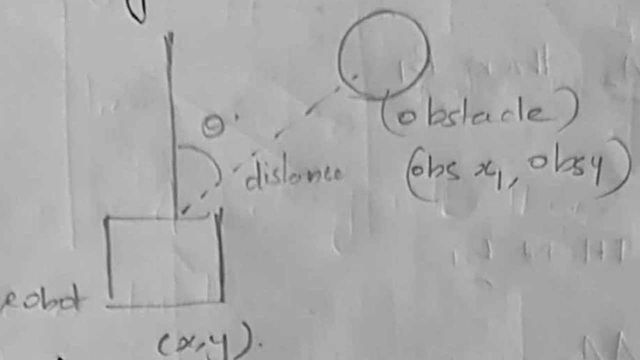


Block diagram of fuzzy inference s/m

- Mamdani. (Intuitionistic, suited for human i/p, wide acceptance)
1975

Step 1: Identifying set of rules.

Robot navigation.



Step 2: Identify fuzzy sets

Rule: can be defined for motion — If the distance from the obstacle is $\underset{\sim}{\text{NEAR}}$ and the angle from the obstacle is $\underset{\sim}{\text{SMALL}}$

THEN turn very sharply.

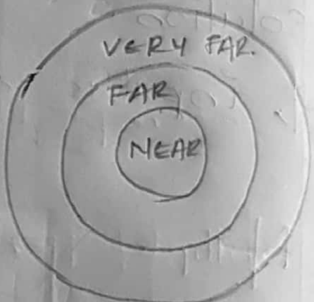
For distance we can define fuzzy sets:

distance:

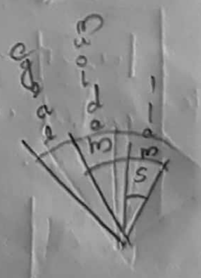
$$\underset{\sim}{NEAR} = \left\{ \frac{0.9}{10cm} + \frac{0.7}{20cm} + \frac{0.1}{100cm} \right\}$$

$$\underset{\sim}{FAR} = \left\{ \frac{0.1}{10} + \cdots \right\}$$

$$\underset{\sim}{VERY\ FAR} = \left\{ \frac{0.9}{100} + \cdots \right\}$$



Angle:
- SMALL
- MEDIUM
- LARGE



The rules can be represented in a table

	Near	far	very far.
Small	Very sharp	sharp turn.	Medium turn.
Medium	Sharp turn	medium turn	mild turn
Large	medium turn	mild turn	No turn

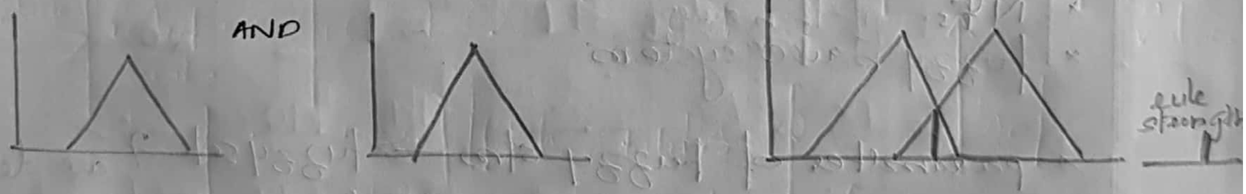
fuzzy system (steering)

	Near	far	very far.
Small	very slow	slow speed	fast
medium	slow speed	fast	very fast
large	fast	very fast	Top speed

speed adjustment.

30/10/2017 Mon.

Step 3: Find rule strength



Step 4: Find consequence of the rule.

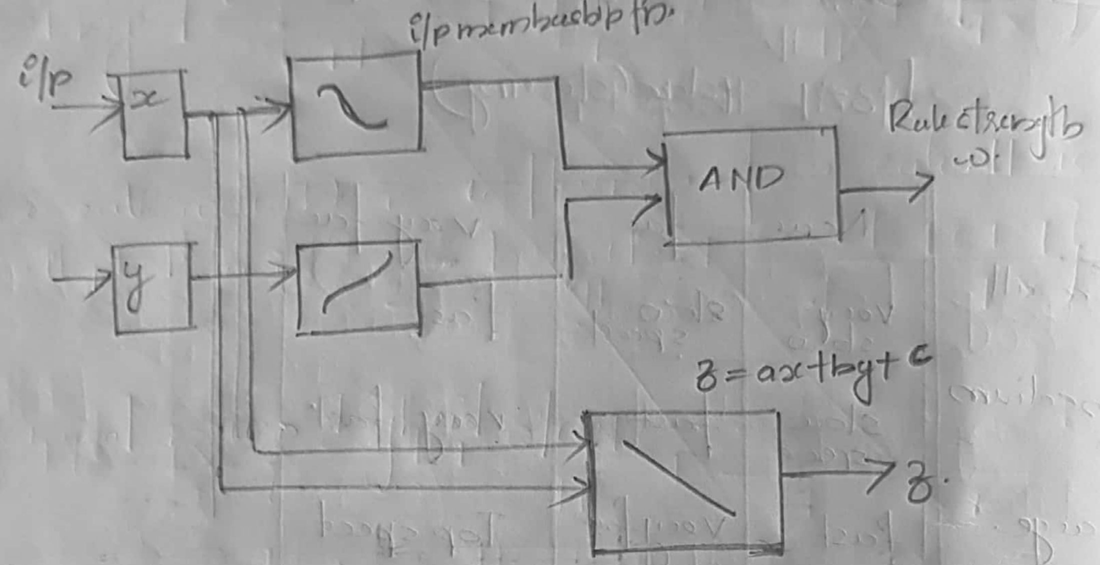
Step 5: Defuzzification

Sugeno system (1985).

→ Linear s/m.

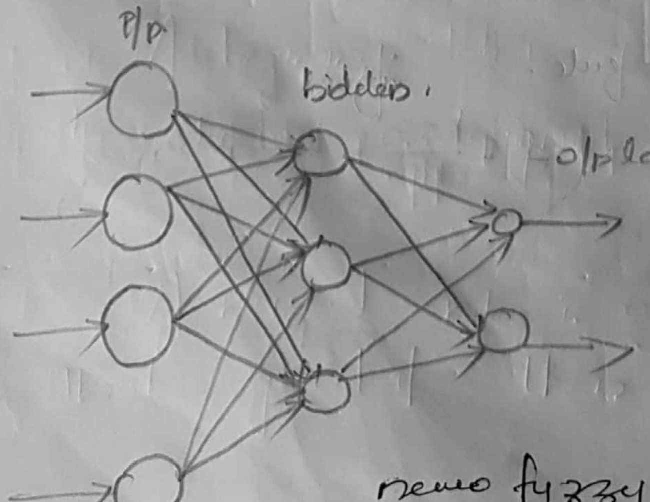
If x is A and y is B then $z = f(x, y)$.

→ Mathematical.



⇒ Neuro fuzzy system:

* Combination of neural n/w and fuzzy.
* Hybrid
* NFs.
* fuzzy neuro system.
* parameters of fuzzy s/m → fuzzy set. } are determined
  → fuzzy rules } by neural n/w.



neuro fuzzy s/m.

→ It is a 3 layered feed forward neural network.

→ First layer corresponds to input variables

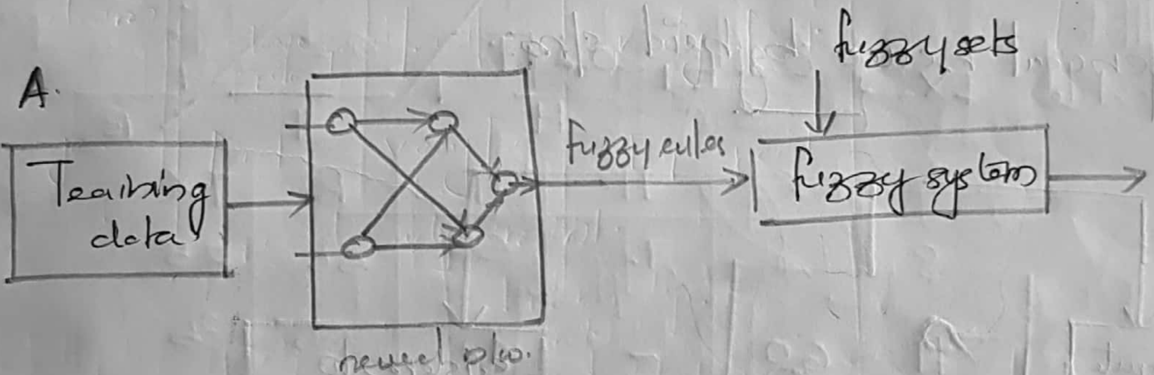
→ Second & third layer corresponds to fuzzy rules and output variables.

→ Some systems may have upto 5 layers.

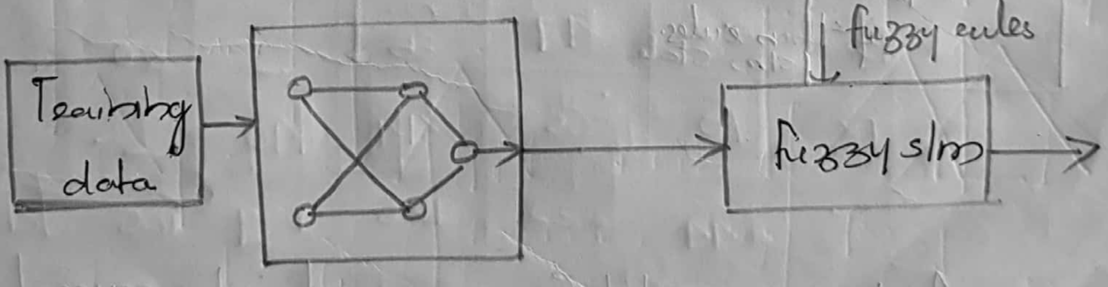
→ NFs
- Cooperative NFs.
- General NFs.

→ Cooperative NFs

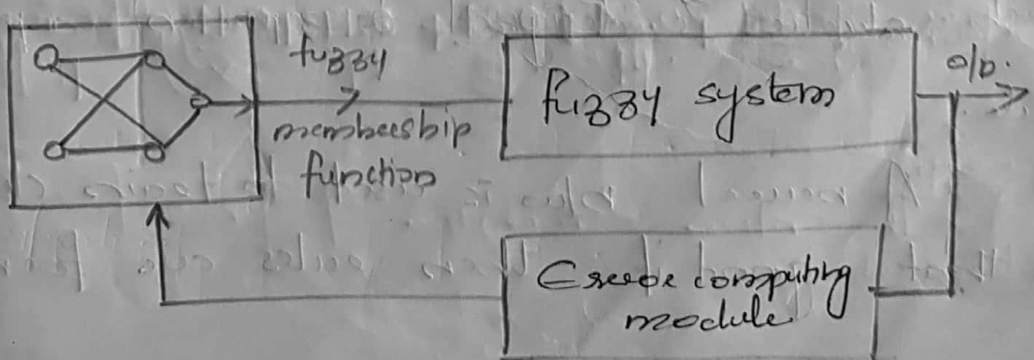
A.



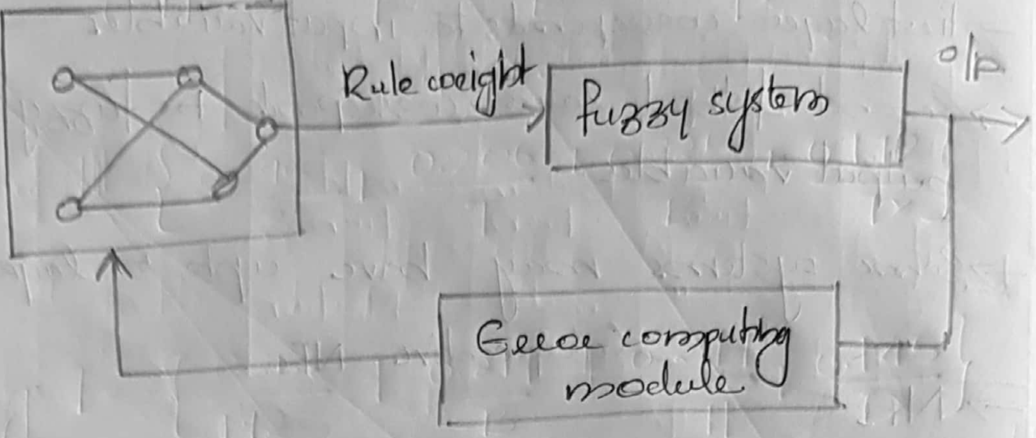
B.



C

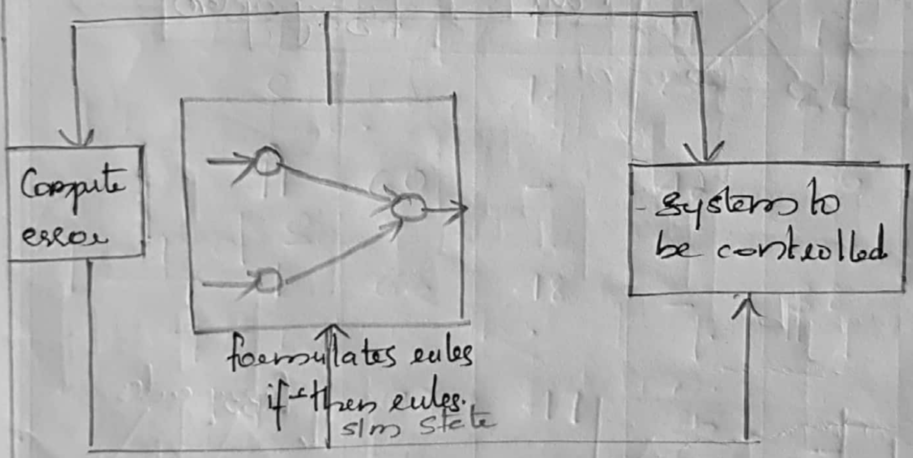


D.



The rule weight is influence of a rule. It is multiplied with rule output.

→ General neuro hybrid s/m:



A training data is grouped into several clusters. Each cluster is designed to represent a particular rule.

A neural n/w is used to train clusters so that good if-then rules are formed.

Every output unit will give an antecedant of the rule. This is used to control the s/m.

The o/p of the s/m (s/m state) is fed back.

31/10/2017 Tue.

The following data was determined by the pairwise comparison of work differences of people. When it was compared with software (S) 72% polled prefered hardware (H). 65 of them required teaching (T). 55 of them prefered Business (B) and 25 prefered Textile (TX). Comparison with b/w the preferences were

60-S

42-T

66-B

35-TX

When compared with teaching

62-S 38-B

48-H 25-TX

Comparison with Business

52-S 47-H 35-T 20-TX

Compared with textile

70-S 65-H ~~44~~-T 40-B

Use rank ordering method & plot the membership function for the most prefered work.

Software $S = \left\{ \frac{0.33}{H} + \frac{0.299}{T} + \frac{0.25}{B} + \frac{0.115}{TX} \right\}$ 217

Hardware $H = \left\{ \frac{0.295}{S} + \frac{0.206}{T} + \frac{0.325}{B} + \frac{0.172}{TX} \right\}$ 208

Teaching $T = \left\{ \frac{0.358}{S} + \frac{0.219}{B} + \frac{0.277}{H} + \frac{0.145}{TX} \right\}$ 173

Business $B = \left\{ \frac{0.338}{S} + \frac{0.305}{H} + \frac{0.227}{T} + \frac{0.129}{TX} \right\}$ 154

Textile $TX = \left\{ \frac{0.319}{S} + \frac{0.297}{H} + \frac{0.2}{T} + \frac{0.18}{B} \right\}$ 219

	S	H	T	B	TX
S	–	72	65	55	25
H	60	–	42	66	35
T	62	48	–	38	25
B	52	47	35	–	20
TX	70	65	44	40	–
	244	232	186	199	105

Total = 966

244
1232

Rank ordering

Preferences $= \left\{ \frac{0.25}{S} + \frac{0.24}{H} + \frac{0.21}{B} + \frac{0.193}{T} + \frac{0.11}{TX} \right\}$ S/co is more preferred.

2> Using inference method design membership functions for

i> quadrilateral

ii> Trapezoid

iii> Parallelograms.

Ans: i> Quadrilateral.

$$\mu_{(Q)} = \frac{p+q+r+s}{360}$$

$=$

Consider two fuzzy sets A and B both defined on X. Find

a> $(\overline{A})_{0.7}$ b> $(B)_{0.2}$ c> $(A \cup B)_{0.6}$ d> $(A \cap B)_{0.5}$

e> $(A \cup \overline{A})_{0.7}$ f> $(B \cap \overline{B})_{0.3}$ g> $(\overline{A \cap B})_{0.6}$ h> $(\overline{A} \cup \overline{B})_{0.8}$

$\mu(x)$	x_1	x_2	x_3	x_4	x_5
$\underset{\sim}{A}$	0.2	0.3	0.4	0.7	0.1
$\underset{\sim}{B}$	0.4	0.5	0.6	0.8	0.9

Scanned by CamScanner

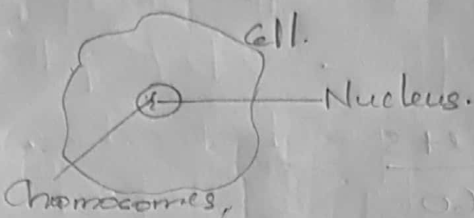
Module 6

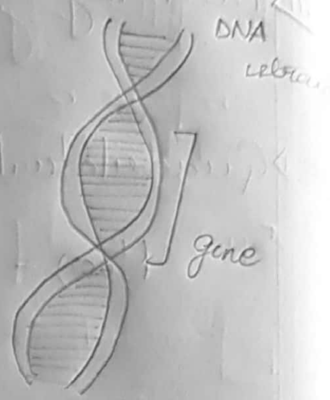
Genetic Algorithm

→ 'Survival of fittest'

- Cell:

  Basic building block.





23 pair of chromosomes

Chromosome / String / Individual.

Allele: combination of a gene which defines a particular ppty.

Eg: eye color
↓
black, brown, grey

Gene pool

Gene pool is the set of all possible alleles in a particular population.

Genome is the set of all possible genes in a particular species.

Locus:

position of a gene

Genotype:

The entire combination of genes for a particular individual.

gene 1	gene 2	...	gene N

genotype

converted to physical ppty

factor 1	factor 2	...	factor N.

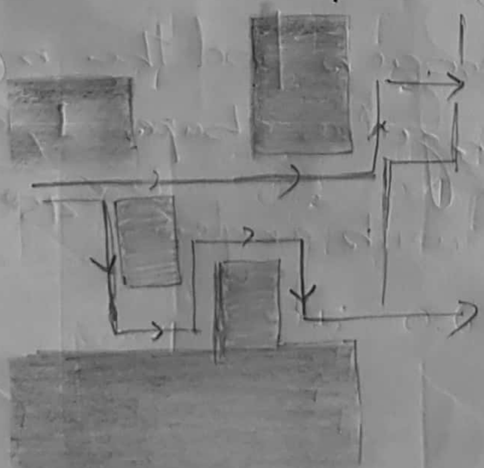
phenotype.

→ Mapping of terms.

Natural evolution	Genetic Algorithm.
* Chromosome	* String.
* Gene	* Feature
* Allele	* Feature value
* Locus	* String position
* Genotype	* Coded string.
* Phenotype	* Parameter set.

In genetic algorithm: new chromosomes are created by (1) mutating / (2) crossover of old chromosomes

Each chromosome represent solution to a problem.



ignore ignore

chrom1: sol 1: (straight) (straight) (straight) (straight) (straight) (left) (right)

chrom2: sol 2: (straight) (right) (left) (left) (right) (right) (r

fitness function = no of turns

$$f(X) = 2$$

$$f(Y) = 6.$$

Assignment
Submit on 13/11

Consider two membership functions

$$\mu_{\tilde{A}}(x) = \frac{|60 - x|}{80} \qquad \mu_{\tilde{B}}(x) = \frac{|40 - x|}{80}$$

Take suitable values of x and find

a) $\tilde{A} \cup \tilde{B}$

b) $\tilde{A} \cap \tilde{B}$.

c) $\tilde{A} / \tilde{B}$

d) $\tilde{B} / \tilde{A}$

2) Using inference approach find the membership values for the triangular shapes I, R, E, IR and T. for a triangle with angles. 45°, 55° and 80°.

chrom 1: sol 1: (straight) (straight) (straight) (straight) (straight) (left) (right)

chrom 2: sol 2: (straight) (right) (left) (left) (right) (right) (left)

fitness function = no of turns

$f(X) = 2$

$f(Y) = 6.$

Assignment Submit on 13/11

Consider two membership functions

$$\mu_{\underset{\sim}{A}}(x) = \frac{|60 - x|}{80} \qquad \mu_{\underset{\sim}{B}}(x) = \frac{|40 - x|}{80}$$

Take suitable values of x and find

a) $\underset{\sim}{A} \cup \underset{\sim}{B}$

b) $\underset{\sim}{A} \cap \underset{\sim}{B}$.

c) $\underset{\sim}{A} / \underset{\sim}{B}$

d) $\underset{\sim}{B} / \underset{\sim}{A}$

2) Using inference approach find the membership values for the triangular shapes I, R, E, IR and T. for a triangle with angles. 45°, 55° and 80°.

(right)
(left)

3) Using neuron intuition and definition of universe of discourse ~~quote~~ plot fuzzy membership function for weight of people.

7/11/2017 Tue

Genetic Algorithm:

Set of chromosomes - population.

Initial population
Evolve population
- ↳ selection of two individuals.
- ↳ Reproduce
- ↳ Evaluation.
- ↳ Replacement.

⇒ Steps in Genetic Algorithm

Step 1: Generate initial population randomly

2: Evolve population

2.a: Selection.
Select individuals for reproduction. The best ones are chosen rather than the poor ones. This is based on the relative fitness of individuals.

2.b: Reproduction.

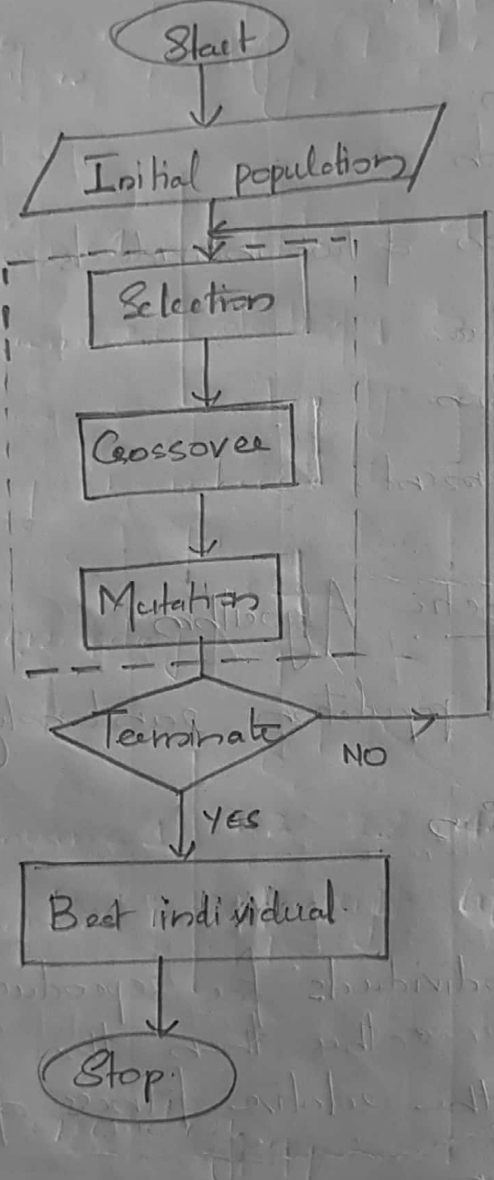
For generating new chromosomes crossover and mutation can be applied.

2.c: Evaluation
fitness of new chromosomes are evaluated

2(d): Replacement

Individuals from old population are killed & replaced by new ones.

Algorithm is stopped when population converges to optimal solution.



Operators in Genetic Algorithm.

1> Encoding → creation of chromosome/string
2> Selection
3> Crossover
4> Mutation

1> Encoding

→ Encoding schemes:-

chromosome length should be same

<i> Binary encoding:

1010	0001	1011	1111
gene1	gene2	....	geneN.

Every bit string is a solution but may not be the best solution.

<ii> Octal encoding.

String is made of numbers: 0-7

chromosome1	0 3 4 6 7 2 1 6
chromosome2.	1 5 7 2 3 0 6 4

<iii> Hexadecimal encoding.

String consists of numbers : 0-9

chromosome 1	9CE7
chromosome 2	3DBA

(iv) Permutation encoding (Real number coding)

chromosome 1	1 5 3 2 4 6 9
chromosome 2.	8 5 3 1 0 2 1.

Integer numbers.

(v) Value encoding

chromosome 1	1.2	2.3	4.5	6.8.
chromosome 2.	left	right	up	right

<vi> Tree encoding

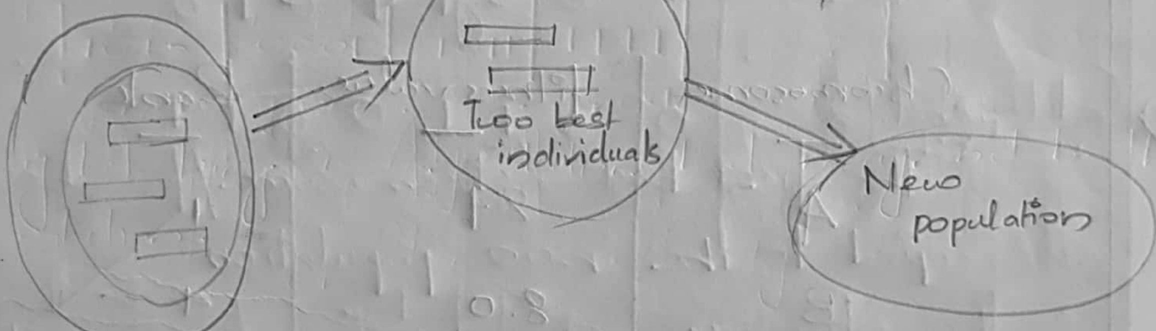
Every chromosome is a tree

<2> Selection

Selection is a process of choosing new parents from the population for crossing, ie, parents is selected for reproduction. According to Darwin's theory of evolution the best ones survive to create new offspring. Higher the fitness function the better chance that an individual will be selected.

→ **Selection Pressure:**

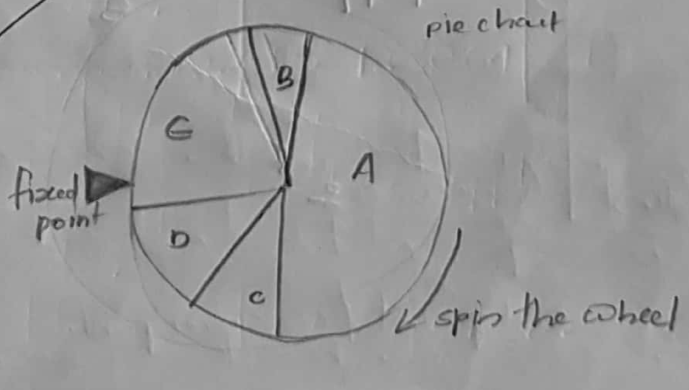
The degree to which best individuals are favoured. This will drive the GA to improve the population fitness over successive generations.



Selection

Selection mechanisms:

<i> Roulette wheel selection



chromosome	fitness
A	8.5
B	1.2
C	2.3
D	3.5
G	5.1

8/11/2019 Thu.

Stochastic universal sampling:

* More fixed points.
* That many parents are selected in a single spin.

<ii> Random Selection:-

* Parent is randomly selected.

<iii> Rank Selection

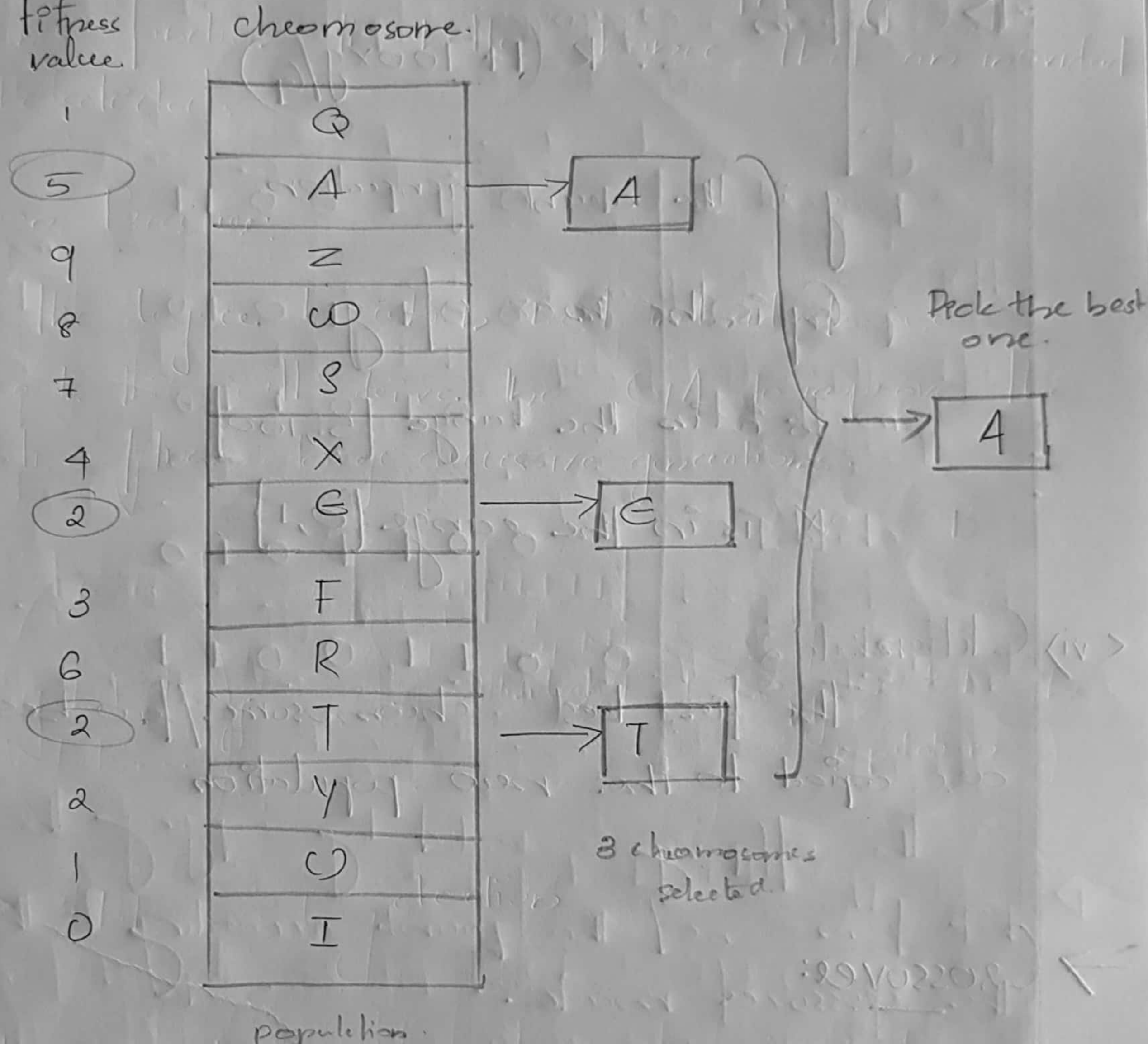
* Based on rank selection.
* Chromosomes vary slightly in fitness.

Chromosome	fitness value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

parents.

<iv> Tournament Selection:

Population ⟹ Randomly select k chromosome ⟹ Select the Best.



(v) Boltzmann Selection

f_{max} be the current maximum value

If $f(x_i) > f_{max}$,

select $f(x_i)$.

otherwise use Boltzmann probability

$$P = \exp\left[-\{f_{max} - f(x_i)/T\}\right]$$

where $T = T_0 (1-\alpha)^k$

$k = (1 + 100 \times g/G)$

g is the current generation.

G is the max of g.

T_0 is in the range $[5, 100]$

α is in the range $[0, 1]$.

(vi) Elitism:

The first best chromosome / few best chromosomes are copied to the new population

→ Crossover:

Process of producing child chromosomes from parent chromosomes.

1) Single Point Crossover:-

cut point / crossover point

	head		tail
Parent 1	1 0 1 1 0		0 1 0
Parent 2	1 0 1 0 1		1 1 1

crossover point

child 1	1 0 1 1 0		1 1 1
child 2	1 0 1 0 1		0 1 0

Two strings are cut once at corresponding point and the section after the cut are exchanged.

2> Two point crossover.

Two cutpoints / crossover

	head		tail
Parent 1	1 1	0 1 1	0 1 0
Parent 2	0 1	1 0 1	1 0 0

middle section is swapped.

swap

child 1	1 1	1 0 1	0 1 0
child 2	0 1	0 1 1	1 0 0

Two crossover points are chosen and the contents between these points are exchanged between the two parents. One advantage is that if both head and tail of a chromosome containing good genetic information, both are carried to next generation. This is not possible in single point crossover

3> Multi point crossover (multiple cut points).

There can be even/odd number of crossover points.

4> Uniform crossover:

parent 1	1 0 1 1 0 0 1 1
parent 2	0 0 0 1 1 0 1 0
mask bit	1 1 0 1 0 1 1 0
child 1	1 0 0 1 1 0 1 0
swap parents. child 2	0 0 1 1 0 0 1 1

mask bit
1 – Take from parent 1
0 – Take from parent 2

The offspring is created by copying the corresponding bit from one/other parent according to the bit in the crossover mask.

Crossover mask is having the same length as that of the chromosome and is randomly generated.

If there is a '1' in the crossover mask the gene is copied from parent 1 and when there is a '0' in the crossover mask gene is copied from second parent

13/11/17

5> Three Parent Crossover:

Parent 1	1	1	0	1	0	0	0	1
Parent 2	0	1	1	0	1	0	0	1
Parent 3	0	1	1	0	1	1	0	0
Child	0	1	1	0	1	0	0	1

Three parents are randomly chosen.
Each bit of first parent is compared with corresponding bit of second parent both are same, then just copy to child the same bit otherwise bit from parent 3 is taken.

6> Crossover with reduced surrogate.

Crossover point occurs when gene values differ.

Parent 1	1	0	0	1	0	1	0	
Parent 2	1	1	0	0	1	1	0	
child	1	1	0	0	0	1	0	
child 2	1	0	0	1	1	1	0	

head & tail the same.

7> Shuffle crossover:

Before swapping, shuffle. A single crossover position is selected but before they are exchanged they are randomly shuffled in both the parents after recombination, the variables in offspring are not shuffled.

Scanned by CamScanner

Crossover (   )

8. Precedence preservation crossover

P_0	3	~~2~~	2	2	3	~~1~~	~~1~~	~~1~~	3
P_1	1	1	~~3~~	~~2~~	2	~~1~~	2	3	3
b	0	0	1	1	1	1	0	0	0
offspring	3	2	1	1	2	1	2	2	3

If $b_i = 0$ select from P_0

if $b_i = 1$ select from P_1

temp bit string b, used to define order in which genes are drawn in P_0 and P_1. $b_i = 0$, take bit from P_0 and append it into offspring. Delete the bit in P_1. $b_i = 1$, take the bit from P_1 and append it into offspring. Delete same bit in P_0

9) Ordered Crossover

A set of consecutive bits are copied from first parent to the child these bits are cut from the second parent and the rest is copied.

parent 1	4	2	1	3	6	5	
parent 2.	2	3	1	4	5	6.	copied
child.	4	2	3	1	6	5	

parent 1	8	4	7	3	6	2	5	1	9	0
parent 2.	0	~~1~~	~~2~~	~~3~~	4	~~5~~	~~6~~	7	8	9.
child	0	4	7	3	6	2	5	1	8	9

10) Partially matched crossover

It is used in a case where string consists of order in which cities are visited by a travelling person. Two crossover points selected. The section b/w them copied to offspring.

swap

swap.

P_1	1	5	2	8	7	4	3	6
P_2	4	2	5	8	1	8	6	7
child 1	1	2	5	8	1	4	3	6
child 2.	4	5	2	8	7	8	6	7

Since 2 is already there in offspring swap b/w 2 and 5.

Since 7 is already there in offspring, swap b/w 1 and 7.

20/11/2017 Mon

→ Crossover Probability (P_c)

0% → no crossover

100% → all are created by crossover.

→ Mutation:

1) flipping:

1 → 0

0 → 1

Scanned by CamScanner

Eg: $C_1 \Rightarrow$ 1 0 0 1 1 0 1 (flip, flip)

mutation crossover $\Rightarrow$ 1 0 0 0 1 0 0 → if 1 → change/flip the bit of C_1 else copy C_1.

mutated crossover 0 0 0 1 0 0 1

2> Interchanging

Select two random positions and interchange the bits.

Eg:

C_1 = 1 0 $\boxed{1}$ 0 1 $\boxed{0}$ 1 1 0 1 (interchanging)

mutated: 1 0 0 0 1 1 1 1 0 1

In mutation only one parent crossover

3> Reversing:

Randomly select a position and reverse the bits after the selected bit position.

C_1 = 1 1 1 1 $\boxed{0}$ 1 0 0 (reverse)

mutated C_1 : 1 1 1 1 $\boxed{0}$ 0 0 1

→ Mutation probability (P_m)

How well mutation is done.

⟹ Stopping conditions for GA:

→ When there is no change in fitness value.

→ After some elapse of time.

→ After some number of generations.

→ There is no change in objective function. $f(x) = x^2$

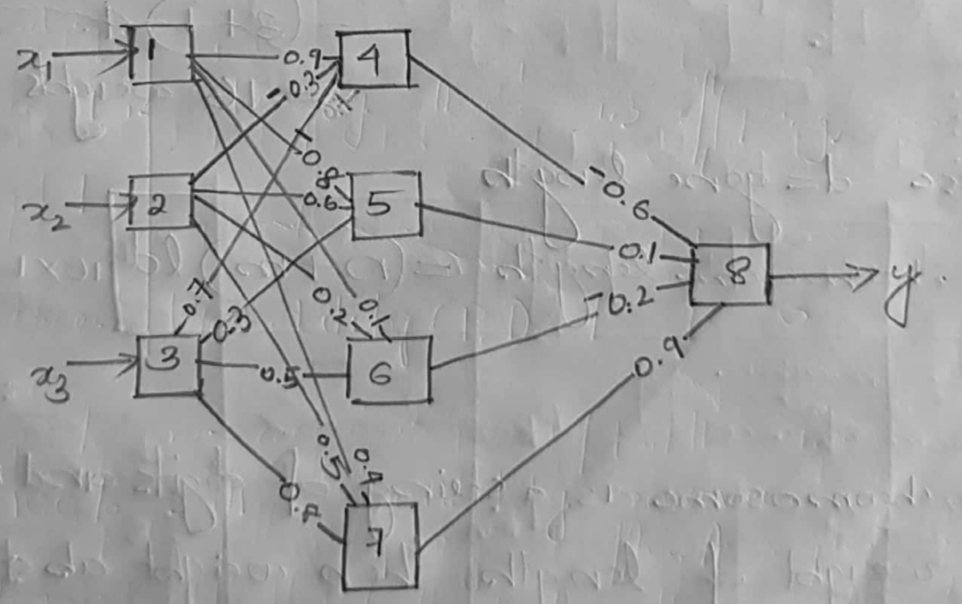
→ There is no improvement in objective function.

⟹ Hybrid Systems:

→ Neuro genetic system.

Parameters of the neural network is set with the help of GA.

<A> Encoding weights.



Creating Chromosome

0.9	−0.3	0.7	−0.8	0.6	0.3	0.1	0.2	0.5	0.4	0.5	0.8

gene 0

−0.6	0.1	−0.2	0.9

gene 15

Now the chromosome is obtained we can do operations like crossover.

$n \rightarrow$ input units

$l \rightarrow$ hidden units

$m \rightarrow$ output units

no of genes = no of weights

$$\boxed{\text{Total number of weights} = (n+m)\,l}$$

$$= (3+1)\,4$$

$$= 16 \text{ weights (from the eg)}$$

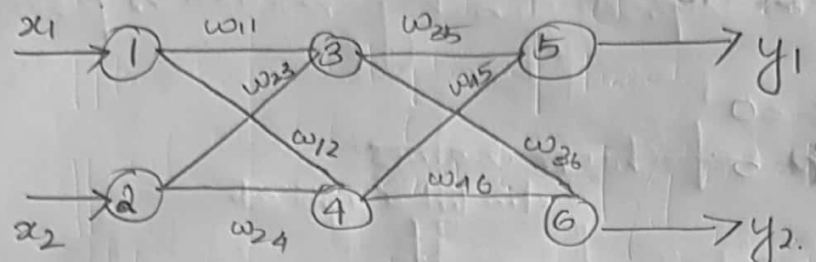
Suppose d = gene length

$$\boxed{\text{Total chromosome length} = (n+m)\,ld} \quad 16 \times L = 16$$

d bits

Note: If the chromosome is a string of digits and not the weight of length then weight abstraction is to be done

< B > Weight Extraction:



$n = 2$

$l = 2$

$m = 2$.

No of weights $= (2+2)2 = 8$ weights.

Number of genes $= 8$

let, d = gene length = 5.

Total length of chromosome $= 8 \times 5 = \underline{\underline{40}}$

$x_1 \to x_4$. range 0→9.

Eg:

gene $k=0$	gene $k=1$	gene $k=2$	gene $k=3$	gene $k=4$	gene $k=5$	gene $k=6$	gene $k=7$
84821	46234	78901	82104	426189	63421	46421	87649

↓ ↓ x_1 x_2 … x_{40}

Weight (w_k) of $gene_k$ is given by.

$$w_k = \begin{cases} \dfrac{+x_{kd+2}10^{d-2} + x_{kd+3}10^{d-3} \dots\dots x_{(k+1)d}}{10^{d-2}} & \text{if } 5 \le x_{kd+1} \le 9 \\[2ex] \dfrac{-x_{kd+2}10^{d-2} + x_{kd+3}10^{d-3} \dots\dots x_{(k+1)d}}{10^{d-2}} & \text{if } 0 \le x_{kd+1} < 5 \end{cases}$$

$$\omega_k = \frac{-x_{kd+2}\,10^{d-2} + x_{kd+3}\,10^{d-3} + \cdots x_{(k+1)d}}{10^{d-2}} \quad \text{if } 0 \le x_{kd+1} < 5$$

find weights:

⇒ gene 0, (84321) $k=0$, $d=5$

$x_{kd+1} = x_1 = 8$. (2nd formula)

$$\omega_0 = \frac{4\times10^3 + 3\times10^2 + 2\times10^1 + 1\times10^0}{10^3} = \underline{4.321}$$

⇒ gene 1, (46234) $k=1$, $d=5$

$x_{kd+1} = x_6 = \underline{4}$. (1st formula)

$$\omega_1 = \frac{-6\times10^3 + 2\times10^2 + 3\times10^1 + 4\times10^0}{10^3} = \underline{-5.766}$$

gene k=2.

⋮

Applications of fuzzy systems:

Washing machine
vaccum cleaner
oven.

(c) Fitness value:

$$= \frac{1}{E}$$

E = root mean square error

Considering the previous example:-

there are 2 inputs and 2 outputs.

Input-target

$(I_{11} \ I_{21})$ $(T_{11} \ T_{21})$

$(I_{12} \ I_{22})$ $(T_{12} \ T_{22})$

$(I_{13} \ I_{23})$ $(T_{13} \ T_{23})$

Actual output

$$\begin{pmatrix} O_{11} & O_{21} \\ O_{12} & O_{22} \\ O_{13} & O_{23} \end{pmatrix}$$

$$E_1 = (T_{11} - O_{11})^2 + (T_{21} - O_{21})^2$$

$$E_2 = (T_{12} - O_{12})^2 + (T_{22} - O_{22})^2$$

$$E_3 = (T_{13} - O_{13})^2 + (T_{23} - O_{23})^2.$$

$$\boxed{E = \sqrt{\frac{E_1 + E_2 + E_3}{3}}}$$

$$\boxed{\text{fitness} = \frac{1}{E}}$$

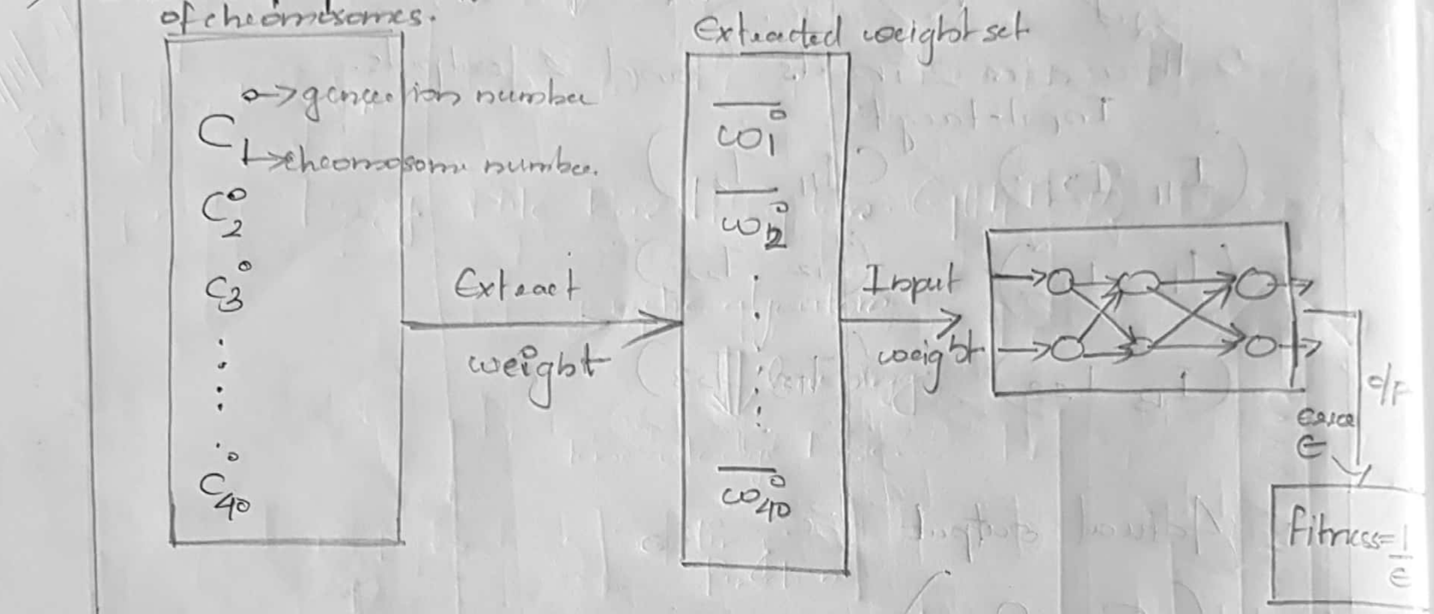
Chromosome length $= \underbrace{8}_{\text{no of gene}} \times \underbrace{5}_{\text{gene length}} = 40$

∴ Initial generation has 40 chromosomes.

Then we apply the fitness function and select the fittest ones:

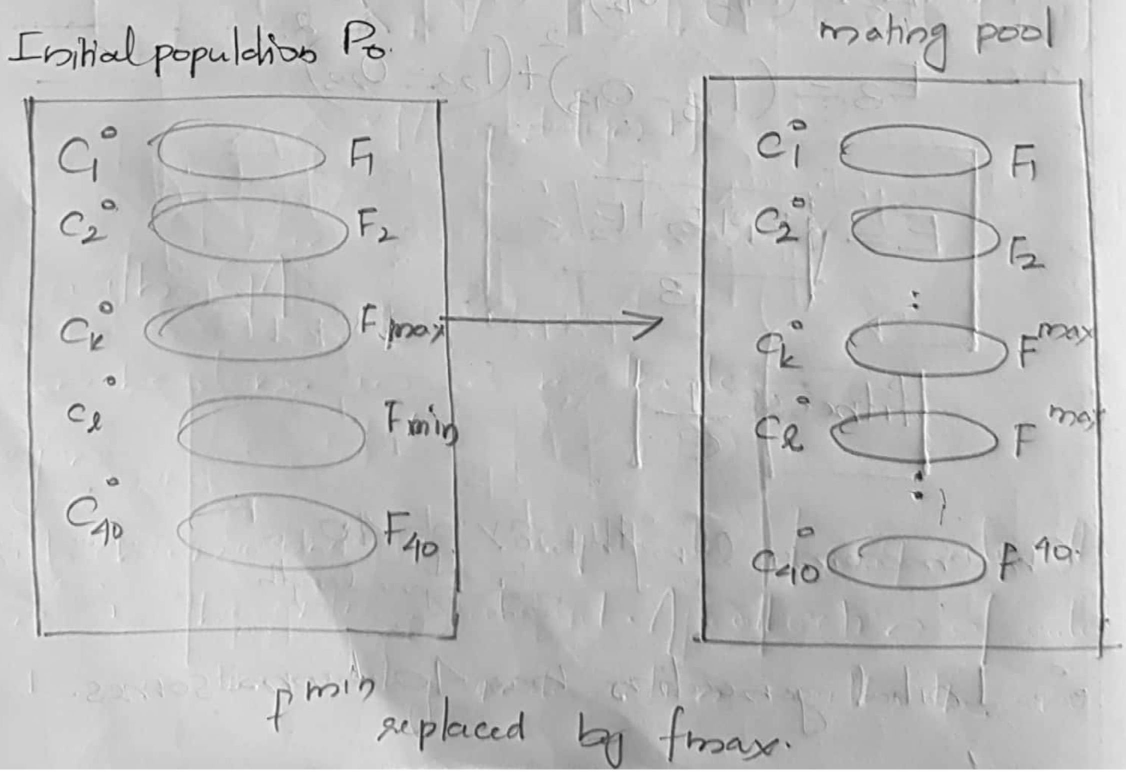
Then we perform reproduction by mutation / crossover.

→ Initial population



hybrid s/m – neuro genetic s/m.

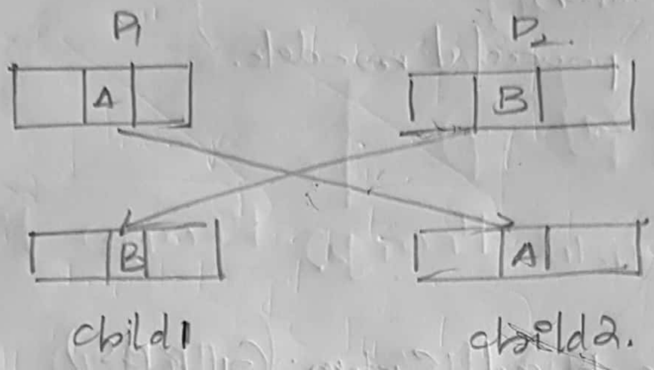
<1> Reproduction of offspring:



F^{min} replaced by fmax.

~~Initial generation has 40 chromosomes.~~

Crossover / mutation.

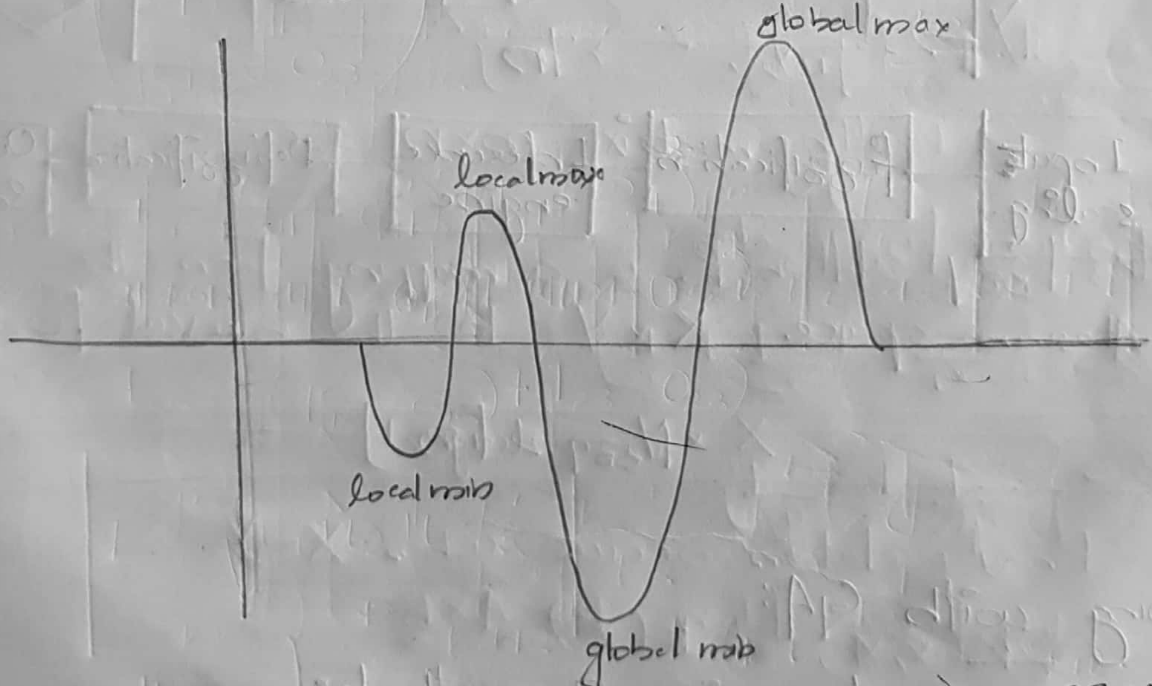


two point crossover

→ Convergence:

A population is said to have converged when 95% of the individuals share the same fitness value.

Global optimum : max / min value out of all (max / min)



The population evolves over successive generation if fitness value increasing towards global optimum (max / min)

Advantages:

1> GA performs optimization of neural n/w parameters.

2> Better fitness calculation.

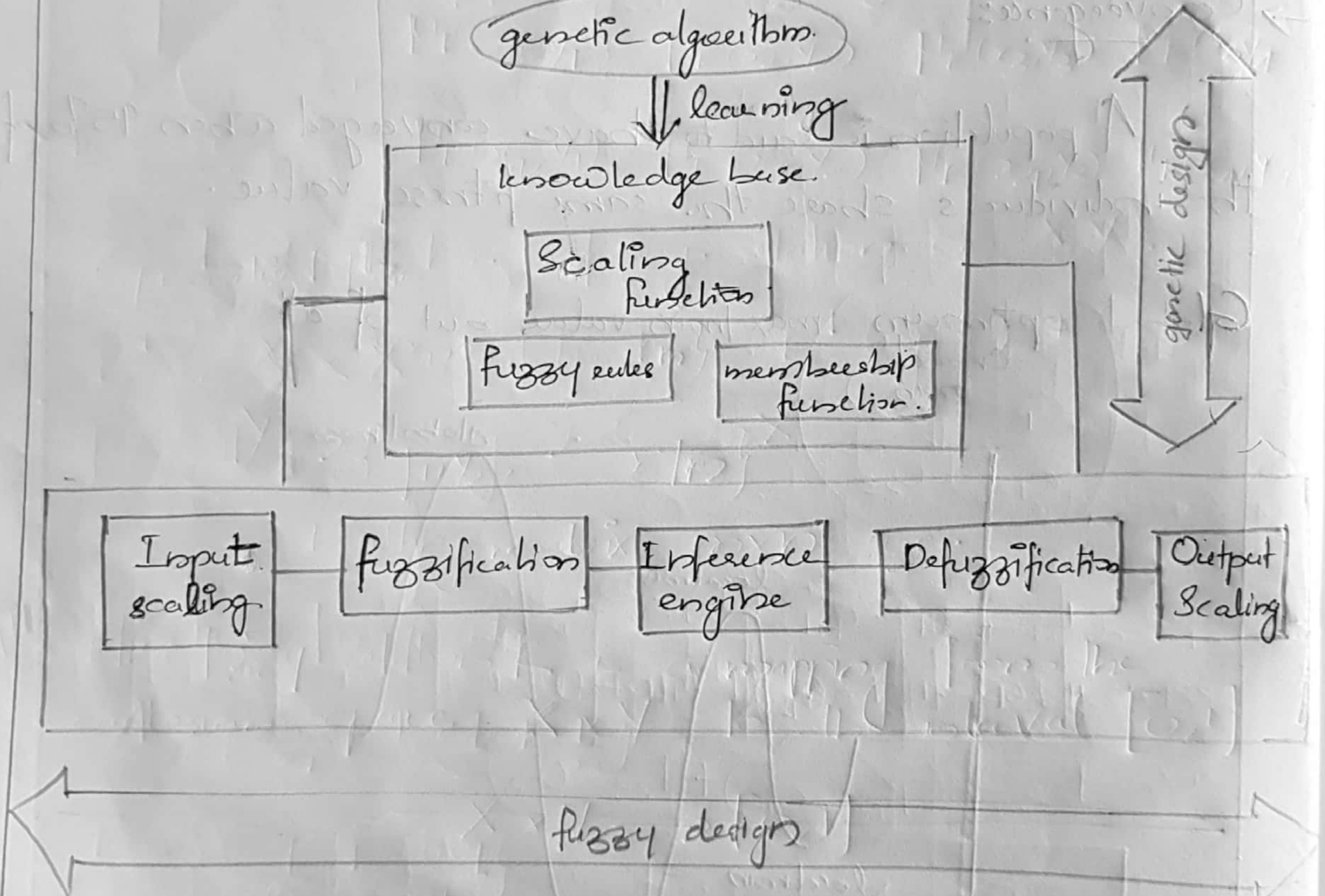
3> Hybrid systems are powerful models.

21/11/2017 Tue

Hybrid Systems

FIS fuzzy inference s/m

Genetic fuzzy Rule Based System: (GFRBS)



→ learning with GA:

can be done in 3 methods:-

1> Michigan:

Each chromosome is a separate fuzzy rule. The population is the entire rule set. A collection of rules ~~have~~ are modified on time by interaction with the environment

2> Pittsburg:

Each chromosome encodes the entire rule base or knowledge base. ~~Ch~~ Crossover will lead to new combination of rules and mutation provides new rules.

3> Iterative rule learning:

Each chromosome is an individual rule. A new rule is adapted and added to the rule set in an iterative fashion in every run of the GA

→ Tuning Scaling function

Here the fuzzy membership functions are normalized by scaling function. applied to input and output variables. The universe of discourse is also scaled

There are two types:

1> Linear scaling

2> Non-linear scaling.

In linear scaling, the scaling function has a single parameter

In non-linear scaling, scaling function has more parameters.

21/11/17 Tue

Module 3 - fuzzy relation:

Consider two universe of discourses X and Y

$X \rightarrow \{x_1, x_2, x_3, x_4\}$

$Y \rightarrow \{y_1, y_2, y_3, y_4\}$.

$$R(X,Y) = \begin{bmatrix} \mu_R(x_1,y_1) & \mu_R(x_1,y_2) & \mu_R(x_1,y_3) & \mu_R(x_1,y_4) \\ \mu_R(x_2,y_1) & \mu_R(x_2,y_2) & \mu_R(x_2,y_3) & \mu_R(x_2,y_4) \\ \mu_R(x_3,y_1) & \mu_R(x_3,y_2) & \mu_R(x_3,y_3) & \mu_R(x_3,y_4) \\ \mu_R(x_4,y_1) & \mu_R(x_4,y_2) & \mu_R(x_4,y_3) & \mu_R(x_4,y_4) \end{bmatrix}$$

Generalised form

$X \rightarrow \{x_1, x_2, \ldots x_m\}$

$Y \rightarrow \{y_1, y_2, \ldots, y_n\}$

Resulting matrix $m \times n$ size.

The fuzzy relation R is a mapping from the cartesian space $X \times Y$ to the interval $[0,1]$

$$X \times Y \longrightarrow [0,1]$$

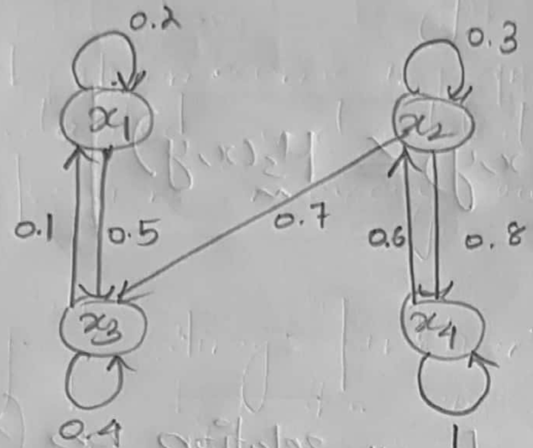
Binary relation occurs when $X = Y$.

$\Rightarrow R(X,X)$ or $R(X^2)$

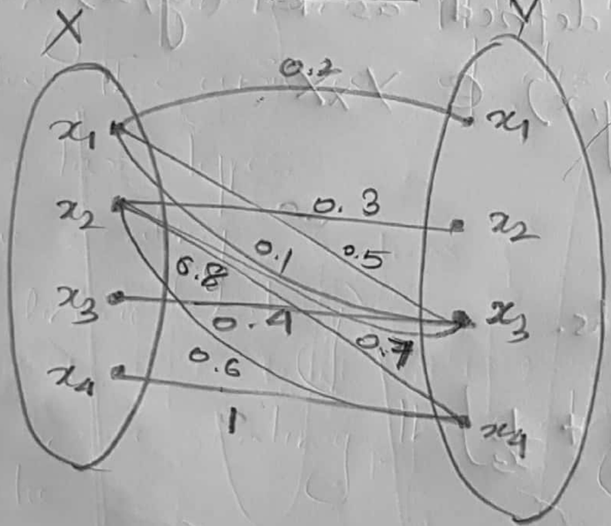
Eg:- $X = \{x_1 \; x_2 \; x_3 \; x_4\}$

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0.2 & 0 & 0.5 & 0 \\ x_2 & 0 & 0.3 & 0.7 & 0.8 \\ x_3 & 0.1 & 0 & 0.4 & 0 \\ x_4 & 0 & 0.6 & 0 & 1 \end{array}$$

fuzzy relation can be represented with a graph.



fuzzy graph



$$R = \left\{ \frac{0.2}{(x_1 y_3)} + \frac{0.4}{(x_1 y_2)} + \cdots\cdots \right\}$$

⇒ Operations on fuzzy Relation:

<1> Union

$$\boxed{\mu_{\underset{\sim}{R} \cup \underset{\sim}{S}}(x,y) = \max\left[\mu_R(x,y),\ \mu_B(x,y)\right]}$$

<2> Intersection

$$\boxed{\mu_{\underset{\sim}{R} \cap \underset{\sim}{S}}(x,y) = \min\left[\mu_{\underset{\sim}{R}}(x,y), \mu_S(x,y)\right].}$$

<3> Complement

$$\boxed{\mu_{\overline{\underset{\sim}{R}}}(x,y) = 1 - \mu_{\underset{\sim}{R}}(x,y).}$$

<4> Containment

$$\boxed{R \subseteq S \Rightarrow \mu_{\underset{\sim}{R}}(x,y) \leq \mu_{\underset{\sim}{S}}(x,y)}$$

<5> Inverse R^{-1}

Let R be a fuzzy relation on (X, Y):

R^{-1} on $Y \times X$ is defined as $R^{-1}(y,x) = R(x,y)$

for all pair $(y,x) \in Y \times X$

23/11/17 Given two fuzzy sets.

$$\underset{\sim}{A} = \left\{ \frac{0.3}{x_1} + \frac{0.7}{x_2} + \frac{1}{x_3} \right\}$$

$$\underset{\sim}{B} = \left\{ \frac{0.4}{y_1} + \frac{0.9}{y_2} \right\}$$

Create fuzzy relation $R = \underset{\sim}{A} \times \underset{\sim}{B}$

Ans: $R = A \times B = \begin{bmatrix} \mu_R(x_1,y_1) & \mu_R(x_1,y_2) \\ \mu_R(x_2, y_1) & \mu_R(x_2,y_2) \\ \mu_R(x_3, y_1) & \mu_R(x_3,y_2) \end{bmatrix}$

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

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

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

$$\mu_R(x_2, y_2) = \min[0.7, 0.9]$$
$$= \underline{\underline{0.7}}$$

$$\mu_R(x_3, y_1) = \min[1, 0.4]$$
$$= \underline{\underline{0.4}}$$

$$\mu_R(x_3, y_2) = \min[1, 0.9]$$
$$= \underline{0.9}$$

$$\therefore A \times B = \begin{bmatrix} 0.3 & 0.3 \\ 0.4 & 0.7 \\ 0.4 & 0.9 \end{bmatrix}$$