International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 8 Issue 4 April - 2024, Pages: 25-27

Hopfield Neural Network Algorithm And Its Practical Application.

Norboyev Bexzod

Master's 1st stage student,
Karshi branch of the Tashkent University of Information Technologies
norboyevbexzod98@gmail.com

Abstract: *Neural Networks are computational models that mimic the complex functions of the human brain. The neural networks consist of interconnected nodes or neurons that process and learn from data, enabling tasks such as pattern recognition and decision making in machine learning.*

Keywords: Linear Function,Activation Function,Hidden Layers,Input Layer,Output Layer.

Introduction:

It is a fully interconnected neural network where each unit is connected to every other unit. It behaves in a discrete manner, i.e. it gives finite distinct output, generally of two types:

- Binary (0/1)
- Bipolar (-1/1)

The weights associated with this network are symmetric in nature and have the following properties.

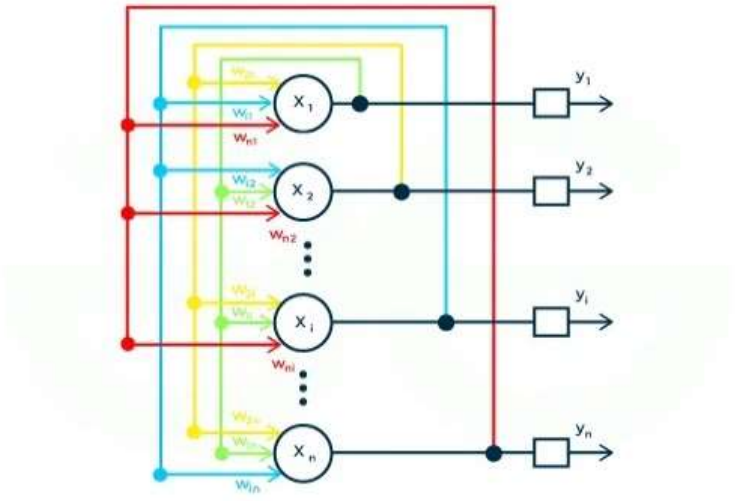
1.w_{ij}=w_{ji}

2. w_{ji}=0

Structure & Architecture of Hopfield Network

- Each neuron has an inverting and a non-inverting output.
- Being fully connected, the output of each neuron is an input to all other neurons but not the self.

The below figure shows a sample representation of a Discrete Hopfield Neural Network architecture having the following elements.



Discrete Hopfield Network Architecture

[x1 , x2 , … , xn] -> Input to the n given neurons.

[y1 , y2 , … , yn] -> Output obtained from the n given neurons

W_{ij} -> weight associated with the connection between the i^{th} and the j^{th} neuron.

Training Algorithm

For storing a set of input patterns S(p) [p = 1 to P], where $S(p) = S_1(p) \dots S_i(p) \dots S_n(p)$, the weight matrix is given by:

- For binary patterns
  $w_{ij}=\sum_{p=1}^{P}[2\,s_i(p)-1][2s_j(p)-1](\,w_{ij} for\ all\ i \neq j)$
- For bipolar patterns
  $w_{ij}=\sum_{p=1}^{P}[\,s_i(p)2s_j(p)](\,where\ w_{ij}=0\ for\ all\ i=j)$

(i.e. weights here have no self-connection)

Steps Involved in the training of a Hopfield Network are as mapped below:

- Initialize weights (w_{ij}) to store patterns (using training algorithm).
- For each input vector y_i, perform steps 3-7.
- Make the initial activators of the network equal to the external input vector x.

$$\boldsymbol{y_i=x_i}\textbf{:(for i=1 to n)}$$

- For each vector y_i, perform steps 5-7.
- Calculate the total input of the network y_{in} using the equation given below.

$$\boldsymbol{y_{in_i}=x_i+\sum_j[\,y_j w_{ji}]}$$

- Apply activation over the total input to calculate the output as per the equation given below:

$$\boldsymbol{y_i=\begin{Bmatrix}1\ if y_{in}>\theta_i\\ y_i\ if\ y_{in}=\theta_i\\ 0\ if\ y_{in}<\theta_i\end{Bmatrix}}$$

(where θ_i (threshold) and is normally taken as 0)

- Now feedback the obtained output y_i to all other units. Thus, the activation vectors are updated.
- Test the network for convergence.

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:

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

and weight matrix with no self-connection is:

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

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)

$$y_{in_i}=x_i+\sum_{j=1}^{4}[y_i\ w_{j1}]=0+[0\ 0\ 1\ 0]\begin{bmatrix}0\\1\\1\\-1\end{bmatrix}=0+1=1$$

Applying activation, $y_{in_1}>0 \Longrightarrow y_1=1$ *giving feedback to other units, we get*

y=[1 0 1 0]

which is not equal to input vector

x=[1 1 1 0]

Conclusion:

In conclusion, the Hopfield neural network algorithm offers a versatile and powerful tool for various computational tasks, including pattern recognition, optimization, and memory retrieval. While it has its limitations, such as susceptibility to spurious states and the requirement for symmetric connections, its simplicity and effectiveness make it a valuable asset in the field of neural networks and computational intelligence.

References

1. Dorigo M., Gambardella L.M. Ant Colony System: A cooperative learning approach to the traveling salesman problem. – IEEE Transactions on Evolutionary Computation. – 1(1):53–66, 1997.

2. Dorigo M., Maniezzo V., Colorni A. Ant System: Optimization by a colony of cooperating agents. – IEEE Transactions on Sys-tems, Man, and Cybernetics – Part B, 26(1):29–41, 1996.

3. Gambardella L.M., Dorigo M. Solving symmetric and asymmet-ric TSPs by ant colonies. – Proc. 96 IEEE Int. Conf. on Evolutionary Computation (ICEC'96). – IEEE Press: New York. – 1996. – P. 622.