



ROHINI

COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE and Affiliated to Anna University (An ISO Certified Institution) | Accredited with A+ Grade by NAAC
Recognized under Section 2(f) of University Grants Commission, UGC ACT 1956
(AUTONOMOUS)

FIXED WEIGHT COMPETITIVE NETS

During training process also the weights remains fixed in these competitive networks. The idea of competition is used among neurons for enhancement of contrast in their activation functions. In this, two networks-

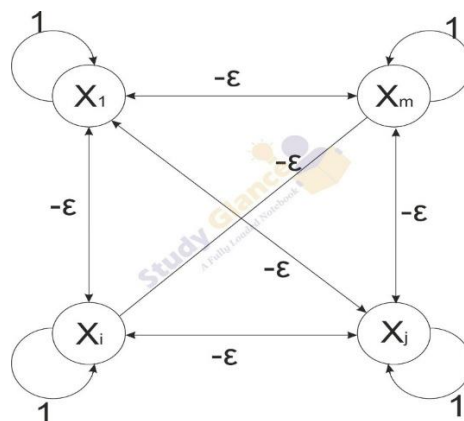
- Maxnet
- Hamming Networks

MAXNET

Maxnet network was developed by Lippmann in 1987. The Maxnet serves as a sub net for picking the node whose input is larger. All the nodes present in this subnet are fully interconnected and there exist symmetrical weights in all these weighted interconnections.

Architecture of Maxnet

The architecture of Maxnet is a fixed symmetrical weights are present over the weighted interconnections. The weights between the neurons are inhibitory and fixed. The Maxnet with this structure can be used as a subnet to select a particular node whose net input is the largest.



Testing Algorithm of Maxnet

The Maxnet uses the following activation function:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Testing algorithm

Step 0: Initial weights and initial activations are set. The weight is set as $[0 < \varepsilon < 1/m]$, where "m" is the total number of nodes. Let

$X_j(0)$ = input the node X_j

and

$$w_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\varepsilon & \text{if } i \neq j \end{cases}$$

Step 1: Perform Steps 2-4, when stopping condition is false.

Step 2: Update the activations of each node. For $j = 1$ to m ,

$$X_j(\text{new}) = F[X_j(\text{old}) - \varepsilon \sum_{i \neq j} X_k(\text{old})]$$

Step 3: Save the activations obtained for use in the next iteration. For $j = 1$ to m ,

$$X_j(\text{new}) = X_j(\text{old})$$

Step 4: Finally, test the stopping condition for convergence of the network. The following is the stopping condition: If more than one node has a nonzero activation, continue; else stop.

HAMMING NETWORK

The Hamming network is a two-layer feedforward neural network for classification of binary bipolar n-tuple input vectors using minimum Hamming distance denoted as D_H (Lippmann, 1987). The first layer is the input layer for the n-tuple input vectors. The second layer (also called the memory layer) stores p memory patterns. A p-class Hamming network has p output neurons in this layer. The strongest response of a neuron is indicative of the minimum Hamming distance between the stored pattern and the input vector.

Hamming Distance

Hamming distance of two vectors, x and y of dimension n

$$x \cdot y = a - d$$

where: a is number of bits in agreement in x & y (No. of Similar bits in x & y), and d is number of bits different in x and y (No. of Dissimilar bits in x & y).

The value "a - d" is the Hamming distance existing between two vectors. Since, the total number of components is n, we have,

$$n = a + d$$

$$\text{i.e., } d = n - a$$

On simplification, we get

$$x \cdot y = a - (n - a)$$

$$x \cdot y = 2a - n$$

$$2a = x \cdot y + n$$

$$a = \frac{1}{2}x \cdot y + \frac{1}{2}n$$

From the above equation, it is clearly understood that the weights can be set to one-half the exemplar vector and bias can be set initially to $n/2$

Testing Algorithm of Hamming Network

Step 0: Initialize the weights. For $i = 1$ to n and $j = 1$ to m ,

$$w_{ij} = \frac{e_i(j)}{2}$$

Initialize the bias for storing the "m" exemplar vectors. For $j = 1$ to m ,

$$b_j = \frac{n}{x}$$

Step 1: Perform Steps 2-4 for each input vector x .

Step 2: Calculate the net input to each unit Y_j , i.e.,

$$y_{inj} = \sum_{i=1}^n x_i w_{ij} + b_j = 1tom$$

Step 3: Initialize the activations for Maxnet, i.e.,

$$y_j(0) = y_{inj} = 1tom$$

Step 4: Maxnet is found to iterate for finding the exemplar that best matches the input patterns.

