Artificial Neural Structures Classification

Outline:

1. **Artificial Neuron Classification**
2. **Artificial Neuron Network characteristics**
3. **Feed forward Network**
4. **Feedback Network**

1. Artificial Neuron Classification

Artificial neural networks can be classified based on the basic of :

1. Pattern of connections between neurons (architecture of network)
2. Activation function
3. Method of determining weights on the connection

1.1 Architecture of Neural Network

The neurons assumed that arranged as a layer:

a) Input layer
b) Hidden layer
c) Output Layer

The neural nets are classified into Single layer networks and multilayer networks

- **Single Layer Networks**

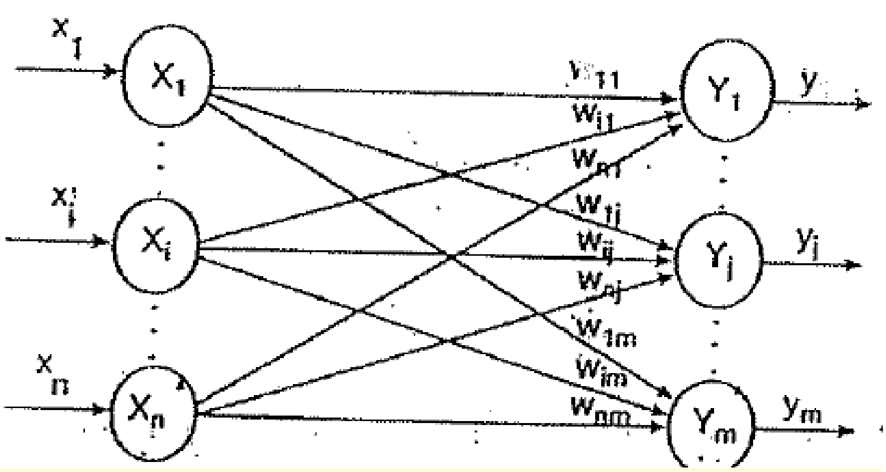

Figure1: Single layer neural network

- Multilayer Networks

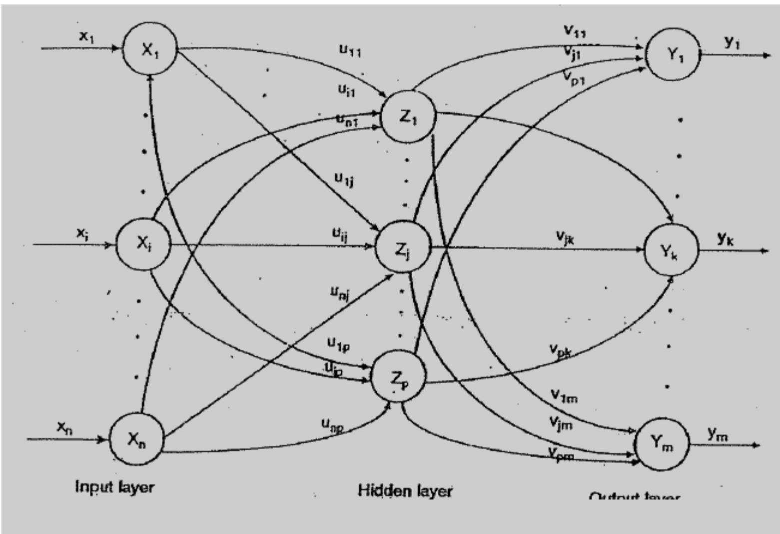

Figure2: Multilayer layer neural network

2. Artificial Neuron Network characteristics

The neuron as a processing node performs the operation of summation of its weighted inputs, or the scalar product computation to obtain net. Subsequently, it performs the nonlinear operation f (net) through its activation function. Typical activation functions used are

$$f(net) \triangleq \frac{2}{1+\exp(-\lambda net)} - 1 \qquad (1)$$

$$f(net) \triangleq \operatorname{sgn}(net) = \begin{cases} +1, & net > 0 \\ -1, & net < 0 \end{cases} \qquad (2)$$

Where $\lambda > 0$ in **(1)** is proportional to the neuron gain determining the steepness of the continuous function f(net) near net = 0. Notice that as $\lambda \to \infty$, the limit of the continuous function becomes the sgn (net) function defined in **(2)**. Activation functions **(1) and (2)** are **called bipolar continuous and bipolar binary functions**, respectively. The word "bipolar" is used to point out that both positive and negative responses of neurons are produced for this definition of the activation function.

By shifting and scaling the bipolar activation functions defined by **(1 &2)**, unipolar continuous and unipolar binary activation functions can be obtained, respectively, as

$$f(net) \triangleq \frac{1}{1 + \exp(-\lambda net)} \quad \textbf{(3)}$$

$$f(net) \triangleq \begin{cases} 1, & net > 0 \\ 0, & net < 0 \end{cases} \quad \textbf{(4)}$$

Function **(3)** is shown. Again, the unipolar binary function is the limit of f(net) in **(3)** when $\lambda \rightarrow \infty$. The soft-limiting activation functions (**1) and (3)** are often called sigmoidal characteristics, as opposed to the hard-limiting activation functions given in **(2) and (4)**. Hard-limiting activation functions describe the discrete neuron model.

If the neuron's activation function has the bipolar binary form of **(2)**, it can be represented by the diagram shown in Figure (**3-a**), which is actually a discrete neuron functional block diagram showing summation performed by the summing node and the hard-limiting thresholding performed by the threshold logic unit (TLU). This model consists of the synaptic weights, a summing node, and the TLU element.

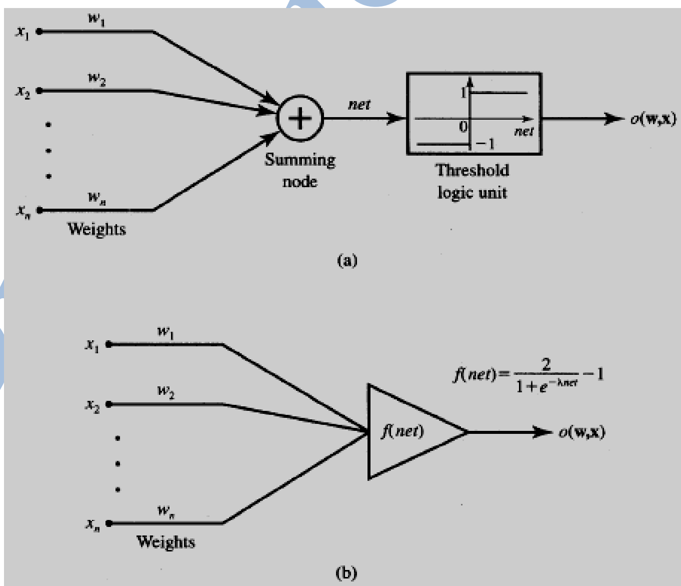

Figure 3: Common models of neurons with synaptic connections: (a) hard-limiting neuron (binary perceptron) and (b) soft-limiting neuron (continuous perceptron).

1. Feed forward Network

Let us consider elementary feed forward architecture of m neurons receiving n inputs as shown in Figure **4(a)**. Its output and input vectors are, respectively

$$\mathbf{o} = [o_1 \quad o_2 \quad \cdots \quad o_m]^t$$
$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^t \qquad \textbf{(5)}$$

Weight w_{ij} connects the i'th neuron with the j'th input. The double subscript convention used for weights such that the first and second subscript denote the index of the destination and source nodes, respectively. We thus can write the activation value for the i'th neuron as

$$net_i = \sum_{j=1}^{n} w_{ij}x_j, \quad \text{for } i = 1, 2, \ldots, m \qquad \textbf{(6)}$$

The following nonlinear transformation Equation **(7)** involving the activation function f(neti), for i = 1, 2, . . . , m, completes the processing of x. The transformation, performed by each of the m neurons in the network, is a strongly nonlinear mapping expressed as

$$o_i = f(\mathbf{w}_i^t\mathbf{x}), \quad \text{for } i = 1, 2, \ldots, m \qquad \textbf{(7)}$$

where weight vector w_i contains weights leading toward the i'th output node and is defined as follows

$$\mathbf{w}_i \triangleq [w_{i1} \quad w_{i2} \quad \cdots \quad w_{in}]^t \qquad \textbf{(8)}$$

Introducing the nonlinear matrix operator Γ , the mapping of input space x to output space o implemented by the network can be expressed as follows

$$\mathbf{o} = \Gamma[\mathbf{W}\mathbf{x}] \qquad \textbf{(9)}$$

where W is the weight matrix, also called the connection matrix:

Note that the nonlinear activation functions f(.) on the diagonal of the matrix operator r operate component wise on the activation values net of each neuron. Each activation value is, in turn, a scalar product of an input with the respective weight vector.

$$\mathbf{W} \triangleq \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \quad (10)$$

$$\Gamma[\cdot] \triangleq \begin{bmatrix} f(\cdot) & 0 & \cdots & 0 \\ 0 & f(\cdot) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & f(\cdot) \end{bmatrix} \quad (11)$$

The input and output vectors x and o are often called input and output patterns, respectively.
Thus, we can rewrite **(9)** in the explicit form involving time t as

$$\mathbf{o}(t) = \Gamma[\mathbf{W}\mathbf{x}(t)] \quad (12)$$

Figure **4(b)** shows the block diagram of the feedforward network. As can be seen, the generic feedforward network is characterized by the lack of feedback. This type of network can be connected in cascade to create a multilayer network

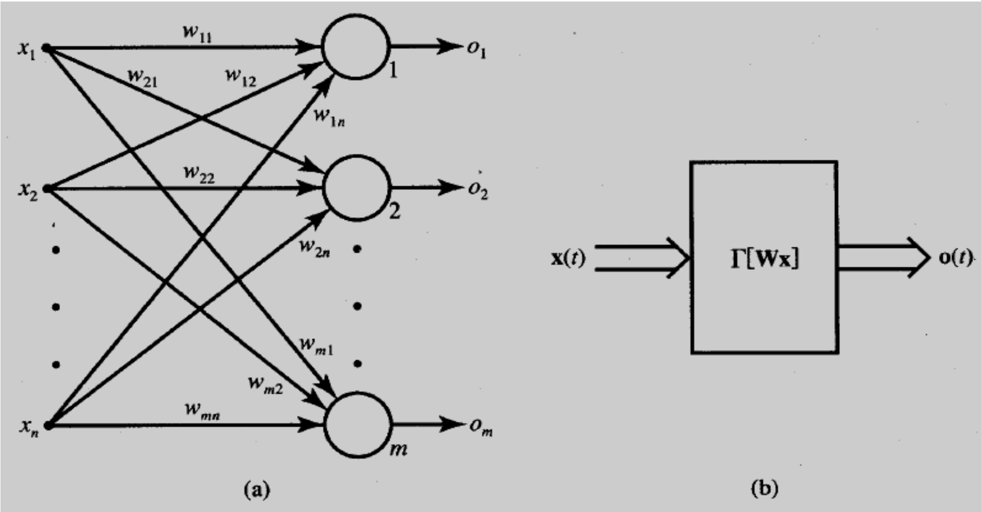

Figure 4: Single-layer feed forward network: (a) interconnection and (b) Scheme and block diagram.

Example1: Two-layer feedforward network using neurons having the bipolar binary activation function , is shown in **Fig (5)**. **Find output o_5** for a given network and input pattern.

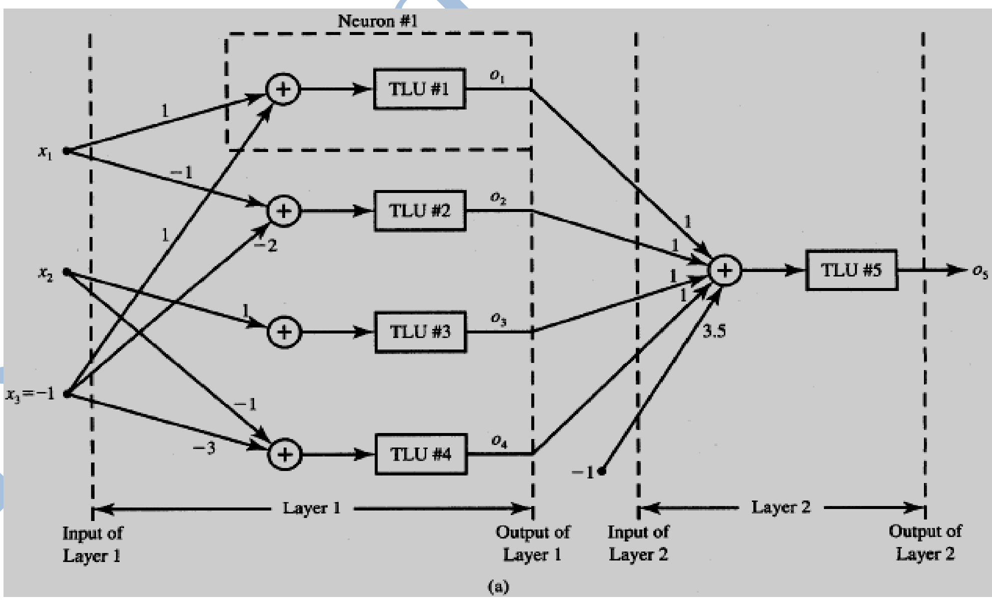

Figure 5: Two-layer feed forward network diagram

Solution:

$$\mathbf{o} = \Gamma[\mathbf{W}\mathbf{x}]$$

By inspection of the network diagram, we obtain the output, input vectors, and the weight matrix **for the first layer**, respectively, as

$$\mathbf{o} = \begin{bmatrix} o_1 & o_2 & o_3 & o_4 \end{bmatrix}^t$$
$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & -1 \end{bmatrix}^t$$
$$\mathbf{W}_1 = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & -1 & -3 \end{bmatrix}$$

Similarly, **for the second layer we can write**

$$\mathbf{o} = [o_5]$$
$$\mathbf{x} = \begin{bmatrix} o_1 & o_2 & o_3 & o_4 & -1 \end{bmatrix}^t$$
$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 3.5 \end{bmatrix}$$

The response of the first layer can be computed for bipolar binary activation functions as below

$$\mathbf{o} = \begin{bmatrix} \mathrm{sgn}(x_1 - 1) & \mathrm{sgn}(-x_1 + 2) & \mathrm{sgn}(x_2) & \mathrm{sgn}(-x_2 + 3) \end{bmatrix}^t$$

The response of the second layer can be easily obtained as

$$o_5 = \mathrm{sgn}(o_1 + o_2 + o_3 + o_4 - 3.5)$$

Note that the fifth neuron responds + 1 if and only if $o_1 = o_2 = o_3 = o_4 = 1$.

Example: Repeat Example **(1)** by using neurons having the sigmoid characteristics activation

Sol:

We obtain for the first layer

$$\mathbf{o} = \begin{bmatrix} \dfrac{2}{1+\exp(1-x_1)\lambda} - 1 \\ \dfrac{2}{1+\exp(x_1-2)\lambda} - 1 \\ \dfrac{2}{1+\exp(-x_2)\lambda} - 1 \\ \dfrac{2}{1+\exp(x_2-3)\lambda} - 1 \end{bmatrix}$$

And for the second layer

$$o_5 = \frac{2}{1+\exp(3.5-o_1-o_2-o_3-o_4)\lambda} - 1$$

2. Feedback Network

A feedback network can be obtained from the feed forward network shown in Figure **6(a)** by connecting the neurons' outputs to their inputs. The result is depicted in Figure **6(a)**. The essence of closing the feedback loop is to enable control of output $\mathbf{o_i}$ through outputs $\mathbf{o_j}$, for $\mathbf{j} = 1, 2, \ldots, m$. Such control is Especially meaningful if the present output, say **o(t)**, controls the output at the following instant, **o(t + Δ)**. The time Δ elapsed between **t** and **t + Δ** is introduced by the delay elements in the feedback loop as shown in Figure **5(a)**. Here the time delay A has a symbolic meaning; it is an analogy to the refractory period of an elementary biological neuron model. Using the notation introduced for feed forward networks, the mapping of **o(t)** into **o(t** +Δ) can now be written as

$$\mathbf{o}(t+\Delta) = \Gamma[\mathbf{W}\mathbf{o}(t)] \quad \textbf{(13)}$$

If we consider time as a discrete variable and decide to observe the network performance at discrete time instantsΔ, 2Δ, 3Δ, . . . , the system is called discrete-time. For notational convenience, the time step in discrete-time networks is equated to unity, and the time instances are indexed by positive integers. Symbol Δ thus has the meaning of unity delay. For a discrete-time artificial neural system, we have Converted **(13)** to the form

$$\mathbf{o}^{k+1} = \Gamma[\mathbf{W}\mathbf{o}^{k}], \quad \text{for } k = 1, 2, \ldots$$

Where k is the instant number. The network in **Figure 5** is called recurrent, since its response at the k + l'th instant depends on the network starting at k = 0. Indeed, we have from **(14)** a series of nested solutions as follows

$$\begin{aligned} \mathbf{o}^{1} &= \Gamma[\mathbf{W}\mathbf{x}^{0}] \\ \mathbf{o}^{2} &= \Gamma\left[\mathbf{W}\Gamma[\mathbf{W}\mathbf{x}^{0}]\right] \\ &\ldots \\ \mathbf{o}^{k+1} &= \Gamma\left[\mathbf{W}\Gamma\left[\ldots\Gamma[\mathbf{W}\mathbf{x}^{0}]\ldots\right]\right] \end{aligned} \qquad \textbf{(15)}$$

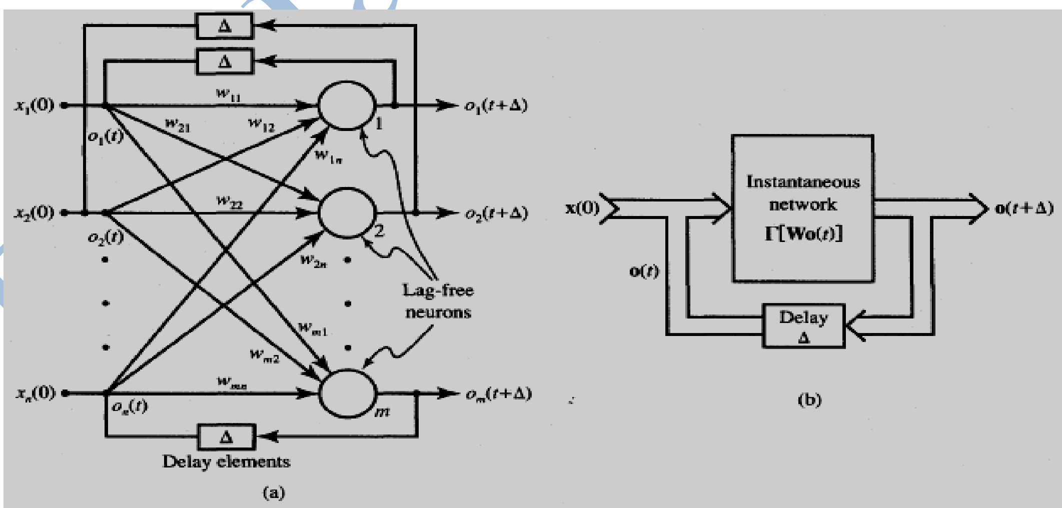

Figure 6: Single-layer discrete-time feedback network: (a) interconnection scheme and (b) block diagram

Example2: Figure (7) shows the Recurrent discrete-time network . Find the output network pattern. Start with $X^0_1 = X^0_2 = X^0_3 = 1$; $X^0_4 = -1$

.

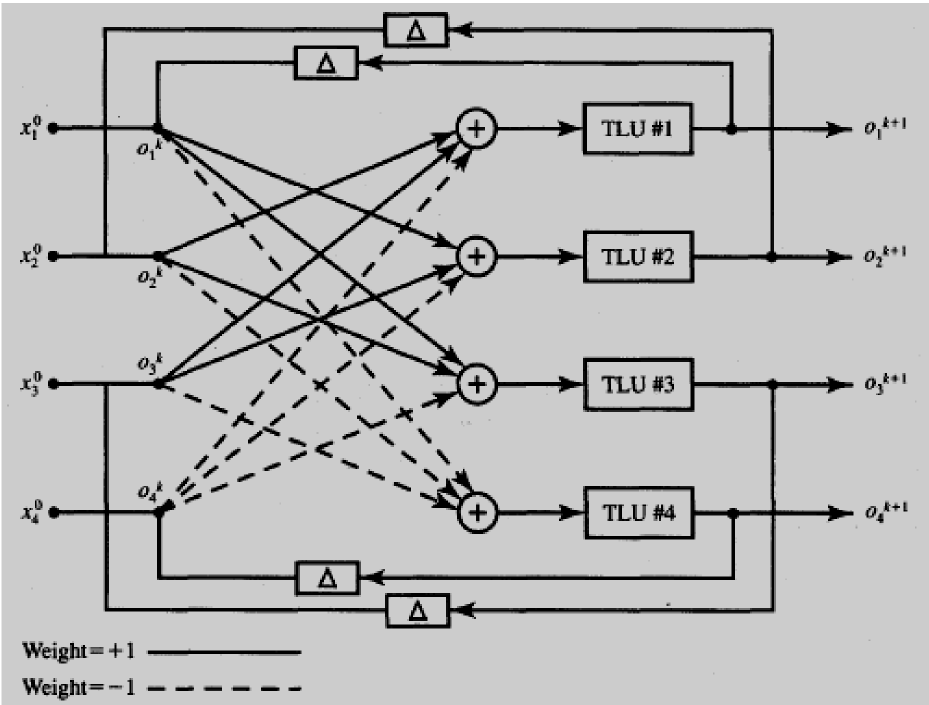

Figure 7: Recurrent network for two-equilibrium state diagram

Solution:

Its weight matrix can be set up by inspection as

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

$$\mathbf{o}^1 = \begin{bmatrix} \operatorname{sgn}(\cdot) & 0 & 0 & 0 \\ 0 & \operatorname{sgn}(\cdot) & 0 & 0 \\ 0 & 0 & \operatorname{sgn}(\cdot) & 0 \\ 0 & 0 & 0 & \operatorname{sgn}(\cdot) \end{bmatrix} \begin{bmatrix} net_1^0 \\ net_2^0 \\ net_3^0 \\ net_4^0 \end{bmatrix}$$

$O_1^1 = [\ \operatorname{sgn}(3)\ \operatorname{sgn}(3)\ \operatorname{sgn}(3)\ \operatorname{sgn}(-3)]^t$

$O_1^1 = O_2^1 = O_3^1 = 1;\ O_4^1 = -1$

Ex3. Repeat the above example with $X^0_1 = X^0_2 = X^0_3 = X^0_4 = 1$

$$\mathbf{o}^1 = [\operatorname{sgn}(1) \quad \operatorname{sgn}(1) \quad \operatorname{sgn}(1) \quad \operatorname{sgn}(-3)]^t$$

Therefore, the transition that takes place is

$$[1 \quad 1 \quad 1 \quad 1] \rightarrow [1 \quad 1 \quad 1 \quad -1]$$

H.W: Repeat Ex3. With $x^0 = [1\ 1\ {-1}\ {-I}]'$,
$x^0 = [1\ {-1}\ 1\ {-1}]'$, and
$x^0 = [-1\ 1\ 1\ {-1}]$