McCulloch and Pitts Neuron

Theory

The McCulloch-Pitts neuron was the earliest neural network discovered in 1943. It is usually called as M-P neuron.

The M-P neurons are connected by directed weighted paths. It should be noted that the activation of a M-P neuron is binary, that is, at any time step the neuron may fire or may not fire.

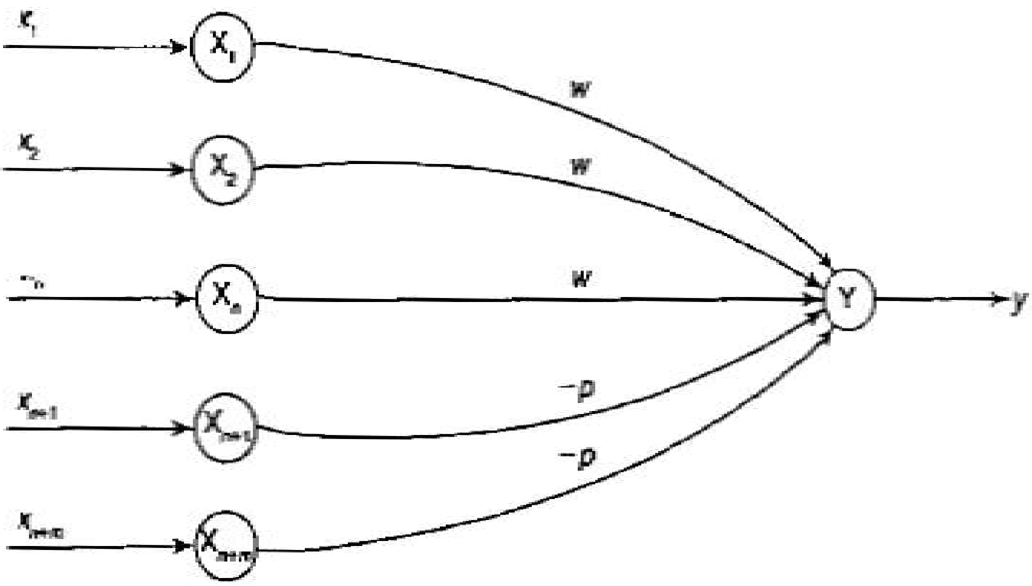
The weights associated with the communication links may be excitatory (weight is positive) or inhibitory (weight is negative). All the excitatory connected weights entering into a particular neuron will have same weights.

Threshold

There is a fixed threshold for each neuron, and if the net input to the neuron is greater than the threshold then the neuron fires. Also, any nonzero inhibitory input would prevent the neuron from firing. The M-P neurons are most widely used in the case of logic functions.

Architecture

A simple M-P neuron is shown in Figure below.



The M-P neuron has both excitatory and inhibitory connections. It is excitatory with weight *(w* > 0) or inhibitory with weight *-p(p* < 0).

In Figure, inputs from x_1 ro x_n possess excitatory weighted connections and inputs from x_{n+1} to x_{n+m} possess inhibitory weighted interconnections. Since the firing of the output neuron is based upon the threshold, the activation function here is defined as

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geqslant \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

For inhibition to be absolute, the threshold with the activation function should satisfy the following condition:

$$\theta > nw - p$$

The output will fire if it receives say k or more excitatory inputs but no inhibitory inputs, where

$$kw \geq \theta > (k-1)w$$

The M-P neuron has no particular training algorithm. An analysis has to be performed to determine the values of the weights and the threshold. Here the weights of the neuron are set along with the threshold to make the neuron "perform a simple logic function.

Hebb network

Theory

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

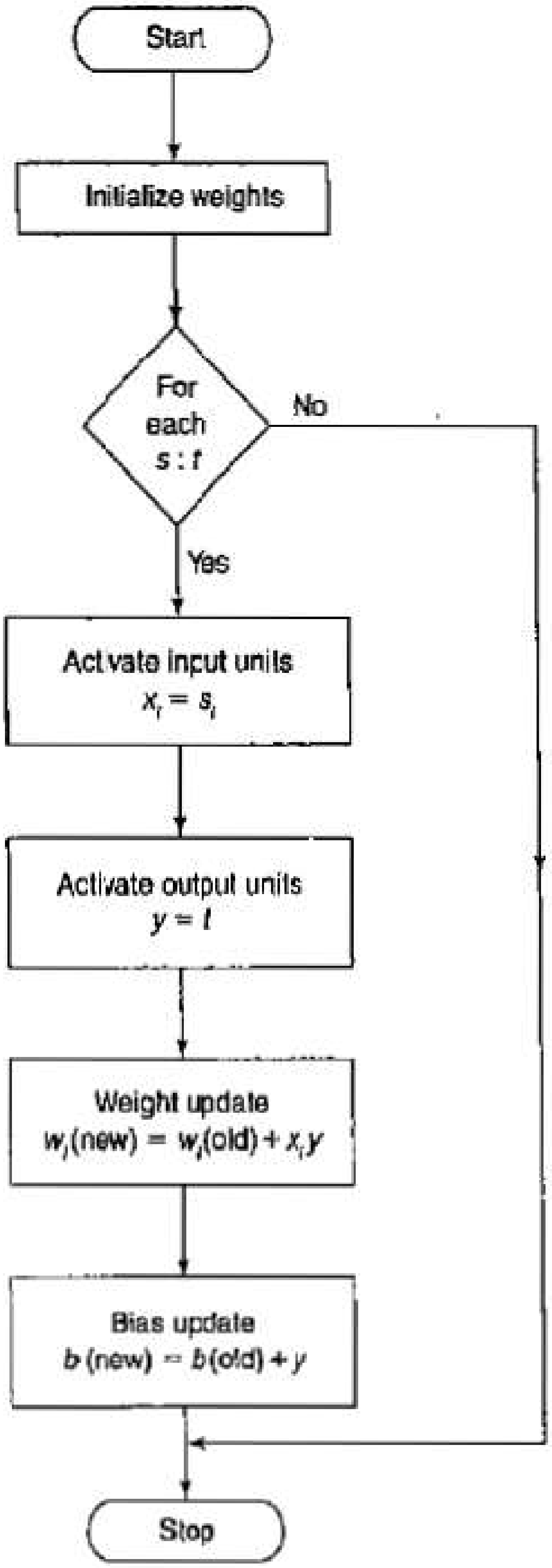
$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

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."

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 below:



s: *t* refers to each training input and target output pair.

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: r.

Step 2: Input units activations are set. Generally, the activation function of input layer is identity function: $x_i=s_i$; for i=1 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

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

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.