**Assignment -4c**

**Title:** To study and implement Hebb’s Rule in MATLAB.

**Theory:**

**Hebb’s Rule:** From the point of view of artificial neurons and artificial neural networks, Hebb's principle can be described as a method of determining how to alter the weights between model neurons. The weight between two neurons increases if the two neurons activate simultaneously—and reduces if they activate separately. Nodes that tend to be either both positive or both negative at the same time have strong positive weights, while those that tend to be opposite have strong negative weights.

* This original principle is perhaps the simplest form of weight selection. The following is a formulaic description of Hebbian learning

\,w_{ij}=x_ix_j

where w_{ij} is the weight of the connection from [neuron](http://en.wikipedia.org/wiki/Artificial_neuron)  j to neuron  i and  x_i the input for neuron  i . Note that this is pattern learning (weights updated after every training example). In a Hopfield network, connections w_{ij} are set to zero if i=j (no reflexive connections allowed). With binary neurons (activations either 0 or 1), connections would be set to 1 if the connected neurons have the same activation for a pattern.

* Another formulaic description is:

w_{ij} = \frac{1}{p} \sum_{k=1}^p x_i^k x_j^k\,

where w_{ij} is the weight of the connection from neuron  j to neuron  i ,  p is the number of training patterns, and x_{i}^kthe  k th input for neuron  i . This is learning by epoch (weights updated after all the training examples are presented). Again, in a [Hopfield network](http://en.wikipedia.org/wiki/Hopfield_network), connections w_{ij} are set to zero if i=j (no reflexive connections).

* If two interconnected neuron are both “on” at the same time, then the weight between those neurons should be increased. However a stronger form of learning occurs if also increases weights, if both neurons are “off” at the same time.

New weights are computed by the formulae:

W(new)=w(old)+delta(w)

delta(w)=X\*Y

X :- Input.

Y:- Target.

b

w1

w2

**Hebb’s Rule Block Diagram**

**Algorithm:**

1. Initialise all weights to 0 and bias to 0 .
2. Compute delta(w) :
3. delta(w)=xi\*y.
4. Update weights by the formulae w(new)=w(old)+delta(w).
5. Repeat the steps 2 to 4 till all inputs are considered.

**Limitations:**

* A disadvantage of the Hebb rule is that if the input vectors are not mutually orthogonal, interference may occur and the network may not be able to learn the associations.
* Hebb’s rule fails for XOR gate problem.

**FAQ’s:**

1. What are the limitations of Hebb’s Law.

2. Define the transfer function used in Adaline.

3. Compare Perceptron and Adaline.

4. Give application of Adaline.

5. Define Pattern Association.