APJ ABDUL KALA TECHNOLOGICALUNIVERSITY

FIFTH SEMESTER B.TECH DEGREE MODEL EXAMINAT NOVEMBER 2017

Department: **Computer Science and Engineering**

Subject: - **CS361**: ***Soft Computing***

Time: 3 hour

PART A

Answer all questions

Max. Marks: 100

Compare supervised ,unsupervised and Reinforcement Learn
----

ing

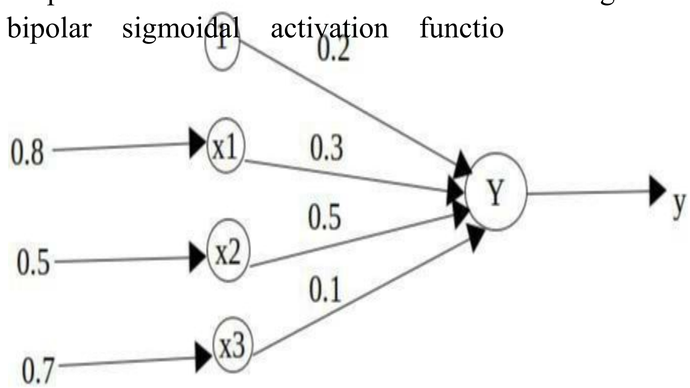
and

1.

2. Calculate the output of neuron Y for the net shown in below Figure . Use binary

bipolar sigmoidal activation functio

ns.



3. Delta ru i Adaline?

a.

(3)

What is (1

b. Write a short note on the Architecture of Back -Propagation Network? (1

1/2) **Total:**

4. Draw the flowchart for perceptron network training algorithm?

(3)

Answer any two questions

(12)

5. a) Exp

PART B

full

McCulloch-Pitts Neuron Model?

(4)

b) Implement AND function using McCulloch-Pitts neuron (consider binary data)

(5)

6. a) Construct a feed-forward network with four input nodes, two hidden nodes and three

output nodes ?

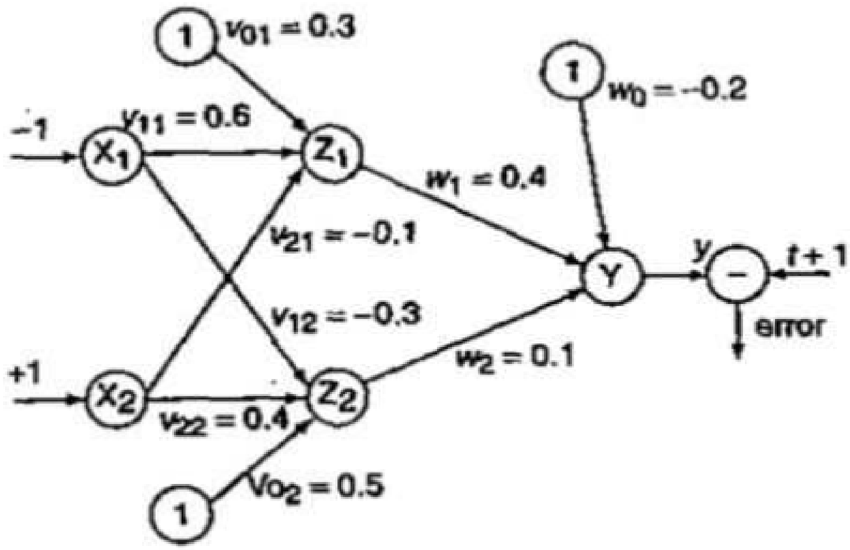
(4) b)Design a Hebb net to implement OR function(consider bipolar inputs and targets).

(5)

7. Fin the new weights, using back-
The network is presented with the input pattern l-1, 1] and the target output is + 1. Use a

learning rate of a = 0.25 and bipolar sigmoidal activation function.

(9)



Total: (18)

PART C

Answer all questions

8. Lis out an three operations on fu sets?

(3)

9.Distinguish inverse and projection operations in Fu relations?

(3)

10.Highlight the maj difference between Core, Support and Bounda

(3)

11.Differentiate Intuition and rank ordering methods of membership value assignments?

(3)

Total: (12)

PART

Answer any tw ful questions

12. a)Mention the properties of Fu sets?

(4)

b)The disc membership functions for a transist and a resistor are given below:

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

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

Find the following: (a) Algebraic sum; (b) algebraic product; (c) bounded sum; (d) bounded difference.

13. Design a computer software to perform image processing to locate objects within a scene. The two fuzzy sets representing a plane and a train image

$$\underset{\sim}{\text{Plane}} = \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

$$\underset{\sim}{\text{Train}} = \left\{ \frac{1}{\text{train}} + \frac{0.2}{\text{bike}} + \frac{0.4}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.2}{\text{house}} \right\}$$

Find the following:

(a) $\underset{\sim}{\text{Plane}} \cup \underset{\sim}{\text{Train}}$; (b) $\underset{\sim}{\text{Plane}} \cap \underset{\sim}{\text{Train}}$;

(c) $\overline{\underset{\sim}{\text{Plane}}}$; (d) $\overline{\underset{\sim}{\text{Train}}}$;

(e) $\underset{\sim}{\text{Plane}} | \underset{\sim}{\text{Train}}$; (f) $\overline{\underset{\sim}{\text{Plane}} \cup \underset{\sim}{\text{Train}}}$;

(g) $\overline{\underset{\sim}{\text{Plane}} \cap \underset{\sim}{\text{Train}}}$; (h) $\underset{\sim}{\text{Plane}} \cup \overline{\underset{\sim}{\text{Plane}}}$;

(i) $\underset{\sim}{\text{Plane}} \cap \overline{\underset{\sim}{\text{Plane}}}$; (j) $\underset{\sim}{\text{Train}} \cup \overline{\underset{\sim}{\text{Train}}}$;

(9)

14. Consider the two fuz sets

$$\underset{\sim}{A} = \left\{ \frac{0}{0.2} + \frac{0.8}{0.4} + \frac{1}{0.6} \right\}$$

$$\text{and} \quad \underset{\sim}{B} = \left\{ \frac{0.9}{0.2} + \frac{0.7}{0.4} + \frac{0.3}{0.6} \right\}$$

Using Zadeh's notations, express the fuzzy sets into $\lambda - cut$ sets for $\lambda = 0.4$ and $\lambda = 0.7$ for the following operations:

(a) $\underset{\sim}{A}$; (b) $\underset{\sim}{B}$; (c) $\underset{\sim}{A} \cup \underset{\sim}{B}$;

(d) $\underset{\sim}{A} \cap \underset{\sim}{B}$; (e) $\overline{\underset{\sim}{A}} \cup \overline{\underset{\sim}{B}}$; (f) $\overline{\underset{\sim}{A}} \cap \overline{\underset{\sim}{B}}$

(9)

Total: (18)

PART

Answer any four full questions

15.Define Mamdani FIS model?

16.Describe all Fuzzy Propositions? (10)

17.Explain about Cooperative Neural Fuzzy Systems? (10)

18.Compare Genetic Learning of Rule Ba and Genetic Learning of Knowledge base?

19.Explain different type of Encoding Techniques (10)

20.a)Mention the stopping condition for genetic Algorith Flow? (5)

b)Disting between single- and Two-point Crossover? (5)

APJ ABDUL KALA TECHNOLOGICA UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE MODEL EXAMINAT NOVEMBER 201

Department: **Computer Science and Engineering**

Subject: - **CS361**: ***Soft Computing***

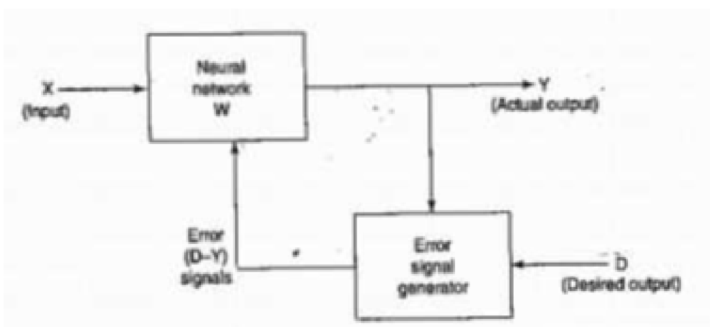
Time: 3 hour   Max. Marks: 100

PART A

Answer all questions

1. **Supervised Learning**

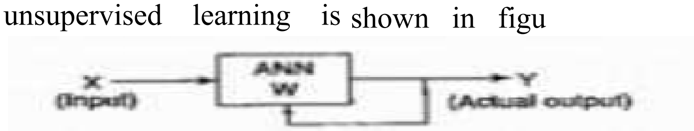
The learning here is performed with the help of a teacher.ANNs following the supervised learning, each input vector requires a corresponding target vector, which represents the desired output.The input vector along with the target vector is called training pair.The network here is informed precisely about what should be emitted as output. The block diagram of unsupervised learning is shown in figu



(1)

Unsupervised Learning

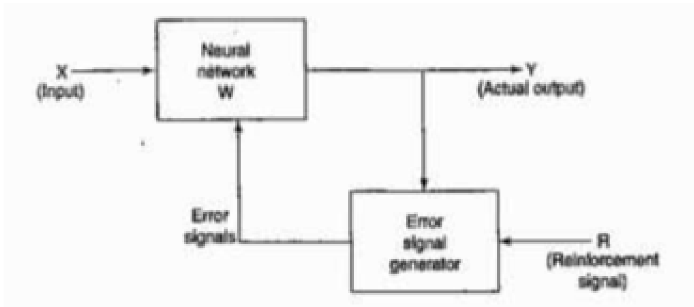
The learning here is performed without the help of a teacher.In ANNs following unsupervised learning, the input vectors of similar type are grouped without the use of training data to specify how a member of each group looks or to which group a number belongs. In the training process,the network receives the input patterns and organizes these patterns to form clusters. When a new input pattern is applied, the neural network gives an output response indicating the class to which the input pattern belongs. If for an input, a pattern class cannot be found then a new class is generated.The block diagram of unsupervised learning is shown in figu



(1)

Reinforcement Learning

This learning process is similar to supervised learning. In the case of supervised learning, the correct target output values are known for each input pattern. But, in some cases, less information might be available. For example, the network might be told chat its actual output is only "50% correct" or so. Thus, here only critic information is available, not the exact information. The learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal. The block diagram of unsupervised learning is shown in figu



2. The given network has three input neurons with bias and one output neuron. These form a single-layer network. The inputs are given as [x1,x2,x3] = [0.8,0.5,0.7] and the weights are [w1, w2, w3] = [0.3,0.5, 0.1] b = 0.2 input always 1)

The net input to the output neuron is

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

(n = because onl 3 input neurons are given)

= 0.2 + 0.8 X 0.3+ 0.5 0.7 X 0.1

= 0.2+ 0.24 + 0.25 + 0.07 = 0.76

(i) For binary sigmoidal activation functi

$y = f(y_{in}) = \frac{1}{(1+e^{-y_{in}})} = 1/1 + e^{-0.76} = 0.68135$

(ii) For bipolar sigmoidal activation fun

$y = f(y_{in}) = (\frac{2}{(1+e^{-y_{in}})}) - 1 = (2/(1 + e^{-0.76})) - 1 = 0.3627$

(3)

3. a) The delta rule is derived from the gradient Descent method.The gradient Descent method approach continues forever, converging only asymptotically to the solution.The delta rule updates the weights between the connections so as to minimize the difference between the net input to the output unit and the target value. The major aim is to minimize the error over all training patterns. This is done by reducing the error for each pattern, one at a time.

The delta rule for adjusting the weight of ith pattern (i = 1 to n) is

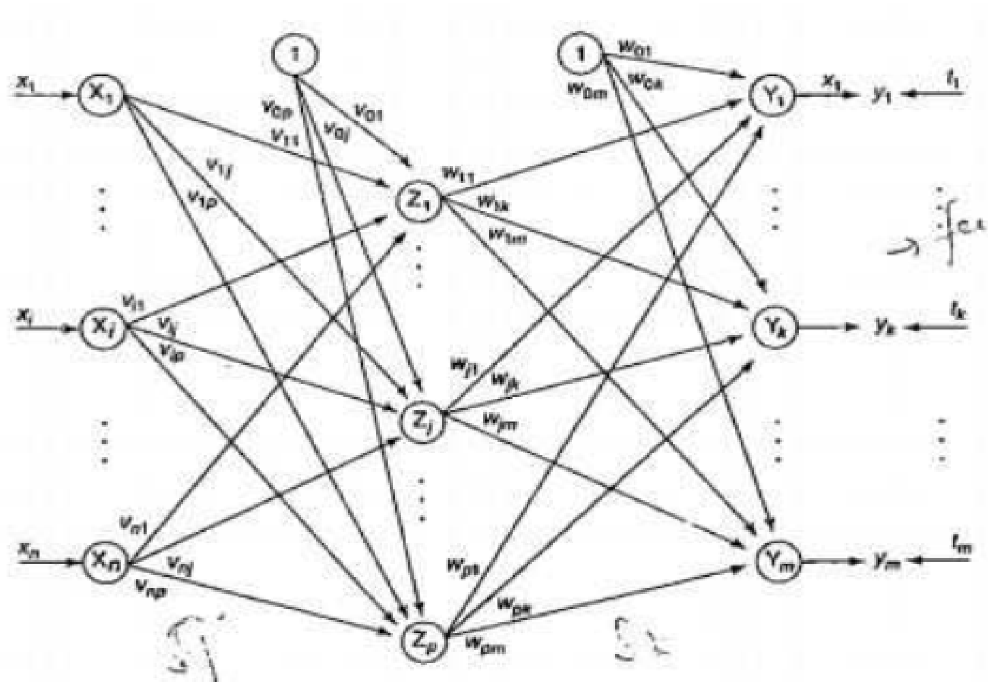
$\Delta w_i = \alpha(t - y_{in})x_i$

where Δw_i is the weight change; α the learning rate; x the vector of activation of input unit; y_{in} the net input to output unit, $Y = \sum_{i=1}^{n} x_i w_i$,t the target output. The deha rule in case of several output units for adjusting the weight from ith input unit to the jth output unit (for each pattern) is

$\Delta w_{ij} = \alpha(t_j - y_{inj})x_i$

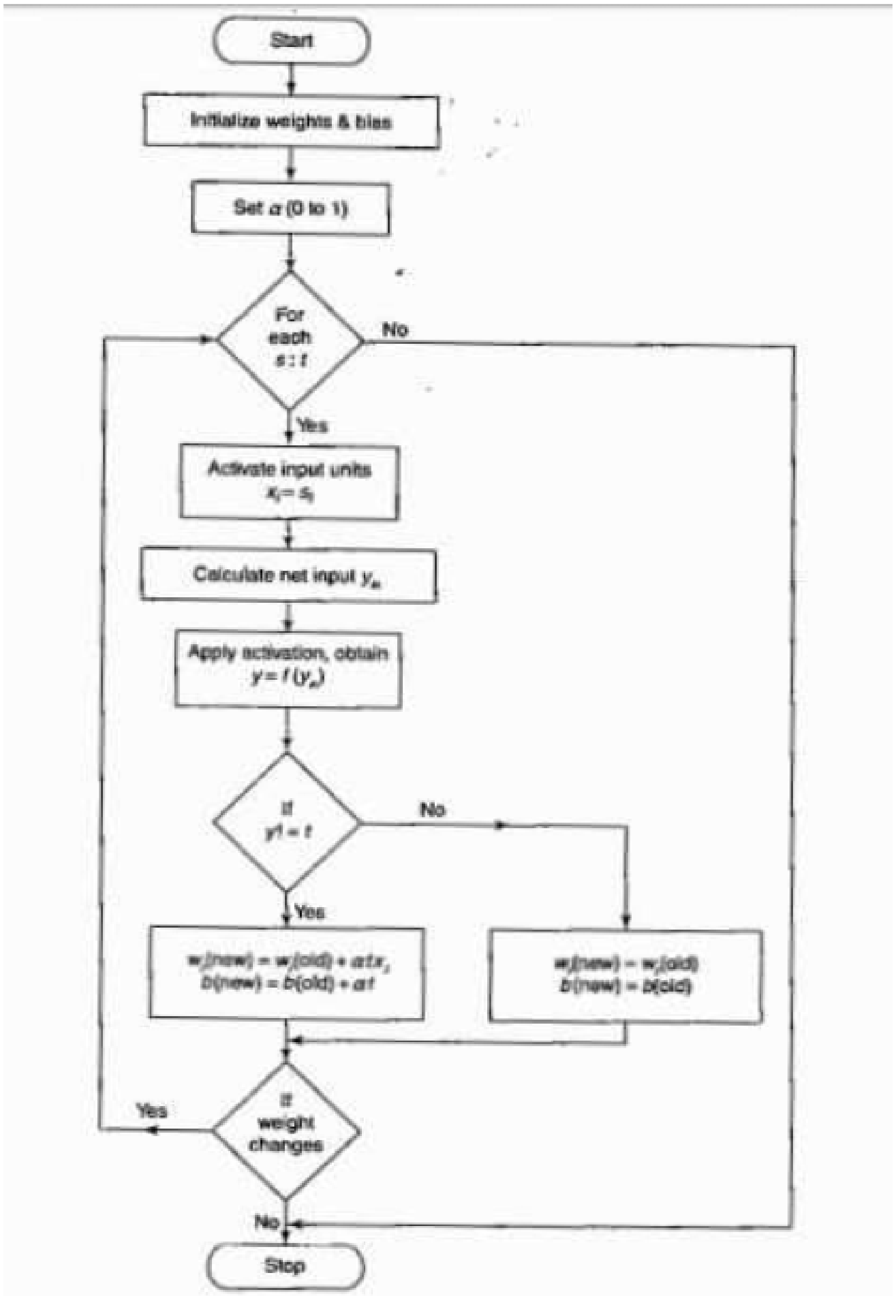
b) A back-propagation neural network is a multilayer, neural network consisting of an input layer, a hidden layer and an output layer. The neurons present in che hidden and output layers have biases, which are the connections from the unit whose activation is always 1. The bias terms also acts as weights.

The inputs sent to the BPN and the output obtained from the net could be either binary (0, I) or bipolar (-1, + 1). The activation function could be any function which increases monotonically and is also differentiable.



x_1
X_1
x_i
X_i
x_n
X_n
1
v_{0p}
v_{11}
v_{0j}
v_{01}
v_{1j}
v_{1p}
v_{i1}
v_{ij}
v_{ip}
v_{n1}
v_{nj}
v_{np}
Z_1
Z_j
Z_p
w_{11}
w_{1k}
w_{1m}
w_{j1}
w_{jk}
w_{jm}
w_{p1}
w_{pk}
w_{pm}
1
w_{01}
w_{0k}
w_{0m}
Y_1
Y_k
Y_m
y_1
y_k
y_m
t_1
t_k
t_m

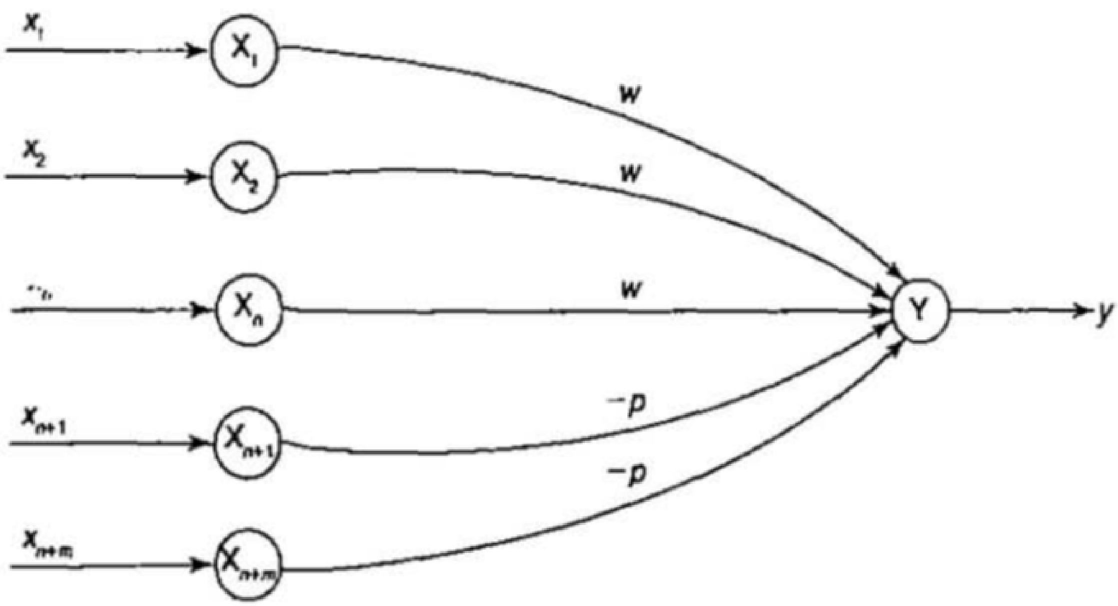
4. Flowchart



.

PART B

5. a) The M-P neuron has both excitator and inhibitory connections. I excitatory weighted (w > 0) or inhibitory with weight p(p < 0). In Figure , inputs fro x1 to Xn possess excitatory weighted connections and inpu fro Xn+ 1 Xn+m posses inhibitor with weighted interconnections.



Since the firi of the output neuron is based upon the threshold, the activation func here is defined a

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

For inhibition to be absolute, the threshold with the activation function should satisfy the following condition:

$$\theta > nw - p$$

The output will f if it receiv say "k or more excitatory inputs but no inhibitory inp

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

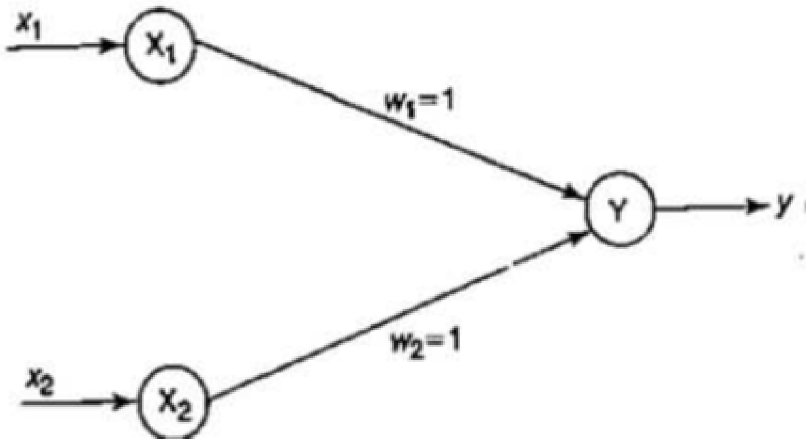
The M- to be performed determine the value no particular training algorithm. An analysis ha set s and the threshold. Here the weig of the weights a simple lo of the neuron a re function

b) along with the threshold to make the neuron perfor m for AND funct

Consider the truth table		
x1	x2	y
1	1	1
1	0	0
0	1	0
0	0	0

IIn McCulloch- onl being performed. Hence, assum che weight analysis is

be WI = 1 w1 = 1. The network architec shown in Figure .



With these assumed weights, the net input is calculated for four inputs: For i

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

For an AND function, the ou high if both the i are high. For this condition, th net input calculated a 2. Hence, based on t net input, the threshold is set, i.e. if threshold value i greater than or equal to 2 then the neuron fires, else it does not fire. the threshold value set equa to 2(θ = 2).This can also be obtained

$$\theta \geqslant nw - p$$

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

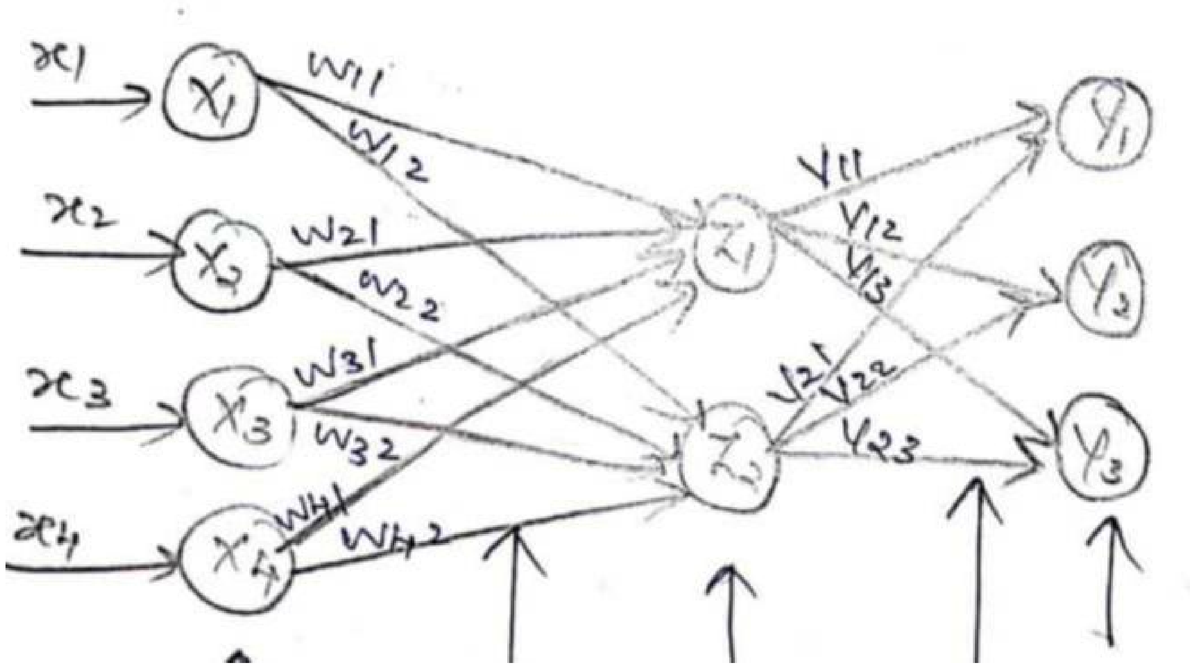
$$\theta \geqslant 2\times 1 - 0 \Rightarrow \theta \geqslant 2$$

Thus, the output of neuron Y can be written as

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

where 2 represents the threshold value.

6 a)



b) The train pair for the OR functi given in Table

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

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

w1=w2=b=0

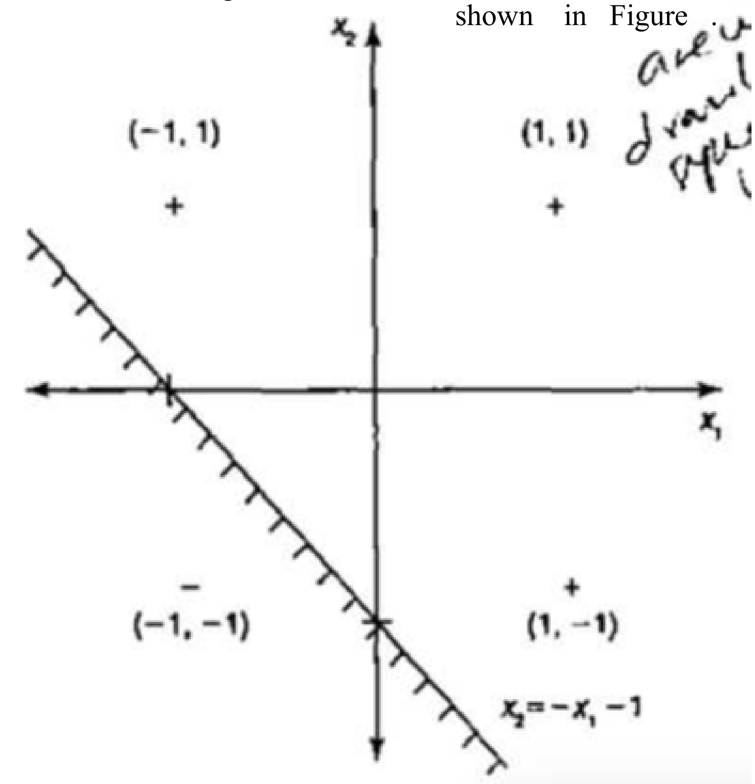
The network trained an the fin weights are out·The weights are considered as final weights if the boundary line obtai fro these weights separates the positiv response region and negative response region.By presenting a the input patterns, the weights are calculated. Table shows the weights calculated for the inputs.

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

Using the fin ine weights, the boundary line equation can be obtained. The separating l ine equation is

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

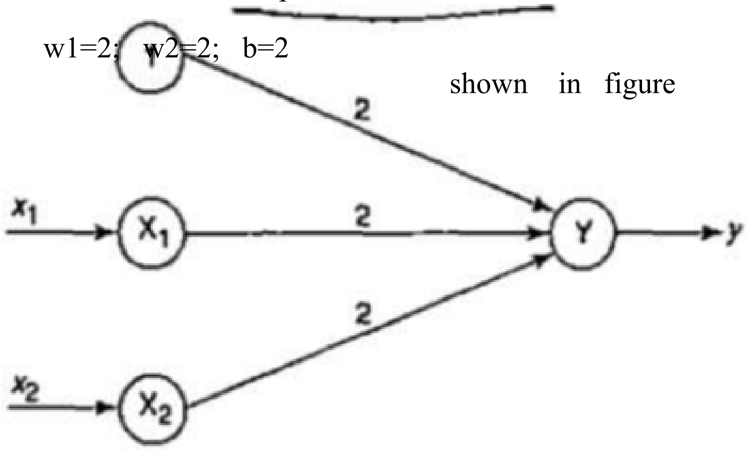
The decision region for net i shown in Figure .



It i observed i Figure that straight line x2 =-x1 - 1 separates the pattern spac into t regions.The input patterns [(1, 1), (1, -1), (-1, 1)] for which the output response is "1" lie o one side of the boundary, and the input pattern (-1, -1) for which the output response is "- the other si of the fi

The network can be represented shown in figure weights are

w1=2; w2=2; b=2



7.

Solution: The initial weights are $[v_{11}\, v_{21}\, v_{01}] = [0.6\ -0.1\ 0.3]$, $[v_{12}\ v_{22}\ v_{02}] = [-0.3\ 0.4\ 0.5]$ and $[w_1\ w_2\ w_0] = [0.4\ 0.1\ -0.2]$, and the learning rate is $\alpha = 0.25$.

Activation function used is binary sigmoidal activation function and is given by

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

Given the input sample $[x_1, x_2] = [-1, 1]$ and target $t = 1$:

- Calculate the net input: For z_1 layer

$$\begin{aligned} z_{in1} &= v_{01} + x_1 v_{11} + x_2 v_{21} \\ &= 0.3 + (-1) \times 0.6 + 1 \times -0.1 = -0.4 \end{aligned}$$

For z_2 layer

$$z_{in2} = v_{02} + x_1 v_{12} + x_2 v_{22}$$
$$= 0.5 + (-1) \times -0.3 + 1 \times 0.4 = 1.2$$

Applying activation to calculate the output, we obtain

$$z_1 = f(z_{in1}) = \frac{1 - e^{-z_{in1}}}{1 + e^{-z_{in1}}} = \frac{1 - e^{0.4}}{1 + e^{0.4}} = -0.1974$$
$$z_2 = f(z_{in2}) = \frac{1 - e^{-z_{in1}}}{1 + e^{-z_{in2}}} = \frac{1 - e^{-1.2}}{1 + e^{-1.2}} = 0.537$$

- Calculate the net input entering the output layer. For y layer

$$y_{in} = w_0 + z_1 w_1 + z_2 w_2$$
$$= -0.2 + (-0.1974) \times 0.4 + 0.537 \times 0.1$$
$$= -0.22526$$

Applying activations to calculate the output, we obtain

$$y = f(y_{in}) = \frac{1 - e^{-y_{in}}}{1 + e^{-y_{in}}} = \frac{1 - e^{0.22526}}{1 + e^{0.22526}} = -0.1122$$

- Compute the error portion δ_k:

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

Now

$$f'(y_{in}) = 0.5[1 + f(y_{in})][1 - f(y_{in})]$$
$$= 0.5[1 - 0.1122][1 + 0.1122] = 0.4937$$

This implies

$$\delta_1 = (1 + 0.1122)\ (0.4937) = 0.5491$$

Find the changes in weights between hidden and output layer:

$$\Delta w_1 = \alpha\delta_1 z_1 = 0.25 \times 0.5491 \times -0.1974$$
$$= -0.0271$$
$$\Delta w_2 = \alpha\delta_1 z_2 = 0.25 \times 0.5491 \times 0.537 = 0.0737$$
$$\Delta w_0 = \alpha\delta_1 = 0.25 \times 0.5491 = 0.1373$$

- Compute the error portion δ_j between input and hidden layer ($j = 1$ to 2):

$$\delta_j = \delta_{inj} f'(z_{inj})$$
$$\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}$$
$$\delta_{inj} = \delta_1 w_{j1} \quad [\because \text{only one output neuron}]$$
$$\Rightarrow \delta_{in1} = \delta_1 w_{11} = 0.5491 \times 0.4 = 0.21964$$
$$\Rightarrow \delta_{in2} = \delta_1 w_{21} = 0.5491 \times 0.1 = 0.05491$$

Error, $\delta_1 = \delta_{in1} f'(z_{in1}) = 0.21964 \times 0.5 \times (1 + 0.1974)(1 - 0.1974) = 0.1056$

Error, $\delta_2 = \delta_{in2} f'(z_{in2}) = 0.05491 \times 0.5 \times (1 - 0.537)(1 + 0.537) = 0.0195$

Now find the changes in weights between input and hidden layer:

$$\Delta v_{11} = \alpha\delta_1 x_1 = 0.25 \times 0.1056 \times -1 = -0.0264$$
$$\Delta v_{21} = \alpha\delta_1 x_2 = 0.25 \times 0.1056 \times 1 = 0.0264$$
$$\Delta v_{01} = \alpha\delta_1 = 0.25 \times 0.1056 = 0.0264$$
$$\Delta v_{12} = \alpha\delta_2 x_1 = 0.25 \times 0.0195 \times -1 = -0.0049$$
$$\Delta v_{22} = \alpha\delta_2 x_2 = 0.25 \times 0.0195 \times 1 = 0.0049$$
$$\Delta v_{02} = \alpha\delta_2 = 0.25 \times 0.0195 = 0.0049$$

- Compute the final weights of the network:

$$v_{11}(\text{new}) = v_{11}(\text{old}) + \Delta v_{11} = 0.6 - 0.0264 = 0.5736$$
$$v_{12}(\text{new}) = v_{12}(\text{old}) + \Delta v_{12} = -0.3 - 0.0049 = -0.3049$$
$$v_{21}(\text{new}) = v_{21}(\text{old}) + \Delta v_{21} = -0.1 + 0.0264 = -0.0736$$
$$v_{22}(\text{new}) = v_{22}(\text{old}) + \Delta v_{22} = 0.4 + 0.0049 = 0.4049$$
$$w_1(\text{new}) = w_1(\text{old}) + \Delta w_1 = 0.4 - 0.0271 = 0.3729$$
$$w_2(\text{new}) = w_2(\text{old}) + \Delta w_2 = 0.1 + 0.0737 = 0.1737$$
$$v_{01}(\text{new}) = v_{01}(\text{old}) + \Delta v_{01} = 0.3 + 0.0264 = 0.3264$$
$$v_{02}(\text{new}) = v_{02}(\text{old}) + \Delta v_{02} = 0.5 + 0.0049 = 0.5049$$
$$w_0(\text{new}) = w_0(\text{old}) + \Delta w_0 = -0.2 + 0.1373 = -0.0627$$

Thus, the final weight has been computed for the network shown in Figure 13.

8.

7.3.1.1 Union

The union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$, denoted by $\underset{\sim}{A} \cup \underset{\sim}{B}$, is defined as

$$\mu_{\underset{\sim}{A} \cup \underset{\sim}{B}}(x) = \max[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)] = \mu_{\underset{\sim}{A}}(x) \vee \mu_{\underset{\sim}{B}}(x) \quad \text{for all } x \in U$$

where $\vee$ indicates max operation. The Venn diagram for union operation of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is shown in Figure 7-10.

7.3.1.2 Intersection

The intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$, denoted by $\underset{\sim}{A} \cap \underset{\sim}{B}$, is defined by

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}}(x) = \min[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)] = \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x) \quad \text{for all } x \in U$$

where $\wedge$ indicates min operator. The Venn diagram for intersection operation of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is shown in Figure 7-11.

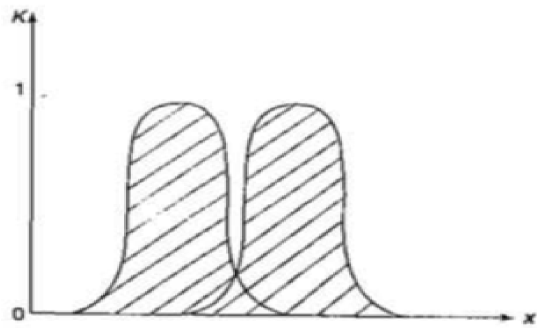


Figure 7-10 Union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

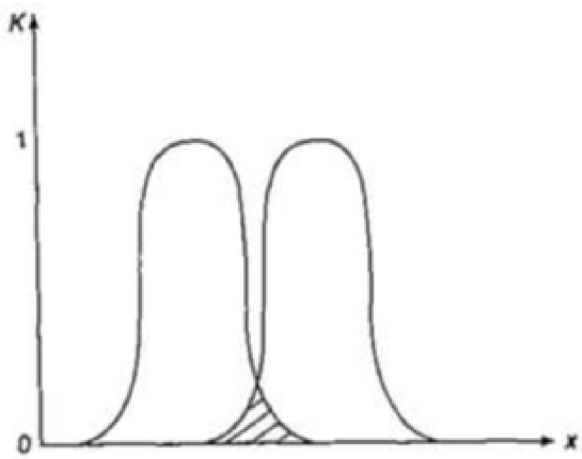


Figure 7-11 Intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

7.3.1.3 Complement

When $\mu_{\underset{\sim}{A}}(x) \in [0, 1]$, the complement of $\underset{\sim}{A}$, denoted as $\overline{\underset{\sim}{A}}$ is defined by

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

The Venn diagram for complement operation of fuzzy set $\underset{\sim}{A}$ is shown in Figure 7-12.

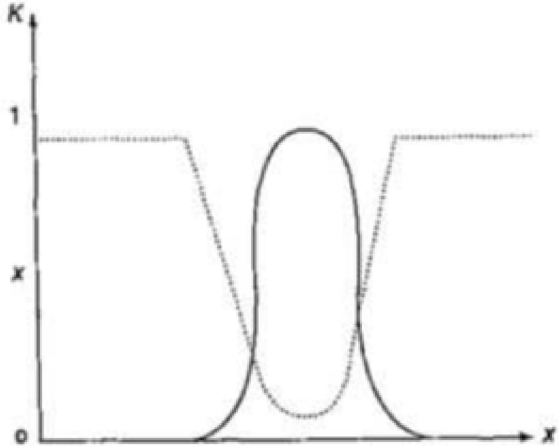


Figure 7-12 Complement of fuzzy set $\underset{\sim}{A}$.

.

9. Inverse: The inverse of a fuzzy relation R on Y denoted b R^{-1} .It i a relation Y X defined b $R^{-1}(y,x) = R(x,y)$ for a pairs (y,x)∈ $Y \times X$

fuzzy relation R(X,Y), let [R $\downarrow Y$ denote dte projection of R ont [R ↓ a fuzz relation i Y whose membership function defined b

R

of a membership function for some f set A defined a that regio of universe that characterized b elements x of the univer such s complete membership in the set A. The core ha that

$\mu_{\underset{\sim}{A}}(x) = 1$

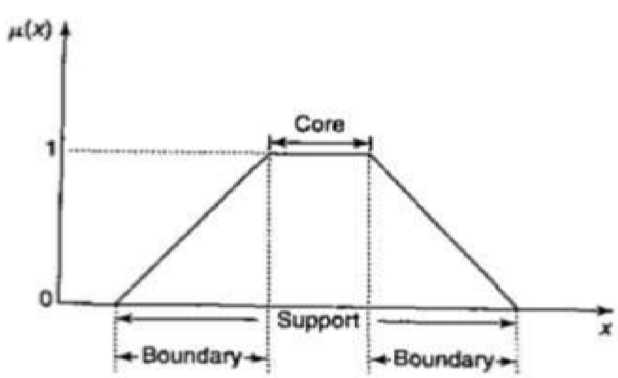
Support: The support of a membership function for a fuz zy set A defined a of universe that characterized b a nonzero membership in the set A. of the universe su that support chat region comprises elements

$\mu_{\underset{\sim}{A}}(x) > 0$

Boundary: The support of a membership function for a fuz zy set A defined containing elements that have a nonzero but not com those elements of of the universe su that region of universe that membership.

The boundary comprises

$0 < \mu_{\underset{\sim}{A}}(x) < 1$



11.Intuition: It based upon the common intelligence of human.It is the capacit of human to develop membership functions on the bas of their own intelligenc e and understanding capability. There should be an in-depth knowledge of the application to which membership value assignment as to be made.

Rank Ordering: The formation of government based on the polling concept; to identify a best student, ranking ma be performed; to a car, one can ask for several opinio and so on. the above mentioned activit are carried out on the basis of th preferences made b an individual, a committee, a poll and other opinion metho

12

a)

1. Commutativity

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup \underset{\sim}{A}; \quad \underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{B} \cap \underset{\sim}{A}$$

2. Associativity

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

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

$$\underset{\sim}{A} \cup \underset{\sim}{A} = \underset{\sim}{A}; \quad \underset{\sim}{A} \cap \underset{\sim}{A} = \underset{\sim}{A}$$

5. Identity

$$\underset{\sim}{A} \cup \phi = \underset{\sim}{A} \text{ and } \underset{\sim}{A} \cup U = U \text{(universal set)}$$
$$\underset{\sim}{A} \cap \phi = \phi \text{ and } \underset{\sim}{A} \cap U = \underset{\sim}{A}$$

6. Involution (double negation)

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

7. Transitivity

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

8. De Morgan's law

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

b)

(a) Algebraic sum

$$\mu_{\underset{\sim}{I}+\underset{\sim}{R}}(x)$$
$$= [\mu_{\underset{\sim}{I}}(x)+\mu_{\underset{\sim}{R}}(x)] - [\mu_{\underset{\sim}{I}}(x)\cdot\mu_{\underset{\sim}{R}}(x)]$$
$$= \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5}\right\}$$
$$- \left\{\frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5}\right\}$$
$$= \left\{\frac{0}{0} + \frac{0.28}{1} + \frac{0.79}{2} + \frac{0.84}{3} + \frac{0.94}{4} + \frac{1}{5}\right\}$$

(b) Algebraic product

$$\mu_{\underset{\sim}{I}\cdot\underset{\sim}{R}}(x)$$
$$=\mu_{\underset{\sim}{I}}(x)\cdot\mu_{\underset{\sim}{R}}(x)$$
$$= \left\{\frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5}\right\}$$

(c) Bounded sum

$$\mu_{\underset{\sim}{I}\oplus\underset{\sim}{R}}(x)$$
$$= \min\{1, \mu_{\underset{\sim}{I}}(x)+\mu_{\underset{\sim}{R}}(x)\}$$
$$= \min\left\{1, \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5}\right\}\right\}$$
$$= \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.0}{4} + \frac{1.0}{5}\right\}$$

(d) Bounded difference

$$\mu_{\underset{\sim}{I}\odot\underset{\sim}{R}}(x)$$
$$= \max\{0, \mu_{\underset{\sim}{I}}(x)-\mu_{\underset{\sim}{R}}(x)\}$$
$$= \max\left\{0, \left\{\frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5}\right\}\right\}$$
$$= \left\{\frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5}\right\}$$

13.

(a) $\text{Plane} \cup \text{Train}$

$$= \max\{\mu_{\text{Plane}}(x), \mu_{\text{Train}}(x)\}$$
$$= \left\{ \frac{1.0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.4}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.2}{\text{house}} \right\}$$

(b) $\text{Plane} \cap \text{Train}$

$$= \min\{\mu_{\text{Plane}}(x), \mu_{\text{Train}}(x)\}$$
$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.2}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

(c) $\overline{\text{Plane}} = 1 - \mu_{\text{Plane}}(x)$
$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(d) $\overline{\text{Train}} = 1 - \mu_{\text{Train}}(x)$
$$= \left\{ \frac{0}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(e) $\text{Plane}|\text{Train}$

$$= \text{Plane} \cap \overline{\text{Train}}$$
$$= \min\{\mu_{\text{Plane}}(x), \mu_{\overline{\text{Train}}}(x)\}$$
$$= \left\{ \frac{0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

(f) $\overline{\text{Plane} \cup \text{Train}}$

$$= 1 - \max\{\mu_{\text{Plane}}(x), \mu_{\text{Train}}(x)\}$$
$$= \left\{ \frac{0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(g) $\overline{\text{Plane} \cap \text{Train}}$

$$= 1 - \min\{\mu_{\text{Plane}}(x), \mu_{\text{Train}}(x)\}$$
$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(h) $\text{Plane} \cup \overline{\text{Plane}}$

$$= \max\{\mu_{\text{Plane}}(x), \mu_{\overline{\text{Plane}}}(x)\}$$
$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(i) $\text{Plane} \cap \overline{\text{Plane}}$

$$= \min\{\mu_{\text{Plane}}(x), \mu_{\overline{\text{Plane}}}(x)\}$$
$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

(j) $\text{Train} \cup \overline{\text{Train}}$

$$= \max\{\mu_{\text{Train}}(x), \mu_{\overline{\text{Train}}}(x)\}$$
$$= \left\{ \frac{1.0}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

14.

Case (i): $\lambda = 0.4$

(a) $$\overline{\underset{\sim}{A}} = 1 - \mu_{\underset{\sim}{A}}(x) = \left\{ \frac{1}{0.2} + \frac{0.2}{0.4} + \frac{0}{0.6} \right\}$$

$(\overline{A})_{0.4} = \{0.2\}$

(b) $$\overline{\underset{\sim}{B}} = 1 - \mu_{\underset{\sim}{B}}(y) = \left\{ \frac{0.1}{0.2} + \frac{0.3}{0.4} + \frac{0.7}{0.6} \right\}$$

$(\overline{B})_{0.4} = \{0.6\}$

(c) $$\underset{\sim}{A} \cup \underset{\sim}{B} = \max[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(y)]$$
$$= \left\{ \frac{0.9}{0.2} + \frac{0.8}{0.4} + \frac{1}{0.6} \right\}$$

$(A \cup B)_{0.4} = \{0.2, 0.4, 0.6\}$

(d) $$\underset{\sim}{A} \cap \underset{\sim}{B} = \min[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(y)]$$
$$= \left\{ \frac{0}{0.2} + \frac{0.7}{0.4} + \frac{0.3}{0.6} \right\}$$

$(A \cap B)_{0.4} = \{0.4\}$

(e) $$\overline{\underset{\sim}{A}} \cup \overline{\underset{\sim}{B}} = \max[\mu_{\overline{\underset{\sim}{A}}}(x), \mu_{\overline{\underset{\sim}{B}}}(y)]$$
$$= \left\{ \frac{1}{0.2} + \frac{0.3}{0.4} + \frac{0.7}{0.6} \right\}$$

$(\overline{A} \cup \overline{B})_{0.4} = \{0.2, 0.6\}$

(f) $\bar{A} \cap \bar{B} = \min[\mu_{\bar{A}}(x), \mu_{\bar{B}}(y)]$

$$= \left\{\frac{0.1}{0.2} + \frac{0.2}{0.4} + \frac{0}{0.6}\right\}$$

$(\bar{A} \cap \bar{B})_{0.4} = \{\phi\}$

Case (ii): $\lambda = 0.7$

(a) $\bar{A} = 1 - \mu_{A}(x) = \left\{\frac{1}{0.2} + \frac{0.2}{0.4} + \frac{0}{0.6}\right\}$

$(\bar{A})_{0.7} = \{0.2\}$

(b) $\bar{B} = 1 - \mu_{B}(y) = \left\{\frac{0.1}{0.2} + \frac{0.3}{0.4} + \frac{0.7}{0.6}\right\}$

$(\bar{B})_{0.7} = \{0.6\}$

(c) $A \cup B = \max[\mu_{A}(x), \mu_{B}(y)]$

$$= \left\{\frac{0.9}{0.2} + \frac{0.8}{0.4} + \frac{1}{0.6}\right\}$$

$(A \cup B)_{0.7} = \{0.2, 0.4, 0.6\}$

(d) $A \cap B = \min[\mu_{A}(x), \mu_{B}(y)]$

$$= \left\{\frac{0}{0.2} + \frac{0.7}{0.4} + \frac{0.3}{0.6}\right\}$$

$(A \cap B)_{0.7} = \{0.4\}$

(e) $\bar{A} \cup \bar{B} = \max[\mu_{\bar{A}}(x), \mu_{\bar{B}}(y)]$

$$= \left\{\frac{1}{0.2} + \frac{0.3}{0.4} + \frac{0.7}{0.6}\right\}$$

$(\bar{A} \cup \bar{B})_{0.7} = \{0.2, 0.6\}$

(f) $\bar{A} \cap \bar{B} = \min[\mu_{\bar{A}}(x), \mu_{\bar{B}}(y)]$

$$= \left\{\frac{0.1}{0.2} + \frac{0.2}{0.4} + \frac{0}{0.6}\right\}$$

$(\bar{A} \cap \bar{B})_{0.7} = \{\phi\}$

PART E

15 . Ebsahim Mamdani proposed system in the year 1975 to contr a stea engine and boiler combination synthesizing a set of fuzzy rules obtained fro people workin on the syste In th case, the output membership functions are expected to be fu sets. After aggregation process, each output variable contains a fuzz set,hence defuzzification is important at the output stage.The following have to be followed to compute the output from this FIS:

Step 1: Determine a set of fuzzy rules.
Step 2: Make the inputs fuzzy using input membership functions.
Step 3: Combine the fuzzified inputs according to the fuzzy rules for establishing a rule strength.
Step 4: Determine the consequent of the rule by combining the rule strength and the output membership function.
Step 5: Combine all the consequents to get an output distribution. aggrigation
Step 6: Finally, a defuzzified output distribution is obtained.

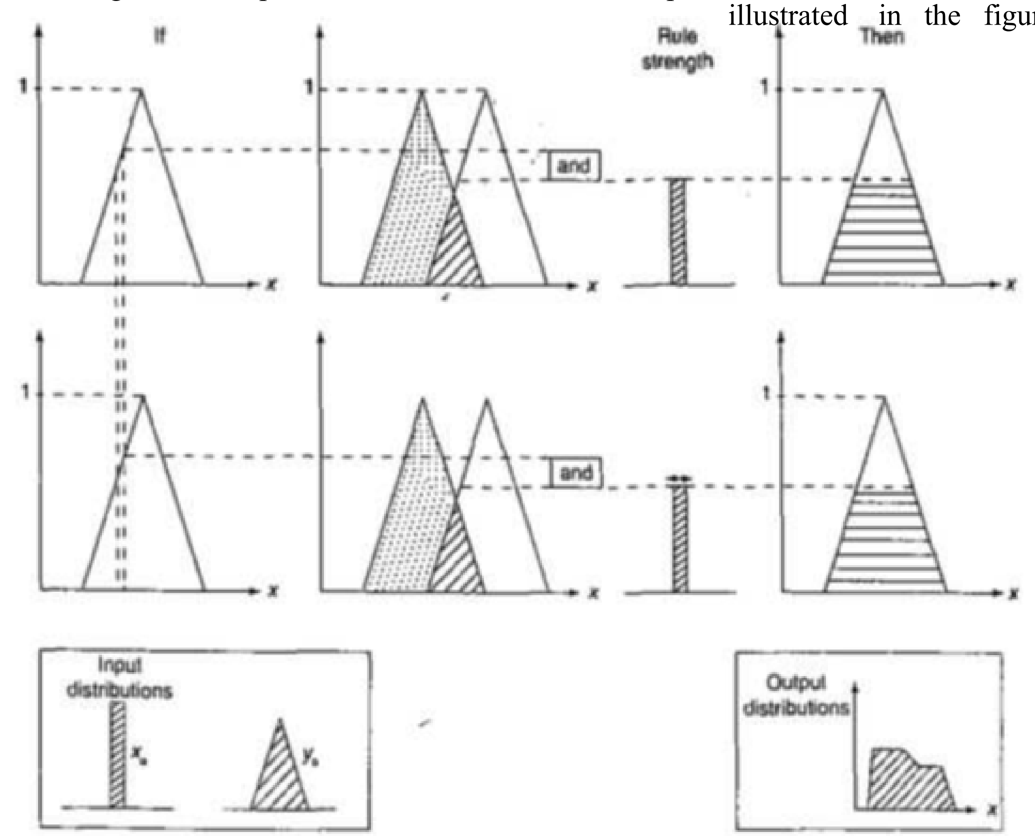
The fuzz rules are forme using "IF-THEN" statements and "AND/OR" connectives. T consequence of the rule ca be obtained in two steps

1.by computing the r strength completely using the fuzzified in from the fuzz combination;
2.by clipping the output membership function at the rule streng th.

The outputs of a the fuz rules are combined to obtain one f output distribucio From FIS, it i desired to get on one crisp output. T crisp output m be obtained from defuzzification process. The common techniques of defuzzification used are center of mass and mean of maximu

Consider a tWo- fuzzifies the two inpu by

used to compute the fuz input "an for combining the fuzzified inputs to obtain a rule strength.The output membership func clipped at t minimum operation is rule strength. Finally, the maximu operator i output "o fo combining the output of the two rules. This p i used to compute the fuzzy illustrated in the figure



16.

1. ***Fuzzy predicates:*** In fuzzy logic the predicates can be fuzzy, for example, tall, short, quick. Hence, we have proposition like "Peter is tall." It is obvious that most of the predicates in natural language are fuzzy rather than crisp.
2. ***Fuzzy-predicate modifiers:*** In fuzzy logic, there exists a wide range of predicate modifiers that act as hedges, for example, very, fairly, moderately, rather, slightly. These predicate modifiers are necessary for generating the values of a linguistic variable. An example can be the proposition "Climate is moderately cool," where "moderately" is the fuzzy predicate modifier.
3. ***Fuzzy quantifiers:*** The fuzzy quantifiers such as most, several, many, frequently are used in fuzzy logic. Employing these, we can have proposition like "Many people are educated." A fuzzy quantifier can be interpreted as a fuzzy number or a fuzzy proposition, which provides an imprecise characterization of the cardinality of one or more fuzzy or nonfuzzy sets. Fuzzy quantifiers can be used to represent the meaning of propositions containing probabilities; as a result, they can be used to manipulate probabilities within fuzzy logic.
4. ***Fuzzy qualifiers:*** There are four modes of qualification in fuzzy logic, which are as follows:
   - ***Fuzzy truth qualification:*** It is expressed as "x is τ," in which τ is a fuzzy truth value. A fuzzy truth value claims the degree of truth of a fuzzy proposition. Consider the example,

     (Paul is Young) is NOT VERY True.

     Here the qualified proposition is (Paul is Young) and the qualifying fuzzy truth value is "NOT Very True."
   - ***Fuzzy probability qualification:*** It is denoted as "x is λ," where λ is fuzzy probability. In conventional logic, probability is either numerical or an interval. In fuzzy logic, fuzzy probability is expressed by terms such as likely, very likely, unlikely, around and so on. Consider the example,

     (Paul is Young) is Likely.

     Here the qualifying fuzzy probability is "Likely." These probabilities may be interpreted as fuzzy numbers, which may be manipulated using fuzzy arithmetic.
   - ***Fuzzy possibility qualification:*** It is expressed as "x is π," where π is a fuzzy possibility and can be of the following forms: possible, quite possible, almost impossible. These values can be interpreted as labels of fuzzy subsets of the real line. Consider the example

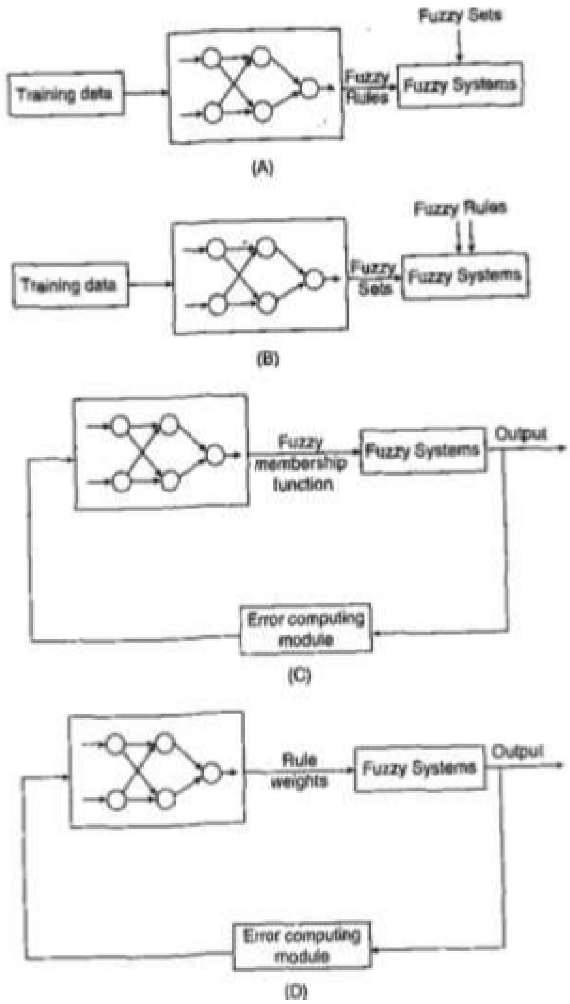
     (Paul is Young) is Almost Impossible.

     Here the qualifying fuzzy possibility is "Almost Impossible."
   - ***Fuzzy usuality qualification:*** It is expressed as "usually (X) = usually (X is F)," in which the subject X is a variable taking values in a universe of discourse U and the predicate F is a fuzzy subset of U and interpreted as a usual value of X denoted by $U(X) = F$. The propositions that are usually true or the events that have high probability of occurrence are related by the concept of usuality qualification.

17

Cooperative Neural Fuzzy Systems

In this type of system, both artificial neural network (ANN) and system wor independently from each other. The ANN attempts to learn the parameter fro the fuzz system. Four different kin of cooperative fuzzy neural networks are shown in Figu



The FNN Figure (A) learns fu set fro the given training data. is
by fittin membership functions with a neural network; the sets then done, usually,
determined offline. Th i followed their utilization to fo the fuzzy system being
rules that are given, not learned. The NFS in Figure (B) determines, fuzzy
network, the fuz rules by fro neural
offline before the fu system the training data. Here again, the neural a networks learn
clustering on self- maps. There i by
clustering methods to obtai initialized. The rule learning happens usually
rules.

For the neuro- model also the possibility of applying fuzzy
are learnt online, while the system i applied. Thi means rules
and membership functions
guide the learning step, the error shown in Figure (C), the parameters of membership functio
The model shown Figure (D) determines the rule weights f fuzzy rules a neura
network. A r i determined b it rule weight- n
that, initially, fuzzy
must be defined beforehand. AJso, in order to improve
and

18

to be measured.

Genetic Learning of

As shown Figure , genetic learning of rule assumes by
membership functions i the DB to which the refer, the influence of a rule. They means
in the approximate approach adapting rules, it
are . then multiplied . with the rule outp ut
equivalent to modifying

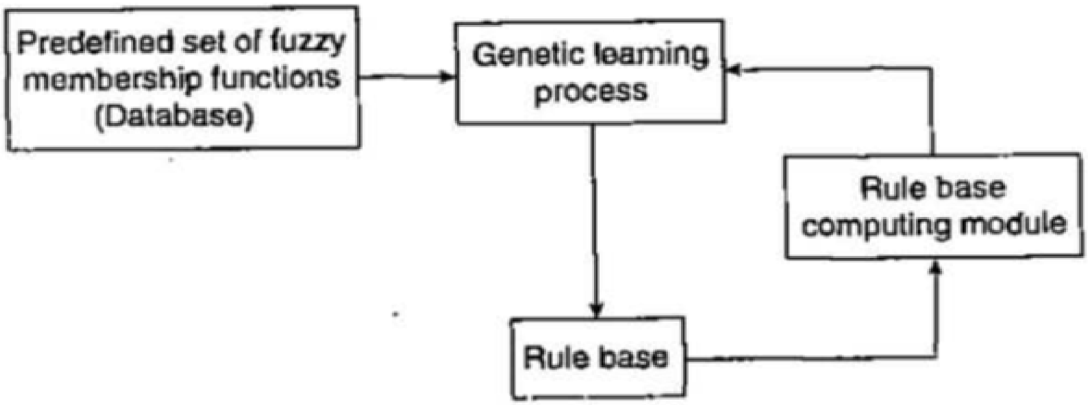
Rule Bases

a predefined set of fuzzy
of linguistic labels. As
applies to descriptive FRBSs, which is
the membership functions.When considering a rule based system

and focusing on learning rules, t n are three main approaches that have been applied i here

1. Pittsburgh
2. approach.
3. Iterative rule learning approach

Michigan approach.



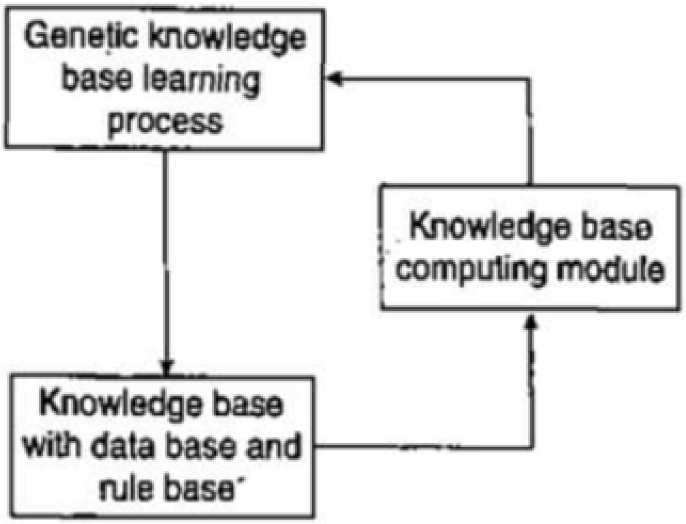
The Pittsburgh approach i characterized b representing an entire rule se code {chromosome),maintaining a population of candidate rule a generic and genetic operators and using selection considers a different model whe the members to produce new generations of rule sets. The Michigan approac a rul set i represented h b ana chromosomes code individual rules, and a new of the rul population are individual rules in the entire population. In the third approach,the iterative an iterative fashion, i every

Genetic Learning of Knowledge Base

adapted and added to the rule set, as variable lengt h run of the genetic algorithm.

instead of a whole K it deal with heterogeneous search spaces. As the complexity the search space increases the computational cost of the gen Genetic learning of a KB includes different generic representing suc h combat this issue an option to maintain a GFRBS that enc individual rul than entire KB. In th one can maintain a flexible, com rule space in which search fo feasible and efficient. The three learning approach used in case a base can also be considered here: Michigan, Pittsburgh, and it search also grows. To rather

rule learning manner solution remains of rule approach. Figure illustrates the generic learning of K B.



19

15.9.1.1 Binary Encoding

The most common way of encoding is a binary string, which would be represented as in Figure 15-15.

Each chromosome encodes a binary (bit) string. Each bit in the string can represent some characteristics of the solution. Every bit string therefore is a solution but not necessarily the best solution. Another possibility is that the whole string can represent a number. The way bit strings can code differs from problem to problem.

Binary encoding gives many possible chromosomes with a smaller number of alleles. On the other hand, this encoding is not natural for many problems and sometimes corrections must be made after genetic operation is completed. Binary coded strings with 1s and 0s are mostly used. The length of the string depends on the accuracy. In such coding

1. Integers are represented exactly.
2. Finite number of real numbers can be represented.
3. Number of real numbers represented increases with string length.

15.9.1.2 Octal Encoding

This encoding uses string made up of octal numbers (0–7) (see Figure 15-16).

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Figure 15-15 Binary encoding.

Chromosome 1	03467216
Chromosome 2	15723314

Figure 15-16 Octal encoding.

Chromosome 1	9CE7
Chromosome 2	3DBA

Figure 15-17 Hexadecimal encoding.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Figure 15-18 Permutation encoding.

15.9.1.3 Hexadecimal Encoding

This encoding uses string made up of hexadecimal numbers (0–9, A–F) (see Figure 15-17).

15.9.1.4 Permutation Encoding (Real Number Coding)

Every chromosome is a string of numbers, represented in a sequence. Sometimes corrections have to be done after genetic operation is complete. In permutation encoding, every chromosome is a string of integer/real values, which represents number in a sequence.

Permutation encoding (Figure 15-18) is only useful for ordering problems. Even for this problem, some types of crossover and mutation corrections must be made to leave the chromosome consistent (i.e., have real sequence in it).

15.9.1.5 Value Encoding

Every chromosome is a string of values and the values can be anything connected to the problem. This encoding produces best results for some special problems. On the other hand, it is often necessary to develop new genetic operator's specific to the problem. Direct value encoding can be used in problems, where some complicated values, such as real numbers, are used. Use of binary encoding for this type of problems would be very difficult.

In value encoding (Figure 15-19), every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or characters to some complicated objects. Value encoding is very good for some special problems. On the other hand, for this encoding it is often necessary to develop some new crossover and mutation specific for the problem.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Figure 15-19 Value encoding.

15.9.1.6 Tree Encoding

This encoding is mainly used for evolving program expressions for genetic programming. Every chromosome is a tree of some objects such as functions and commands of a programming language.

20 a)

1. *Maximum generations*: The GA stops when the specified number of generations has evolved.
2. *Elapsed time*: The genetic process will end when a specified time has elapsed.
   Note: If the maximum number of generation has been reached before the specified time has elapsed, the process will end.
3. *No change in fitness*: The genetic process will end if there is no change to the population's best fitness for a specified number of generations.
   Note: If the maximum number of generation has been reached before the specified number of generation with no changes has been reached, the process will end.

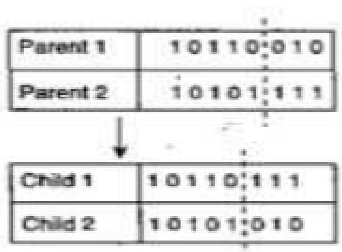
4. *Stall generations*: The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generations of length "Stall generations."
5. *Stall time limit*: The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to "Stall time limit."

b)

Single-Point Crossover

The traditional geneti algorithm use single-point crossover, where the two mating chromosomes are cut onc at corresponding point and che section after the cuts exchanged. Here, a cross site or crossove point selected randomly along the length of the mated strings and bits next to the cro sites are exchanged.Here,a cross site or crossover point is selected randomly along the mated and bits the cross sites are exchanged.If appropriate site chosen, bener children can be next to obtained by combining good parents, else it severely hamp string quality.

Figure illustrates single- crossover point are exchanged to produce children. The cros point can be chosen and it can be observed that th next to the randomly.

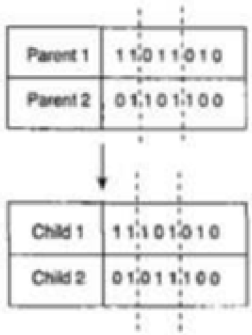


Two-Point Crossover

Apart from single- different crossove algorithms have been devised, often involving mo than one cut point. It should be noted that adding further cro ssover points reduces the performance of the GA. The pr with adding additional crossover points i that building bloc are more likely to be disrupted. However, an advantage of having more crossover points i that the problem spa may be searched more

In two-points crossover, two crossov points are chosen and the conte between these points are exchanged between two mated parents.

In Figure the dotted lines indicate the cro points. Thu the contents between these are exchanged between the parents to produce new children for mating in th e next generation.


Parent 1
1 1 0 1 1 0 1 0
Parent 2
0 1 1 0 1 1 0 0
Child 1
1 1 1 0 1 0 1 0
Child 2
0 1 0 1 1 1 0 0

END