1. The Categories of Neural Network Learning Rules

There are many types of Neural Network Learning Rules, they fall into two broad categories: supervised learning, and unsupervised learning. Block diagrams of the learning types are illustrated in Figures (1) and Figure(2).

1.1 Supervised Learning

The learning rule is provided with a set of examples (the training set) of proper network behavior:

$$\{x_1, d_1\} , \{x_2, d_2\} , \ldots, \{x_n, d_n\} \quad \ldots (1)$$

where (x_n) is an input to the network, (d_n) is the corresponding correct target (desired) output. As the inputs are applied to the network, the network outputs are compared with the targets. The learning rule is then used to adjust the weights and the biases of the network in order to move the network outputs closer to the targets (desired).

In supervised learning we assume that at each instant of time when the input is applied, the desired response (d) of the system is provided by the teacher. This is illustrated in figure (1), the distance between the actual and the desired response serves as an error measure and is used to correct network parameter externally.

For instance, in learning classifications of input patterns or situations with known responses, the error can be used to modify the weights so that the error decreases. This mode of learning is very pervasive. Also it is used in many situations of natural learning. A set of input and output patterns called training set is required for this learning mode .

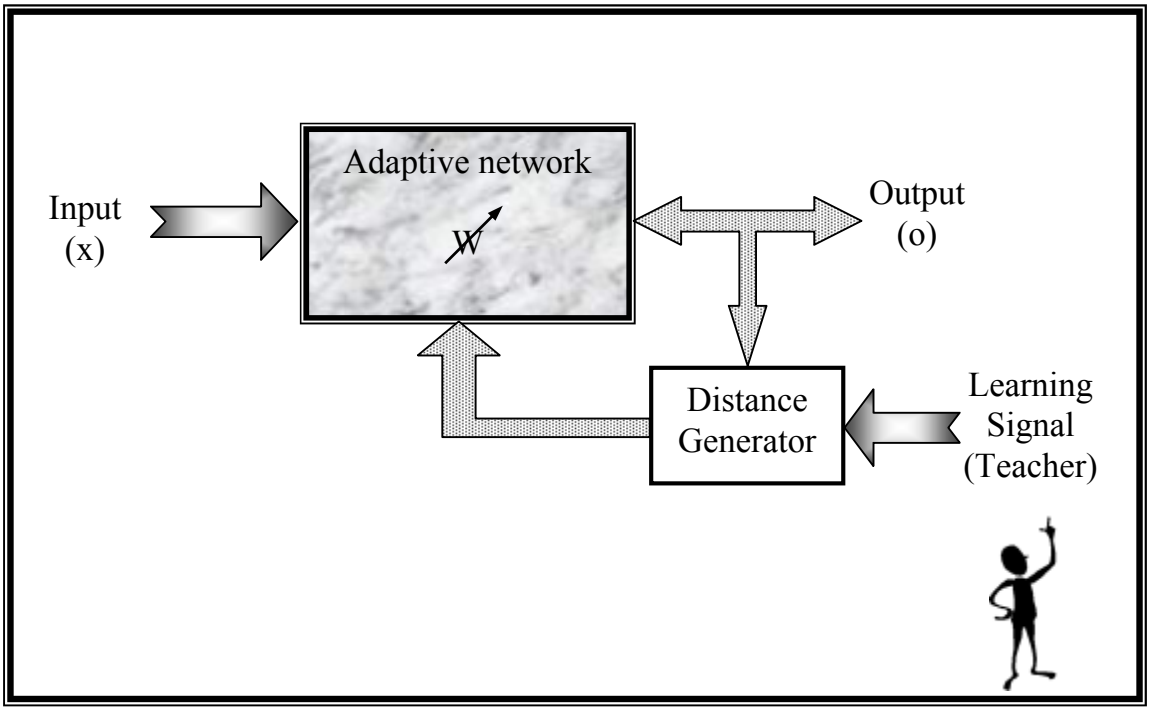

Figure (1) Block diagram for explaining of supervised learning.

1.2 Unsupervised Learning

In unsupervised learning, the weights and biases are modified in response to network input only. There are no target outputs available. At first glance this might seem to be impractical. How can you train a network if you don't know what is supposed to do? Most of these algorithms perform some kind of clustering operation. They learn to categorize the input patterns into a finite number of classes. This is useful in such applications such as vector quantization .

Figure (2) shows the block diagram of unsupervised learning rule. In learning without supervision the desired response is not known; thus, explicit error information cannot be used to improve network behavior. Since no information is available as to correctness or incorrectness of responses, learning must somehow be accomplished based on observations of responses to inputs that we have marginal or no knowledge about.

Unsupervised learning algorithms use patterns that are typically redundant raw data having no label regarding their class membership, or

associations. In this mode of learning, a network must discover for itself any possibly existing patterns, regularities, separating properties, etc., while discovering these the network undergoes change in its parameters, unsupervised learning is sometimes called learning without teacher. This terminology is not the most appropriate because learning without a teacher is not possible at all. Although, the teacher does not have to be involved in every training step, he has to set goals even in an unsupervised learning mode.

Learning with feedback, either from the teacher or from environment, however, is more typical for neural network. Such learning is called incremental and is usually performed in steps. The concept of feedback plays a central role in learning.

The concept is highly elusive and somewhat paradoxical. In a broad sense it can be understood as an introduction of a pattern of relationships into the cause-and-effect path.

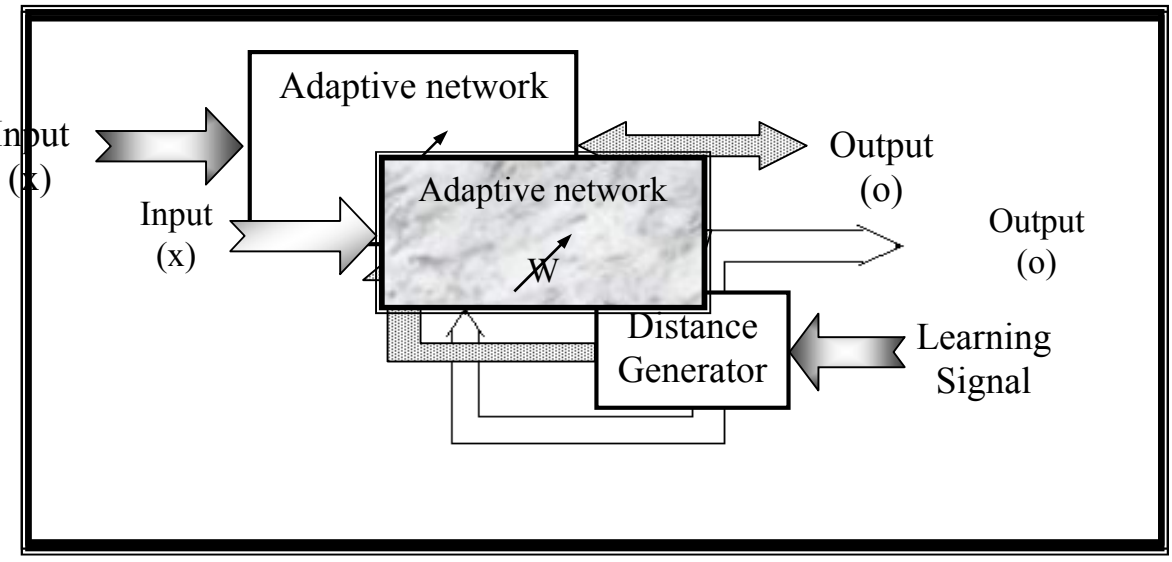

Figure (2) Block diagram for explaining of unsupervised learning.

2. Neural Network Learning Rules

Our focus in this section will be on artificial neural network learning rules. A neuron is considered to be an adaptive element. Its weights are modifiable depending on the input signal it receives, its output value, and the associated teacher response. In some cases the teacher signal is not available and no error information can be used, thus a neuron will modify its weights, based only on the input and / or output. This is the case for unsupervised learning.

The trained network is show in Figure (3), It study the weight vector (w_i) or its component (w_{ij}), which connecting the (j'th) input with neuron (i). The output of another neurons can be the (j'th) input to the neuron (i). Our discussion in this section will cover single neuron and single layer network supervised learning and simple cases of unsupervised learning. The form of neuron activation function may be different when different learning rules is considered.

The learning ability of human beings is properly incorporated in the facility of the changing the transmission efficiency of the synapses which corresponding to adaptation of the weight. The convergence has always been a major problem in the neural network learning algorithms. In most cases, to avoid this, impractical applications different initial conditions are used until one case would converge to the desirable target.

The weight vector $w_i=[w_{i1}\ w_{i2}\ w_{i3}\ \ldots.\ w_{in}]^t$ increases in proportion to the product of input (x) and learning signal (r). The learning signal (r) is, in general, a function of (w_i,x), and sometimes of the teacher's signal (d_i). So for the network shown in Figure (3.3): -

$$r = r(w_i,x,d_i) \quad \ldots (2)$$

The increment of the weight vector (w_i) product by the learning step at time (t) according to the general learning rule is given by

$$\Delta w_i(t) = c\ r[w_i(t), x(t), d_i(t)]\ x(t) \quad \ldots (3)$$

where (c) is constant called the learning constant that determines the rate of learning. At the next instant learning step, the weight vector adapt at time (t) becomes:

$$w_i(t+1) = w_i(t) + c\, r[w_i(t), x(t), d_i(t)]\, x(t) \quad \ldots (4)$$

The superscript convention will be used in this context to index the discrete – time training steps as in equation (3.4). For the (k'th) step it can be had from equation (3.4) using this convention:

$$w_i^{k+1} = w_i^k + cr\,(w_i^k, x^k, d_i^k)\, x^k \quad \ldots (5)$$

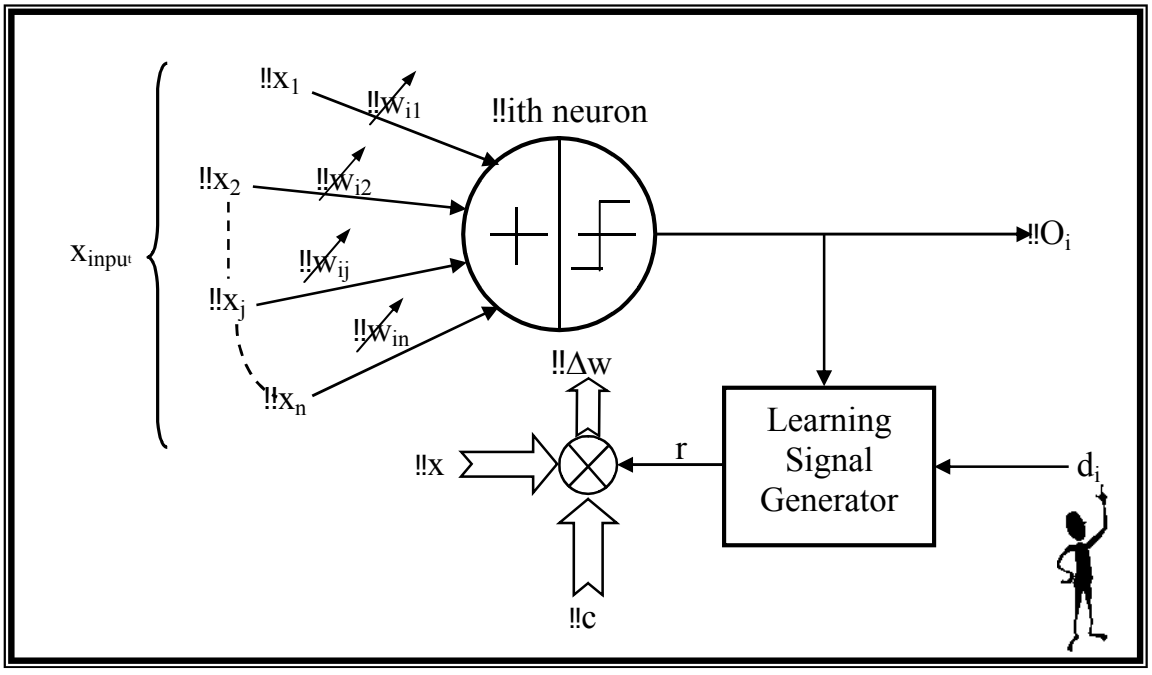

Figure (3) Illustration of weight learning values
(d_i) provided only for supervised learning mode)

3 Hebbian Learning Rule

For the Hebbian learning rule the learning is equal simply to the neuron's output as shown in Figure (3.4), so it can be seen that:

$$r = f(w_i^t x) \quad \ldots (6)$$

The increment (Δw_i)of the weight vector becomes:

$$\Delta w_i = c\, f(w_i^t x) x \quad \ldots (7)$$

The single weight (w_i) is adapted using the following increment:

$$\Delta w_{ij} = c\, f(w_i^t x) x_j \quad \ldots (8)$$

This can be written briefly as:

$$\Delta w_{ij} = co_i x_j \quad , \quad \text{For j= 1, 2, 3, …, n.} \quad \ldots (9)$$

This learning rule requires the weight initialization at small random values around (w_{ij}=0) prior to learning. The Hebbian learning rule represents a purely unsupervised learning. The rule implements the interpretation of the classic statement; "when an axon of cell (a) is near enough to the exit of a cell (b) and repeatedly or persistently takes place in firing it, some growth process or metabolic change takes place in one or both cells such that cell (a) efficiency, as one of the cells firing cell (b), is increased".

The rule states that if the cross product of output and input, or correlation term ($o_i x_j$) is positive, this results in an increase of weight w_{ij}; otherwise the weight decreases. It can be seen that the output is strengthened in turn for each input presented. Therefore, frequent input patterns will have most influence at the neurons weight vectors and will eventually produce the largest output.

A persistence worry with computational model of unsupervised learning is that learning will become more difficult as problem is scaled. The Hebbian rule has evolved in a number of directions, in some cases, the Hebbian rule needs to be modified to counteract unconstrained growth of weight values, which takes place when excitations and responses consistently agree in sign. This corresponds to Hebbian learning rule with saturation of the weights at certain, preset level.

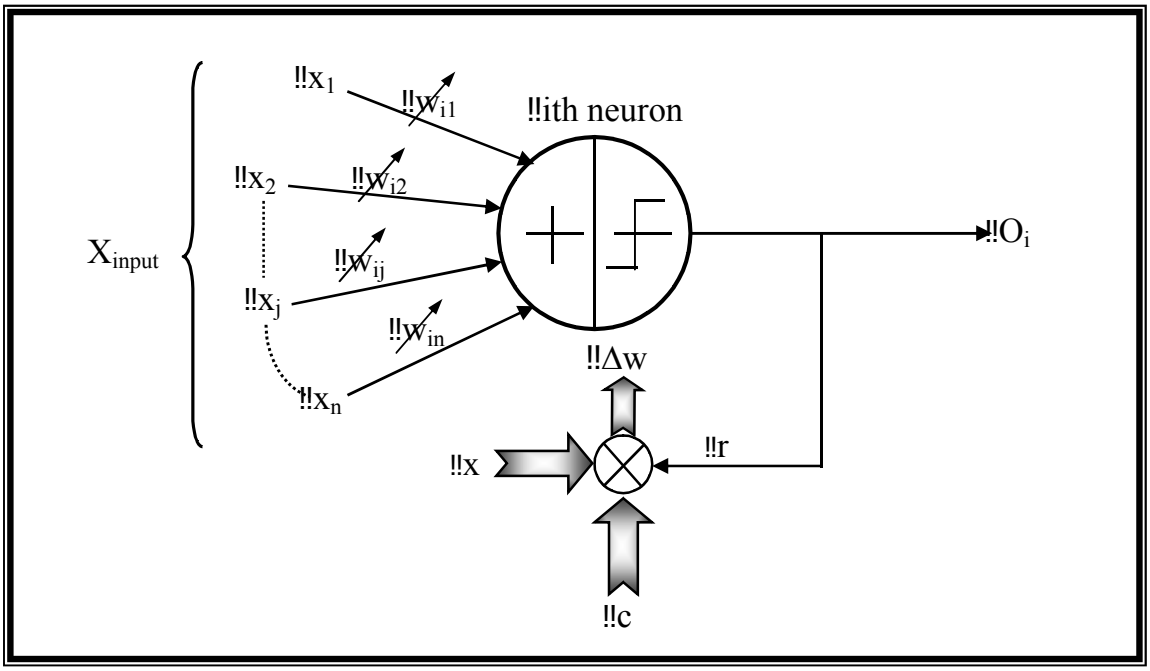

Figure (4) The Hebbian learning rule structure.

It can be seen from Figure (4) that:

$$w_i^1 = w_i^0 + \Delta w_i^0 \quad \ldots (10)$$

$$w_i^2 = w_i^1 + \Delta w_i^1 \quad \ldots (11)$$

$$\vdots$$

$$w_i^k = w_i^{k-1} + \Delta w_i^{k-1} \quad \ldots (12)$$

where (k) is the number of steps that the Hebbian learning rule need to learn the input signals.