2.7 Hebb Network

2.7.1 Theory

For a neural net, the Hebb learning rule is a simple one. Let us understand it. Donald Hebb stated in 1949 that in the brain, the learning is performed by the change in the synaptic gap. Hebb explained it: "When an axon of cell A is near enough to excite cell B, and repeatedly or permanently takes place in firing it, some growth process or metabolic change takes place in one or both the cells such that A's efficiency, as one of the cells firing B, is increased."

According to the Hebb rule, the weight vector is found to increase proportionately to the product of the input and the learning signal. Here the learning signal is equal to the neuron's output. In Hebb learning, if two interconnected neurons are 'on' simultaneously then the weights associated with these neurons can be increased by the modification made in their synaptic gap (strength). The weight update in Hebb rule is given by

$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

The Hebb rule is more suited for bipolar data than binary data. If binary data is used, the above weight updation formula cannot distinguish two conditions namely:

1. A training pair in which an input unit is "on" and target value is "'off."
2. A training pair in which both the input unit and the target value are "off."

Thus, there are limitations in Hebb rule application over binary data. Hence, the representation using bipolar data is advantageous.

2.7.2 Flowchart of Training Algorithm

The training algorithm is used for the calculation and adjustment of weights. The flowchart for the training algorithm of Hebb network is given in Figure 2-21. The notations used in the flowchart have already been discussed in Section 2.4.7.

In Figure 2-21, $s:t$ refers to each training input and target output pair. Till there exists a pair of training input and target output, the training process takes place; else, it is stopped.

2.7.3 Training Algorithm

The training algorithm of Hebb network is given below:

Step 0: First initialize the weights. Basically in this network they may be set to zero, i.e., $w_i = 0$ for $i = 1$ to n where "n" may be the total number of input neurons.

Step 1: Steps 2–4 have to be performed for each input training vector and target output pair, $s:t$.

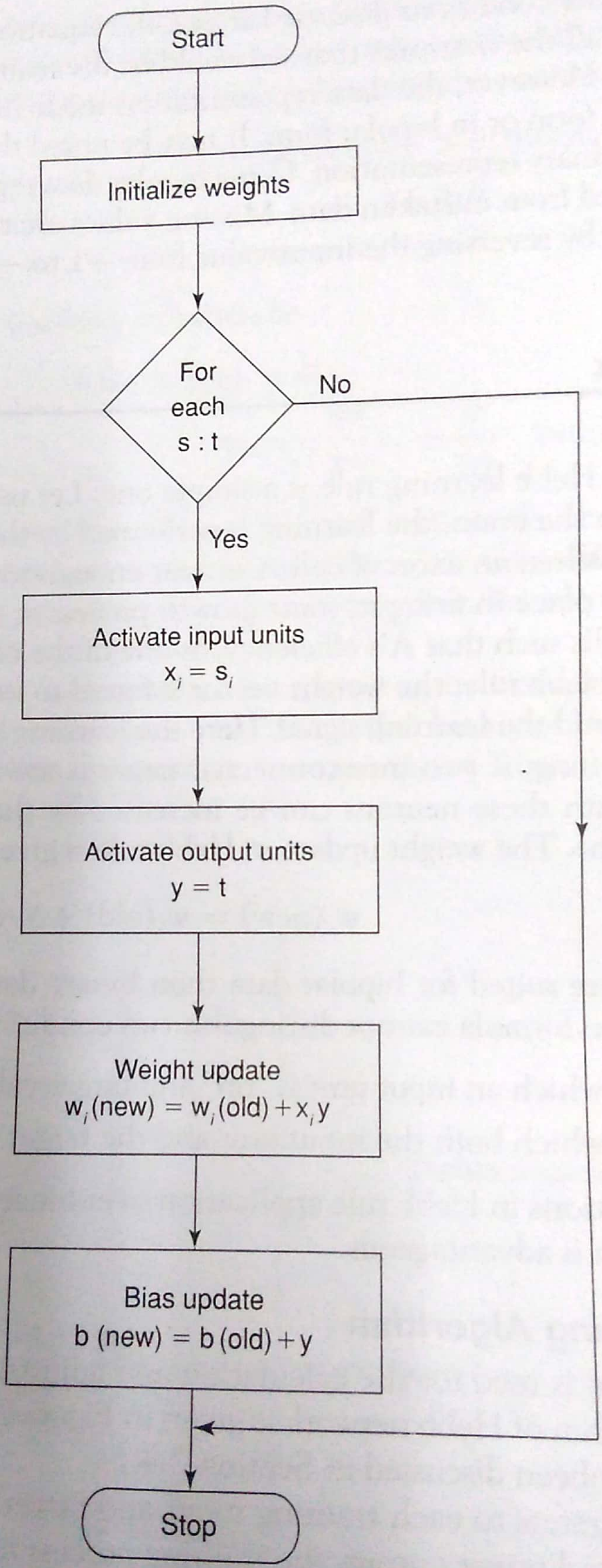

Figure 2-21 Flowchart of Hebb training algorithm.

Step 2: Input units activations are set. Generally the activation function of input layer is identity function.

$$x_i = s_i \quad \text{for } i = 1 \text{ to } n$$

Step 3: Output units activations are set.

$$y = t$$

Step 4: Weight adjustments and bias adjustments are performed:

$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$
$$b(\text{new}) = b(\text{old}) + y$$

The above five steps complete the algorithmic process. In Step 4, the weight updation formula can also be given in vector form as

$$w(\text{new}) = w(\text{old}) + xy$$

Here the change in weight can be expressed as

$$\Delta w = xy$$

As a result,

$$w(\text{new}) = w(\text{old}) + \Delta w$$

The Hebb rule can be used for pattern association, pattern categorization, pattern classification and over a range of other areas.

2.9 Solved Problems

1. For the network shown in Figure 1, calculate the net input to the output neuron.

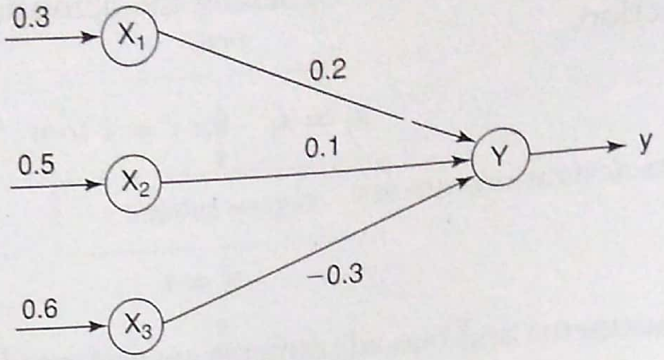

Figure 1 Neural net.

Solution: The given neural net consists of three input neurons and one output neuron. The inputs and weights are

$$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$$
$$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$$

The net input can be calculated as

$$\begin{aligned} y_{in} &= x_1 w_1 + x_2 w_2 + x_3 w_3 \\ &= 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times (-0.3) \\ &= 0.06 + 0.05 - 0.18 \\ y_{in} &= -0.07 \end{aligned}$$

2. Calculate the net input for the network shown in Figure 2 with bias included in the network.

Solution: The given net consists of two input neurons, a bias and an output neuron. The inputs are $[x_1, x_2] = [0.2, 0.6]$ and the weights are $[w_1, w_2] = [0.3, 0.7]$.

Since the bias is included $b = 0.45$ and bias input x_0 is equal to 1.

The net input is calculated as

$$\begin{aligned} y_{in} &= b + x_1 w_1 + x_2 w_2 \\ &= 0.45 + 0.2 \times 0.3 + 0.6 \times 0.7 \\ &= 0.45 + 0.06 + 0.42 \\ &= 0.93 \end{aligned}$$

Therefore $y_{in} = 0.93$ is the net input.

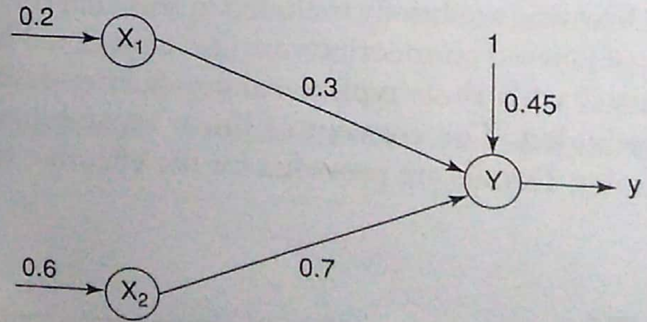

Figure 2 Simple neural net.

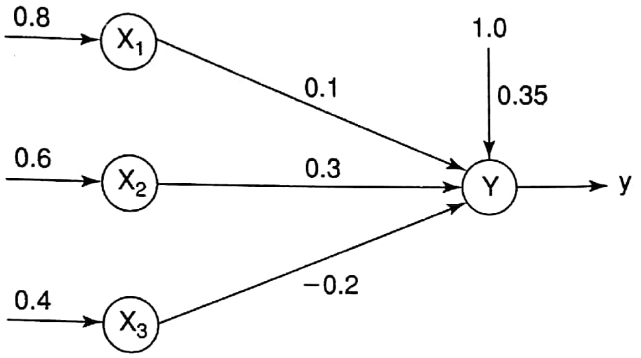

Figure 3 Neural net.

3. Obtain the output of the neuron Y for the network shown in Figure 3 using activation functions as:
   - binary sigmoidal;
   - bipolar sigmoidal.

Solution: The given network has three input neurons with bias and one output neuron. These form a single-layer network.

The inputs are given as $[x_1, x_2, x_3] = [0.8, 0.6, 0.4]$ and the weights are $[w_1, w_2, w_3] = [0.1, 0.3, -0.2]$ with bias $b = 0.35$ (its input is always 1).

The net input to the output neuron is

$$\begin{aligned}
y_{in} &= b + \sum_{i=1}^{n} x_i w_i \qquad [n = 3,\ \text{because only 3 input neurons are given}] \\
&= b + x_1 w_1 + x_2 w_2 + x_3 w_3 \\
&= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times (-0.2) \\
&= 0.35 + 0.08 + 0.18 - 0.08 \\
y_{in} &= 0.53
\end{aligned}$$

- For binary sigmoidal activation function,

$$y = f(y_{in}) = \frac{1}{1 + e^{-y_{in}}} = \frac{1}{1 + e^{-0.53}} = 0.625$$

- For bipolar sigmoidal activation function,

$$y = f(y_{in}) = \frac{2}{1 + e^{-y_{in}}} - 1 = \frac{2}{1 + e^{-0.53}} - 1 = 0.259$$