Associative Memories

Hopfield Networks

Hopfield neural network was invented by Dr. John J. Hopfield in 1982. It consists of a single layer which contains one or more fully connected recurrent neurons. The Hopfield network is commonly used for auto-association and optimization tasks.

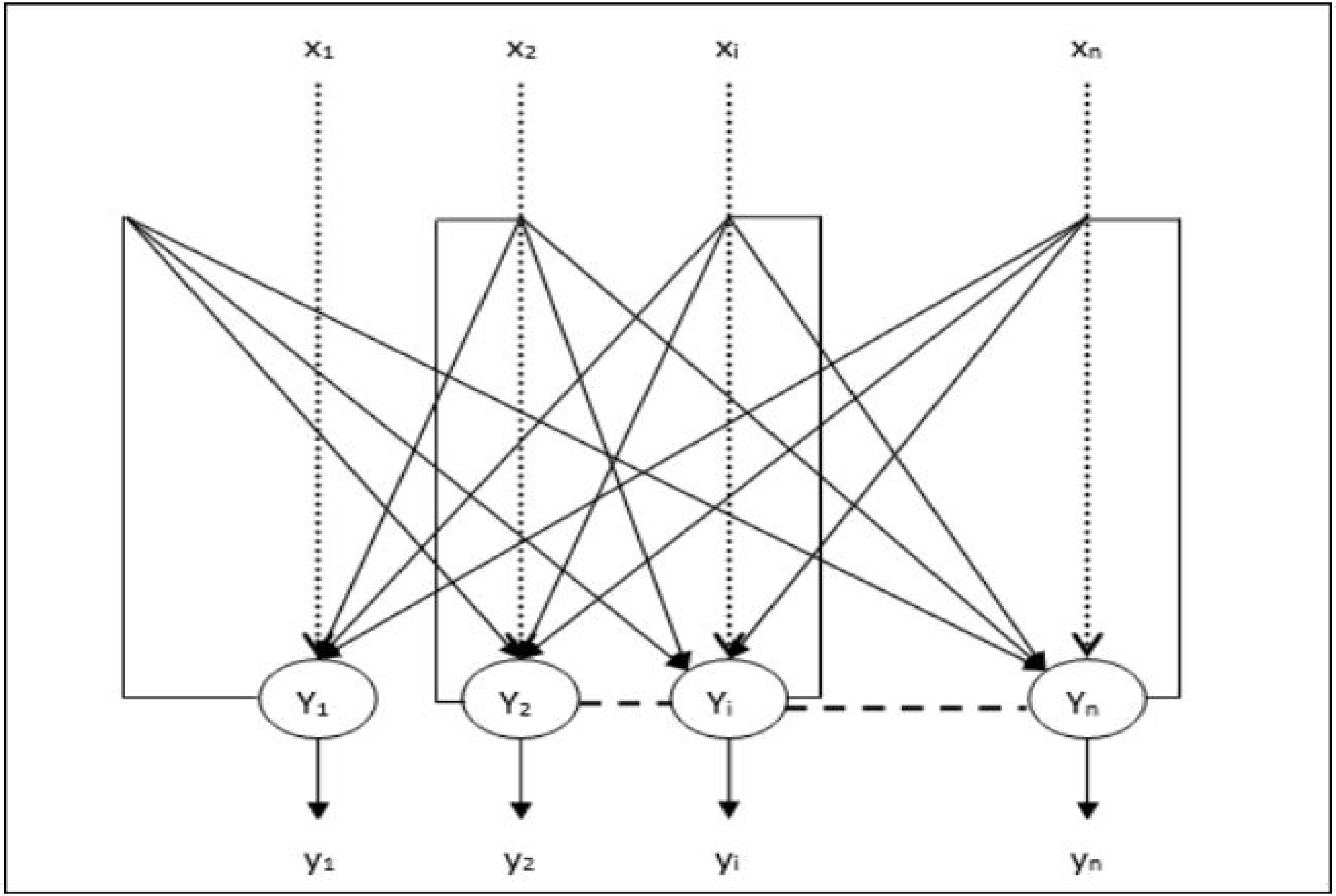
Discrete Hopfield Network

A Hopfield network which operates in a discrete line fashion or in other words, it can be said the input and output patterns are discrete vector, which can be either binary $0, 1$ or bipolar $+1, -1$ in nature. The network has symmetrical weights with no self-connections i.e., $\mathbf{w_{ij} = w_{ji}}$ and $\mathbf{w_{ii} = 0}$.

Architecture

Following are some important points to keep in mind about discrete Hopfield network −

- This model consists of neurons with one inverting and one non-inverting output.
- The output of each neuron should be the input of other neurons but not the input of self.
- Weight/connection strength is represented by $\mathbf{w_{ij}}$.
- Connections can be excitatory as well as inhibitory. It would be excitatory, if the output of the neuron is same as the input, otherwise inhibitory.
- Weights should be symmetrical, i.e. $\mathbf{w_{ij} = w_{ji}}$



The output from $\mathbf{Y_1}$ going to $\mathbf{Y_2}$, $\mathbf{Y_i}$ and $\mathbf{Y_n}$ have the weights $\mathbf{w_{12}}$, $\mathbf{w_{1i}}$ and $\mathbf{w_{1n}}$ respectively. Similarly, other arcs have the weights on them.

Training Algorithm

During training of discrete Hopfield network, weights will be updated. As we know that we can have the binary input vectors as well as bipolar input vectors. Hence, in both the cases, weight updates can be done with the following relation

Case 1 − Binary input patterns

For a set of binary patterns **s_p, p = 1 to P**

Here, **$s_p = s_1p, s_2p, \ldots, s_ip, \ldots, s_np$**

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P}[2s_i(p) - 1][2s_j(p) - 1] \quad for\ i \neq j$$

Case 2 − Bipolar input patterns

For a set of binary patterns **s_p, p = 1 to P**

Here, **$s_p = s_1p, s_2p, \ldots, s_ip, \ldots, s_np$**

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P}[s_i(p)][s_j(p)] \quad for\ i \neq j$$

Testing Algorithm

Step 1 – Initialize the weights, which are obtained from training algorithm by using Hebbian principle.

Step 2 – Perform steps 3-9, if the activations of the network is not consolidated.

Step 3 – For each input vector **X**, perform steps 4-8.

Step 4 – Make initial activation of the network equal to the external input vector **X** as follows –

$$y_i \; = \; x_i \quad for\, i \; = \; 1\, to\, n$$

Step 5 – For each unit $\mathbf{Y_i}$, perform steps 6-9.

Step 6 – Calculate the net input of the network as follows –

$$y_{ini} \; = \; x_i \; + \; \sum_j y_j w_{ji}$$

Step 6 – Calculate the net input of the network as follows –

$$y_{ini} = x_i + \sum_j y_j w_{ji}$$

Step 7 – Apply the activation as follows over the net input to calculate the output –

$$y_i = \begin{cases} 1 & if\, y_{ini} > \theta_i \\ y_i & if\, y_{ini} = \theta_i \\ 0 & if\, y_{ini} < \theta_i \end{cases}$$

Here θ_i is the threshold.

Step 8 – Broadcast this output $\mathbf{y_i}$ to all other units.

Step 9 – Test the network for conjunction.

Energy Function Evaluation

An energy function is defined as a function that is bonded and non-increasing function of the state of the system.

Energy function **E_f**, also called **Lyapunov function** determines the stability of discrete Hopfield network, and is characterized as follows −

$$E_f = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j w_{ij} - \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} \theta_i y_i$$

Condition − In a stable network, whenever the state of node changes, the above energy function will decrease.

Suppose when node **i** has changed state from $y_i^{(k)}$ to $y_i^{(k\,+\,1)}$ then the Energy change ΔE_f is given by the following relation

$$\Delta E_f \;=\; E_f(y_i^{(k+1)}) \;-\; E_f(y_i^{(k)})$$

$$= \;-\left(\sum_{j=1}^{n} w_{ij} y_i^{(k)} \;+\; x_i \;-\; \theta_i\right)(y_i^{(k+1)} \;-\; y_i^{(k)})$$

$$= \;-(net_i)\Delta y_i$$

Here $\Delta y_i \;=\; y_i^{(k\,+\,1)} \;-\; y_i^{(k)}$

The change in energy depends on the fact that only one unit can update its activation at a time.

Problem

Consider the following problem. We are required to create a Discrete Hopfield Network with the bipolar representation of the input vector as [1 1 1 -1] or [1 1 1 0] (in case of binary representation) is stored in the network. Test the Hopfield network with missing entries in the first and second components of the stored vector (i.e. [0 0 1 0]).

Given the input vector, x = [1 1 1 -1] (bipolar) and we initialize the weight matrix **(w_{ij})** as:

$$w_{ij} = \sum[s^T(p)t(p)]$$
$$= \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

and weight matrix with no self-connection is:

$$w_{ij} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

As per the question, input vector x with missing entries, x = [0 0 1 0] (**[x1 x2 x3 x4]**) (binary). Make y_i = x = [0 0 1 0] **([y1 y2 y3 y4]).** Choosing unit y_i **(order doesn't matter)** for updating its activation. Take the i^{th} column of the weight matrix for calculation.

(we will do the next steps for all values of yi and check if there is convergence or not)

$$y_{in_1} = x_1 + \sum_{j=1}^{4}[y_j w_{j1}]$$

$$= 0 + \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}0\\1\\1\\-1\end{bmatrix}$$

$$= 0 + 1$$

$$= 1$$

$$Applying\ activation,\ y_{in_1} > 0 \implies y_1 = 1$$

$$giving\ feedback\ to\ other\ units,\ we\ get$$

$$y = \begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix}$$

$$which\ is\ not\ equal\ to\ input\ vector$$

$$x = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}$$

$$Hence,\ no\ covergence.$$

Now for next unit, we will take updated value via feedback. (**i.e. y = [1 0 1 0]**)

$$y_{in_3} = x_3 + \sum_{j=1}^{4} [y_j w_{j3}]$$

$$= 1 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

$$= 1 + 1$$

$$= 2$$

$$Applying\ activation,\ y_{in_3} > 0 \implies y_3 = 1$$

$$giving\ feedback\ to\ other\ units,\ we\ get$$

$$y = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$$

$$which\ is\ not\ equal\ to\ input\ vector$$

$$x = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

$$Hence,\ no\ covergence.$$

Now for next unit, we will take updated value via feedback. **(i.e. y = [1 0 1 0])**

$$y_{in_4} = x_4 + \sum_{j=1}^{4}[y_j w_{j4}]$$

$$= 0 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

$$= 0 + (-1) + (-1)$$

$$= -2$$

$$Applying\ activation,\ y_{in_4} < 0 \implies y_4 = 0$$

$$giving\ feedback\ to\ other\ units,\ we\ get$$

$$y = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$$

$$which\ is\ not\ equal\ to\ input\ vector$$

$$x = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

$$Hence,\ no\ covergence.$$

Now for next unit, we will take updated value via feedback. **(i.e. y = [1 0 1 0])**

$$y_{in_2} = x_2 + \sum_{j=1}^{4}[y_j w_{j2}]$$

$$= 0 \ + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

$$= 0 \ + 1 \ + 1$$

$$= 2$$

$$Applying\ activation,\ y_{in_2} > 0 \implies y_2 = 1$$

$$giving\ feedback\ to\ other\ units,\ we\ get$$

$$y = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

$$which\ is\ equal\ to\ input\ vector$$

$$x = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

$$Hence,\ covergence\ with\ vector\ x.$$

Continuous Hopfield Network

Unlike the discrete Hopfield networks, here the time parameter is treated as a continuous variable. So, instead of getting binary/bipolar outputs, we can obtain values that lie between 0 and 1. It can be used to solve constrained optimization and associative memory problems. The output is defined as:

$$v_i = g(u_i)$$

where,

- v_i = *output from the continuous hopfield network*
- u_i = *internal activity of a node in continuous hopfield network.*

Energy Function

The Hopfield networks have an energy function associated with them. It either diminishes or remains unchanged on update (feedback) after every iteration. The energy function for a continuous Hopfield network is defined as:

$$E = 0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} v_i v_j + \sum_{i=1}^{n} \theta_i v_i$$

To determine if the network will converge to a stable configuration, we see if the energy function reaches its minimum by:

$$\frac{d}{dt} E \leq 0$$

The network is bound to converge if the activity of each neuron wrt time is given by the following differential equation:

$$\frac{d}{dt} u_i = \frac{-u_i}{\tau} + \sum_{j=1}^{n} w_{ij} v_j + \theta_i$$

Brain-State-in-a-Box Network

The Brain-State-in-a-Box BSB neural network is a nonlinear auto-associative neural network and can be extended to hetero-association with two or more layers. It is also similar to Hopfield network. It was proposed by J.A. Anderson, J.W. Silverstein, S.A. Ritz and R.S. Jones in 1977.

Some important points to remember about BSB Network –

- It is a fully connected network with the maximum number of nodes depending upon the dimensionality **n** of the input space.
- All the neurons are updated simultaneously.
- Neurons take values between -1 to +1.

Mathematical Formulations

The node function used in BSB network is a ramp function, which can be defined as follows −

$$f(net) = min(1, max(-1, net))$$

This ramp function is bounded and continuous.

As we know that each node would change its state, it can be done with the help of the following mathematical relation −

$$x_t(t + 1) = f\left(\sum_{j=1}^{n} w_{i,j} x_j(t)\right)$$

Here, $\mathbf{x_i}t$ is the state of the $\mathbf{i_{th}}$ node at time **t**.

Weights from $\mathbf{i_{th}}$ node to $\mathbf{j_{th}}$ node can be measured with the following relation −

$$w_{ij} = \frac{1}{P} \sum_{p=1}^{P} (v_{p,i}\ v_{p,j})$$

Here, **P** is the number of training patterns, which are bipolar.