

**B. E.**

Seventh Semester Examination, Dec.-2006

**NEURAL NETWORKS**

**Note :** Attempt any five questions.

**Q. 1. What is competitive learning? How is it different from conventional learning? Discuss its significance and usefulness in neural networks.**

**Ans. Competitive learning :** Another modification of the winner-take all learning rule is that both the winners and losers weights are adjusted in proportion to their level of response. This may be called pleaky competitive learning and should provide more subtle learning in the cases for which clusters may be hard to distinguish.

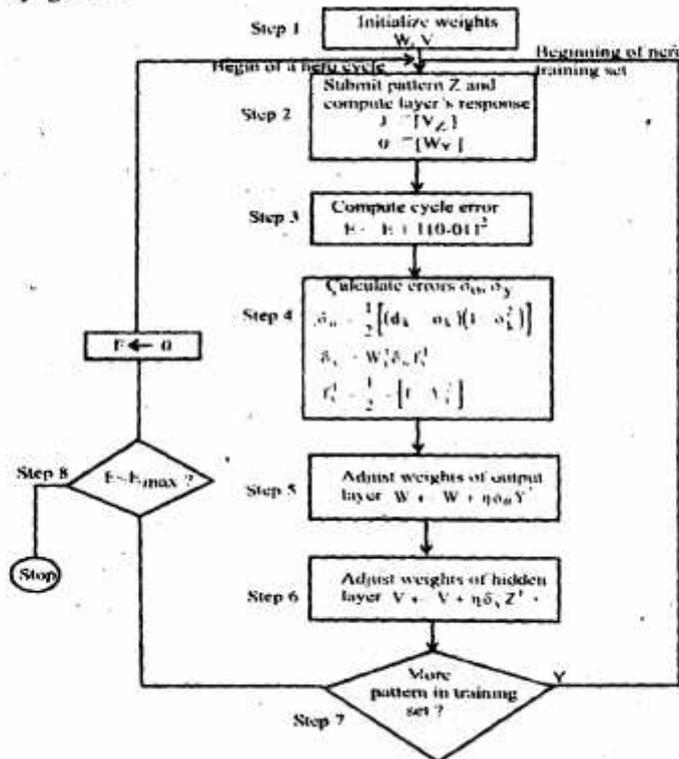
**Winner-Take-all learning rule :** The rule differs substantially from any of the rules discussed so far in the section. It can only be demonstrated and explained for an assemble of neurons, preferably arranged in a layer of  $p$  units. This rule is an example of competitive learning, and it is used for unsupervised network training. Typically winner-take-all learning is used for learning statistical properties of inputs. The binary is based on the premise that one of neurons in this layer.

**Q. 2. Explain the following :**

**(a) Back propagation**

**(b) Perception training rule.**

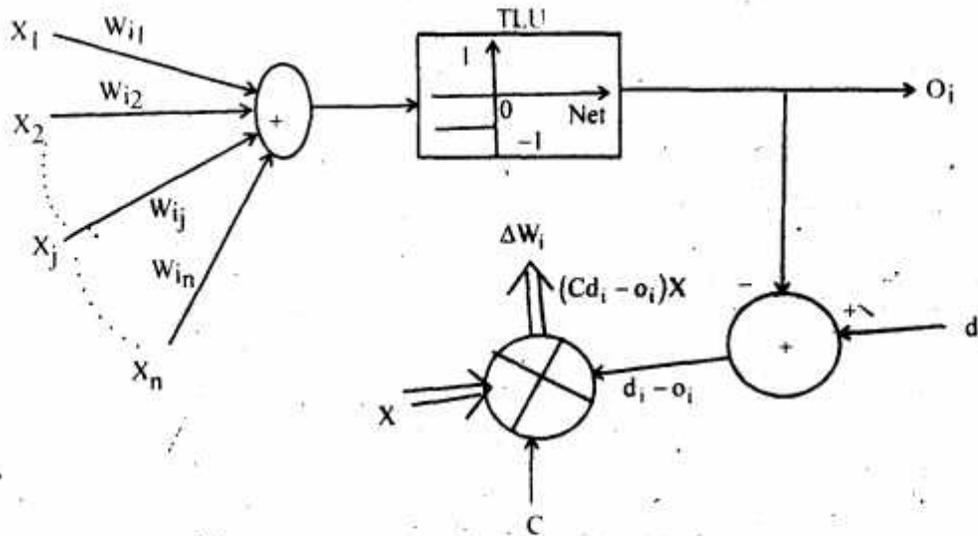
**Ans. (a) Back Propagation :**



(b) **Perception learning rule** : For perception learning rule, the learning signal is the difference between the desired and actual neurons' response. Thus, learning is supervised and learning signal is equal to

$$r \Delta d_i - O_i$$

Where  $O_i = \text{sgn}(w_c^t x)$  and  $d_i$  is the desired response as shown in figure.



Weight adjustment is achieved as follows :

$$\Delta w_i = c [d_i - \text{sgn}(w_i^t + x)] x$$

$$\Delta w_{ij} = c [d_i - \text{sgn}(w_i^t x)] x_j \text{ for } j = 1, 2, \dots, n.$$

This rule is applicable for binary neuron response.

**Q. 3. What are biological neurons? How they help in creating artificial neuron models? Also discuss significance of such models.**

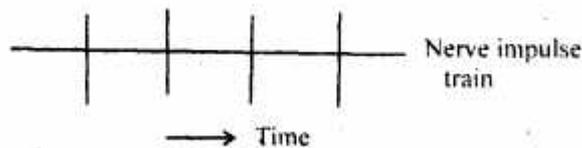
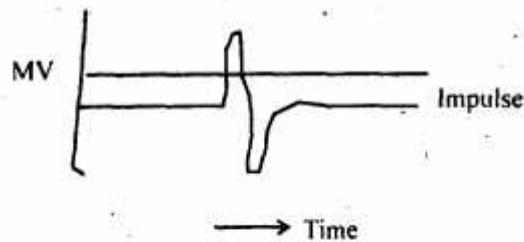
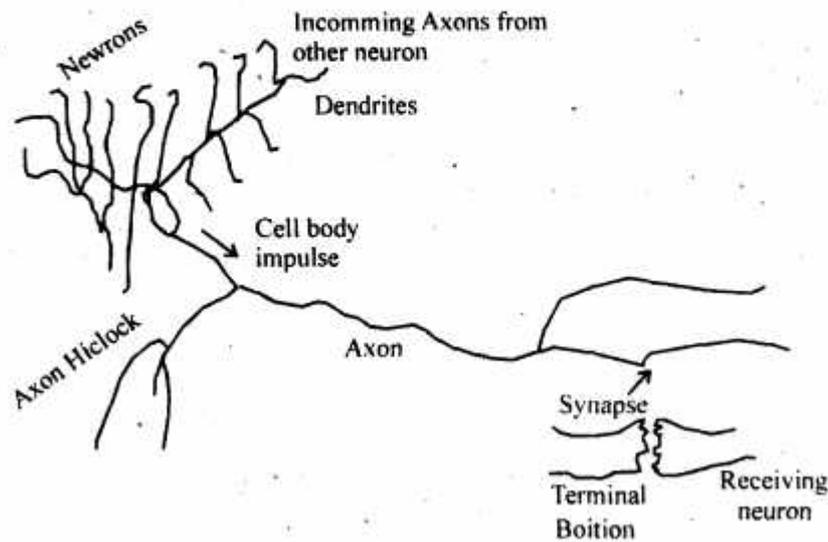
**Ans. Biological neurons :**

The elementary nerve cell called a neuron is the fundamental building block of biological neural network.

**A typical cell has 3 major regions :**

The cell body, which is called soma, the axon and dendrites. Dendrites form dendritic tree, which is very

fine bush of thin fibers around the neuron's body. Dendrites receive informations from neurons through axons-long fibers that serves as transmission lines. An axon is a long cylindrical connection that carries impulses from neuron. The end part of an axon split into fine aborization. The axon dendrite contact organ is called a synapse.

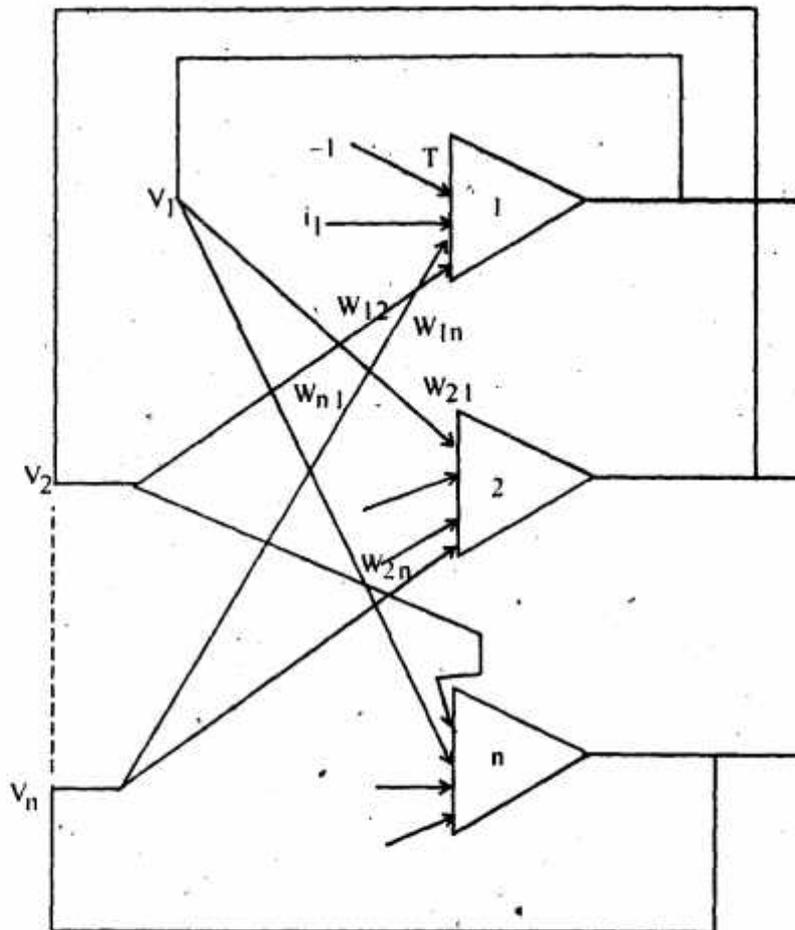


Let us consider the conditions necessary for firing of a neuron. Incoming impulses can be excitatory if they cause the firing or inhibitory. If they hinder the firing of the response. A more precise conditions for firing is that the excitation should exceed the inhibition by the amount called the threshold of neuron. Since a synaptic connection cause the excitatory or inshibitory reactions of receiving neurons, it is practical to assign +ve and -ve unity weight values to such connections. This allows us to reformulate the neurons' firing

conditions. The neuron fires when the total weights to receive impulses exceeds the threshold value during the latent summations period.

**Q. 4. What do you understand by Hop field networks? Discuss their significance and usefulness in neural networks.**

**Ans. Hopfield N/ws :** The postulates of hopfield, the single-layer feedback NN is shown. It consists of  $n$  neurone having threshold values  $T_i$ . The feedback input to the  $i$ th neuron is equal to the weighted sum of neuron outputs  $V_j$ , where  $j = 1, 2, \dots, n$ .



Denoting  $w_{ij}$  as the weight value connecting the output of  $j$ th neuron with the input of  $i$ th neuron, we express the total input

$$\text{net}_i = \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} V_j + i_i - T_i \text{ for } i = 1, 2, \dots, n$$

Introducing vector notation for synaptic weights and neuron output can be written as,

$$\text{net}_i = w_i^T v + i_i - T_i \text{ for } i = 1, 2, \dots, n$$

Where

$$w_i \Delta \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{bmatrix}$$

It is the weight vector containing weights connected to the input of  $i^{\text{th}}$  neuron and

$$v \Delta \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

It is NN output vector called output vector. The complete matrix descriptions of linear portion is shown in figure is given by

$$\text{net} = w v + i - t$$

Where

$$\text{net} \Delta \begin{bmatrix} \text{net}_1 \\ \text{net}_2 \\ \vdots \\ \text{net}_n \end{bmatrix}, i \Delta \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix}$$

Are vectors containing activations and external inputs to each neuron respect. The threshold vector it has been defined here as,

$$t \Delta \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{bmatrix}$$

Matrix  $w$ , sometimes called connectivity matrix is an  $n \times n$  matrix containing network weights arranged in row of vectors  $w_i^t$  as defined and it is equal to,

$$w = \begin{bmatrix} w_1^t \\ w_2^t \\ \vdots \\ w_n^t \end{bmatrix}$$

Matrix will be

$$w = \begin{bmatrix} 0 & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & 0 & w_{23} & \dots & w_{2n} \\ w_{31} & w_{32} & 0 & \dots & w_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & w_{n3} & 0 \end{bmatrix}$$

**Q. 5. Explain the following :**

**(a) Multilayer nets**

**(b) Associative memories**

**Ans. Multilayer nets :**

Multilayer networks are often called layered network. They can implement arbitrary complex input/output mappings or decisions surfaces separating pattern classes. The important attribute of a multilayer feed forward network is that it can learn a mapping of any complexity. The network learning is based on repeated presentations of training samples as has been the case for single layer networks. The trained network often produces surprising results and generalizations in applications where explicit derivation of mappings and discovery of relationships is impossible. Assume two training sets  $y_1$  and  $y_2$  of argumented patterns are available for training. If no weights vector exists such that,

$$y^t w > 0 \text{ for each } y \in y_1 \text{ and}$$

$$y^t w < 0 \text{ for each } y \in y_2$$

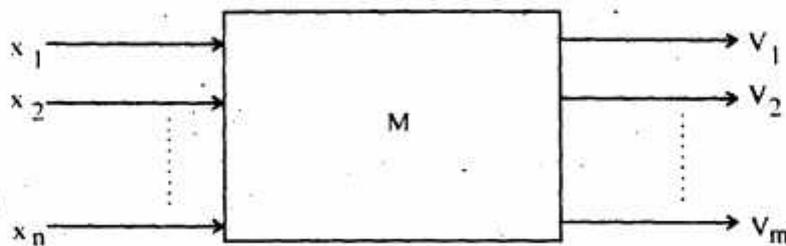
Then the pattern set  $y_1$  and  $y_2$  are linearly & non-separable.

**(b) Associative memories :**

Associative memory can store a large set of patterns as memory. During recall, the memory is excited with a key pattern containing a portion of information about a particular member of a stored pattern set. This particular stored prototype can be recalled through association of key pattern and the information memorized.

Associative memory belong to a class of NN that learns according to a certain recording algorithm. They usually acquire information a prior, and their connectivity matrices most of Fen need to be formed in advance.

- Writing into memory produces changes in neural interconnection. Reading of stored information from memory can be considered as the transformation of the input signals by the networks.



Associative memory usually enables a parallel search with is a stored data file. The purpose of search is to output either one or all stored items that match the given search arguments and to retrain it either entirely or partially. It is also believed that biological memory operates according to associative memory principle.

**Q. 6. What are neural networks? Discuss the applications of neural networks in vision and speech.**

**Ans. Neural networks :** Its ability to perform computations is based on the hope that we can reproduce some of the flexibility and power of the human brain by artificial means.

Network computation is performed by a dense mesh of computing nodes and connection. They operate collectively and simultaneously on most or all data & inputs. The basic processing elements of neural networks are called Artificial neurons or simply neurons after the simply call them nodes. Neurons perform as summing and nonlinear mapping functions. In some cases they can be considered as threshold units that fire when their total input exceeds certain bias levels. Neurons usually operate in parallel and are configured in signal architectures. They are often organized in layers and feedback connections both within the layer and towards adjacent layers are allowed. Each connection strength is expressed by a numerical value called a weight, which can be modified.

The effort is not only called neural network, but also noncomplying, network computation, connections, parallel distributed processing layered adaptive system, self organising network or memomorphic systems or network.

**Q. 7. Explain the following :**

(a) Linear separability

(b) Delta rule.

**Ans. (a) Linear separability :**

Here we divide the pattern set  $x$ . This set is divided into subsets  $\Pi_1, \Pi_2, \dots, \Pi_R$ , respectively. If a linear m/c can classify the pattern from  $\Pi_i$  as belong to class  $i$  for  $i = 1, 2, \dots, R$ , then the pattern sets are linearly separate. Using this perfectly we have :

$$g_i(x) \geq g_j(x) \text{ for all } x \in I_i$$

$$i = 1, 2, \dots, R$$

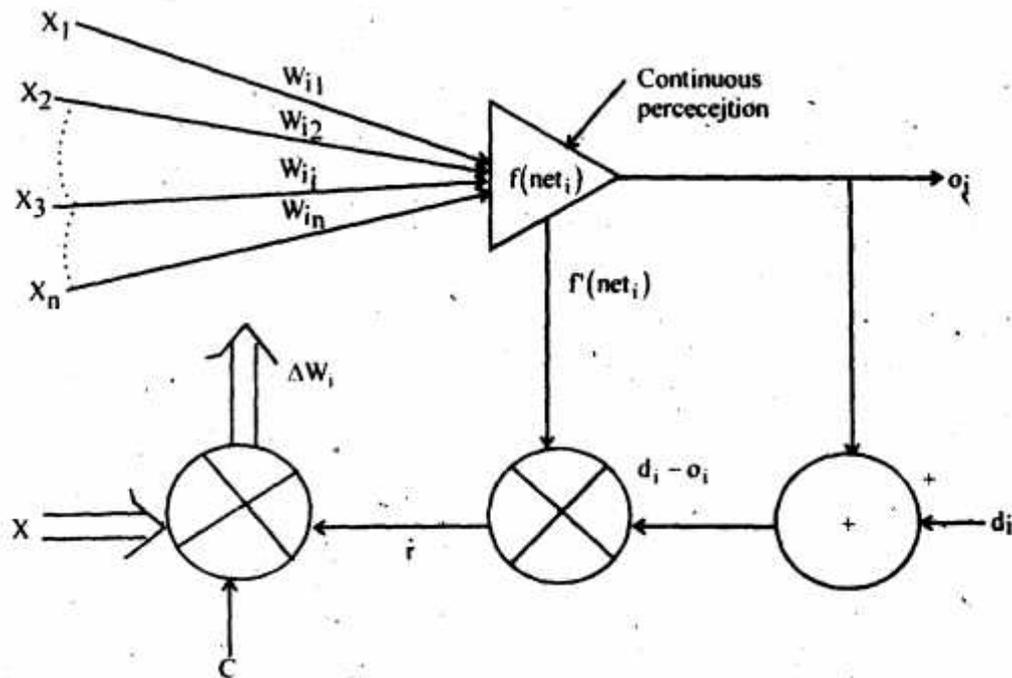
$$j = 1, 2, \dots, R$$

$$i \neq j.$$

(b) **Delta rule** : Delta rule is only valid for continuous activation function and it supervised in bring modes. The leaving for this rule called delta rule, it defined as :

$$\Delta w_i = [d_i - f(w_i^T x)] f'(w_i^T x)$$

Where from  $f'(w_i^T x)$  is derivate of  $f(\text{net}_i) = w_i^T x$ . The explanation of delta rule is as shown as below figure.



The learning rule can be readily derived from the condition of least squared error between  $0_i$  and  $d_i$  calculating the gradient vector with respect to  $w_i$  of the squared error defined as :

$$E = \frac{1}{2} (d_i - o_i)^2$$

**Q. 8. What do you mean by Kohonen self-organizing feature maps? How are these useful? Illustrate through suitable examples.**

**Ans. Kohonen self-organizing features :** The network to be trained is called Kohonen network. The processing of input data  $X$  from the training set  $\{X, X_1, \dots, X_N\}$  which relation  $P$  clusters, follows the customary expression.

$$y = T(Wx)$$

With diagonal elements of the operator  $T$  being continuous activation function operating componentwise on this of vector  $w_x$ . The processing by the layer of neurons is instantaneous and feed forward. To analyze network performance, we average the matrix  $W$  to the following form :

$$W = \begin{bmatrix} w_1^t \\ w_2^t \\ \vdots \\ w_p^t \end{bmatrix}$$

Where

$$w_i = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{bmatrix} \text{ For } i = 1, 2, \dots, P.$$

It is the column vector equal to  $i$ th row of weight matrix  $w$ . Component weights of  $w_x$  are highlighted in fig below as shown, it shows a winner-take-all bearing network.

The learning algorithm treats a set of  $P$  weight vectors as variable vectors that need to be learned prior to the learning, the normalization of all weight vectors is required,

$$\hat{w}_i = \frac{w_i}{\|w_i\|} \text{ for } i = 1, 2, \dots, P$$

