MODULE - 4
ATTRACTOR NEURAL NETWORKS

Associative Learning Attractor Associative Memory, Linear Associative memory, Hopfield Network, application of Hopfield Network, Brain State in a Box neural Network, Simulated Annealing, Boltzmann Machine, Bidirectional Associative Memory.

<u>Characteristics of associative memory :</u>

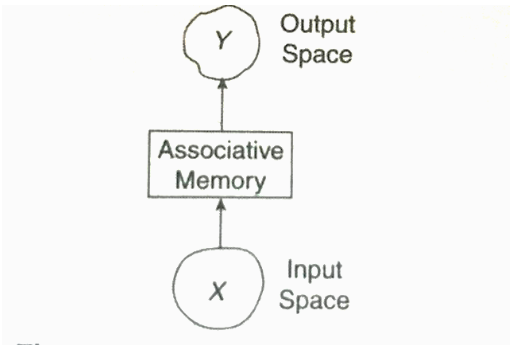
1. Memories are distributed.
2. Stimulus pattern and response pattern are in the form of data vectors.
3. Information contained in stimulus also determines the address for retrieval (place where it is found).
4. Associative memory has a high degree of resistance to noise. (though neurons are noisy)
5. High degree of interaction between the patterns and hence there is a chance for error during the recall.

NOTE: **Hebbian learning** principle implies that synaptic dynamics are governed by the conjunction of free semantic and post semantic packages and leads to powerful associative memory that relate or associate one concept or idea with another.

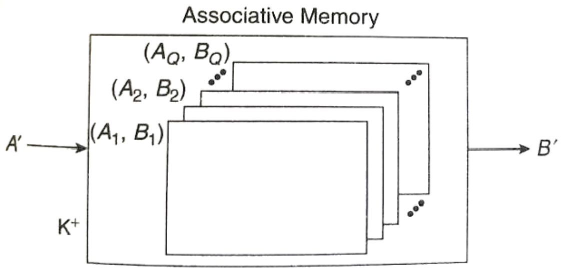
The dynamics of the system can be visualised by imagining a higher dimensional wavy energy surface generated from Lyapunov function much in a way that undulation can be pinched onto a rubber sheet by holding it down at various places in space. The ball that is left at some point on the rubber sheet rolls to the closest local minima under the influence of gravity.
To make the ball go to a specific place, give multiple forces or keep extracting information stored in the network.
In neural network dynamics, the state of the networks maps to a point on the energy surface and the attractor neural network falls to an energy state that represents the closest local minima under the influence of the vector field. This vector field is characterized by the structure of connection , weights and signal function. Network states evolve in time until a local minima is reached after which the state stops evolving further.
To perform useful computation in an attractor neural network, the point of local minima at which network state stabilises is to be interpreted as memory of the system.
An input vector sets the initial state of the network which is represented by a point on an energy surface. The network state evolves in time till the system hits the local energy minimum. This final state is the recalled memory. Vector. The minimum energy points of the Lyapunov surface have therefore to be mapped to the desired memory state.
The energy surface gives the state of the network.

1. Write a note on associative memory and its models.

Associative memory model :



Associative memory model maps the data from input space to the data in the output space from unknown domain points to known range points.
Memory learns an underlying association from a training data set.
In a common associative memory, an input vector generates an output response on the basis of a set of associations programmed into the system.
When one vector maps to itself we say the memory is autoassociative. When a vector maps to another vector, we say memory is heteroassociative.
Associative memory models considered have their origin in additive neuronal dynamics and are non-learning in nature, that is connection strengths are programmed apriori based on association that are coded in the system.



The connection matrix W encodes all programs association $\{A_i , B_i\}^{Q}_{i=1}$ where A_i belongs to R^n and B_i to R^m . as shown in figure.
If A_i , B_i is one of the programmes in memory then B_i is called associant of A_i .

Presentation of some input vector A' produces an output or response vector B' . If A' is close to some encoded association A_i then response B' generated is close to B_i . A simple associative memory is generated by a two layer neural network as shown in figure.

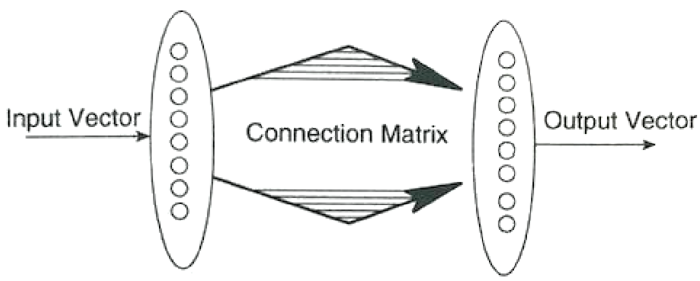
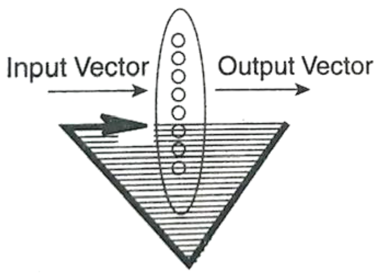


Fig. 10.4 *A simple associative memory is generated by a two-layer neural network.*

(a)



(b)
Connection Matrix

The associative memory generated by using 2 layer are assumed to be connected through n x m connection matrix 'W'

If all the neurons are assumed to be linear then response $B=A^TW$ where $A \in R^n$ and $B \in R^n$. Additional component can be introduced to the competition of same network by employing non linear signal function which transforms the response equation to

$$B = S(A^T W)$$

The feedback signal in the model introduces dynamics in the network operation.

HEBB ASSOCIATION MATRIX :

Hebb postulate states that = "The change in synaptic efficacy is proportional to the product of pre and postsynaptic neuron signals . The connection matrix W is built using this principle. Consider a 2 layer L_x and L_y of n and m neurons. The neurons from L_x connect to L_y using a weight matrix 'W' . The Hebbian Learning generates weight values in accordance with pre and post synaptic activity of neurons. The change in weight is computed as

$$\Delta W_{ij} = \eta a_i b_j \quad \text{— (1)}$$

If a_i b_j are correlated , then the synaptic weight strengthens.

Eqn (1) is referred to as the generalised Hebb rule.

If signal vector $A = [a_1,....a_n]^T$ and $B = [b_1....b_m]^T$ then according to Hebbian Learning , connection matrix

$W = [a_1b_1 \ldots.. A_1b_m]$

$[.a_nb_1 \ldots. A_nb_m]$

$W = AB^T$

Which is nothing but the outer product of signal vectors A and B.

The weight W_{ij} that connects neuron i of L_x to neuron j of L_y is the product of respective signals a_i and b_j of both neurons, The association a,b is thus encoded into the matrix.

2. What are linear associative memories

Linear Associative Memory :

Assume that a single association (A,B) has been encoded into the connection matrix W using the outer product $W = AB^T$. If we present the vector A at L_x then the activity that develops across L_y is given by

$$(B')^T = A^T W$$
$$= A^T AB^T$$
$$= \|A\|^2 B^T \text{ (scaled and normalized)}$$

That is output that is recalled across L_y is scaled version of B (associant of A) If vector A is normalized then recall is exact $\|A\|^2 =1$

If more than one association needs to be encoded then the natural action to follow is to superpose the individual association matrices.

Assuming Q associations $\{ A_k , B_k \}^Q_{i=1}$ that are to be encoded, resulting connection matrix is then given by

$$W = \sum_{k=1}^{Q} A_k B_k^T$$

Such superposition will certainly cause the individual memories to interfere with each other leading to degradation of memory recall or performance.

A special case arises if the vectors A_k are orthogonal and normalized to unity magnitude.

In this case for input A_i

$$(B')^T = A_i^T W$$

$$= A_i^T \sum_{k=1}^{Q} A_k B_k^T$$

$$= A_i^T \sum_{k=i}^{Q} A_k B_k^T + A_i^T \sum_{k\neq i}^{Q} A_k B_k^T$$

When vector A_k are orthogonal $A_i^T A_i = 1$ and $A_i^T A_j = 0$

$$(B')^T = B_i^T$$

This means that the recall is perfect. Such associative memories are called orthogonal linear associative memory (OLAMS)

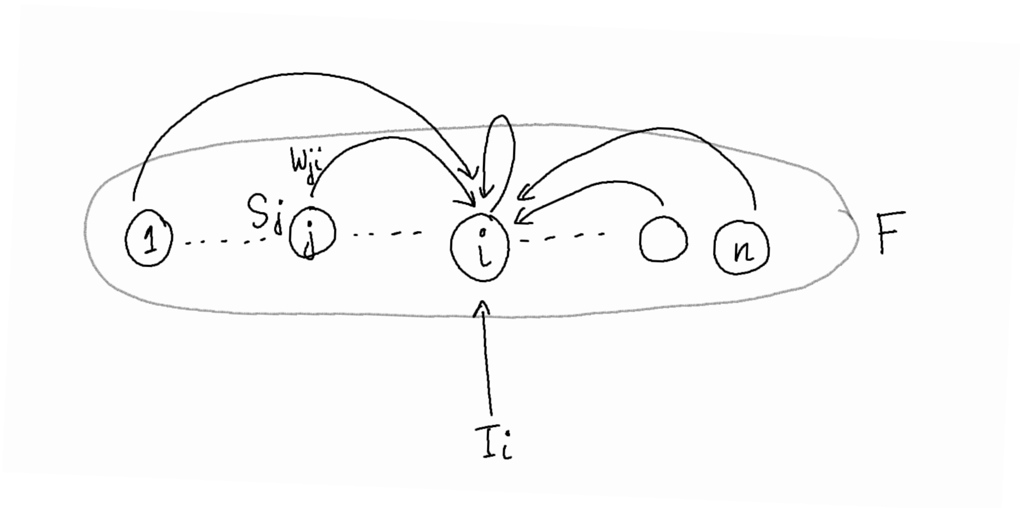
3. Write about Hopfield networks and its equivalent circuit interpretation

HOP FIELD NETWORK :

Recurrent single layer neural networks are essentially dynamical systems that feed signals back to themselves. These models possess a rich class of dynamics characterized by the existence of several stable states, each with its own basin of attraction.

If we map these stable states to correspond to certain desired memory vectors, then time evolution of dynamic systems leads to stable states where the outputs of the network correspond to a specific memory.
The Hebbian matrix provides a way to encode association as memories into the neural network.

Architecture detail of HopField network :



$$X_i = -A_i x_i \quad + \quad \sum_{j=1}^{n} W_{ji}\, S_j\,(x_j) \quad + \quad I_i$$

Passive decay signal feedback external input

Hopfield network embodies additive dynamics.
Layer F comprises n neurons which receive external input I_i $i=1..,n$. Each neuron feeds back signals $S_i(x_i)$ through weights W_{ij} $j=1\ldots n$, The simplest form of dynamics is the additive activation dynamics model that is

$$X_i = -A_i x_i + \sum_{j=1}^{n} W_{ji}\, S_j\,(x_j) + I_i \quad \text{—(1)}$$

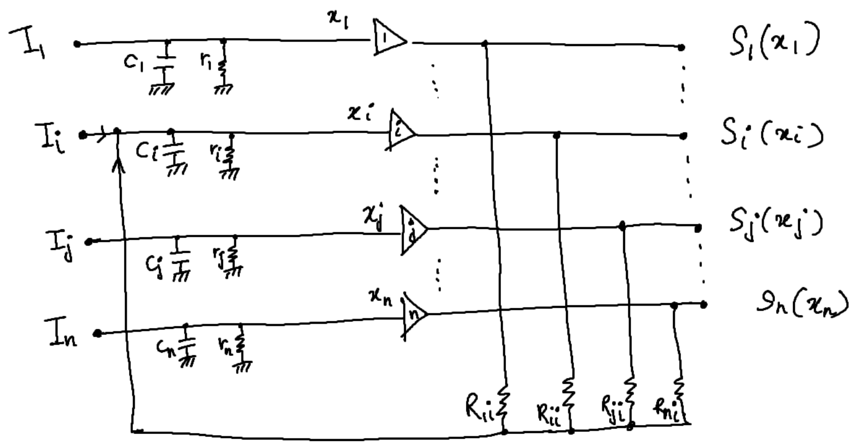
where x_i is the state $= \frac{dx_i}{dt}$

The individual neurons can have $S_i(x)$ distinct sigmoidal characteristics having transfer function :

$$S_i(x) = \frac{1-e^{-\lambda_i x}}{1+e^{-\lambda_i x}}$$

Equation 1 represents a high dimensional cross coupled nonlinear dynamic system And usually $W_{ii} = 0$ (no self feedback) and $W_{ij} = W_{ji}$ (symmetric)

Electrical interpretation of additive dynamics :



Equation 1 has a circuit equivalent interpretation which has n amplifiers which feedback current through Resistance R_{ji} . Each amplifier is associated with input capacitor C_i and resistor r_i and each amplifier receives an external current input I_i . Assume ith input and output of amplifier voltage to be x_i and $S_i(x_i)$

Where S(.) is the transfer function of the amplifier. One applying KCL at the input node of ith.

$$C_i x_i + \frac{x_i}{r_i} = \sum_{j=1}^{n} \frac{S_j(x_j) - x_i}{R_{ji}} + I_i$$

On rearranging above equation :

$$X_i = -\frac{x_i}{R_i C_i} + \sum_{j=1}^{n} \frac{S_j(x_j)}{C_i R_{ji}} + \frac{I_i}{C_i} \quad \text{—(2)}$$

$$\frac{1}{R_i} = \frac{1}{r_i} + \sum_{j=1}^{n} \frac{1}{R_{ji}}$$

Equation 1 and equation 2 are similar,

$$A_i = \frac{1}{R_i C_i},\ W_{ji} = \frac{1}{C_i R_{ji}}$$

External input $I_i = \frac{1}{C_i}$

The energy E is a function of network states and decreases monotonically with time until the system settles down into any one local minima of the energy surface .

At the local minimum the energy stops changing and network state stabilizes. The class of recurrent networks therefore possess stable dynamics. In summary, the network relaxes to an attractor whose basin of attraction the initial state lies. It is this phenomenon of attraction to fixed network states that is to be interpreted as memory recall.

4. How Hopfield network is used as content addressable memory

Content Addressable Memory : (memory that can be accessed by content and not by the address)

A CAM recalls data based on the input that partially resembles the data itself. Primary application of the Hopfield network is CAM.
Eg: a binary number 11001010 can be recalled given an input that resembles it such as 11001000 where one bit has been distorted. The key to the memory is a partially distorted version of the data stored in that memory. Recall is essentially a cleanup operation on a partially distorted input. In other words, CAM solves the pattern completion problems.

Now imagine that the attractor states represented by the minima of the energy surface are memories that have been programmed into the network. Then the application of a distorted version of a programmed memory to the network should result in distortion being removed to recall the memory correctly. This correct recall operation however depends on the Hamming distance of the initially applied input state vector from the programmed memory. To understand this, observe that movement from an initial point (initial state vector) away from an attractor point (memory state vector) on the energy surface amounts to distortion of certain bits in the state vector. Increasing distortion amounts to increasing the distance from the attractor. For distortion to a certain extent, the initial point remains within the basin of attraction of the memory state vector and it falls towards the attractor as time evolves and the network is allowed to relax. For larger distortion in the input vector, the initial point might fall outside the basin of attraction of the programmed memory and enter the basin of attraction of the same other memory.

Encoding memories via outer products :

In Hopfield network, it is required to map a set of desired memories $\{A_i\}$ to stable states . The state space will then contain a set of fixed stable points representing those desired memories.
To encode binary vectors $\{A_i\}$ from i=1 to q into the Hopfield CAM , the weight matrix is generated as :

$$W = \sum_{k=1}^{Q} X_K X_k^T - QI$$

The retrieval of memory may happen synchronously or asynchronously. In standard Hopfield CAM neuron states are updated or updates occur asynchronously as per the activation $S(Y^{K+1}) =$
1 for $Y_I^K > 0$ and
-1 for $Y_I^K < 0$
0 for $Y_I^K = 0$

5. Write about Hopfield network as pattern completion and 3D object recognition

Hopfield network and BAM are used to implement associative memory applications. This associative property emerges by virtue of the symmetry of internal connections and from the fact that desired memories are programmed to be stable states of the network. So computation is viewed as an error correcting motion towards the closest memory on an energy landscape. In the context of ANN, if solutions of the optimized problem can be mapped onto the stable states of the neural network then the network can be made to evolve towards one of these solutions from an initial state.

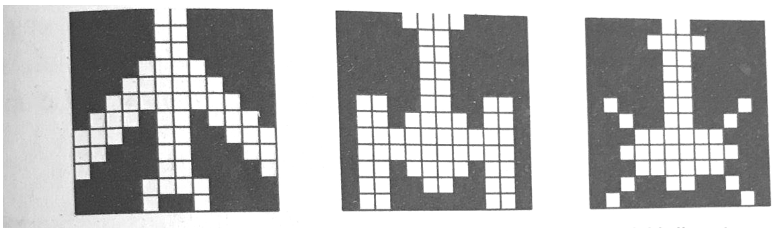
Pattern Completion :

Hopfield network is used to encode 3 graphic patterns : a plane, a tank, a helicopter as memories. These images are 12x12 pixel grids. Each figure is represented as 144 dimensional bipolar vector. 1 for white pixels and -1 for black pixels. These vectors are encoded into a 144- dimensional Hopfield network using bipolar outer product encoding which also encodes their complements. This means that there are going to be 3 complement state spurious attractors amongst a number of other mixed states. Partial distorted patterns will be cleaned up to their original form.

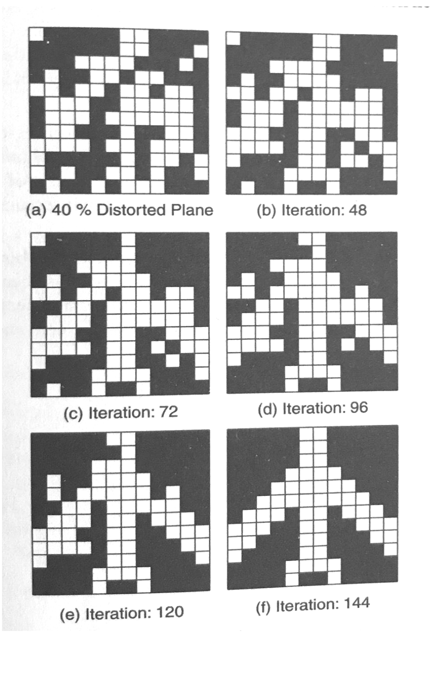
In other words as long as the distortion level is such that the initial point lies within the basin of attraction of the encoded memory, image clean up and proper recall are guaranteed.

Among the samples of the three images, a random sample of 58 pixels of the original 144 pixels of the plane image have been inverted. We use pure asynchronous updates and ensure that each neuron in the network is updated the same number of times on average.

The order of update of the neurons was randomized, but it was ensured that each neuron was updated at least once during a field update cycle involving all 144 neurons.



(a) Plane (b) Tank (c) Helicopter



(a) 40 % Distorted Plane (b) Iteration: 48

(c) Iteration: 72 (d) Iteration: 96

(e) Iteration: 120 (f) Iteration: 144

Three dimensional Object Recognition :

3D object recognition involves matching an object to a description of a scene in order to determine its position and orientation in space.

The solution to the problem employs a database of object models which comprises 2D projections of 3D objects viewed from different angles. These topologically different views are referred to as characteristic views and are usually generated by an automated procedure.

It is usually a two step recognition process. The first step is coarse grained recognition of the object is performed based on polygonal region matching of the 2D projections. Having reduced the search space, the second step is a fine grained recognition is performed based on matching of vertex sequences.

Both stages employ a Hopfield network that comprises a 2D array of neurons.

(i) Feature selection : 3D objects are described by 2D in the form of line drawings obtained by segmentation of the original image. The starting point is to describe each projection by a set of features.

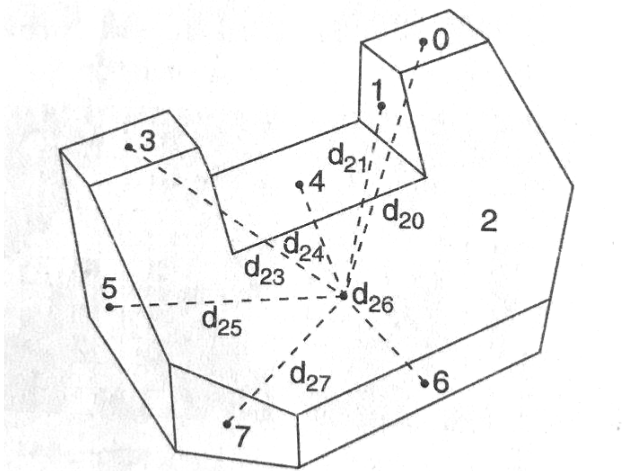


Diagram shows a 2D projection of a slotted wedge with different regions numbered from 0 through 7. This is a labeled image that involves calculating the area of each region and tracing the boundary to determine corner points. The area is a local feature of each region. Once the region is labeled, a global relational feature is computed.

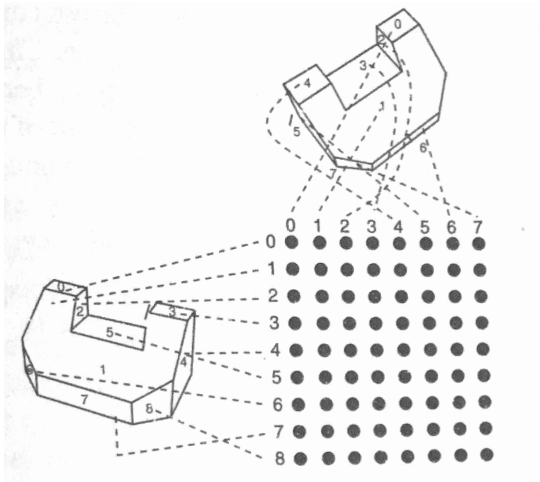
In the image, for region 2, the global feature is the distance from the centroid of region 2 to the centroids of all other regions. Both the area and distances are suitably normalized.

(ii) Architecture of the network : A neuron signal reflects the degree of similarity between image and model region for the corresponding row and column. In order to design connection strengths for the Hopfield network, one requires a suitable energy function that is to be minimized.

The present application employs 2D arrays of neurons that have a Hopfield network architecture. Each row corresponds to a region of an input image and each column corresponds to a region of an object model.

(iii) Implementation : Each region of an input image corresponds to a row of Hopfield network. Each column of the network corresponds to a region in the model. With the constraints embedded into the connections of the network, we allow the network to relax to a stable state by continuous time update procedure of the Hopfield network.

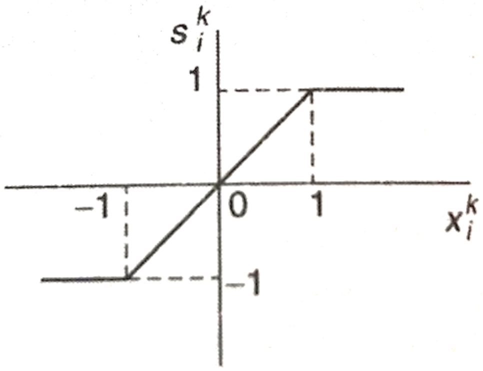
(iv) Fine grained search : The application employs search based on fine grained features such as vertex sequence correspondence. Each row of the Hopfield network represents a vertex on the input image and each column represents a vertex on the model object. Vertices in the object are numbered clockwise starting at the smallest x coordinate. Constraints are set up to the network now derived from vertex correspondence establishment. Correspondence between vertex views of two polygons is easily established by fixing one of them and rotating the other by one vertex and accumulating the errors that arise from the corresponding differences at each step until the rotating image is brought back to its original position.



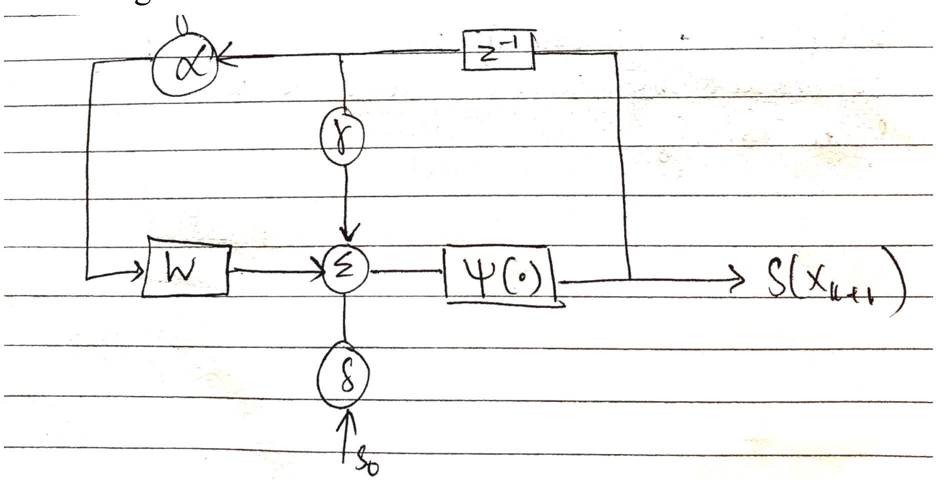
6. Explain the BSB NN and its operational details

<u>Brain state in a box NN (BSB) :</u>

BSB is a predecessor of Hopfield network . It is an autoassociative model where the connection matrix is computed in the usual way . The difference is that it uses linear threshold signal function.



Block diagram of BSB :



Here neurons are connected in a positive feedback fashion through a weight matrix W. Signals across the fields update synchronously using the same iterative procedure until the signal stabilizes.
Let $X = \{x_1^k{}_{, \ldots}\, x_n^k\}$ and
$S_k = \{S_1^k{}_{, \ldots}\, S_n^k\}$ where X is an accuration vector and S_k is a signal vector, at time constant 'k' /
The neuron updates occurs as per the equation :

$$X^T{}_{k+1} = \gamma\, S_k^T + \alpha\, S_k^T\, W + \delta S_0^T \text{ —(1)}$$

$$S^T{}_{k+1} = S(X^T{}_{k+1})$$

In the activation computation signal vector scales by a factor γ and feedback is controlled by scaled vector α and the initial state is S_o .

For $\gamma = \alpha = 1$ and $\delta = 0$

$$X^T{}_{k+1} = S_k^T + S_k^T\, W$$

That is, a new activation vector is a sum of signal vectors and net feedback.
Signal function is given by :

$$S(x_i^k) = \begin{cases} -1 & \text{for } x_i^K \\ x_i^K & \text{for } -1 \le x_i^K \le 1 \\ +1 & \text{for } x_i^K > 1 \end{cases}$$

The linear threshold signal function clips the signal as they exceed lower and upper limit and thus constrains the BSB state vector to remain within the bipolar n dimensional unit hypercube centered at the origin.
An activation pattern S_o is input into the BSB model as the initial state vector from this, we obtain the state vector S_1 and is cycled through equation (1) and (2)

The process is repeated until the BSB model reaches a stable state represented by a particular corner of the unit hypercube.
Intuitively, positive feedback in BSB model , causes initial state vactor s_o to increase in euclidean length with an increasing no. of iterations until it hits a wall of the box. Then, slides along the wall and eventually ends up in a stable corner of the box, where it keeps on pushing but cannot get out of the box hence the name of the model.

7. Explain the concept of Simulated Annealing(SA). Also write the basic steps used in SA

Simulated Annealing :

Simulated Annealing provides a general framework for optimisation of the behaviour of complex systems. It operates by introducing noise in a controllable fashion into the operational dynamics of the system. If specific combination of neuron state is the configuration of the network, different configuration of the network means a particular instance of signal vector, indexed by gamma in 'n' node network with 2^n different configurations, the basic idea underlying SA is to generate different configurations of the system at various values of control parameters (temperature) and to gradually reduce the value of this parameter to search for an optimal state solution.
Consider a classic quadratic energy optimisation problem in a hopfiled network with bipolar signal vectors

$$S_i^{\gamma} = \{ -1, +1\} \qquad 1 \leq i \leq n$$

Here our objective is to find the values S_i^{γ} and S_j^{γ} that minimizes the Lyapunov Energy .

$$E_{\gamma} = - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} S_i^{\gamma} S_j^{\gamma}$$

The simulated annealing algorithm :
Step 1 : Randomise neuron states in the beginning and initialise the temperature to a high value.
Step 2 : Choose neuron 'I' randomly from the network .
Step 3 : Compute the energy E_A of present configuration 'A'
Step 4 : Flip the state of neuron 'I' to generate a new configuration 'B' .
Step 5: Compute energy E_B of configuration 'B'
Step 6: If $E_B < E_A$ then accept the state change of neuron I
Otherwise accept the state change of neuron I with probability

$$P = e^{\frac{-\Delta E}{T}}$$

Where $\Delta E = E_B - E_A$

Step 7 : Continue selecting and testing neurons randomly and set their states several times in this way until equilibrium is reached.
Step 8 : Finally lower the temperature and repeat the procedure
$T_{K+1} = C\, T_K \qquad 0 < C < 1$

8. Write about Boltzmann machine architecture and its operational details

<u>BOLTZMANN MACHINE :</u>
This is an extension to the discrete Hopfield network. BM replaces the deterministic local search dynamics of Hopfield network by a randomised local search dynamics.
The model introduces stochastic learning algorithms in place of simple Hebbian rule employed in Hopfield network.
The relaxation in BM is done using Simulated Annealing.

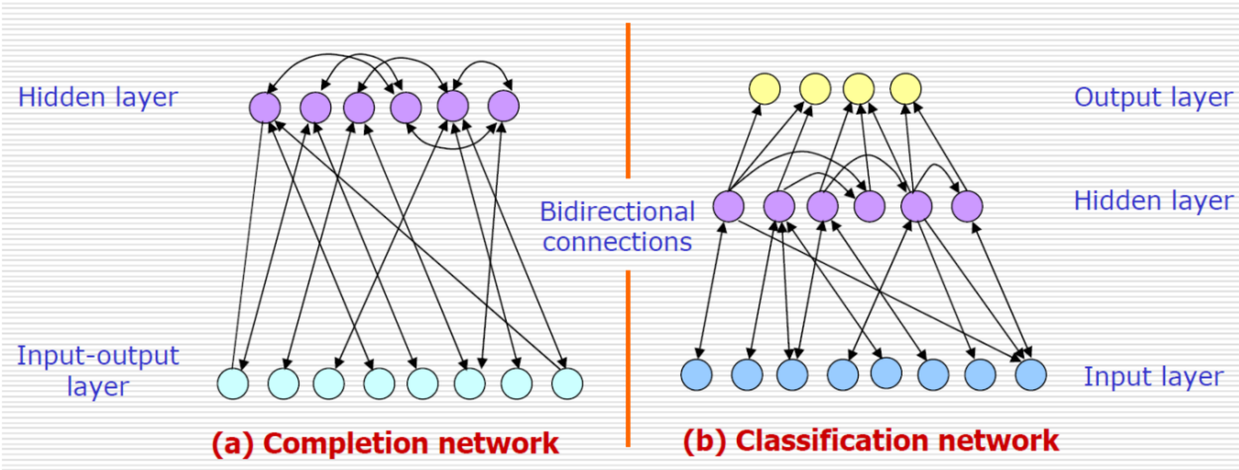


Figure (a) is a completion network and figure (b) is the classification network for Hopfield BM.
Hidden neurons do not receive any external inputs and all connections are bidirectional.
Neurons are classified as visible neurons and hidden neurons.
Visible neurons provide an interface to the network and environment. These neurons are clamped to specific states determined by the environment during the training.
Hidden neurons are always free and they are designated to extract underlying features. By clamping procedure, the probability distribution of clamping patterns are modelled and the network can perform a pattern completion task.
In completion network figure (a) architecture, a partially unspecified vector is expected to be completed (corrected) by clamping (holding fixed) the neuron signals to the corresponding known values and allowing the remaining unknown values to be determined through a network relaxation process that is based on simulated annealing.
On the other hand, in an input- output network i.e. figure (b), the inputs are clamped to a particular value and outputs are determined through a relaxation process (SA).

Operational details of BM relaxation procedure :

1. Computation of energy difference.
   Suppose the current state of neuron 'I' has an energy state

$$E_A = - S_I \sum_{j=1}^{j} W_{Ij} S_j$$

   Similarly if E_B is the energy when a neuron flips from a state S_I to - S_I (+1,-1)
   Then ,

$$E_B = - (- S_I \sum_{j=1}^{j} W_{Ij} S_j)$$

   So energy difference, $\Delta E = E_B - E_A$ (substitute values)

$$\Delta E = 2 S_I \sum_{j=1}^{j} W_{Ij} S_j$$

$$\Delta E = 2 S_I x_I$$

   Where x_I is the net activation

$$x_I = \sum_{j=1}^{j} W_{Ij}S_j \quad \text{—(1)}$$

Find the probability,

$$P(S_I \rightarrow -S_I) = \frac{1}{1 + e^{\frac{\Delta E}{T}}}$$

$$P(S_I \rightarrow -S_I) = \frac{1}{1 + e^{\frac{2S_I x_I}{T}}} \quad \text{—(3)}$$

2. Basic steps in BM relaxation procedure :
   We assume that a partially specified input vector S' has to be computed by the network and following procedure is followed for this :
   (i) Clamp the signal values of visibles units to the corresponding known values.
   (ii) Randomize the signals of neurons corresponding to unknown input values and signals of hidden neurons for the set {-1,+1} .
   (iii) Initialise the temperature $T = T_0$ (high value)
3. Select a neuron I at random for update.

   Compute its activation x_I as given in equation (1) and use $P(S_I = +- 1) = \dfrac{1}{1 + e^{\frac{+-2S_I x_I}{T}}}$

   To compute signal state.

   Alternatively, flip the signal value, in accordance $P(S_I \rightarrow -S_I)$ in equation (3) .

   Note that signals for clamped neurons will not be allowed to change during the update cycles.
4. Repeat the procedure until all the neurons are polled several times.
5. Reduce the temperature and repeat the polling cycle. Stop when the temperature is low (simulated annealing) procedure : $T_{K+1} = C\, T_K$
   Weight update rule Boltzmann Learning:
   Equation is given by
   $\Delta W = \eta(\delta_{ij}^{+} - \delta_{ij}^{-})$

   δ_{ij}^{+} = correlation of neuron state in clamped state

   δ_{ij}^{-} = correlation of neuron state in free state

9. Explain how BAM can be used as heteroassociative memory

We will see how to encode Q binary associations $\{A_k , B_k\}^{Q}_{k=1}$ into a BAM.
Associations can be programmed into the weights using Hebb learning rules and by subsequently superimposing the individual association matrices.

$$W = \sum_{k=1}^{Q} X_k Y_K^T$$

Where $X_k = 2A_k - 1$ and $Y_k = 2B_k - 1$ are bipolar equivalents of A_k , B_k .
The diagonal is not zeroed out in BAM connection matrices as is done in Hopfield CAMs.
Bipolar encoding can generate synaptic strengths which can be positive or negative.
The sequence converse to vector pairs (A_k,B_k) which is one of the closest programmed memory of the system.
Formalizing the model using bidirectional update procedure : L_x, L_y layers update their states depending upon their activity levels either synchronously or subset asynchronously.
Let x_i and y_i denote neuronal activations and let a_i and B_i denote corresponding neuronal activations on layers L_x and L_y.
At iteration k, the forward direction L_x and L_y

$$Y_j^k = \sum_{i=1}^{n} w_{ij} a_i^k$$

L_y activations :

$$b_j^k = S(y_j^k) = \begin{cases} 1 & \text{if } y_j^k > 0 \\ 0 \text{ binary} & \text{if } y_j^k < 0 \\ -1 \text{ bipolar} & \text{if } y_j^k = 0 \\ b_j^{k-1} & \text{if } y_j^k = 0 \end{cases}$$

In the reverse direction L_y and L_x

$$X_i^{k+1} = \sum_{j=1}^{m} w_{ji} b_j^k$$

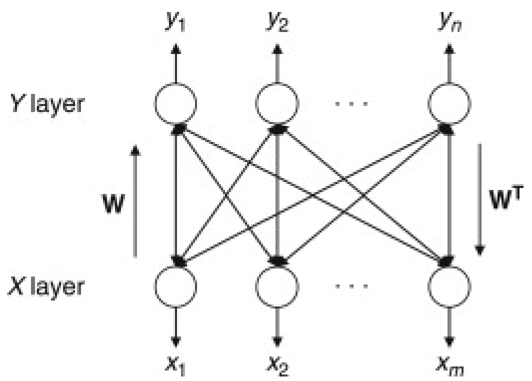
L_x is

$$Er_i^{k+1} = S(x_i^{k+1}) = \begin{cases} 1 & x_j^{k+1}>0 \\ 0 & x_j^{k+1} < 0 \\ -1 & x_j^{k+1} = 0 \end{cases}$$

Iteration time increments over one in a complete forward reverse cycle of updates. Neurons continue to update these states until a bidirectional state equilibrium $(A_f B_f)$ is reached.

Bidirectional Association Memory (BAM) :

BAM is a two field attractor neural network is a variety of association memory models continuous BAMs employ additive dynamics in a fashion parallel to that of Hopfield network.
Architecture and operation :



Consider the forward chain of activity transmission from L_x to L_y

$(A')^T.W = B'$

Signal vector A' of L_x is composed with weight matrix W to set up activations in L_y which are then thresholded to generate the signal vector B' across L_y .

Assuming that A' is closer to an encoded association A_i than to all other encoded associations A_j , it is desired that the vector B' i.e generated is closer to B_i than to any other encoded B_j ,B_i being the vector associated with A_i . If we wish to feed back B' to L_x in order to improve the accuracy of recall we may do so by using W^T , This reverse chain activity transmission from L_y to L_x :

$A'' = (B')^T W^T$

A'' can then be fed forward through W to generate B'' . Continuing this bidirectional process generates the sequence of vector pairs.

$(A')^T W = B'$

$A'' = (B')^T W^T$

$(A'')^T W = B''$

$A_f^T W = B_f$

$A_f = B_f^T W^T$