Neural Networks & fuzzy logic :-

1(a) Ans. **Applications of Neural Networks :-** (2 Marks)

(i) character Recognition :-
(ii) Image Compression :-
(iii) Stock market Prediction :-

(1 Mark)

(iv) Travelling salesman Problem :-
(v) Medicine, electronic nose, security and loan applications :-
(vi) Miscellaneous applications :-

(1m)

1(b) Ans. **Training methods of Artificial neural network :-** (2 Marks)

→ 2 - types
i) supervised learning defination
ii) unsupervised learning defination.

(1m)

→ training methods of ANN Consists of two Parts
ⓐ Activation state ⓑ dynamic synaptic weights.

(1M)

which are interms represented as short term and long term.

short term :- neural n/w is modeled by activation state of n/w.

long term :- neural n/w is Encoded by information in synaptic weights due to learning.

(2 Marks)

1(c):- **Delta learning Rule :-**

Ans

Delta learning rule introduced for training neural network. which is applicable only for Continous activation function.

→ This rule can be derived from Condition of least square error between desired o/p (d_i) and actual o/p (O_i)

→ the squared error defined as 1m

$$E = \frac{1}{2}(d_i - O_i)^2 = \frac{1}{2}[d_i - f(w_i x)]^2 \rightarrow (a)$$

the error gradient vector w.r.t W_i written as

$$\nabla E = -(d_i - O_i) f'(w_i x) x \rightarrow (b)$$ 1m

∴ Elements of gradient vector are $\frac{\partial E}{\partial w_{ij}} = -(d_i - O_i) f'(w_i x_i) x_j \rightarrow (c)$

for $j = 1 \rightarrow n$.

∴ The weight adjustment given by

$$\Delta w_{ij} = \eta (d_i - O_i) f(w_i x_i) x_j \text{ for } i = 1 \rightarrow n.$$

(2Marks)

1(d):- **Perceptron:-**

Ans

→ The Perceptron is an model of artificial neuron, which is a single layer ~~network~~ neural network whose weights and biases could be trained to produce a correct target (or) desired response.

→ Perceptron is a program that learn concepts i.e it can learn to respond with (True (1)) and (false (0)).

→ The training technique used is called perceptron learning rule.

(i) Pattern representations:-

Classification involves { Spatial Pattern, temporal Pattern. } 1m

Scanned by CamScanner

Examples:- of Spatial Patterns:-

Pictures, Video images, finger prints, weather maps.

Eg:- of Temporal Patterns:-

Speech signals, signal vs time produced by sensors, ...etc,

1(e):- Recurrent Auto associate Memory:- (2 Marks)

Ans.

→ Associative memories can be implemented using networks with (or) without feedback but in recurrent autoassociate the simplest kind of feedback that the output of a network is used repetitively as a new i/p untill the process converges.

→ The function of an associative memory is to recognize previously learned input vectors; even in the case where some noise has been added. in finding cluster centroids in input spaces.

1(f):- Spatio-Temporal Patterns:- (2 Marks)

Ans

A spatio temporal neural network differs from other neural networks in two ways

(i) Propagation delay along links and recurrence play an important role in the computation carried out by the network.

(ii) The representational state of network depends not only on which nodes are firing, but also on relative firing times of nodes.

→ The representational significance of node varies with time and depends on firing state of other nodes. The use of recurrence and multiple links with variable Propagation delay Provides rich mechanism for integration, Context sensitivity, feature Extraction and pattern recognition. Recurrent links Enable node to integrate and differentiate i/p, detect the onset of features and measure their duration. At the same time, multiple links with variable Propagation delays between node serves as a short term memory and allow the network to maintain Context over a window of time.

1 (g) (i) Classical sets in Operations :- (2 marks)

Union → $A \cup B = \{x \mid x \in A \text{ (or) } x \in B\}$

intersection → $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

Complement → $\bar{A} = \{x \mid x \notin A, x \in X\}$

Difference → $A \mid B = \{x \mid x \in A \text{ and } x \notin B\}$

→ (1 mark) with venn diagrams

(ii) Fuzzy Sets Operations :-

union → $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$

intersection → $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$

Complement → $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

→ (1 mark) with venn diagram.

1 (h) Ans. Fuzz Logic :- → (2 marks)

Definations ; Logical Connectives.

↳ (i) negation

↳ (ii) disjunction

↳ (iii) Conjunction

↳ (iv) implication

notations.

1 (i) **Fuzzyfication :-** $\longrightarrow$ 2 marks

Defination ~~and~~ Explanation $\longrightarrow$ 1 mark

membership function representing with example $\longrightarrow$ 1m

1(j):- **Problem:-** 2 marks

Ans

Given $X = (a, b, c, d)$ $\quad Y = (1, 2, 3, 4)$

$\tilde{A} = \{(a,0)(b,0)(c,0.6)(d,1)\}$; $\tilde{B} = \{(1,0.2)(2,1)(3,0.8)(4,0)\}$

$\tilde{C} = \{(1,0)(2,0.4)(3,1)(4,0.8)\}$

(i) If x is $\tilde{A}$ THEN y is $\tilde{B}$ $= (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times Y)$

By using Zadeh's notation.

$$\tilde{A} = \left\{\frac{0}{a} + \frac{0}{b} + \frac{0.6}{c} + \frac{1}{d}\right\} = \left\{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}\right\}$$

$$\tilde{B} = \left\{\frac{0.2}{1} + \frac{1}{2} + \frac{0.8}{3} + \frac{0}{4}\right\} = \left\{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{c}\right\}$$

$$\tilde{C} = \left\{\frac{0}{1} + \frac{0.4}{2} + \frac{1}{3} + \frac{0.8}{4}\right\} = \left\{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}\right\}$$

$$(\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times Y) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

1m

(ii) If x is $\tilde{A}$ THEN y is $\tilde{B}$ ELSE y is $\tilde{C}$

$$R = (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times \tilde{C})$$

$$\therefore (\tilde{A} \times \tilde{B}) \cup (\tilde{A}^c \times \tilde{C}) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

1m

Scanned by CamScanner

2(a) **Structure of Biological neuron :-** (5 marks)

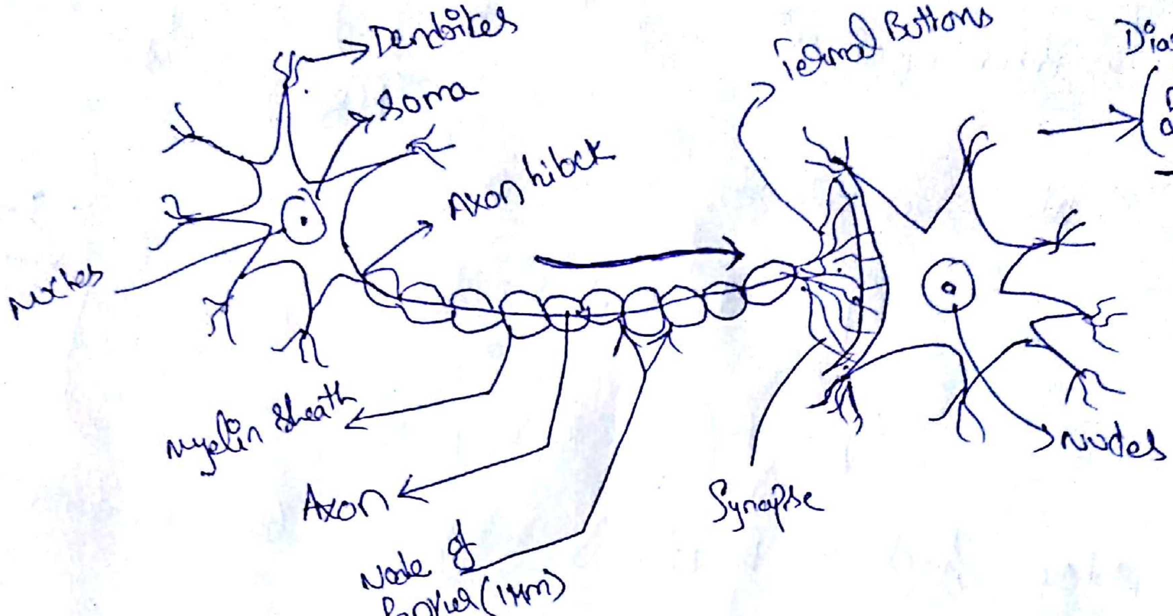


Diagram (2 marks)

(i) Dendrites :-
(ii) Soma (or) Cellbody :-
(iii) Axon :-
(iv) Axon hillock :-
(v) Myelin sheath :-
(vi) Node of Ranvier :-
(v) Synapse :-
(vii) Thermal Buttons :-

→ (2 marks)

→ Information flow in neural cells :- (1 mark):

2(b):- **Neuron modeling for artificial neural systems :-** (5 marks).

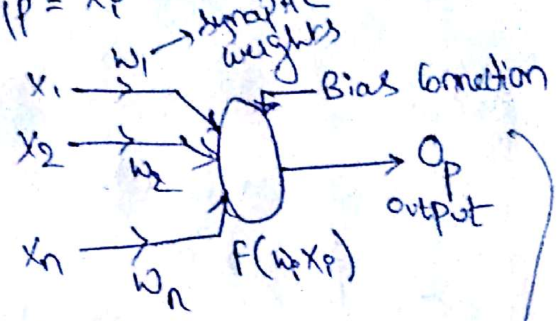
McCulloch pitts model of neuron is characterized by its formalism and its elegant precise mathematical model.

→ This model allows 0, 1 states only operates under a discrete time assumptions. in larger networks.

→ weights and neuron threshold fixed in model and number interaction among network neurons take place except signal.

Let $x_1, x_2, x_3 \ldots x_n$ are i/p = 'x_i'

'$w^t = [w_1, w_2 \ldots w_n]^t$ are weighted synaptics



∴ Synaptic representation of o/p is

$$O_p = f(w^t . x) \rightarrow (a)$$

(or)

$$O_p = f\left(\sum_{i=1}^{n} w_i x_i\right) \rightarrow (b)$$

1 mark

where function $f(w^t . x)$ → referred as Activation function.

Q-models :-

(i) neural network are meant to be artificial neural n/w consisting of neuron models

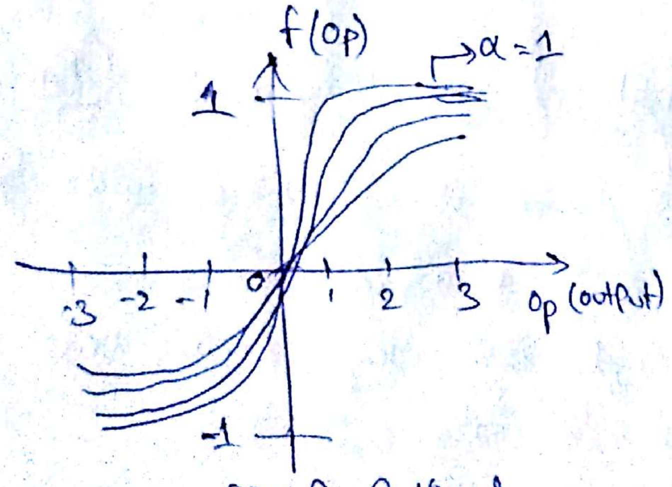
(ii) neuron are meant to be artificial neuron models.

→ Typical activation function used is

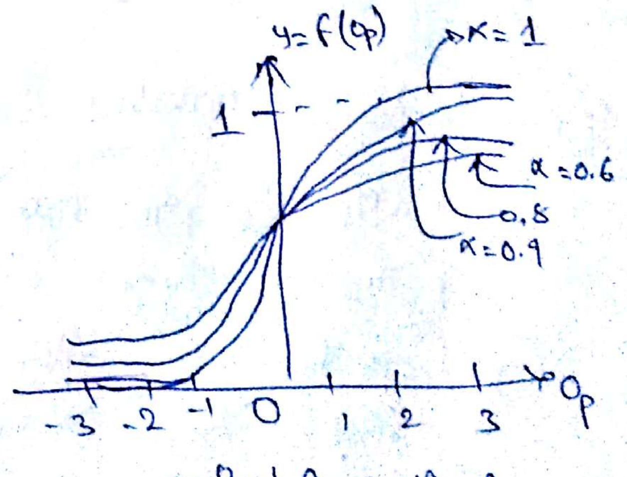
(1 mark)

$$f(O_p) = \frac{2}{1+e^{(-\alpha O_p)}} - 1 \rightarrow (c)$$

$$f(O_p) = Sgn(O_p) = \begin{cases} +1 & O_p > 0 \\ -1 & O_p < 0 \end{cases} \rightarrow (d)$$



Bipolar Continous

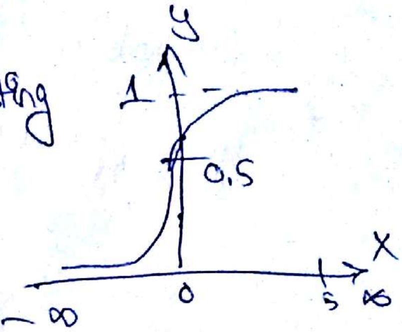


unipolar Continous.

Activation function:- (3 marks)

(i) Sigmoid function :- (unipolar sigmoid)

used in Prediction & Process forecasting



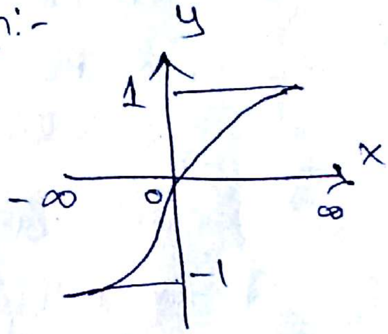
$$y(x) = f(x) = \frac{1}{1+e^{-x}}$$

range is $0 < y < 1$

The derivative function is $y'(x) = f'(x) = f(x) \cdot (1 - f(x))$.

(ii) Hyperbolic tangent (bipolar sigmoid) function:-

used in Prediction & Process forecasting



$$y(x) = F(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

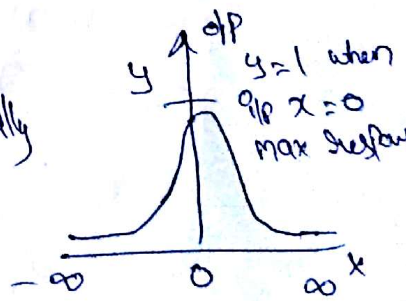
range is $-1 < y < 1$

derivative $y'(x) = F'(x) = \left[1 - F(x)\right]^2$ "

(iii) Radial biasis function:-

→ used for fault diagnosis Problems.

→ Gaussian function is mostly Common used in radially Symmetric function



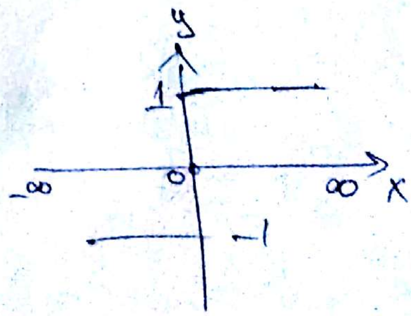
$$y(x) = f(x) = e^{(-x^2/2)}$$

range $0 < y < 1$

derivative $y'(x) = f'(x) = -x \cdot e^{-x^2/2}$

(iv) Hard limiter:-

used in classification of patterns

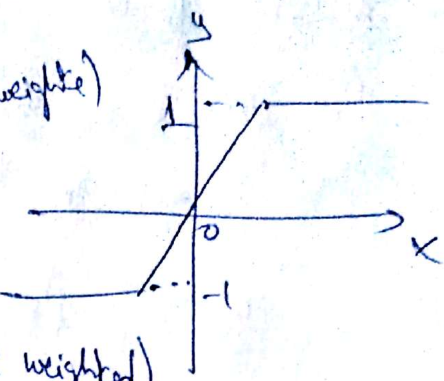


$$f(x) = \operatorname{Sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

(v) Piecewise Linear function:-

→ Saturation linear function.

$$y(x) = f(x) = \begin{cases} -1 & \text{if } Sgn(x) < 0 \ \text{(-ve weighted)} \\ Sgn(x) & \text{if } -1 \geq x \geq 1 \\ 1 & \text{if } Sgn(x) \geq 0 \ \text{(+ve weighted)} \end{cases}$$



3(a) Learning mechanisms:- (Any two types) (5 marks)

(i) Error Correction mechanism → Perceptron, Delta, Widrow Hoff.
(ii) memory Based learning mechanism.
(iii) Hebbian learning mechanism.
(iv) Competitive learning mechanism.
(v) Boltzmann learning mechanism.

(i) Error Correction mechanism:- (used in Recognization of error).

Consider n time index, then the error at n^{th} instant case obtained as function of actual o/p be $O_k(n)$ and desired target response be $d_k(n)$ for k^{th} neuron.

∴ o/p layer error signal is

$$e_k(n) = d_k(n) - O_k(n) \rightarrow (a)$$

cost function (or) Performance index $E(n)$ is

$$E(n) = \frac{1}{2} \cdot e_k^2(n) \rightarrow (b)$$

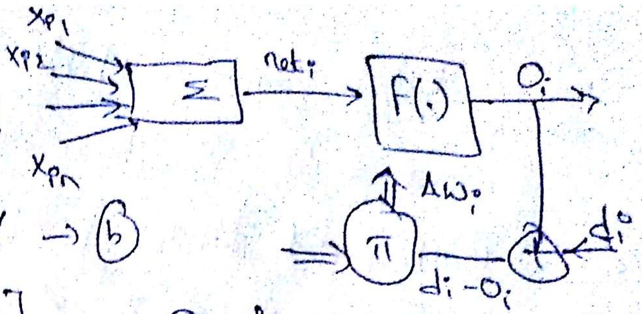
(a) Perceptron learning rule:- (Applicable for binary response).

let 'i' be neuron then error learning is

$$e_i = d_i - O_i \rightarrow (a)$$

$$O_i = Sgn(w_i x_i)$$

weight adjustments given as



ΔW_i and Δw_{ij} are defined

$$\Delta W_i = C\left[d_i - \text{Sgn}(w_i x)\right] x \rightarrow (b)$$

$$\Delta w_{ij} = C\left[d_i - \text{Sgn}(w_i x)\right] x_j \rightarrow (c) \text{ for } j = 1 \rightarrow n.$$

The desired response is either 1 (or) -1

C = learning Constant.

(b) Delta learning rule:- (used for Continuous activation function)

The squared error defined as

$$E = \frac{1}{2}(d_i - O_i)^2 = \frac{1}{2}\left[d_i - f(w_i x)\right]^2 \rightarrow (a)$$

Error gradient

$$\nabla E = -(d_i - O_i)\, f'(w_i x)\, x \rightarrow (b)$$

gradient vector $\frac{\partial E}{\partial w_{ij}} = -(d_i - O_i)\, f'(w_i x) x_j$ for $j = 1 \rightarrow n$. $\rightarrow$ (c)

$\therefore$ The weight adjustment

$$\Delta w_{ij} = \eta(d_i - O_i)\, f(\text{net}_i)\, x_j \text{ for } j = 1 \rightarrow n.$$

(c) window-Hoff learning rule:-

used for independent activation function.

Error represent for ith neuron be

$$e_i = d_i - O_i = d_i - W_i x \rightarrow (a)$$

using learning rule change in weight vector.

$$\Delta W_i = \eta(d_i - W_i x)\, x \rightarrow (b)$$

weight adjustment

$$\Delta w_{ij} = \eta(d_i - W_i x) \text{ for } j = 1 \rightarrow n. \rightarrow (c)$$

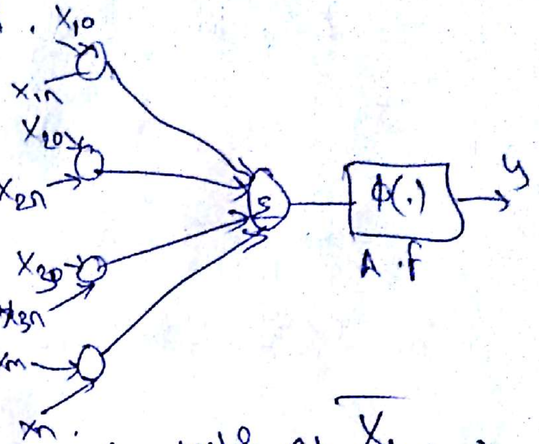
(ii) memory based learning mechanism:-

used in identifying the error.

let us consider 1-Pattern of x then.

$\bar{x}$ is vector and o/p of x is

$d_i \rightarrow$ vector.



$\{\bar{x}_i, d_i\}_{i=1}^{N} \rightarrow$ for 1-Pattern.

let an i/p be $\bar{x}_{test}$ and o/p of system be vector as $\bar{X}_{test}$.

for nearest vector among m vector i/p.

∴ Closest among 'm' of 1-Pattern is.

$$x'_N \in \{\bar{x}_1, \bar{x}_2, \bar{x}_3 \dots \bar{x}_N\}$$

↓ nearest neighbour.

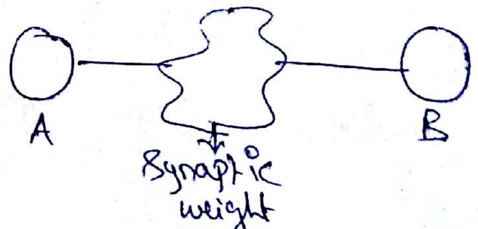
∴ euclident distance minimum is related as

$d(\bar{x}_i \,\&\, \bar{x}_{test})$ is ~~searc~~ searched by valying $i = 1 \rightarrow N$

$$\min d(\bar{x}_i, \bar{x}_{test}) = d(\bar{x}'_N, \bar{x}_{test}).$$

(iii) Hebbian learning mechanism:-

node A's ~~fo~~ efficiency in firing B metabolic changes between ~~as~~ two nodes increases.



→ Positive Correlated leads Synaptic strengthening.

→ negative Correlated leads Synaptic weakening.

mathematic relations:- in weight adjustments

$$\Delta w_{kj}(n) = f(y_k(n) \cdot x_j(n))$$

$y_k(n)$ ↓ Post Synaptic

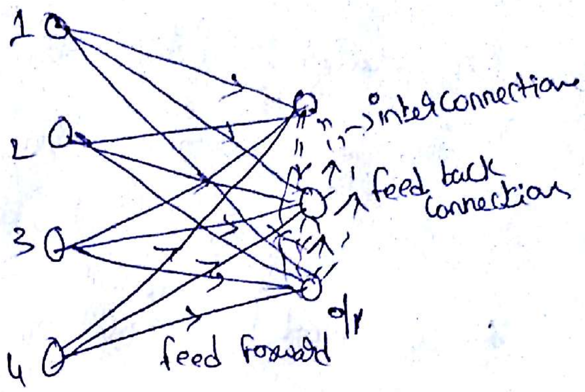
$x_j(n)$ → Pre Synaptic signal.

$$\boxed{\therefore \Delta w_{kj}(n) = \eta \, y_k(n) \cdot x_j(n)}$$

$\eta \rightarrow$ learning Constant.

(iv) Competitive learning :-

In this learning mechanisum only a single o/p neuron is active at any one time.



$$y_k = \begin{cases} 1 & \text{if } V_k > V_j \text{ for } J \neq K \\ 0 & \text{other wise.} \end{cases}$$

where the sum of total of synaptic weights = 1

$$\sum_j W_{kj} = 1 \quad \text{for all } K^{th} \text{ neuron.}$$

(v) Boltzmann learning mechanisim :-

In Boltzmann machine no hidden layer is present and it is characterized by Energy function E.

$$E = -\frac{1}{2} \sum_j \sum_K W_{kj} x_k \cdot x_j \rightarrow (a) \quad \text{for all } J \neq K$$

If particular neuron K pick up then K neuron state changes from (-1) to (+1) state.

Probability at neuron k flips state is

$$P(x_K \rightarrow -x_K) = \frac{1}{1+e^{(-\Delta E_K/T)}}$$

$\Delta E_K \rightarrow$ change of energy resulting from such a flip (constant).

$T \rightarrow$ Pseduo temperature.

Boltzmann learning categorized in 2 states. → visible ((+1/-1) state) → non visible (free state

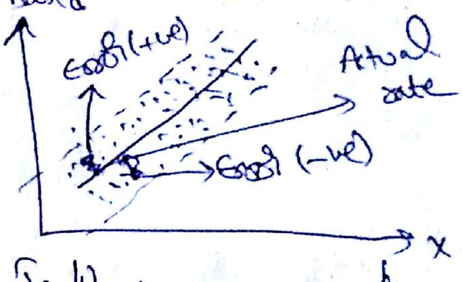
The weight adjustment b/w free running condition of neurons

$$\Delta W_{Kj} = \eta\,(\rho^{+}_{kj} - \rho^{-}_{kj}) \quad J \neq K.$$

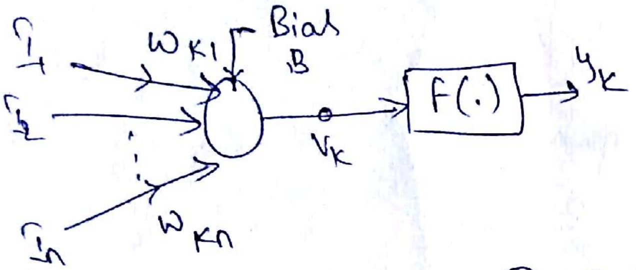
Scanned by CamScanner

3(b):- Use of linear model and non linear model in ANN:- (5 marks)

for concept of single layer (or) multilayer neural network to identify the active neurons in network based on Activation function process it is identified.



Linear model:- used in data fit model

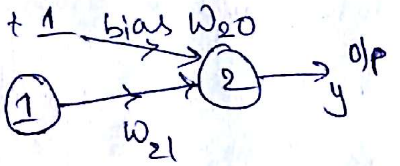


$$V_K = \sum_{j=1}^{n} I_j W_{Kj}$$

$$V_K = I_1 W_{K1} + I_2 W_{K2} + \dots I_j W_{Kj}$$

→ (1 mark)

if 2-models ① & ② considered



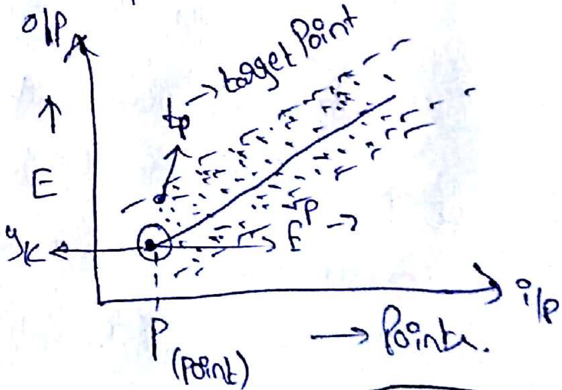
$$y_{21} = W_{21} x + W_{20}$$

x → o/p from neuron 1

W_{21} → slope

↳ Bias is intercept

(i) Representation of Error:-



let E^P → Error for particular point p

t^P → target o/p for point 'p'

y^P → filled o/p (or) neural o/p response

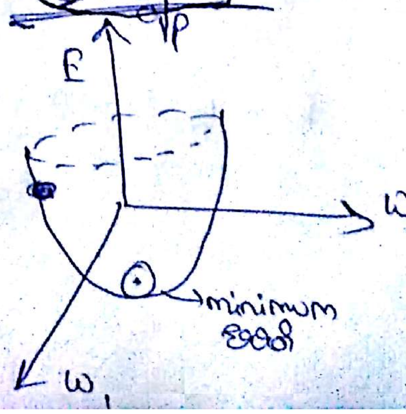
$$\therefore E^P = (t^P - y^P) \rightarrow ①$$

→ (1 mark)

∴ Total error of all set up points are.

$$E = \sum_P E^P = \sum_P (t^P - y^P)^2 \rightarrow ②$$

(ii) Gradient Descent algorithm. → (3 mark)



(Derivation)

$$\boxed{\Delta W_{0i} = \eta (t_0 - y_0) . x_i}$$ //.

Scanned by CamScanner

(5 marks) ⑧

4(a) Ans

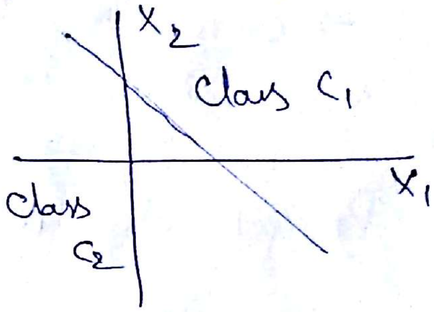
Single layer Perceptron :- (2 marks)

→ Perceptron learning is of supervised type, which is trained by presenting a set of patterns to its i/p, one at a time and adjusting the weights untill the desired o/p occurs. for each of them.

→ for discrete perceptron the activation function should be hard limiter (or) sgn() function.

→ The popular application of discrete perceptron is pattern classification.

Let the decision region in n-dimensional space, spanned by n-inputs variables.



The two regions seperated by hyper plane is

$$\sum_{i=0}^{n} W_i X_i = 0 \rightarrow (a)$$

The i/p variables of perceptron originates from two linearly separable classes.

Let $\mathcal{X}_1$ be subset of training vectors $X_1(1), X_2(2) \ldots$ that belong 'C_1' class.

and $\mathcal{X}_2$ be subset of training vectors $X_2(1), X_2(2) \ldots$ that belong 'C_2' class.

The weight vector 'W' is written as.

$WX > 0$ for every vector X belong class C_1

$WX \leq 0$ for every vector X belonging to class C_2.

Algorithm for updating weights formulated as

1.) If k^{th} neuron of training set X_K is correctly classified by weight vector $W(K)$ computed at K^{th} iteration of algorithm.

$$W^{K+1} = W^K \quad \text{if} \quad W^K X_K > 0 \text{ and } X_K \text{ belong to class } C_1$$

$$W^{K+1} = W^K \quad \text{if} \quad W^K X_K \leq 0 \text{ and } X_K \text{ belongs to class } C_2.$$

2) weight vector of perceptron is updated with rule.

$$W^{(K+1)T} = W^{KT} - \eta X_K \quad \text{if} \quad W^K X_K > 0 \text{ and } X_K \text{ belong to class } C_2$$

$$W^{(K+1)T} = W^{KT} + \eta X_K \quad \text{if} \quad W^K X_K \leq 0 \text{ and } X_K \text{ belongs to class } C_1.$$

Multilayer Perceptron:- (3 marks)

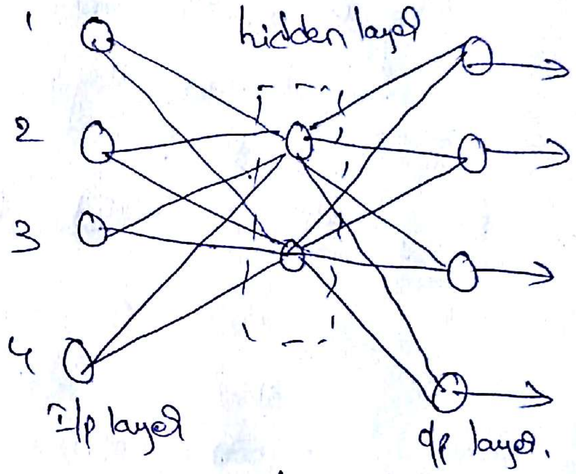
A multilayer neural network is a feed forward neural network with at least one hidden layer. It can deal with non-linear classification problems because of more complex decisions region.

→ Each node in first layer can create hyper plane

→ Each node in second layer can combine hyper plane to create convex region.

→ Each node in third layer combine convex region to form concave region.

→ Learning rule used for multilayer neural network is generalized delta rule. (or) back propagation rule. (or) Back propagation algorithm.



(1 → mark)

(*) Back Propagation algorithm:-

↳ Diagram → (1 mark)

↳ steps → (1 marks)

(5 marks)

4(b):-

Ans

The error correcting algorithm to task an multicategory classification of single layer networks. The assumption needed is that classes are linearly pairwise separable, (or) that each class is linearly separable from each other class.

The assumptions exits R linear discriminant functions.

$$g_i(x) > g_j(x) \quad \text{for } i, j = 1 \to R \; ; \; i \neq j$$

let weights vector W_q

$$W_q = [W_{q1}, W_{q2} \cdots W_{qn+1}]^t \to (a)$$

The 'R'-decision functions $w_1^t y, w_2^t y \ldots w_R^t y$ are Evaluated. (9)

If $w_i^t y$ is larger than any remaining R-1 discriminant function no adjustment of weight needed.

indicates $w_1' = w_1$; $w_2' = w_2 \ldots w_R' = w_R$.

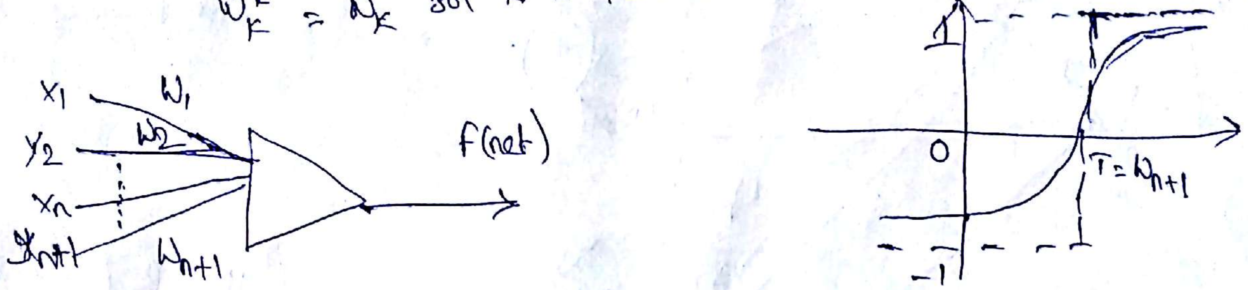
If for some m-values we have $w_i^t y \leq w_m^t y$ then update weights vector become.

1 mark

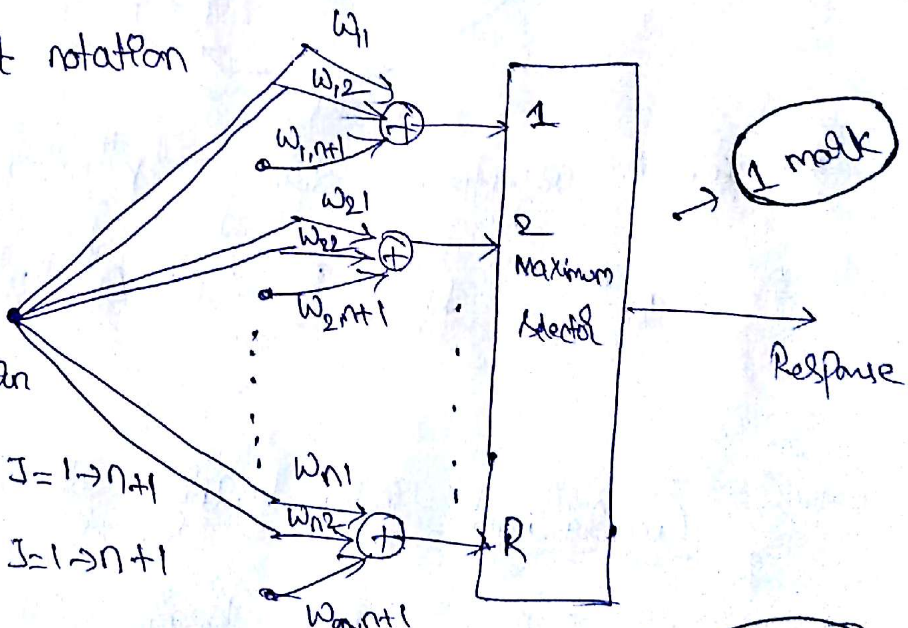
$w_i' = w_i + cy$

$w_M' = w_m - cy$

$w_k' = w_k$ for $k = 1,2 \ldots R$ $k \neq i, m$.



The double subscript notation for weight is



$w_{ij}' = w_{ij} + cy_j$ for $j = 1 \to n+1$

$w_{mj}' = w_{mj} - cy_j$ for $j = 1 \to n+1$

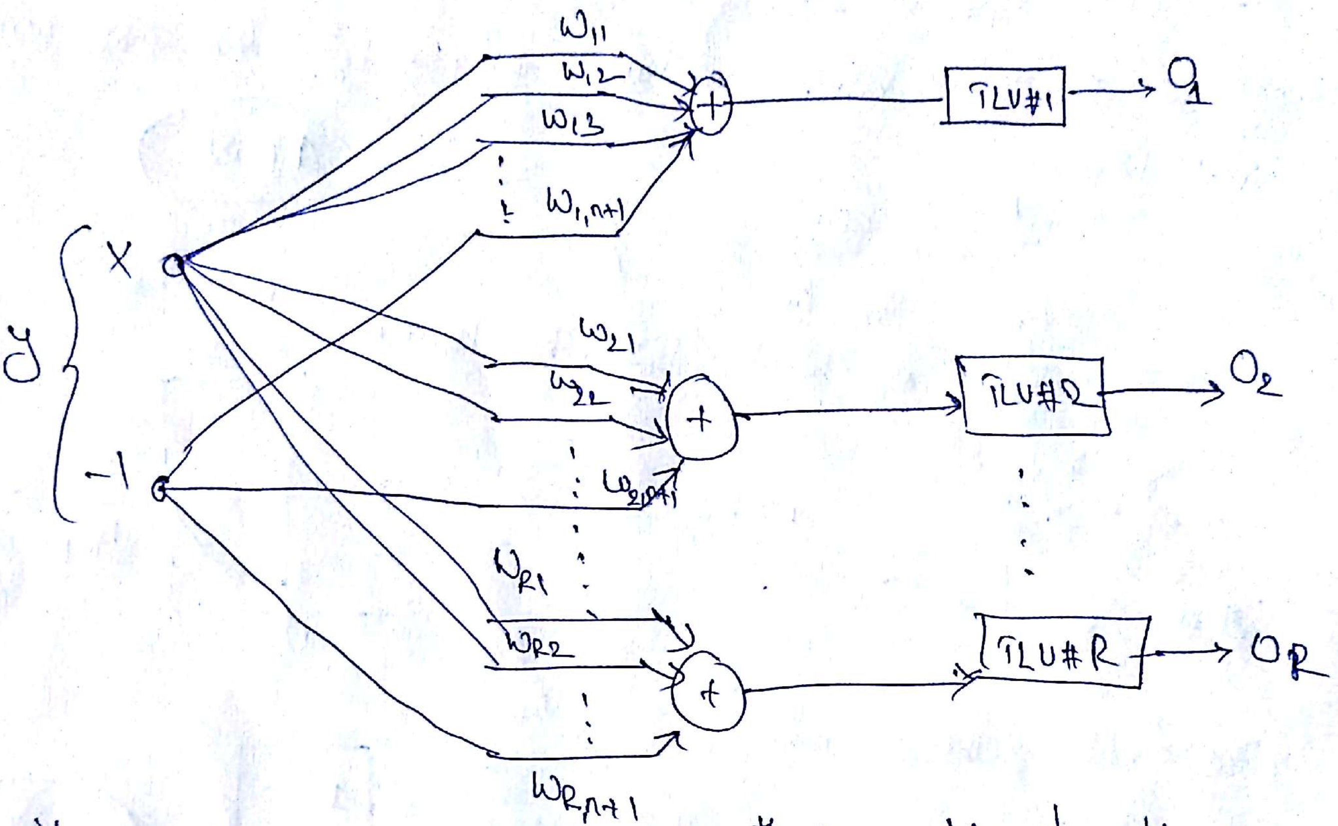
$w_{kj}' = w_{kj}$ for $k = 1,2 \ldots R$
where $k \neq i$; $m, j = 1 \to n+1$.

1 mark

By Eliminating maximum selector in a R-class linear classifier and replacing in first with R-Discrete Perceptrons.

→ The properly trained classifier should respond $i_0 = 1$, when $g_1(x)$ is larger than discriminant function $g_i(x)$, for $i = 1 \to R$.

Instead of signaling output maximum selector responding with $i_0 = 1$ a threshold limit of '1' may be used and output $O_1 = 1$ and $O_2 = O_3 = \dots O_R = -1$. (2 marks)



The weight adjustment during k^{th} step for n/w is

$$w_i^{k+1} = w_i^{k} + \frac{c^k}{2}\left(d_i^k - O_i^k\right) y^k \quad \text{for } i = 1, \to R.$$

5(a) Ans

Generalized Delta learning rule:- (5 Marks)

Generalized delta learning rule for feedforward layered neural networks.

Let us consider an multi layered network for representing an Delta learning rule. to minimize the error correction.

Diagram → (3 marks)

The negative gradient descent formulae for hidden layer

$$\Delta W_{ji} = -\eta \frac{\partial E}{\partial W_{ij}} \quad \text{for } j = 1 \to J \text{ and } i = 1 \to I$$

$$\therefore \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial (net_j)} \cdot \frac{\partial (net_j)}{\partial W_{ij}}$$

→ (2 marks) for derivation

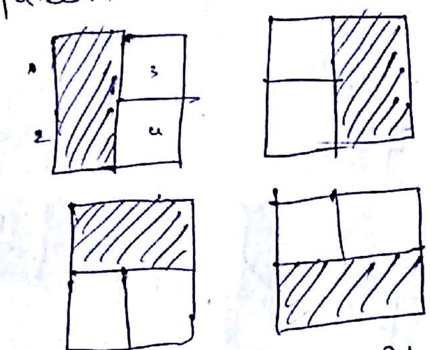
∴ The weight adjustments in hidden layer

$$\boxed{\Delta W_{ij} = \eta f'_j (net_j) z_i \sum_{K=1}^{K} \delta_{oK} W_{Kj}} \quad \text{for } j = 1 \to J \text{ and } i = 1 \to I$$

→ (1 mark)

5(B)
Ans

Given Pattern (4 marks)



class - I

class - II

(i)
Let Pattern vector be of order 2x2 and Each white column be represented as '1' and remaining square as '-1'.

$$\therefore P_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \text{ and } P_2 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$ for class-I vertical lines.

for class - II horizontal lines be → (1 m)

$$P_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \text{ and } P_4 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

The weight of matrix W linearly separable by a hyper plane b/w two categories.

$$WP_1 + b > 0 \; ; \; WP_2 + b > 0 \; ; \; WP_3 + b < 0 \; ; \; WP_4 + b < 0$$

$$\Rightarrow \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} w_{1,1} + w_{1,2} - w_{1,3} - w_{1,4} \end{bmatrix} > 0$$

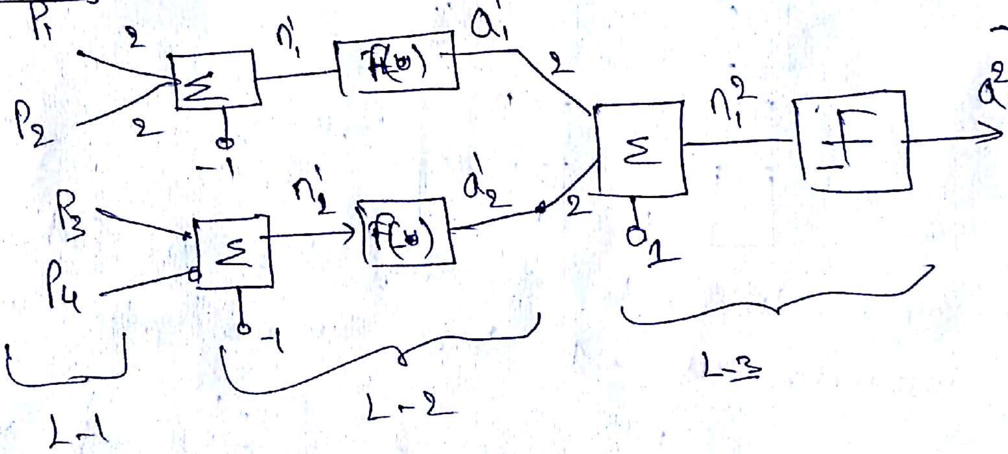
$$\left.\begin{array}{l} \Rightarrow \begin{bmatrix} -w_{1,1} - w_{1,2} + w_{1,3} + w_{1,4} \end{bmatrix} > 0, \\ \begin{bmatrix} w_{1,1} - w_{1,2} + w_{1,3} - w_{1,4} \end{bmatrix} < 0 \\ \begin{bmatrix} -w_{1,1} + w_{1,2} - w_{1,3} + w_{1,4} \end{bmatrix} < 0. \end{array}\right\} \rightarrow 1m$$

the first condition reduced to

$w_{1,1} + w_{1,2} > w_{1,3} + w_{1,4}$ and $w_{1,3} + w_{1,4} > w_{1,1} + w_{1,2}$ → (1m)

$\therefore \Rightarrow w_{1,1} + w_{1,3} > w_{1,2} + w_{1,4}$ and $w_{1,2} + w_{1,4} > w_{1,1} + w_{1,3}$ "

(ii) Multilayer network:-



→ 1m

6(a):- Recurrent Associative Memory:- 6 Marks

Ans.

→(i) Expanded view of Hopfield network → 2m
↳ Diagram

→ (ii) Simplified view of Hopfield network → 2m
↳ diagram

→ Performance Consideration:- → 2m

Problem :- (4 marks)

6(b):- Given vector sequences $s^{(1)}, s^{(2)}, s^{(3)}$

Ans.

$s^{(1)} = [1 \ -1 \ -1 \ 1 \ -1]^t$; $s^{(2)} = [1 \ 1 \ -1 \ -1 \ 1]^t$

$s^{(3)} = [1 \ -1 \ 1 \ -1 \ 1]^t$

The memory weight matrix W is written as

$$W = \sum_{i=1}^{p-1} s^{(i+1)} \cdot s^{(i)t} + s^{(1)} \cdot s^{(p)t},$$

where $s^{(i+1)} = \Gamma[W s^{(i)}]$

1m

$$\Rightarrow W = s^{(2)} \cdot s^{(1)t} + s^{(3)} \cdot s^{(2)t} + s^{(1)} \cdot s^{(3)t}$$

$$\therefore W = \begin{bmatrix} 3 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 3 & -3 \\ -1 & 3 & -1 & -1 & 1 \\ -1 & -1 & 3 & -1 & 1 \\ 1 & 1 & -3 & 1 & -1 \end{bmatrix}$$

To recall encoded sequence 's'. with input $s^{(1)}$ the layer response is 'a'.

1m

$$a = \Gamma[W s^{(1)}] = \ldots$$

$$\text{where } a = \Gamma\left[s^{(2)} \cdot s^{(1)t} + \ldots + s^{(k+1)} \cdot s^{(k)t} + \ldots + s^{(1)} \cdot s^{(p)t}\right] s^{(k)}$$

$$\therefore \Rightarrow a = \Gamma[W s^{(1)}] = \Gamma[3 \ 7 \ -5 \ -5 \ 5]^t$$

$\therefore$ the noise vector η is given as

$$\eta = \sum_{i \neq k}^{p} (s^{(i+1)})(s^{(i)t} \cdot s^{(k)}).$$

1m

$$\Rightarrow \eta_1 = \begin{bmatrix} 3 \\ 7 \\ -5 \\ -5 \\ 5 \end{bmatrix} - 5 \cdot s^{(2)}$$

$$= \begin{bmatrix} 3 \\ 7 \\ -5 \\ -5 \\ 5 \end{bmatrix} - \begin{bmatrix} 5 \\ 5 \\ -5 \\ -5 \\ 5 \end{bmatrix} = [-2 \ 2 \ 0 \ 0 \ 0]^t$$

if magnitude of each component η_1 is smaller than magnitude of corresponding of $5\,S^{(2)}$ which recall further update in memory

$$\Gamma\left[W S^{(2)}\right] = \Gamma\left[5 \quad -7 \quad 5 \quad -3 \quad 3\right]^t$$

the recall required with noise term.

$$\eta_2 = \left[0 \quad -2 \quad 0 \quad 2 \quad -2\right]^t$$

now $S^{(1)}$ is recalled due to thresholding operation

1M

$$\Gamma\left[W S^{(3)}\right] = \Gamma\left[5 \quad -7 \quad -3 \quad 5 \quad -5\right]^t$$

with the noise vector

$$\eta_3 = \left[0 \quad 2 \quad 2 \quad 0 \quad 0\right]^t$$

$\therefore$ The recall sequence of $S^{(1)}, S^{(2)}, S^{(3)}$ terminates.

7(a):- **Bidirectional Associative memory:-** (6 marks)

Ans

Architecture :- (2 marks)

Mathematical modeling of BAM:- (4 Marks)

The additive dynamic of BAM can be written as

$$\dot{x}_i(t) = -A_i x_i(t) + \sum_{j=1}^{m} f_j(y_j(t))\, w_{ij} + I_i \quad ; i = 1 \to N \to (a)$$

$$\dot{y}_j(t) = -A_j y_j(t) + \sum_{i=1}^{N} f_i(x_i(t))\, w_{ji} + J_j , \quad J = 1 \to N \to (b)$$

where I_i & I_j are net external i/p to units i & j

A_i and $f_i(\cdot)$ could be different for each other.

If neuron reaches equilibrium state $\dot{x}_i(t) = 0$ and $\dot{y}_j(t) = 0$ the output of network is computed.

Let us consider x-layer as i/p layer and y-layer as output layer.

(i) Computing net input values of nodes on y-layer

$$Net^y = WX \to (c)$$

$$\Rightarrow Net_j^y = \sum_{i=1}^{N} w_{ji} x_i \to (d) \quad \to 1m$$

(ii) Computing the outputs (new values of y-layer units) of output layer

$$y_j = f_j\left(\sum_{i=1}^{N} w_{ji} x_i\right). \to (e)$$

where $f_j \to$ is non linear activation function of neuron 'j'.

$\therefore$ typical activation function is $\quad \to$ 1M.

$$y_j(l+1) = \begin{cases} +1 & net_j^y > 0 \\ y_j(l) & net_j^y = 0 \\ -1 & net_j^y < 0 \end{cases} \to (f)$$

where l is step number and $J = 1 \to M$.

(iii) Compute the net input value of nodes on X-layer

$$Net^X = WY \rightarrow (g) \longrightarrow 1m$$

$$\Rightarrow net_i^X = \sum_{j=1}^{M} y_j w_{ij} \rightarrow (h)$$

(iv) Compute the outputs (new values of X-layer units) of output layer nodes

$$x_i = f_i\left(\sum_{j=1}^{M} w_{ij} y_j\right) \rightarrow (i) \longrightarrow 1M.$$

where $f_i(\,)$ is non linear activation function of neuron i on X-layer.

$$x_i(l+1) = \begin{cases} +1 & net_i^X > 0 \\ x_i(l) & net_i^X = 0 \\ -1 & net_i^X < 0. \end{cases}$$

$l \rightarrow$ step number

$i = 1 \rightarrow N$.

7(b) Ans

Improved Coding memories:- 4 Marks

The generic bidirectional associative memory described in defining error-free retrieval of stored pairs.

let the stability of updates within bidirectional associative memory (a, b).

the performance can be improved by possibly reshaping its energy function.

During decoding energy function decreases until reaches minimum at (a^f, b^f)

$$\therefore E(a,b) = -a^{ft} W . b^f \rightarrow (a) \longrightarrow 1M$$

The Energy E has local minimum at (a^f, b^f) when no vector at $HD = 1$ from a^f and b^f.

The gradient energy function at m^{th} association pair is

$$\nabla_a E\left(a^{(m)}, b^{(m)}\right) = -\sum_{i=1}^{P} a^{(i)} . b^{(i)t} . b^{(m)} \rightarrow (b)$$

Scanned by CamScanner

the corresponding energy change ΔE_{ak} is

$$\Delta E_{ak}\left(a^{(m)}, b^{(m)}\right) = -\left(\sum_{i=1}^{p} a^{(i)} b^{(i)t} \cdot b^{(m)}\right) \Delta a_k^{(m)} \rightarrow (c) \rightarrow 1M$$

To ensure local minima of energy $E\left(a^{(m)}, b^{(m)}\right)$ multiple encoding has been proposed as. for multiple training order 'q' for $\left(a^{(m)}, b^{(m)}\right)$ pair

$$\Delta W = (q-1)\, a^{(m)} \cdot b^{(m)t} \rightarrow (d)$$

$\therefore$ weight matrix obtained for q training is

$$W = q\, a^{(m)} \cdot b^{(m)t} + \sum_{i \neq m}^{p} a^{(i)} \cdot b^{(i)t} \rightarrow (e) \quad //.$$

$\Rightarrow$ $\boxed{W \simeq q\, a^{(m)} \cdot b^{(m)t}}$ //.

$\rightarrow$ 2M

8(a): Ans.

Relations between fuzzy and crisp sets:- 5 marks

(i) Crisp Relations:- $\longrightarrow$ 2 marks

↳ Cardinality of crisp relations

↳ operations on crisp

↳ union ; intersection ; complement ; containment

↳ Composition of Relations.

(ii) Fuzzy relations:- $\longrightarrow$ 3 marks

↳ Cardinality of fuzzy relations

↳ Operations

↳ union ; intersection ; complement ; containment.

↳ Cartesian product and composition.

Problem :- (5 Marks)

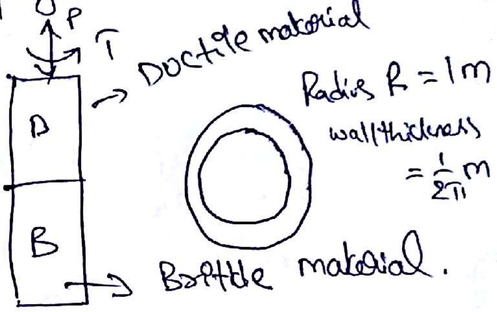
8(b) :- Let fuzzy set A be the region in (P, T) space for material surface 'D'.

using merit failure function $\mu_A = f\left(\left[P^2 + 4T^2\right]^{1/2}\right)$. → (1m)

Let fuzzy set 'B' to be the region in (P, T) space for which material 'B' is.

using merit failure function $\mu_B = g\left(P - \beta|T|\right)$ → (1m)

Let the function f and g be membership function interval [0, 1]



Problems defined below :- (3m)

1) $\underset{\sim}{A} \cup \underset{\sim}{B}$ is set of loading for one expects either B (or) D materials is "safe"

2) $\underset{\sim}{A} \cap \underset{\sim}{B}$ is set of loading that both material B and D are "safe".

3) $\overline{\underset{\sim}{A}}$ and $\overline{\underset{\sim}{B}}$ are set loading for D and B are "unsafe"

4) $\underset{\sim}{A} | \underset{\sim}{B}$ is set of loading for D is safe and B is unsafe

5) $\underset{\sim}{B} | \underset{\sim}{A}$ is set of " for B is safe but D is unsafe

6) $\overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = \overline{\underset{\sim}{A}} \cup \overline{\underset{\sim}{B}}$ loadings are not safe for both 'B' and 'D'

7) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = \overline{\underset{\sim}{A}} \cap \overline{\underset{\sim}{B}}$ loading both are safe. either B (or) D.

9(a) Ans. Fuzzy sets :- (5m)

Defination, notations → (2M)

fuzzy set operations → (1m)

Properties of fuzzy sets → (2m)

Scanned by CamScanner

9(b):- Classical sets:- 5 marks

Defination:-

Operation on classical sets $\Big\} \rightarrow$ 2m

Properties:-

(i) Associative
(ii) Distributive
(iii) idempotency
(iv) Identity
(v) Transitivity
(vi) Involution.

$\Big\} \rightarrow$ 3m

Venn diagrams:-

10(a):- Ans

5 marks

Given $\tilde{A} = \left\{ \frac{1}{1.0} + \frac{0.65}{1.5} + \frac{0.4}{2.0} + \frac{0.35}{2.5} + \frac{0}{3.0} \right\}$

$\tilde{B} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.6}{2.0} + \frac{0.25}{2.5} + \frac{1}{3.0} \right\}$

$\tilde{C} = \left\{ \frac{0.5}{1.0} + \frac{0.25}{1.5} + \frac{0}{2.0} + \frac{0.25}{2.5} + \frac{0.5}{3.0} \right\}$

(i) $\tilde{A} \cap \tilde{B} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.4}{2.0} + \frac{0.25}{2.5} + \frac{0}{3.0} \right\}$

(ii) $\tilde{A} \cup \tilde{B} = \left\{ \frac{1}{1.0} + \frac{0.65}{1.5} + \frac{0.6}{2.0} + \frac{0.35}{2.5} + \frac{1}{3.0} \right\}$

(iii) $\tilde{A}^c = \left\{ \frac{0}{1.0} + \frac{0.35}{1.5} + \frac{0.6}{2.0} + \frac{0.65}{2.5} + \frac{1}{3.0} \right\}$

(iv) $\tilde{B}^c = \left\{ \frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.4}{2.0} + \frac{0.75}{2.5} + \frac{0}{3.0} \right\}$

(v) $\tilde{A}^c \cup \tilde{B}^c = \left\{ \frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.6}{2.0} + \frac{0.75}{2.5} + \frac{1}{3.0} \right\}$.

10(b) Membership function:- (5 marks)

Defination:- → 1m

features of the membership function:-

(i) Core of membership:-

(ii) Support of membership:-

(iii) Boundaries of membership:-

(iv) normal fuzzy set:-

(v) Convex fuzzy set:-

3m

various form of membership functions:- → 1m

~~(or)~~:-

11(a):- Process of fuzzification and De-fuzzification:- 6m

(i) fuzzification:- (explanation)

↳ membership function representation:- (explanation)

2m

(ii) De-fuzzification:- methods:-

(i) maximum membership Principle

(ii) Centroid method

(iii) weighted method

(iv) mean maximum membership

(v) Center of sums

(vi) Center of largest area

4m

11 (b) Ans

Problem :-

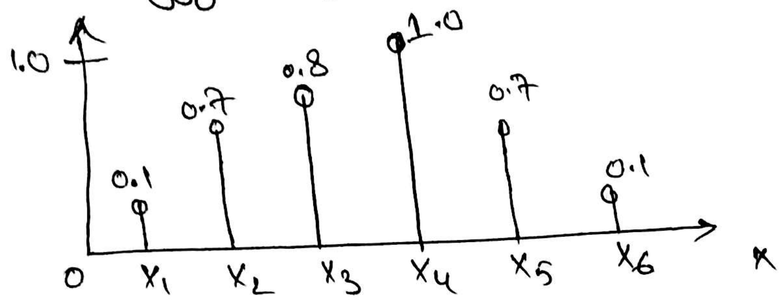
Given

$\mu(x_i)$	x_1	x_2	x_3	x_4	x_5	x_6
$\tilde{A}$	0.1	0.7	0.8	1.0	0.7	0.1
$\tilde{B}$	1.0	0.9	0.5	0.2	0.1	0

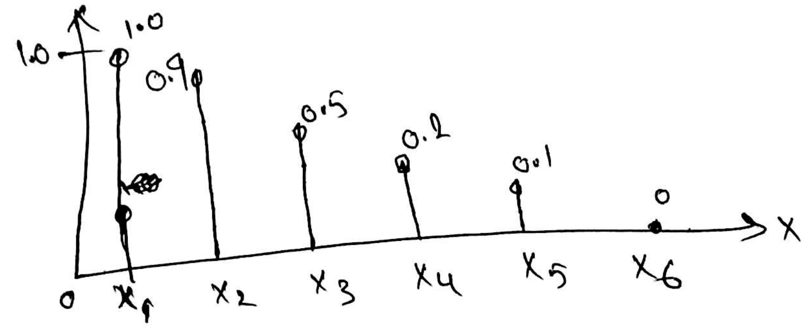
4m

To express cut sets for given fuzzy sets using Zadeh's notation

(a) A discrete fuzzy set of 'A' represented as



(b) A discrete fuzzy set of 'B' represented as



→ 2m

2m

Given to find :-

(i) $(\tilde{A})_{0.6} = \left\{ \frac{1}{x_1} + \frac{0}{x_2} + \frac{0}{x_3} + \frac{0}{x_4} + \frac{1}{x_5} + \frac{0}{x_6} \right\}_{\tilde{A}} \Rightarrow \{x_1, x_5\}$

(ii) $(\tilde{B})_{0.5} = \left\{ \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \frac{0}{x_4} + \frac{0}{x_5} + \frac{0}{x_6} \right\}_{\tilde{B}} \Rightarrow \{x_1\ x_2\ x_3\}$

(iii) $(\tilde{A} \cup \tilde{B})_{0.8} = \left\{ \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} \right\}_{\tilde{A}} \cup \left\{ \frac{1}{x_1} + \frac{1}{x_2} \right\}_{\tilde{B}}$

$= \left\{ \frac{1}{x_1} + \frac{1}{x_2} \right\}_{\tilde{A} \cup \tilde{B}}$

Scanned by CamScanner

(iv) $(\tilde{A} \cap \tilde{B})_{0.9} = \left\{ \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \frac{1}{x_4} \right\}_{\tilde{A}} \cap \left\{ \frac{1}{x_1} + \frac{1}{x_2} \right\}_{\tilde{B}}$

$= \left\{ \frac{1}{x_1} + \frac{1}{x_2} \right\}_{\tilde{A} \cap \tilde{B}}$ //.