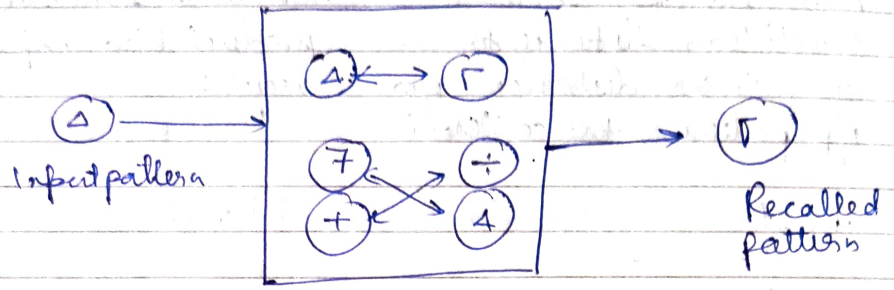
ASSOCIATIVE MEMORY

DATE: / /

An associative memory which belongs to the class of single layer feedforward or recurrent network architecture depending on its association capability, exhibits Hebbian learning.



(Δ, F), (7, 4) & (+, ÷) are associated pattern pairs.

⟷ associations symbols

⇒ An associative memory is a storehouse of associated patterns which are encoded in some form. When the storehouse is triggered or incited with pattern, the associated pattern pair is recalled or output.

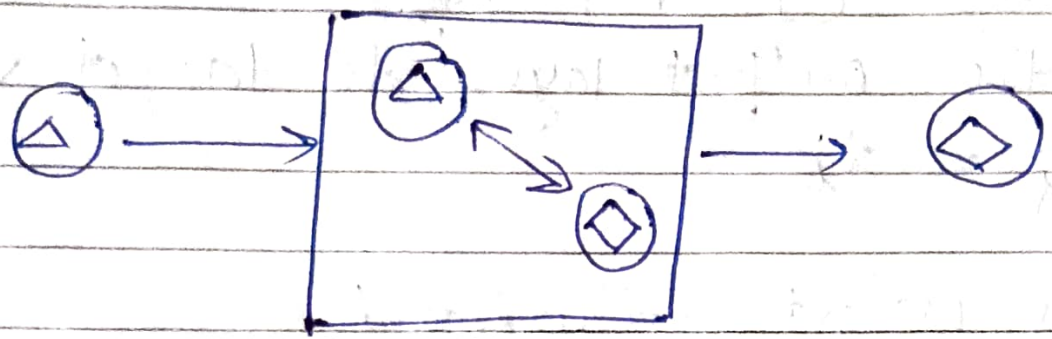
⇒ If the associated pattern pairs (x, y) are different and if the model recalls a y given an x or vice versa, then it termed as heteroassociative memory. useful for association of patterns
fiber defect identification

⇒ If x & y refers to the same pattern, then model is termed as autoassociative memory.
useful for image refinement
character recognition

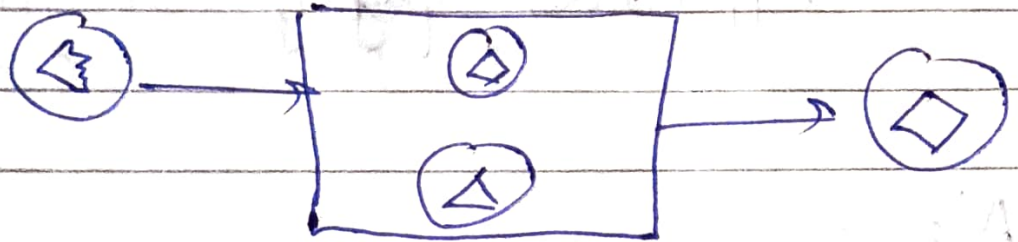
B+ TM
Be Positive.....

DATE: __/__/__

Autoassociative correlation memories are known as auto correlators & other hetero correlators.



Heteroassociative memory



Autoassociative memory.

⇒ Autoassociative Memory :-

It uses Hebb's Rule.
There associative net has same no. of neurons in the input & output layers as the training and the target output vectors are same.
The association b/w the patterns is stored by their weights

→ Training algo

(1) Set all the weights to zero
$W_{ij} = 0 \quad (j = 1, 2, \ldots n \;\&\; i = 1, 2, 3, \ldots m)$

(2) Repeat step (3) to 5 for all input-output vectors.

B+
Be Positive.....

DATE: / /

(3) Activate the input layer to training input vector.

$$x_j = (j = 1 - n)$$

(4) Activate the output layer to target vector

$$y_j = t_j$$

(5) Perform weight adjustment.

$$w_{ij}(new) = w_{ij}(old) + \eta x_j y_j \quad \text{Hebb Rule}$$

→ Testing Algo –

The weight obtained from training algo is considered for testing.

Set the activation function for the network.

$$net = w^T x$$

The activation function can be binary or bipolar step function.

→ Outer Product Rule –

Outer product rule is an alternative for Hebb's rule for weight adjustment.

Input vector $x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$, $t = \begin{Bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{Bmatrix}$

Be Positive.....

DATE: __/__/____

The outer product is obtained by performing multiplication of input vector with transpose of target vector.

$$Y = x t^T = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}_{n \times 1} \cdot \begin{bmatrix} t_1 & \cdots & t_j & \cdots & t_m \end{bmatrix}_{1 \times m}$$

$$W_{ij} = \begin{bmatrix} x_1 t_1 & \cdots & x_1 t_j & \cdots & x_1 t_m \\ \vdots & & \vdots & & \vdots \\ x_i t_1 & \cdots & x_i t_j & \cdots & x_i t_m \\ \vdots & & \vdots & & \vdots \\ x_n t_1 & \cdots & x_n t_j & \cdots & x_n t_m \end{bmatrix}_{n \times m}$$

So weight is

$$W_{ij} = \sum_{P=1}^{P} x_i(P)\, t_j^T(P) \quad \text{or}$$

$$\boxed{W = \sum_{P=1}^{P} x(P)\, t^T(P)}$$

Ex → $x = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$ $t = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$

Output vector $Y = x \cdot t^T$

B+ Be Positive.....

$$W = \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix}_{4\times4}$$

(i) Test the network

$$net = W^T X = \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ -4 \\ 4 \\ 4 \end{bmatrix}$$

Activation function:

$$y_i = f(net) = \begin{cases} 1 & \text{if } net > 0 \\ -1 & \text{if } net \leq 0 \end{cases}$$

$$= \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

(ii) testing the network with one missing [illegible]

DATE: / /

⇒ Hetro-associative Memory —

It retrieves the pattern from memory where input and output (target) patterns are different.
The correlation b/w the input pattern & the output pattern is determined by the weights.
Training algo is similar.

Ex →

	x_1	x_2	x_3	x_4	t_1	t_2
Pattern 1	1	0	1	0	1	0
" 2	1	0	0	1	1	0
" 3	1	1	0	0	0	1
" 4	0	0	1	1	0	1

Outer Product. $W = \sum_{P=1}^{P} x(P)\, t^T(P)$

for Pattern 1. , for 2 for 3 for 4

$W_{11} = \begin{bmatrix} 1 \end{bmatrix}$ W_{22} W_{33} W_{44}

final weights $W = W_{11} + W_{22} + W_{33} + W_{44}$