UNIT 3

BI-DIRECTIONAL ASSOCIATIVE MEMORIES AND ART

3.1 AUTO ASSOCIATIVE MEMORY NETWORKS –

Q.1 What is an Auto associative net? State the application algorithm of an auto associative net.

Ans. In auto associative net the training input and target output vectors should be identical or save. The auto associative net training is often called storing the vectors which may be binary or bipolar. A stored vector can be retrieved from noisy input if the input is sufficiently similar to it. The ability of the net to reproduce a stored pattern from a noisy input is used to determine the performance of the net. The performance of the net is better for bipolar vectors than the binary vectors. Setting weights to zero improves the net is ability to generalize or increase the plansibility of the net.

Architecture– figure 3.1 shows the architecture of a auto associative neural net. This archtecture resembles a single layer feed forward network as discused before. It consists of only one layer of weighted interconnections. There exists 'n' number of input neurons in the input layer and 'n' number of output neurons in the output layer. The training process is based on the Hebb learning rule. This is a fully inerconnected network, where in the inputs and the output are same, hence called an auto associative network.

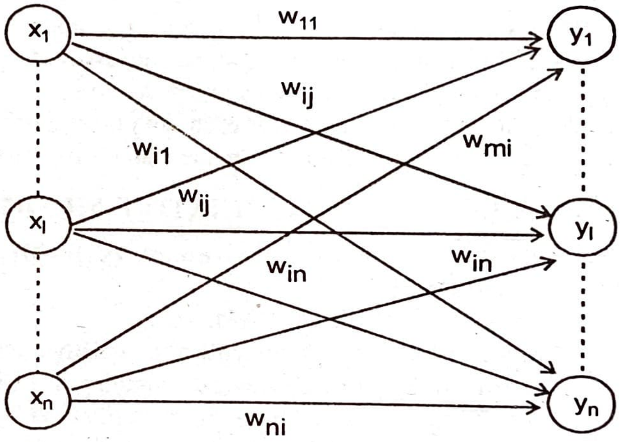

Fig. 3.1 Architecture of an auto associative net

Training Algorithm – Initially, the weights of the auto associative network are taken as zero. This is a pattern associative supervised leaning network where in both the inputs and outputs are known, as a result the activations are set for both the input and output units. Then the final weight are calculated based on the hebb learning rule. The algorithm is given below.

Step 1: initialize allweights i = 1,........n j= 1,.............n

$$wij = 0;$$

Step 2: For each vector to be stored follow steps 3-4.

Step 3: Set activation for each input unit i = 1,........n.

$$xi = si$$

Step 4: Set activation for each output unit j = 1,.........n

$$y_j = s_j$$

Step 5: Adjust the weight for i = 1 to n and j = 1 to n.

$$W_{ij\,(new)} = W_{ij\,(old)} + x_i y_i$$

The weight can also be determined from Hebb learning as

$$W = \sum_{p=1}^{P} S^T(P)\, S(P)$$

Application Algorithm– based on the application procedure only the performance of the network is evaluated. This stepwise application algorithm is given below. This is purely based on the calculation of the net input for the final weights obtained and applying the activations over the calculated net input.

Step 1: Initialize the weights.

Step 2: For each testing input follow steps 3 to 5.

Step 3: Set the activations of the input units equal to the input vector.

Step 4: Compute net input to each output Unit $j = 1, \ldots\ldots n$

$$y_{-inj} = \Sigma x_i w_{ij}$$

Step 5: Apply Activation function $(j = 1, \ldots\ldots n)$

$$f_j = f(y_{-inj}) = \begin{Bmatrix} 1 \text{ if } y_{-inj} > 0 \\ 0 \text{ if } y_{-inj} \leq 0 \end{Bmatrix}$$

Storage Capacity – An important criterion for an associtive network is the number of patterns or pattern pair it can store. The capacity of an auto associative net depends on the number of components in the stored vector. More number of vectors can be stored if they are mutually orthogonal. Note that n-1 mutually orthogonal bipolar vector each with n components can be stored using the sum of the outer product weight matrices. The two vectors a and b are orthogonal if $\Sigma a_i b_i = 0$

3.2 BIDIRECTIONAL ASSOCIATIVE MEMORY AND STRUCTURE OF BAM

Q.2 Define Bidirectional Associative memory (BAM). Draw the architecture of BAM and discuss training Algorithm. **(KUK, June 2010)**

Q.3 Explain the structure of BAM. **(KUK, May 2010)**

Ans. Kosko developed the Bi-directional associative memory (BAM) in the year 1988. It is a hetro associative recent neural network consisting of two layers. The net iterated by sending a signal back and forth between the two layers until each neuron's activations remains constant for several steps. The net associates a set of patterns by summing bipolar correction matrices. This type of net consists of two layers of neurons, which are connected by means of directional weighted connection paths. The iteration is carried out by sending signals back and forth between two layers until all neurons reach equilibrium. The net can respond to input on either layer. The layers in this case are referred as X-layer and Y- layer instead of input and output layer, because the weights are bi-directional and the algorithm alternates between updating the activations of each layer. Three forms of BAM are (i) Binary (ii) Bipolar (ii) Continuous. But the architecture for the three types remains the same.

Structure–A BAM network has n-units in A-layer & m-units in B-layer. It is a fully inter-connected network where in the input and output are different.

* An initializing vector b is applied at input to layer A of neurons. The neurons are assumed to be bipolar binary. The input passes through linear connection layer and bipolar threshold functions as:

$$A = F(W b)$$

or

$$a_i = F \sum_j w_{ij} b_j \qquad \ldots\ldots(1)$$

* Now, this output vector A is applied as input to the layer B of neurons. It is processed in layer B through similar matrix multiplication and bipolar threshold but the processing now uses transposed matrix W^t of layer B.

$$B = F(W^t A)$$

or

$$b_j = F\left(\sum_i W_{ij} a_i\right)$$

Where

A = Vector of output from layer A
B = Vector of output from layer B
W = Weight matrix between layers A & B.
W^t = Transpose of matrix W
F = Activation function

This process continus until the network arrives to a stable point or A& B stop changing.

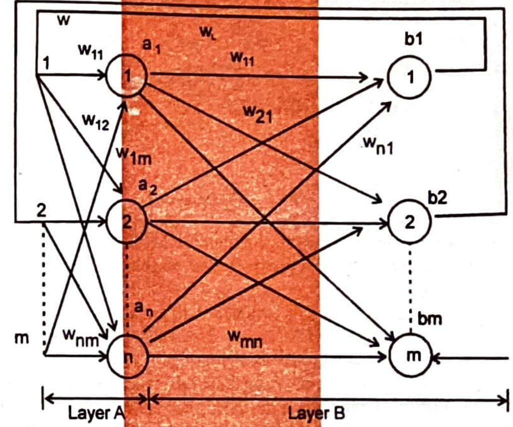

Fig. 3.2 (a) General diagram

The activation function used is a simple binary-bipolar or simple step function. It is given by :

$$OUT_i = \begin{Bmatrix} 1, \text{ if } a_i > 0 \\ 0, \text{ if } a_i < 0 \\ OUT_i, \text{ if } a_i = 0 \end{Bmatrix}$$

Layer 0 performs no computation & serves only as distribution points for outputs of layer 2 to matrix W^t.

In the simplified diagram, layers A & B operate in an alternate fashion-first transferring neurons output signals to right using matrix W then to the left by using matrix W^t, respectively. The input-output transformation is highly non-linear due to the threshold based state transitions.

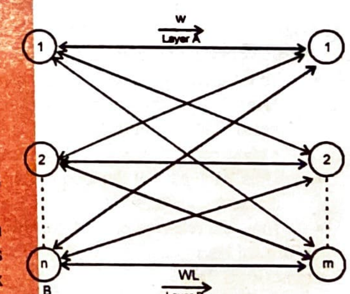

Fig. 3.2(b) Simplified diagram

3.3 ENCODING ASSOCIATIONS AND RETRIEVING A STORED ASSOCIATION–

Q.4 Explain the procedure for Encoding and retrieving associations in BAM structure by taking a suitable example of your choice.

(KUK, June 2007)

Q.5 Using Suitable diagram explain the Basic Bidirectional associative memory configuration. Also describe the procedure for encoding association & retrieving the stored associations.

(KUK, June 2008)

Ans. Encoding the Associations– A network is usually trained to recognize a set of memories. These comprise a training set composed of vector pairs A & B.

- The coding of information into the bi-directional associative memory is done using customary outer product rule or by adding p cross- correlation matrices. The formula for weight matrix is :

$$W = \sum_{i=1}^{p} a^{(i)} b^{(i)\,t}$$

Where $a^{(i)}$ & $b^{(i)}$ are bipolar binary vectors.

- The above equation is equivalent to the Hebbian learning rule yielding the following weight values:

$$W_{ij} = \sum_{m=1}^{p} a_i^{(m)} b_j^{(m)}$$

- Suppose, we wish to train a network to remember 2 binary vector pairs. Here, vector A_i have same number of components as vector B_j. The association can also been formed between vectors of different sizes.
- Here, an input vector is applied to A through W to produce its associated output vector at B, which in turn is applied through W^t to again produce A. Thus, forming resonance around the loop.
- BAM has the capability to generalise since, ever if an incomplete or partially incorrect vector is applied memory at B. which in turn tends to correct errors at A.
- Although, several passes may be required but the network converges to the nearest stored memory.
- BAM is unconditionally stable to any weight network which arise from the transpose relationship P between the two weight network which means that any set of associations can be learned with out risk of instability.

Retrieving a stored Association– Long term memories are stored in the weight arrays W and W^t. Each memory consists of two vectors, which approx at the outputs of layer A & B.

- To retrieve a stored memory, all or part of vector 1 is momentially forced on to the outputs of layer A. 1 is then removal and network is allowed to stablize, producing the associated vector 2 at he output of layer B. 2 then operates through the transpose matrix W^t to produce a close replica of original input vector 1 at the output of layer A.
- Thus, each pass of the loop will allow vector outputs to come closer to stored memory until it reaches stable point
- Thus, the retrieval process consists of the following steps:

First forward Pass $a' = F[Wb^0]$

First backward Pass $b' = F[W^t a']$

Second backward pass $b' = F[W^t a']$

$\frac{k}{2}$ backward pass : $b^k = F[W^t a^{k-1}]$

- Suppose one of the stored patterns a (m') is presented memory.

The retrieval process proceeds form equ ...(1)

$$b = F\left[\sum_{m=1}^{p} \left(b^{(m)} a^{(m)t}\right) \alpha^{(m')}\right]$$

which further reduces to

$$b = F\left[nb^{(m')} + \sum_{m \neq m'}^{p} b^{(m)} a^{(m)t} a^{(m')}\right]$$

The above equation contains is signal term $nb^{(m')}$ addition with noise term n of values–

$$\eta = \sum_{m \neq m'}^{p} b^{(m)} \left(a^{(m)t} a^{(m')}\right)$$

- Suppose
- Assume that for m = 1, noise term η reduces to zero.

  Therefore, immediate stabilization exact association b = f(m') occur with in only a single pass through layer B.
- If the input vector is a distorted version of pattern a (m'), the stabilization at b(m) is not imminent and depends upon the hamming distance between the key vectors & prototype vectors.

- For a perfect retrieval in a single pass, hamming distance has to be $\frac{n}{2}$; for n=1, 2.........P, m≠m',

  then n = 0.

Procedure for Encoding & Retrieving a stored Association–

Step 1: Initialize activations to 0.

Step 2: Initialize the weight to store a set of P patterns.

$$W_{ij} = \sum_{m=1}^{10} a_i^{(xi)} b_j^{(xn)}$$

Step 3: Set activation of layer A to current input pattern

Step 4: Input pattern b is presented to layer B.

Step 5: Activation unit in B-layer and net input is calculated as :

B =ΣWA. compute activation b = F(B) send signals to layer A.

Step 6: Update activation unit in layer A. Net input is calculated by A = ΣWB. Then compute activations a = F(A). Send signals to layer B.

Step 7: Test for convergence.

The stopping condition may be that activation vector a & b have reached equilibrium.

Example : Use BAM to store following 3 pair of vectors.

$X_1 = (1\ -1\ -1)$ $y_1 = (1\ -1)$

$X_2 = (-1\ 1\ -1)$ $y_2 = (-1\ 1)$

$X_3 = (-1\ -1\ 1)$ $y_3 = (1\ -1)$

Examine network for test vector (-1 -1 -1)

Solution : The weight vector

$$W = W_1 + W_2 + W_3$$

$$W = X_1^T y_1 + X_2^T y_3 + y_3$$

$$W = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}[1\ -1] + \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}[-1\ 1] + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}[1\ -1]$$

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

Output for test vector $X = (-1\ -1\ -1)$ is

$$y = XW = [-1\ -1\ -1]\begin{bmatrix} 1 & -1 \\ -3 & 3 \\ 1 & -1 \end{bmatrix}$$

$$y = [1\ -1]$$

The test paths associates with I[st] pattern.

3.4 MEMORY CAPACITY

Like the hopfield network, BAM has restriction on the maximum number of associations that can be stored. If this iimit is exceeded, the network may produce incorrect outputs.

Kasko should that the upper limit on the number P of pattern pair that can be stored & successfully retireved is min [n,m]. If n < p or P > n then the noise term exceeds the signal term.

(a) Similarly, for processing of a signals & noise mixture by layer B, it is expected that the dominance of signal term over noise component is maintained for P<m. Hence, a rough heuristic estimate on memory storage capacity is,

$$P \le \min [\min]$$

(b) This assumes that capacity has been maximize through "even coding" i.e. the number of "+1" components equals the number of "-1" components in bipolar vector.

(c) The capacity of a hopfield net can also be extended to BAM. For eg. L is randomly closer & evenly coded & if $L < \dfrac{n}{(2\log_2 n)}$ where n is the no. of neurons in smaller layer, then only a small fraction of memories could be retrieved. For eg : if n = 1024 then,

L < 51

This implies that large systems could store only modest number of associations.

(d) Recent works have shown that a BAM can have up to 2 stable states, if a threshold term is selected for each neuron. This configuration of BAM is termed as non-homogeneous BAMs the modified neuron transfer functions becomes.

$$OUT_i(n+1) = \begin{Bmatrix} 1, \text{ if } NET_i(n) > T_i \\ 0, \text{ if } NET_i(n) < T_i \\ OUT_i(n), \text{ if } NET_i(n) = T_i \end{Bmatrix}$$

(e) By choosing an appropriate threshold for each neuron the number of states can be made anything from 1 to 2^n but these states cannot be selected at random, where

$$L < \frac{(0.68)n^2}{[(\log_2 n)] + 4]2}$$

and each vector has $4 + \log_2 n$ entries equal to its rest equal to -1 then, it is possible to construct a non-homogenous BAM with 98% of each vectors as stable states.

(f) The restrictions on the number of ones in the input vectors constitutes a serious problem. There was no theory that enables to code an arbitrary set of vectors into 'sparse sets".

(g) Perhaps, the more serious is the problem of incorrect convergence, which means that the net may not produce the closest association.

Q.6 Differetiate between continuous Bi-directional Associative memory (BAM) and discrete BAM and state the algorithm of a discrete BAM. (KUK, May 2011)

Ans. There exist two types BAM. They are

Discrete BAM

Continuous BAM

1. **Discrete BAM–** The binary and bipolar forms of BAM are closely related. In each case the weight are found from the sum of the Hebb outer product of the bipolar form of the training vectors pairs. The activation function used is a step activations function with the non-zero threshold. But generally, we know that the bipolar vectors improve the performance of the net.
The weight matrix to store the set of input and target factors SCP : t(P), P = 1.........P, where

$$S(P) = (S_1(P) \ldots S_i(P), \ldots \ldots Sn(P))$$

and can be determined with the help of Hebb rule. for binary input vectors, the weight matrix can be determined by the formuls.

$$Wij = \sum_p (2s_i\ (p) - 1)'(2t_j(p) - 1)$$

For bipolar input vectors, the weight matrix is given by:

$$Wij = \sum_p s_i(p)t_j(p)$$

The activation function for discrete BAM depends on whether binary of bipolar vectors are used. The activation function is an appropriate step function.

2. **Continuous BAM–** The continuous BAM was introduced by KOSKO, 1988/ A capability to transfer the input smoothly and continuously into the respective output in the range between **[0,1]**. The continuous BAM uses logistic sigmoid function aas the activation function for all units.
For binary input vectors (SCP), t (P), P= 1, 2,.....P the weights are determine by the formula.

$$Wij = \sum_p (2s_i\ (p) - 1)\ (2t_j(p) - 1)$$

y-layer : The logistic sigmoid activation function is given by:

$$f(y_{inj}) = \frac{1}{1 + \exp(-y_{inj})}$$

If bias is included in calculating the net input then

$$f(f_{inj}) = b_j + \sum x_i W_{ij}$$

X-layer : The logistic sogmoid activation function is given by:

$$f(x_{inj}) = \frac{1}{1 + \exp(-x_{inj})}$$

if bias is included in calculating the input then

$$X_{ini} = b_i + \sum y_j W_{ij}$$

3.5 INTRODUCTION TO ART

Q.7 Explain the concept behind Adaptive Resonance Theory (ART). How is an ART net designed for both stability and plasticity? **(KUK, May 2011)**

Ans. Adaptive Resource Theory or ART refers to a class of self organizing neural architecture that cluster the pattern space and produce appropriate weight vector template convential artificial neural networks have failed to solve the stability-plasticity dilemma. A network remains open to new learning without washing away previously learned codes. Too often, learning a new pattern erases or modifies previous training. If there is only a fixed set of training vectors, the network can be cycled through these repeatedly and may eventually learn all.

ART network and algorithms maintain the plasticity required to learn new patterns, while preventing the modification of pattern that have been learned previously.

A stable net will not return the pattern to a previous cluster. Some nets achieve stability by gradually reducing the learning rate as the same set of training patterns is presented many times. However this does not allow the net to readily learn a new pattern that is presented for the first time after a number of training epochs have already taken place. The ability of a net to learn a new pattern equally well at any stage of learning is called plasticity. ART nets are designed to be both stable and plastic.

The ART network is a vector classifier. It accepts an input vector and classifies it into one of the categories depending upon which of the stored pattern it most resembles. if the input vector does not match any stored pattern, a new category is created by storing pattern, that is like the input vector. Once a stored pattern is gound that matches the input vector within a specified tolerance, that pattern is adjusted to make it still more like the input vector.

In ART, the changes in the activation of units and weights are governed by differential equations. The net is continuously changing dynamical system, but the complexity is reduced because the activations are to chase much more rapidly than the weights. Once an acceptable cluster unit is selected for learning, the weights may be maintained over are extended period. During that period only weight changes should be done. This period is called 'resonance period'. Thus the net gets the name 'resonance' in it.

3.6 ART ARCHITECTURE–

Q.8 Explain ART architecture? **(KUK, May 2009)**

Ans. ART architecture- The architecture of an ART network divided into categories:

(a) Typical architecture (b) Simplified Architecture

(a) Typical Architecture– The typical or basic architecture of ART involves following group of neurons:

1. Input processing field- F1 layer
2. Cluster units- F2 layer
3. Reset mechanism- which controls the degree of patterns in each cluster.

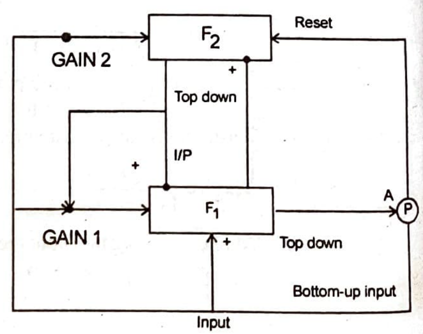

Fig. 3.3 Typical Architecture

1. Input processing (F1layer)- The F1 field receives input from possibly three sources bottom up input to F1, top-down from F2 & gain control signal. To avoid the possibity that mere feedback from F2 can generate spontaneous activity at level F1, system dynamics are limited in such a way that at least two out of three inputs must be active to generate activity at F1 field. This is called 2/3 rule in ART

2. **Cluster Units[F2 layer]–** This is a competitive layer. The cluster unit with largest net input is selected to learn I/P pattern. The activations of all other F2 units are set to zero.
3. **Reset Mechanism–** The information from I/P cluster units is combined in the interface units. Depending upon the similarity between top-down weights I/P vector, the cluster unit may or may not be allowed to learn the pattern. This is done at the reset unit based on the signals it receives from the input.

(b) Simplified Architecture– It consists of two layers of neurons labelled 'comparison & recognization needed for training & classification.

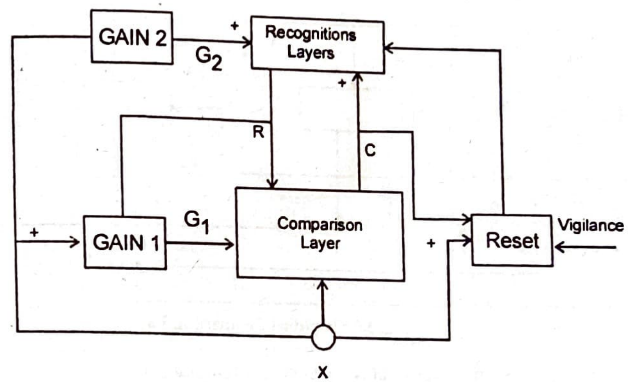

Fig. 3.4 Simplified ART Architecture

It consists of five functional modules which are explained one by one below.

(a) Comparison layer– It receives binary input vector xs initially passes it unchanged through to becomes vector.

---Each neuron in the comparison layer receives three binary inputs.

1. Component X_i from input vector X.
2. Feedback. signal P_i from recognition layer.
3. Input from gain signal gain 1.

---To output a one, atleast two a three neuron's inputs must be one, otherwise it is zero. This is called as two thirds rule.

---Initially :

* G =1
* $R [P_1, P_2 ...... P_3] = 0$

Vector

* Vector C =X

G = gain signal input

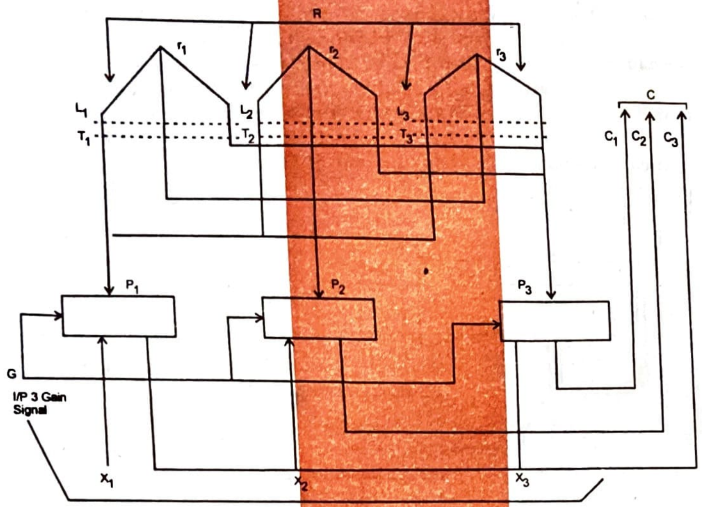

Fig. 3.5 Simplified Comparison Layer

(b) Recognition Layer– It serves to classify the input vector each recognition layer neuron has no associated weight vectors.

--- only neuron with weight vector best matching the input vector "fires."

----A neuron in the recognition layer responds maximally when the vector C matches its set of weights.

--- Each recognition layer neuron computes a dot product between its weights and vector C. The neuron that has weights most like the vector will have largest output thus, winning the competition.

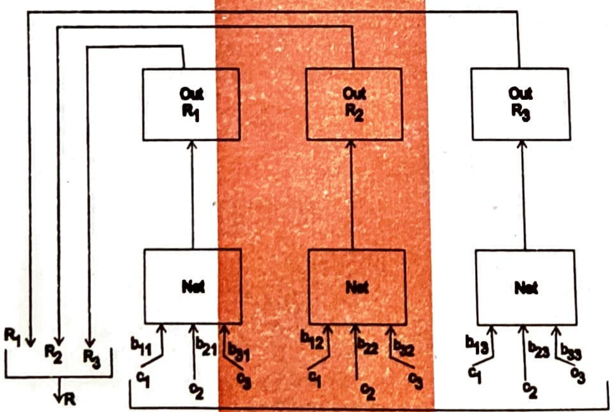

Fig. 3.6 Simplied Recognition Layer

(c) Gain 2 : G2o, the output of gain 2 is one if any component of input vector x is one. Or,

$$G_2 = OR\ (x)$$

G_2 is the logical OR of components of x.

(d) Gain 1 : G1, the output of gain 1 is one if any components of input vector X is one however if any component of R is one is G1 is forced to be zero.

0	0	0
1	0	1
0	1	0

OR (x') = 1

OR (R) = 0

(e) Reset– It measures the similarity between vector R & C. If they differ by more than the vigilance parameter, are sent signal is sent to disable the firing neurons. It calculates the ratio between number of ones on vector C to number of ones in vector X.

3.7 ART CLASSIFICATION OPERATION–

Q.9 What is the function of ART network and explain its operation with relevant equations. (KUK, June 2008)

Ans. Function of ART– ART network vector is a vector classifier. It accepts an input vector and classifies it into one of a number of categories depending upon which number of stored patterns it most resenbles.

- Its classification is indicated by a single recognition layer that fires.
- If the input vector does not match any stored pattern then a new category is created storing the pattern that is like the input vector.
- Once a stored pattern is found that matches the input vector with in a specified tolerance, that pattern adjusted to make it still more like the input vector stored pattern is ever modified if it doesn't match the current input pattern with in specified tolerance.

ART Classification Operation– The ART classification process consists of 3 major phases:

1. **Recognition Phase–** Initially when no input vector is applied, all components of vector X are zero, which set G2 to zero thus, all recognition layer neurons are disabled with output zero.

--All recognition layer neurons have an equal chance to win the competition since they all start in same state.

Thus, initially

$$X = 0$$
$$G_2 = 0$$
$$R_j = 0$$

---Now vector X is applied with one or more component are one which makes both G1 & G2 one. This provides are of the two input required by 2/3 rule.

---Thus, during this phase vector C is exact duplicate of X.

---Thus, X is applied and

$$OR\ (x) = 1$$

=> G_1, and $G_2 = 1$

=> $\frac{2}{3}$ rule

$$C = x$$

--- Now, for each neuron is recognition layer, a dot product is forced between B_j and C where B_j is weiggld vector.

--- The neuron with largest dot product has weights that best match input vector. Thus, it wins & fires & makes a single component r_j of vector R one and all other components equal to zero.

-- Thus, dot product of B_j & C is calculated

- largest value => best match
- wins & fires

$\Rightarrow r_j = 1$

2. **Comparison Phase–** The neuron firing in the recognition layer passes a one back to comparison layer on its output signal r_j. This signal fan outs through a seprate binary weight t_{ji} to each neuron in campharison layer with a signal Pp.
-- Each vector by consititutes a scalar version of vector T_j which means that all components of phase binary valued.
--- Since vector R is no longer zero G1 is inhibited & in output set to zero. This is in accordance with 2/3rd rule.
--Thus, if the input doesn't match stored pattern, the top down feedback from recognition layer forces components of C to be zero.
--- If there is a substantial match between X & D, few neurons in comparison layer will fire Sc will contain many zeros.

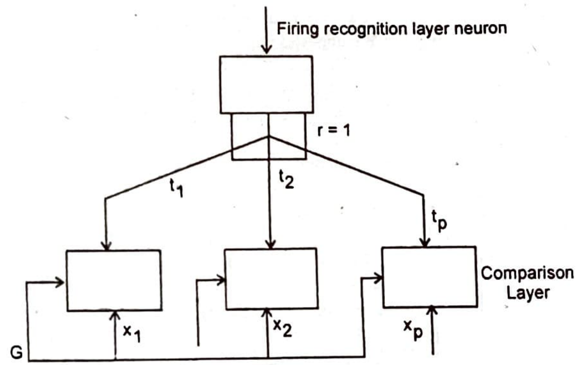

Fig. 3.7 Signal path for firing neuron

--Thus, the neuron firing in the layer should be inhibited which is done b_g reset block which causes the reset signal to occur if the degree of similarity is less than vigilance level.

3. **Search Phase–** If there is no reset signal generated, the match is adequate & classification is finished . Else the stored patterns must be searched to seek a better match.
--In latter case, inhibition of neuron in the recognition layer causes all components of R to be zero. G1 goes to ones X again appears at C. As a result, different neurons wins in the recognition layer, and pattern is fed back to comparison layer.
---This process is repeated untill one of two events occur.
(a) A stored pattern is found that matches above the vigilance parameter S>P, than both T_j & B_j are adjusted.
(b) All stored patterns mismatch input vector than a previous unallocated neuron in recognition layer is applied to pattern & B_j & T_j are set to match input pattern.

3.8 ART IMPLEMENTATION

The detailed operatiom of an ART consists of following five phases–
1. Initialization 2. Recognition 3. Comparison, 4. Search., 5. Training.

1. **Initialized :-** During this phase, the weight vectors B_j & T_j, vigilance parameter are set to initial values.
---The weights of bottom up vectors B_j are all initialized to same value as–

$$b_{ij} < L\frac{1}{[L-1+p]} \text{ for all } 1 \text{ \& } j$$

where p-no. of components in I/P vector L-a constant >1
-- The weights of all top down vectors are initialized to 1 as:

$$t_{ji} = 1 \text{ for all}_j$$

--The vigilance parameter is set between 0 & 1depending upon the degree of mismatch that is to be accepted between stored pattern I/P vector.

$$P = 0 \text{ to } 1$$

2. **Recognition–** Initially there is no output from recognition layer thus, G1 is set to one providing all comparison layer neurons with one of two inputs needed for it to fire.
-- As a result any component of Z that provides the second Input which is one will causes associated comparison layer neuron to fires at output a one.
--- Thus, $NET = [B_j.C]$
where B_j - weight associated recognition layer neuron.
C- output vector of comparison layer neuron F is the threshold function that follows:

$$OUT_j = 1 \text{ if } NET_j > T$$
$$= 0 \text{ otherwise}$$

Where T is threshold value.
---Lateral inhibition ensures that only recognition layer neuron with highest value of NET will output is other will output 0.

3. **Comparison-** The feedback signal from recognition layer causes G1 to go to zero. Thus, G1 = 0.
--- Comparison layer neuron fire only if

$$P = X = 1 \text{ as per } 2/3^{rd} \text{ rule.}$$

-- The reset block compares vector C to input vector X producing a rest output whenever the similarity is below vigilance threshold. The procedure that follows is–
--- Call D as number of 1's in X vector.
-- call N as number of 1's in X vector.
--Compute similarity as

$$S = \frac{N}{D}$$

For eg. suppose

$X = 1011101$ then $D = 5$
$C = 0011101$ then $N = 4$

$$S = \frac{N}{D} = \frac{4}{5} = 0.8$$

S will vary from 1 to 0.

4. **Search–** If the similarities of winning neuron is greaster than vigilance the S>P then no search is required.

— Due to training algorithm, it is possible that a different recognition layer neuron will provide a better match exceeding the vigilance level though the dot product may be lower.

— If the similarity is below the vigilance level then the stored patterns must be searched seeking the one that matches more closely.

— To initiate the search, reset signal temporarily disables the firing neuron in recognition layer for duration of search i.e. G=1 & a different recognition layer neuron wins the competition.

— An unsuccessful search will automatically terminate on a uncommitted neuron as its t_{ji} are all ones, their initial values.

5. **Training**- Training is a process in which input vectors are presented to the network & the weights are adjusted so that similar activate the same recognition layer neuron.

⇒ There are two types of learning or training in ART.

1. **Fast Training :** It is used in ART network. Here input is binary weight changes during resonance period occur more rapidly.

⇒ The network is stablized each time the pattern has to choose the correct cluster in it where it is presented.

⇒ The weight associated which cluster units are also stabilized in last traning.

2. **Slow Training :** It is used in ART 2 network. The weight changes during resonance period occur slowly.

⇒ The weight produced are continuously changing hence the will not be stabilized. It will become stabilized only after presenting a large number of inputs & method for equilbrium is very difficult.

Training Algorithm : The ART network has two main layers, comparison layer-F1 having P nodes, the output recognition layer having K nodes as shown.

⇒ The best ART algorithm is described by following sequence of operation.

Algorithm:—Step.1 : Select maximum number of output nodes k to exceed maximum expected number of categories.

Step. 2 : While network has not stabilized do.

Step. 3 : Apply new input Vector x = $[x_1 x_2 x_i \ldots\ldots\ldots x_4]$

Step. 4 : Compute outputs

$$Net_j = \sum_{i=1}^{p} b_{ij}(k)\, x_i$$

Step. 5 : Select best watch Net_j = max (Net)

Step 6 : Calculate vigiliance or similarity ratio w.r.t. best watch

$$S = \frac{\sum_{i=1}^{p} b_{ij}(k)\, x_i}{\sum_{i=1}^{p} x_i}$$

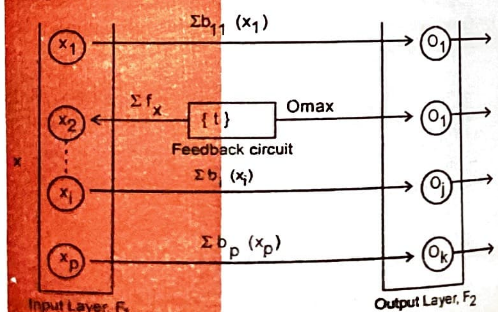

Fig. 3.8 ART Network Model

Step. 7 : It S≤ p then go to step 8 else go to step 9 where P = Vigilance threshold.

Step. 8 : Disable the best match set the output of best match to until next uector is applied.

Step. 9 : Adept best match

bij (k + 1) tji (I) xI

$$0.5 + \sum_{i=1}^{p} t_{ji}(k)x_i$$

$t_{ji}(k+1) = t_{ji}(k)\, x_i$

Example : P = 0.7, P = 7

$x(1) = [1, 1, 0; 0, 0, 0, 1]$ $x(2) = [0, 0, 1, 1, 1, 1, 0]$

$x(3) = [1, 0, 1, 1, 1, 1, 0]$ $x(4) = [0, 0, 0, 1, 1, 1, 0]$

$x(5) = [1, 1, 0, 1, 1, 1, 0]$

Initially : $t_{ji}(0) = 1,\ b_{ji}(0) = \frac{1}{8}$

for input x_1, Net_1 $= \frac{1}{8}\times 1 + \frac{1}{8}\times 1 + \frac{1}{8}\times 0 \ldots + \frac{1}{8}\times 1 = \frac{3}{8}$

Node 1 Wins $\dfrac{\sum_{i=1}^{7} b_{ji}.x_i}{\sum_{i=1}^{7} xi} = \frac{3}{3} = 1 > 0.7$

$b_{ij}(k) = \left[\frac{1}{0.5+3} = \frac{1}{3.5} \text{ For i 1, 2 .....7 otherwise}\right]$

$t_{ji}(1) = t_{ji}(0)x_i$

$B(1) = \left[\frac{1}{3.5}\frac{1}{3.5}0000\frac{1}{3.5}\right]$

$T(1) = [1\,1\,0\,0\,0\,0\,1]$

For second sample [0, 0, 1, 1, 1, 1, 0]

$NET_1 = [0, but]$

$S = 0, < P$. A new node i generated with

$$B(2) = \begin{bmatrix} \frac{1}{3.5} & \frac{1}{3.5} & 0 & 0 & 0 & 0 & \frac{1}{3.5} \\ 0 & 0 & \frac{1}{4.5} & \frac{1}{4.5} & \frac{1}{4.5} & \frac{1}{4.5} & 0 \end{bmatrix}^T$$

$$T(2) = \begin{bmatrix} 1\,1\,0\,0\,0\,0\,1 \\ 0\,0\,1\,1\,1\,1\,0 \end{bmatrix}^T$$

The Network stabilizes with 3 nodes in two presentations of each input vector.

Q10. Describe ART training with the help of an example. **(KUK May 2009)**

Ans. See section 3.8, Subheading (5) : Training

3-9. ART Characteristics : The ART system has a number of characteristics that are not obvious. Some of the important characteristics are listed below :

⇒ **top down weight initiazation :** The top down weight must be initialized to ones. if they were initialised too their input vector C would also be zero.

⇒ Training may be viewed as a process of pruning componets of stored vectors that do not match input vectors. This process is irreversible i.e. once a top-down weight has been set to zero, the training algorithm can never restore it to a one.

⇒ This characteristics has importance implication for the learning process. Suppose that a group of closely related vectors should be classified into same category indicated by firing the same recognition layer neuron if they were presented sequentially to netwoks, first will be assigned a recognition layer neuron, its weights will be trained to match the network input vector. Training with rest of vectors will set the weights of stored vector to zero in all postition.

⇒ Thus, the stored vectors come to represent the logical intersection of all training vectors. A new vectors consisting only their essential features will be assigned to this category.

2. **Bottom up weights adjustments :** The summation in the denominator of weight adjustment formula respresents the number of ones in the output of comparision layer. It also represent the "size" of the vector.

⇒ Thus, large C Vectors produce smaller weight values than do small c vectors. This property makes possible to separate vectors where one is a subset of other.

⇒ Suppose network has been trained on two input vectors with a recognition layer neuron assigned to each. Here X1 is a subset of X2. Without this property, weight would be trained to the same value for each pattern.

⇒ If X1 is applied once more both recognition layer neurons recive the same activation hence the wrong neuron 2 will win the competition.

⇒ Scaling the bottom up weight prevents this undesirable behaviour.

3. **Bottom-up weight initialization :** For correct functioning of ART, bottom up weights must be intialized to small values.

⇒ If they are too high input vectors that already have been learned will activate an uncommitted recognition layer neuron rather the one previous trained.

⇒ Setting these weights to low values ensures that are uncommited neuron will not over power a trained recognition layer neuron.

⇒ Foreg : For L = 2, m = 5, bij < $\frac{1}{3}$ & we set brij = $\frac{1}{6}$ with these weights, applying a vector for which network has been trained, will cause correctly trained recognition layer neurron to win over uncommitted neuron.

Searching : Direct access obviates the need for search except who an uncommitted recognition layer neuron is to be assigned. Application of an input vector that is simillar but not identical to one stored patterns but not indentical to one of stored patterns may not on first trial select a recognition buyers neuron st, S > P.

⇒ Suppose network has been trained to input vectors that are : x_1 = 10110, x_2 = 11001.
Now an input vectors x_3 = 11000 is applied. The excitation to recognition layer neuron i will be 1.0 while of neuron 2 is $\frac{2}{3}$. Neuron i will win & C will be set to 10000 & s = $\frac{1}{2}$. If p = $\frac{3}{4}$ then neuron 2 will win the competition, C will now become 11000 & S = 1 & search will stop.

3.10. Image comression using ART :

Q.11 What is ART? Explain Image compression using ART.

Q.12 How the image compression can be done using ART?

Ans. ART : See 3.5 (Introduction to ART)

Image Compression Using ART : Like CPN's, ART are also used for image data compression.

⇒ The unique ability of ART to control the trade offs between compression ratio and image quality makes ART suitable for this application.

⇒ The large amount of time & storage required to transmit pictorial data brings the need of image data compression.

⇒ In this work, ART is studied and its performance compared with CPN'S ART has CPN advantages such as faster tearning & self organising in addition ART has vigilance parameter to directly control the trade off between compression ratio & distortion ratio.

⇒ Another factor is that we are concerned only with the quality of reconstructed pictorial data & length of time needed to transmit data from source to destination & not the relation among nodes.

⇒ Also, modifications are made in the ART learning algorithm to achieve error free compression.

⇒ In the process of compression a pictures is split into small frames. Those sub-images are than fed into ART networks for self organization & outputs represent category indices for those sub-images.

⇒ After learning, those indices & their corresponding prototypes are transmitted to destination. Picture is then reconstructed according to these indices & proto types.

⇒ The benefit received from this operation is measured by compression ratio (Q) which is computed as :

$$Q = \frac{NF}{[C.N + F.\text{Log}_2 C]}$$

Where N—number of dimension of sub images
F—total no. of frames
C—Total number of categories formed during learning.

⇒ The price we pay is measured by distortion ratio defined as follow :

$$F = \sum_{ji}^{E}\sum_{i=1}^{N}(x_{ij} - x'_{ij})^2$$

Where X_{ij} – Value in original image
X'_{ij} – Corresponding value inreconstructed image

⇒ By adjusting the vigilance parameter between O&1, ART Performs highly error free compression. The picture quality control & parameter value assignment are more direct in ART which makes it attractive for use in real time environment.

Q.13+ Describe the Architecture of ART Network. Sate in detail the computational and suphlemental units. **(KUK May 2001)**

Ans. ART is designed for clustering binary vectors. It is the first members of the ART family. It is a two layer network that discover pattern clusters templates in arbitrary boolean pattern sets.

Architecture : The ARTI network has computational units and supplemental units. It is architecture is shown in fig. 3.9

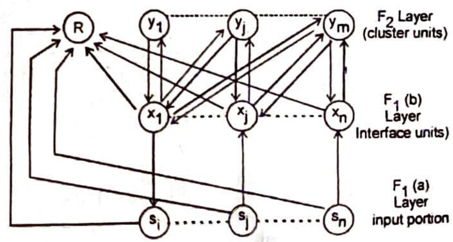

Fig. 3.9 Architecture of ART 1

Computational Unit : The computational Unit has F_1 and F_2 Units and reset units. The F_1 (a) is connected to F_1(b) interface unit both input and interface units are connected to reset we mechanism unit. By means of top down and bottom up weight the interface layer units are connected to the cluster units and the reciprocity is also achieved.

Suplemental Units : The structure of supplemental unit is shown in Fig. 3.10 Difficulty posed in computational units (F1, F2 and reset unit) is that these units are required to respond differently at different stages of the process. Also the problem with reset mechanism is that, F_2 Units must be inhibited under certain conditions, but retruned to availability on subsequent learning trails.

To avoid the above mentioned problems, two supplemental units (called again control G_1 and G_2) are present in addition to reset unit, R. These special units receive signal forms and send their signal to, all the units present in occupational structure. In Figure 3.10, excitatory signals are indicated by '+' and inhibitory signals by '–'. The signal may be sent, wherever any unit in interface or cluster layer has three sources from which it can receive a signal. Each of these units also receives two excitatory signals in order to be on. Hence, due to this, the requirement is called the two-thirds rule. This rule plays a role in the choice of parameter and initial weight. the reset unit R controls the vigilance matching also.

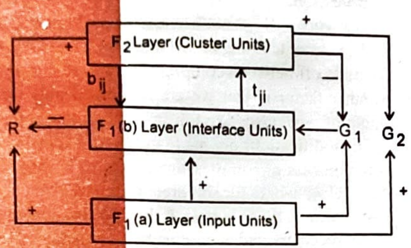

Fig. 3.10 Structure of supplemental units

Q.14. What is the basic concept behind adaptive resonance. Theory (ART)? Explain Basic Architecture and operation of ART network. **(KUK June 2010)**

Ans. ART : Section 3.5

Basic Architecture : Section 3.6, question No.8

Operating ART Network : Section 3.7, Question No.9

Q.15 Calculate the weights of 2 * 2 BAM, Where 2 pairs of patterns and give below :

$$A_1 = (+1, +1, -1) \ \& \ B_1 = (-1, +1, -1, +1)$$
$$A_2 = (+1, -1, +1) \ \& \ B_2 = (+1, -1, +1, -1)$$

Also recall pattern B_1 for A_1 **(KUK May 2010)**

Ans.

$$A_1 = (+1, +1, -1) \ \& \ B_1 = (-1, +1, -1, +1)$$
$$A_2 = (+1, -1, +1) \ \& \ B_2 = (+1, -1, +1, -1)$$

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} [-1 \ 1 \ -1 \ 1] = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} [1 \ -1 \ 1 \ -1] = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$W_1 + W_2 = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -2 & 2 & -2 & 2 \\ 2 & -2 & 2 & -2 \end{bmatrix}$$

$$B = A/w = [1 \ 1 \ -1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ -2 & 2 & -2 & 2 \\ 2 & -2 & 2 & -2 \end{bmatrix}$$
$$= [-2 -2 \ \ 2+2 \ \ -2 -2 \ \ 2+2]$$
$$= [-4 \ \ 4 \ \ -4 \ \ 4]$$

Hence proved = [–1 +1 –1 +1]

• • •