PREPARED BY: Dr.S.Kishore

SATHYABAMA UNIVERSITY

SECX1048 FUNDAMENTALS OF FUZZY LOGIC AND ARTIFICIAL NETWORK

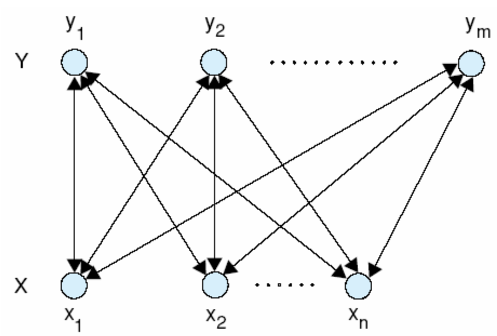
UNIT V BAM, ART & OPTICAL NETWORK

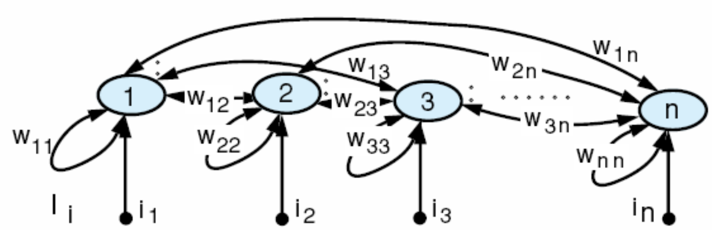
BAM – Structure – Types – Encoding and Retrieving

Bi-Directional Associative Memory (BAM)

Developed in the year 1988 by KOSKO. It's Hetro-Associative neural network with Two Layers. The signals are transmitted between these two layers until each neurons Activation is constant for several steps.

A Bi-directional Associative Memory (BAM) consists of bi-directional edges so that information can flow in either direction. Nodes can also have recurrent edges that is, edges that connect to themselves. Two different BAM networks are shown below





BAM consists of two layers .Layer X with n no. of neurons and Layer Y with m no. of neurons.

Here the connections are bidirectional and wij = wji.

Introduce an input and propagate to the other layer, a node's activation (state) will be = 1 if its activation function value > 0 stay the same state if its activation function value = 0, = -1 if its activation function value < 0, take the activation values (states) of the computed layer and use them as input and feed back into the previous layer to modify those nodes' states, repeat until a full iteration occurs where no node changes state , this is a stable state , the output is whatever the non-input layer values are indicating.

Two Forms of BAM – (1) Discrete BAM

(2) Continuous BAM

Discrete BAM: Uses Binary or Bipolar i/p's. Step function with non-zero threshold is used as Activation function. A bipolar vector improves the performance of the network. The weight matrix is S(p):T(p) where S(p) = (s1(p)…si(p)…sn(p)). A bi-directional associative memory (BAM) network is one with two fully connected layers, in which the links are all bi-directional. There can also be a feedback link connecting a node to itself. A BAM network may be trained, or its weights may be worked out in advance. It is used to map a set of vectors Xi (input layer) to a set of vectors Yi (output layer).

Algorithm:

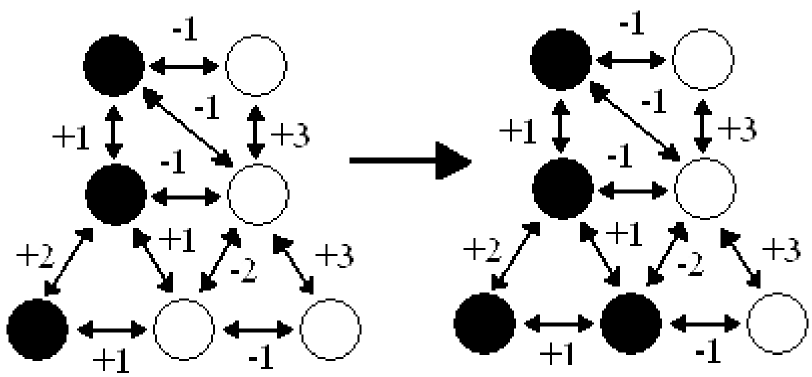
1. Initialize the weights to store a set of P vectors.Initialize activations to zero. 2. Set activation of X layer to current input pattern 3. Present it to Y layer 4. While activations are not converged do the following. 5. Cal .net value of Y layer and send it to X layer. 6. Update activations in X layer and cal. Net value.send dignals to Y layer. 7. Repeat the steps 5 to 6 till convergence is reached.

Since the BAM network has bidirectional edges, propagation moves in both directions, first from one layer to another, and then back to the first layer. We need edge weights for both directions of an edge, $w_{ij} = w_{ji}$ for all edges. Propagation continues until the nodes are no longer changing values, that is, once all nodes stay the same for one cycle (a stable state), BAM networks as attractor networks which provide a form of content addressable memory , given an input, we reach the nearest stable state

Edge weights are worked out in advance without training by computing a vector matrix, this is the same process as the linear associator.

Hopfield Network:

This is a form of BAM network, in this case, the Hopfield network has four stable states, no matter what input is introduced, the network will settle into one of these four states, the idea is that this becomes a content addressable, or autoassociative memory, the stable state we reach is whatever state is "closest" to the input, closest here is not defined by Hamming distance but instead by minimal energy – the *least amount of work* to reach a stable state .



If the Hopfield network shown to the right starts in the state on the left side of the figure, it will relax to the state shown on the right – this Hopfield network has 4 stable states, one of which is the one to the right

A Hopfield Network is identical in structure to an autoassociative BAM network – one layer of fully connected neurons. The activation function is

$+1$, if net $> T_i$,

$x^{new} =$ x^{old}, if net $= T_i$,

-1, if net < T_i,

where net = $\Sigma_j w_j * x_j$.

The are restrictions on the weights: $w_{ii} = 0$ for all i, and $w_{ij} = w_{ji}$ for i.j.

Usually the weights are calculated in advance, rather than having the net trained.

The behavior of the net is characterized as an energy function, $H(X) = - \Sigma_i \Sigma_j w_{ij} w_i w_j + 2 \Sigma_i T_i x_i$, decreases from every network transition.

Two goals:

Guarantee that the network converges to a stable state, no matter what input is given. The stable state should be the closest one to the input state according to some distance metric. Thus, the network must converge, and converge to a local energy minimum, but there is no guarantee that in converges to a state near the input state. Can be used for optimization problems such a TSP (map the cost function of the optimization problem to the energy function of the Hopfield net).

Associative memory is used for memory retrieval, returning one pattern given another. There are three types of associative memory.

1 Heteroassociative: Mapping from X to Y s.t. if an arbitrary vector is closer to Xi than to any other Xj, the vector Yi associated with Xi is returned.
2 Autoassociative: Same as above except that Xi = Yi for all exemplar pairs. Useful in retrieving a full pattern from a degraded one.
3 Interpolative: If X differs from the exemplar Xi by an amount Δ, then the retrieved vector Y differs from Yi by some function of Δ. A linear associative network (one input layer, one output layer, fully connected) can be used to implement interpolative memory.
4 Hamming vectors are vectors composed of just the numbers +1 and -1. Assume all vectors are size n.

5 The Hamming distance between two vectors is just the number of components which differ.
6 An orthonormal set of vectors is a set of vectors where are all unit length and each pair of distinct vectors is orthogonal (the cross-product of the vectors is 0).
7 If a BAM network is used to implement an autoassocative memory then the input layer is the same as the output layer, i.e., there is just one layer with feedback links connecting nodes to themselves in addition to the links between nodes. This network can be used to retrieve a pattern given a noisy or incomplete pattern.

BAM processing

Apply an initial vector pair (X,Y) to the processing elements. X is the pattern we wish to retrieve and Y is random. Propagate the information from the X layer to the Y layer and update the values at the Y layer. Send the information back to the X layer, updating those nodes. Continue until equilibrium is reached.

Adaptive Resonance Theory – Architecture of ART 1 & ART 2 – Implementation – Training – Characteristics

Adaptive Resonance Theory (ART)

Stability: system behaviour doesn't change after irrelevant events

Plasticity: System adapts its behaviour according to significant events

Ongoing learning capability, Preservation of learned knowledge

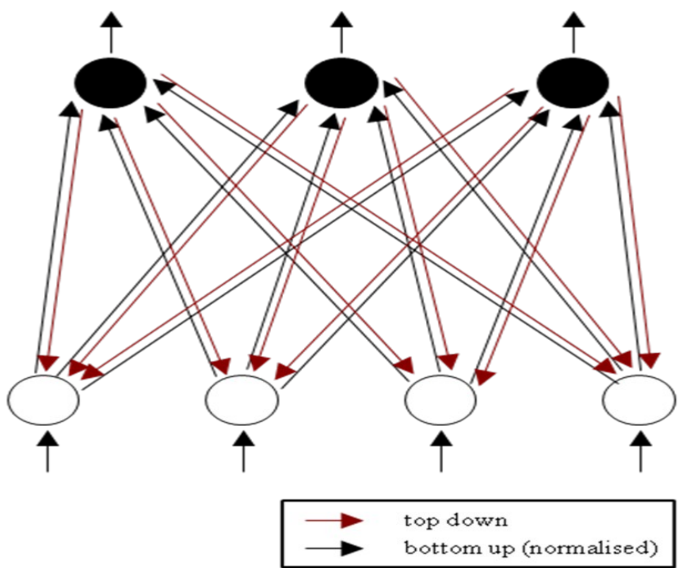
Invented by Grossberg in 1976 and based on unsupervised learning model. - Resonance means a target vector matches close enough the input vector. - ART matching leads to resonance and only in resonance state the ART network learns. Suitable for problems that uses online dynamic large databases. ART 1- classifies binary input vectors, ART 2 clusters real valued input (continuous valued) vectors. - Used to solve Plasticity stability dilemma.

Plasticity –stability dilemma:

When there is change in the patterns (plasticity) how to remember previously learned vectors (stability problem) is a problem. ART uses competitive law (self-regulating control) to solve this PLACITICITY – STABILITY Dilemma.

ART consists of

(1) F1 Layer: I/P processing unit also called comparison layer. (2) F2 Layer: clustering or competitive layer. (3) Reset mechanism.



Comparison Layer: Take 1D i/p vector and transfers it to the best match in recognition field (best match - neuron in recognition unit whose weight closely matches with i/p vector). Recognition Unit: produces an output proportional to the quality of match. In this way recognition field allows a neuron to represent a category to which the input vectors are classified. Vigilance parameter: After the i/p vectors are classified the reset module compares the strength of match to vigilance parameter (defined by the user).

Higher vigilance produces fine detailed memories and lower vigilance value gives more general memory. Reset module: compares the strength of recognition phase. When vigilance threshold is met then training starts otherwise neurons are inhibited until a new i/p is provided. There are two set of weights (i) bottom up weight - from F1 layer to F2 Layer (2) Top –Down weight – F2 to F1 Layer.

Fast learning: Happens in ART 1 – Weight changes are rapid and takes place during resonance. The network is stabilized when correct match at cluster unit is reached.

Slow Learning: Used in ART 2 .weight change is slow and does not reach equilibrium in each learning iteration.so more memory to store more i/p patterns (to reach stability) is required.

BASIC ART Training steps:

1. Initialize the parameters.
2. If no stop condition do step 3 to 10.
3. For each i/p vector do steps 4 to 9.
4. F1 Layer process starts.
5. If reset condition = true do step 6 to 8.
6. Find f2 unit with largest i/p (to learn current pattern).
7. F1(b) units combine their /p's F1(a) & F2.
8. Test for reset condition. (differs for ART1 & ART2)
9. If reset true candidate is rejected (i.e., neuron is inhibited). Return to step 5.
   If reset is false candidate unit is accepted for learning.
10. Learning starts. Weight updation starts as per diff equation.
11. Test for stop condition.

ART ALGORITHM:

Incoming pattern matched with stored cluster templates. If close enough to stored template joins best matching cluster, weights adapted. If not, a new cluster is initialised with pattern as template.

PREPARED BY: Dr.S.Kishore

ART TYPES

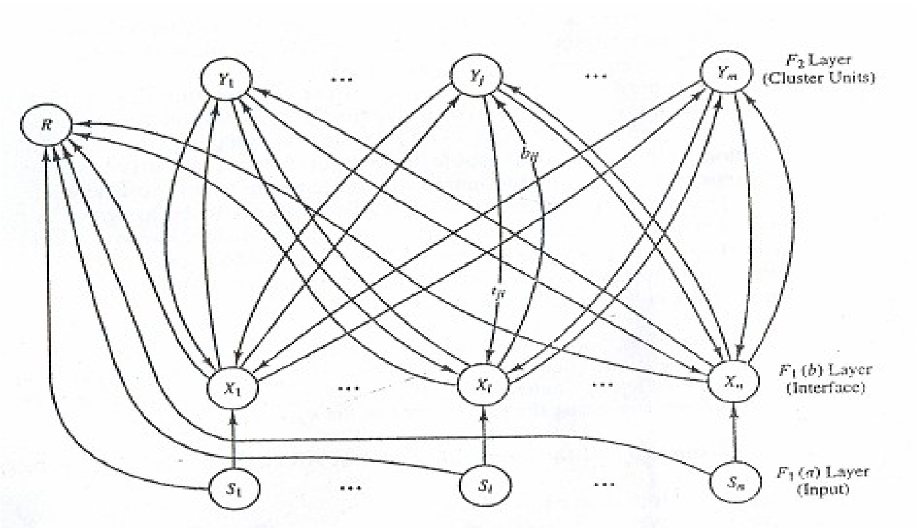
ART1: Unsupervised Clustering of binary input vectors.

ART2: Unsupervised Clustering of real-valued input vectors.

ART3: Incorporates "chemical transmitters" to control the search process in a hierarchical ART structure.

ARTMAP: Supervised version of ART that can learn arbitrary mappings of binary patterns. Fuzzy ART: Synthesis of ART and fuzzy logic. Fuzzy ARTMAP: Supervised fuzzy ART, dART and dARTMAP: Distributed code representations in the F2 layer (extension of winner take all approach). Gaussian ARTMAP

ART1 ARCHITECTURE:



RESET MODULE

Fixed connection weights

Implements the vigilance test

Excitatory connection from F1(b)

Inhibitory connection from F1(a)

Output of reset module inhibitory to output layer

Disables firing output node if match with pattern is not close enough

Duration of reset signal lasts until pattern is present

GAIN MODULE

Fixed connection weights

Controls activation cycle of input layer

Excitatory connection from input lines

Inhibitory connection from output layer

Output of gain module excitatory to input layer

2/3 rule for input layer

Recognition Phase

Forward transmission via bottom-up weights. Input pattern matched with bottom-up weights (normalised template) of output nodes. Inner product $x \bullet b_i$. Best matching node fires (winner-take-all layer). Similar to Kohonen's SOM algorithm, pattern associated to closest, Comparison Phase. ART1: fraction of bits of template also in input pattern. Backward transmission via top-down weights. Vigilance test: class template matched with input pattern. If pattern close enough to template, categorisation was successful and "resonance" achieved. If not close enough reset winner neuron and try next best matching. Repeat until, vigilance test passed, Or (all committed neurons) exhausted.

Vigilance Threshold

Vigilance threshold sets granularity of clustering, it defines amount of attraction of each prototype. Low threshold, large mismatch accepted, few large clusters, misclassifications more likely. High threshold, small mismatch accepted, many small clusters, higher precision.

Adaptation

Only weights of winner node are updated, only features common to all members of cluster are kept. Prototype is intersection set of members.

$$t_{ji} = x_i$$

$$b_{ij} = \frac{L \times x_i}{L - 1 + \|x_i\|}$$

Issues about ART1

- Learned knowledge can be retrieved
- Fast learning algorithm
- Difficult to tune vigilance threshold
- New noisy patterns tend to “erode” templates
- ART1 is sensitive to order of presentation of data
- Accuracy sometimes not optimal
- Only winner neuron is updated, more “point-to-point” mapping than SOM

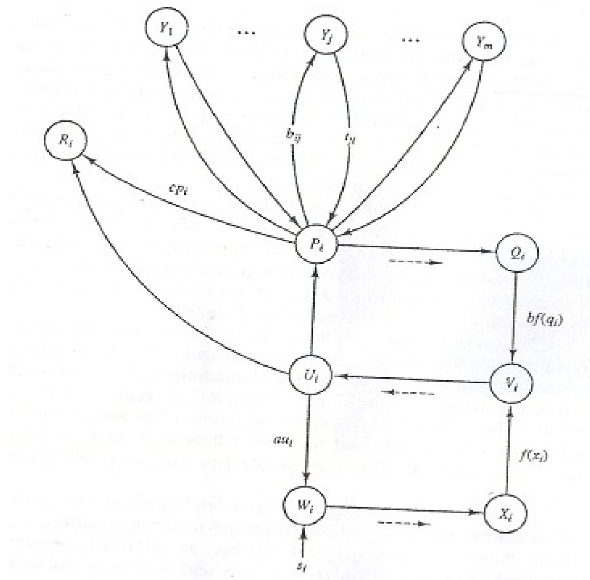
ART2

Unsupervised Clustering for:

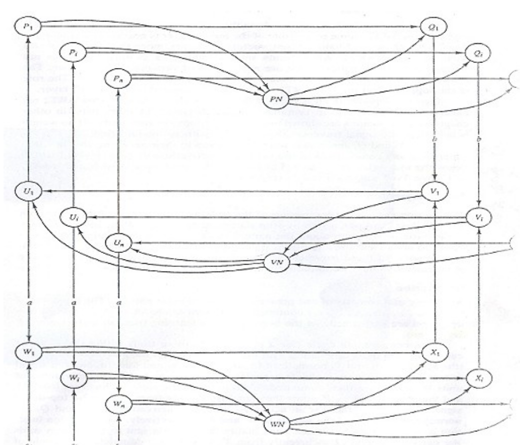
Real-valued input vectors

Binary input vectors that are noisy

Includes a combination of normalization and noise suppression



NORMALIZATION



Fast Learning

Weights reach equilibrium in each learning trial. Have some of the same characteristics as the weight found by ART1. More appropriate for data in which the primary information is contained in the pattern of components that are ‘small’ or ‘large’ .

Slow Learning

Only one weight update iteration performed on each learning trial. Needs more epochs than fast learning. More appropriate for data in which the relative size of the nonzero components is important .

ART2 ALGORITHM

Step 0: Initialize parameters:

a, b, ɵ, c, d, e, α, ρ.

Step 1: Do Steps 2-12 N-EP times.

(Perform the specified number of epochs of training.)

Step 2: For each input vector s, do steps 3-11.

Step 3: Update F1 unit activations:

$$u_i = 0$$
$$w_i = s_i$$
$$p_i = 0$$
$$x_i = \frac{s_i}{e + \|s\|}$$
$$q_i = 0$$
$$v_i = f(x_i)$$

Update F1 unit activations again:

$$u_i = \frac{v_i}{e + \|v\|}$$
$$w_i = s_i + au_i$$
$$p_i = u_i$$
$$x_i = \frac{w_i}{e + \|w\|}$$
$$q_i = \frac{p_i}{e + \|p\|}$$
$$v_i = f(x_i) + bf(q_i)$$

Step 4: Compute signals to F2 units:

$$y_j = \sum_i b_{ij} p_i$$

Step 5: While reset is true, do Steps 6-7.

Step 6: Find F2 unit Y_J with largest signal .(Define J such that $y_J \geq y_j$ for j=1…m.)

Step 7: Check for reset:

$$u_i = \frac{v_i}{e + \|v\|}$$

$$p_i = u_i + dt_{Ji}$$

$$r_i = \frac{u_i + cp_i}{e + \|u\| + c\|p\|}$$

If $\|r\| \leq \rho - e$ then

y_J=-1 (inhibit J)

(reset is true; repeat Step 5);

If $\|r\| \geq \rho - e$ then

$$w_i = s_i + au_i$$

$$x_i = \frac{w_i}{e + \|w\|}$$

$$q_i = \frac{p_i}{e + \|p\|}$$

$$v_i = f(x_i) + bf(q_i)$$

Reset is false; proceed to Step 8.

Step 8: Do Steps 9-11 N_IT times.

(Performs the specified number of learning iterations.)

Step 9. Update weights for winning unit J:

t_{Ji} = αdui+{1+αd(d-1)}t_{Ji}

b_{iJ}= αdui+{1+αd(d-1)}b_{Ji}

Step 10: Update F1 activations:

$$u_i = \frac{v_i}{e + \|v\|}$$

$$w_i = s_i + au_i$$

$$p_i = u_i + dt_{Ji}$$

$$x_i = \frac{w_i}{e + \|w\|}$$

$$q_i = \frac{p_i}{e + \|P\|}$$

$$v_i = f(x_i) + bf(q_i)$$

Step11: Test stopping condition for weight updates.

Step 12: Test stopping condition for number of epochs.

In fast learning: $t_{Ji} = \frac{1}{1-d} u_i$

In slow learning:

After adequate epochs, top-down weights converge to average of learned patterns by that cluster.

ART APPLICATIONS

Natural language processing

Document clustering

Document retrieval

Automatic query

Image segmentation

Character recognition

Data mining

Data set partitioning

Detection of emerging clusters

Fuzzy partitioning

Condition-action association

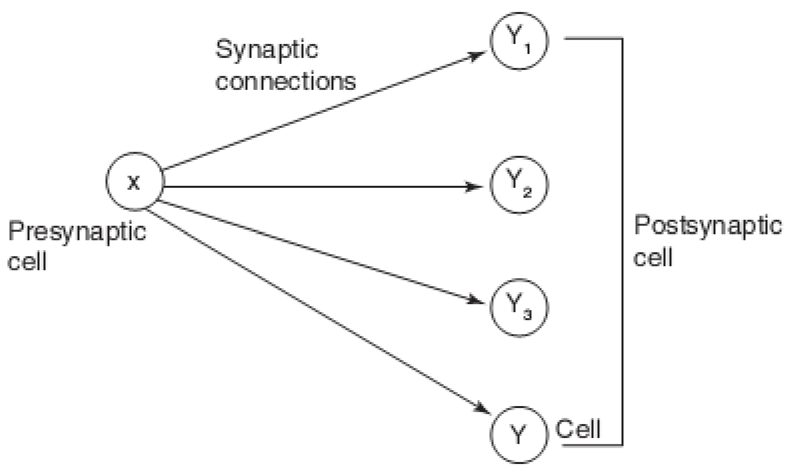
Introduction to Optical Neural Network – Cognitron & Neocognitron.

COGNITRON

Developed by FUKUSHIMA in 1975. Cognitron is just a hypothetical mathematical model of Human. Cognitron network was proposed by Fukushima in 1975. The synaptic strength from cell X to cell Y is reinforced if and only if the following two conditions are true:

1. Cell X: presynaptic cell fires.
2. None of the postsynaptic cells present near cell Y fire stronger than Y.

The connection between presynaptic and postsynaptic cells is as follows:



Cognitron is a self-organizing. Each layer receives input from previous layer and also from its own connection area. Two types of neural cells are present. as shown in figure 1.It consists of presynaptic neuron which feeds the next layer post synaptic layer. There are two types of neurons- Inhibitory & excitatory.

Excitatory cells make post synaptic neurons to fire. Inhibitory – used to reduce the firing of post synaptic neurons. Final firing is calculated as weighted sum of excitatory & inhibitory i/p. Each neuron connects to nearby area neurons only. This area is known as connection area. Unsupervised Training is performed. For any given set of i/p patterns the network, self organizes by adjusting the synaptic weights.

Competition among the (neural) cells is adopted. When connection region overlaps, the group of cells has similar response. To avoid this, competition among cells is used.

Excitatory neuron

Output is calculated as the ratio of excitatory input to that of inhibitory input. Excitatory input E is the weighted sum of input from excitatory neuron and from previous layer. This is same for inhibitory neuron'salso.E = $\sum$ Wi Xi and I = $\sum$ VjYj . Weights are always positive.Net input = [{(1+E) / (1+I)}-1] . Final Output = E-I, for I <<<1. When E = aP, I = bP where a,b are constantsthen using some transformations .o/p = [(a-b)/2b] { 1+tanh[log (ab)/2]} .From the above equation we can tell that cognitron closely emulates the response of Biological neural network.

NEOCOGNITRON

Human Brain is capable of recognising different alphabets in different font & size even if they are distorted. This is not the case in machine vision processing. Hence there is a need to develop a network which recognises patterns even with distortion and disorientation. This is the difference between cognitron and neocognitron. Neocognitron network is capable of recognising patterns that are disoriented.

Neocognitron is a multilayer feedforward network model for visual pattern recognition. It is an extension of cognitron network. Neocognitron net can be used for recognizing handwritten characters. The algorithm used in cognitron and neocognitron is same, except that neocognitron model can recognize patterns that are position-shifted or shape-distorted. The cells used in neocognitron are of two types: S-cell: Cells that are trained suitably to respond to only certain features in the previous layer.

C-cell: A C-cell displaces the result of an S-cell in space, i.e., sort of “spreads” the features recognized by the S-cell.

Training is found to progress layer by layer. The weights from the input units to the first layer are first trained and then frozen. Then the next trainable weights are adjusted and so on. When the net is designed, the weights between some layers are fixed as they are connection patterns.

Architecture:
Neocognitron is a feedforward network which uses competitive Learning. As like human visual processing (working of human Retina & visual cortex regions) it accepts 2D images.Each layer two types of cells are present .S Cell & C- Cell.
S-CELL: Simple cell arrays trained for a particular patterns.Each s cell is sensitive to a restricted region (Receptive range) in the input image. Receptive range sometimes can overlap to cover the entire input image.All S cells respond to the same pattern.
C-CELL:Complex C cell combines the output from S cell's.It makes the network less sensitive to the changes in the position of the input patterns. Each layer of Complex cell responds to the larger range by combining receptive layer of S cell's than in the previous layer.

Optical Neural networks

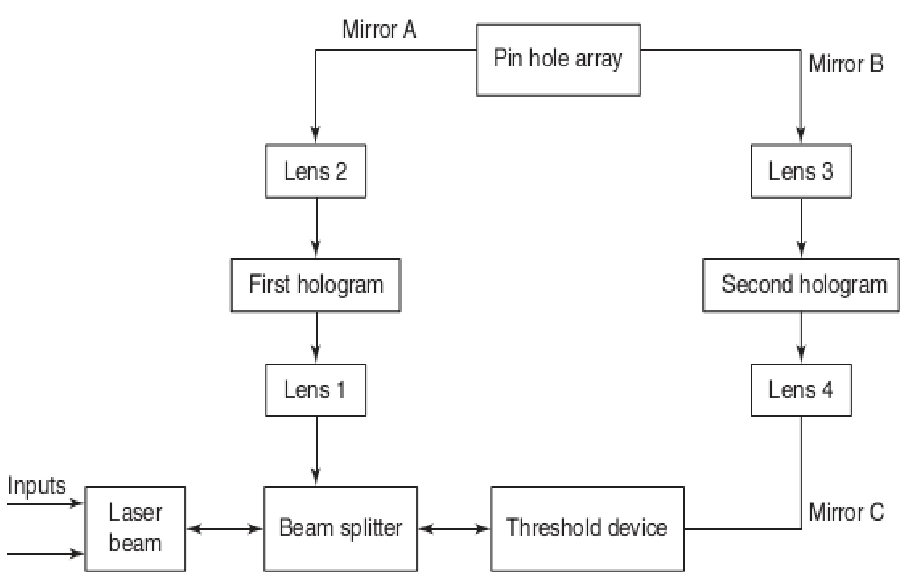
Optical neural networks interconnect neurons with light beams. There are two classes of optical neural networks. They are: Electro-optical multipliers, Holographic correlators.

Optical neural networks are designed by using optical light sources and detectors. Each neuron is represented by a light source and a detector for transmitting & receiving the signals. The main Advantage is the High speed of signal transmission and less error. Moreover designing of these are easy in this technologically improved days.

Holographic correlators:

Holographic correlator is a device which compares stored images with input image. By using Hopfield network this device can be trained to increase its efficiency. The reference images are stored in a thin hologram and can be retrieved by using an illuminated feedback loop. The noisy or incomplete input signal is continuously correlated optically with all the stored reference images. These correlations are threshold and fed back to the input side, where the stronger correlation reinforces the corrupt distorted input image.

In holographic correlators, the reference images are stored in a thin hologram and are retrieved in a coherently illuminated feedback loop. The input signal, either noisy or incomplete, may be applied to the system and can simultaneously be correlated optically with all the stored reference images. These correlations can be threshold and are fed back to the input, where the strongest correlation reinforces the input image. The enhanced image passes around the loop repeatedly, which approaches the stored image more closely on each pass, until the system gets stabilized on the desired image**.**



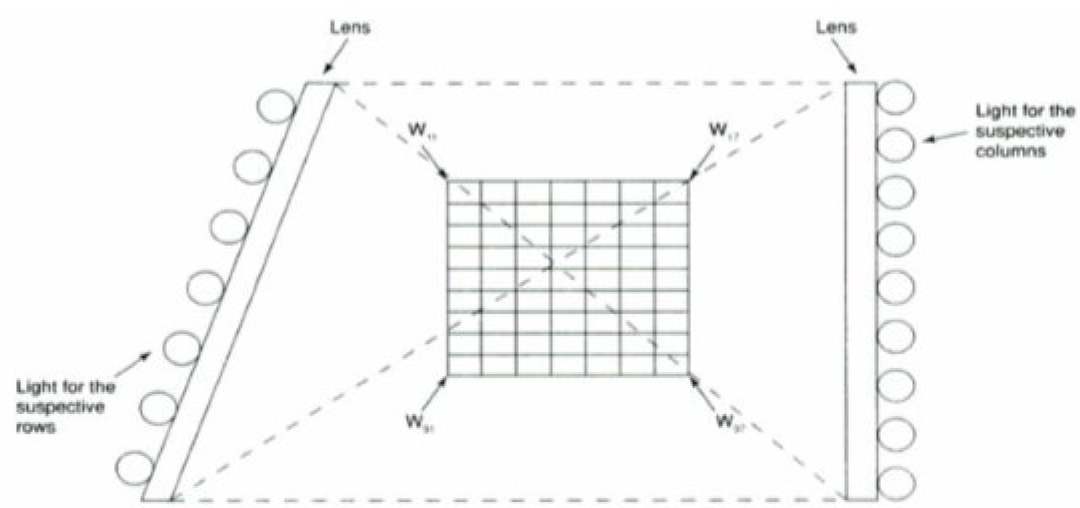
ELECTRO – OPTICAL MULTIPLIERS:

Electro-optical multipliers, also called electro-optical matrix multipliers, perform matrix multiplication in parallel. The network speed is limited only by the available electro-optical components; here the computation time is potentially in the nanosecond range.

An electro – optical matrix multiplierperforms parallel multiplication in a faster way. The n/w speed is limited by the available electro-optical components and the computation time is in nanosecond ranges. The figure represents the electro-optical matrix multiplier. The NET output vale is calculated as follows.

$$NET = \sum W_{ij}X_i$$

Thus the light falling on the photo detector is the sum of product of light intencities transmitted column wise.Each photodetector output is the dot product of input and column weight matrix.



Typical electro-optic multiplier

NEURO PROCESSOR CHIPS

Neural networks implemented in hardware can take advantage of their inherent parallelism and run orders of magnitude faster than software simulations. There exists a wide variety of commercial neural network chips and neuro computers. A few are listed below:

Probabilistic RAM, pRAM-256 neural net processor.

Neuro Accelerator Chip (NAC).

Neural Network Processor (NNP), developed by Accurate Automation Corporation.

CNAPS- 1064 digital parallel processor chip.

IBM ZISC036.

INTEL 80170NX Electrically Trainable Analog Neural Network and so on.

UNIT 5 BAM, ART & OPTICAL NETWORK

PREPARED BY: Dr.S.Kishore

QUESTIONS FOR PRACTICE

PART A

1. What is BAM?
2. What is encoding?
3. Define ART?
4. What are the different types of ART?
5. Brief about optical neural networks?
6. What is Cognitron?
7. Brief about Neocognitron?
8. What is Plasticity- Stability dilemma?
9. What are Neuroprocessor chips?

PART B

1. Explain about associative memory and its types?
2. Discuss about different types of ART in detail?
3. Elaborate on the different models of optical neural networks?
4. Discuss about ART1 architecture and training?
5. Explain ART 2 architecture, algorithm and training?
6. Elaborate on the Cognitron and Neocognitron with necessary diagrams?

SECX1048 FUNDAMENTALS OF FUZZY LOGIC AND ARTIFICIAL NETWORK

UNIT 5 BAM, ART & OPTICAL NETWORK

PREPARED BY: Dr.S.Kishore