## Neural networks viewed as directed graph

1- Signal flow graph:

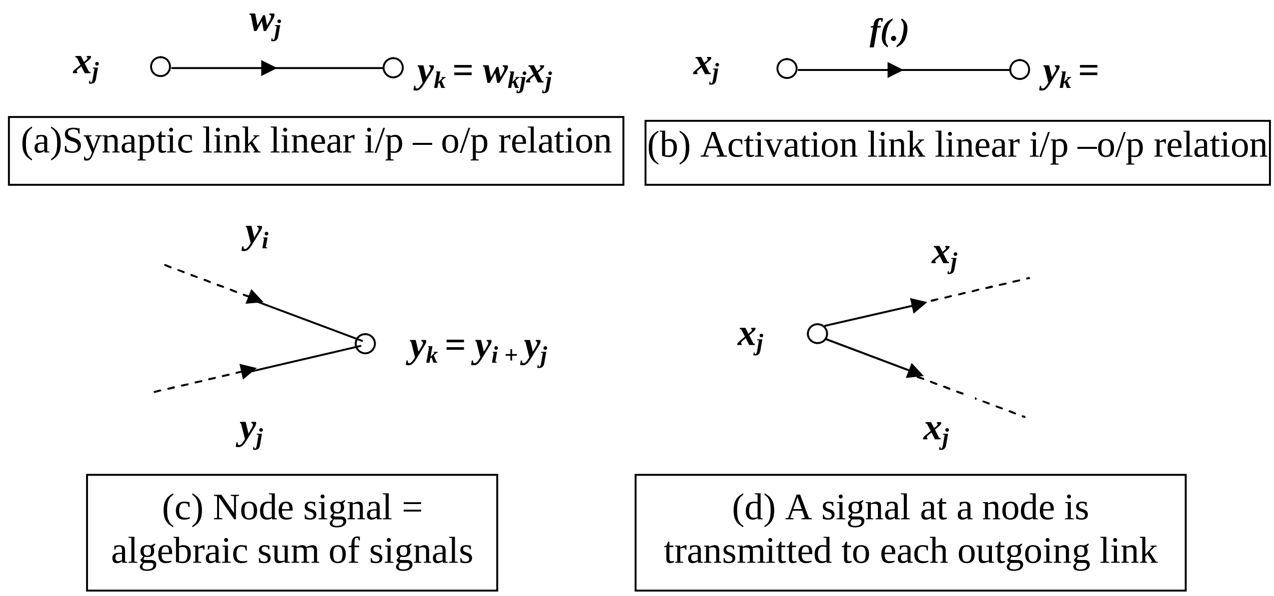



**Fig. (6) Signal – flow graph component**

Using these rules we may construct the signal – flow diagram

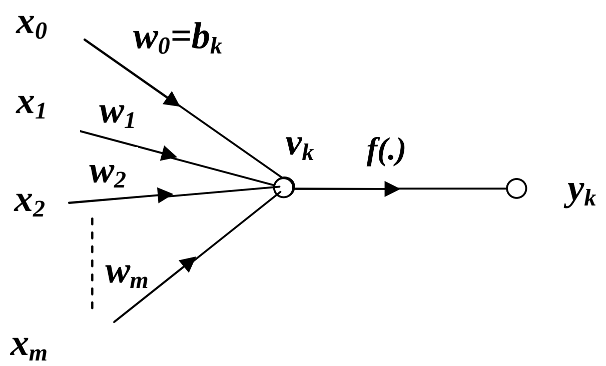



**Fig. (7) signal – flow graph of a neuron**

A reduced form of this graph is shown in fig. (8). This graph describes the layout of the neural network referred to as an “architectural graph”.

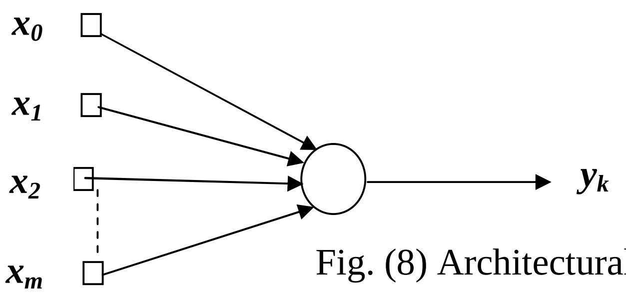



Fig. (8) Architectural graph of a neural

## Feedback

Feedback is said to exist in a dynamic system whenever the output of an element in the system influences in part the input applied to that particular element, thereby giving rise to one or more closed paths for the transmission of signals around the system. Fig. (9) shows the signal – flow graph of a single loop feedback

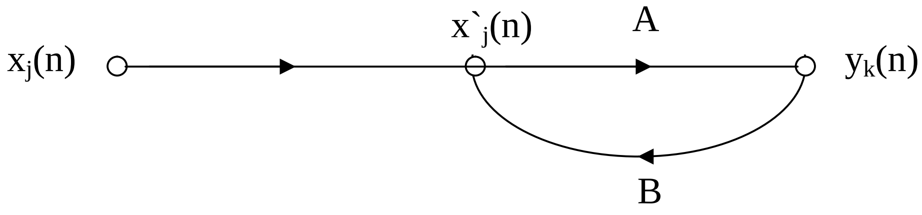



Fig. (9) Signal–flow graph of a single loop feedback system

Where, $x_j(n)$ is the input signal,

$x`_j(n)$ is the internal signal

$y_k(n)$ is the output signal

These signals are functions of the discrete – time variable (n). The system is assumed to be linear, consisting of forward path and feedback path that are characterized by the operators **A** and **B** respectively.

From figure (9)

$$\boldsymbol{y_k(n) = \mathbf{A}\,[x`_j(n)]} \qquad \ldots\ldots(1)$$

and

$$\boldsymbol{x`_j(n) = x_j(n) + \mathbf{B}[y_k(n)]} \qquad \ldots\ldots\ldots\ldots..(2)$$

from eq. (1)

$$y_k(n) = \frac{A}{1-AB}[x_j(n)] \qquad \ldots\ldots\ldots(3)$$

A/(1-AB) is the closed loop operator of the system and AB is the open loop operator which non-commutative in that BA ≠ AB.

Now consider A as a fixed weight W, B is a unit-delay operator $z^{-1}$, whose output is delayed with respect to the input by "one time unit". We may then express the closed loop operator of the system as:

$$\frac{A}{1-AB} = \frac{w}{1-wz^{-1}} = w(1-wz^{-1})^{-1} \quad \text{…………(4)}$$

Using binomial expansion for $(1 - wz^{-1})^{-1}$ , we may rewrite the closed-loop operator of the system as:

$$\frac{A}{1-AB} = w\sum_{l=0}^{\infty} w^{l} z^{-l} \quad \text{…………….(5)}$$

Substituting eq. (5) into eq. (3),

$$y_k(n) = w\sum_{l=0}^{\infty} w^{l} z^{-l}[x_j(n)] \quad \text{…………(6)}$$

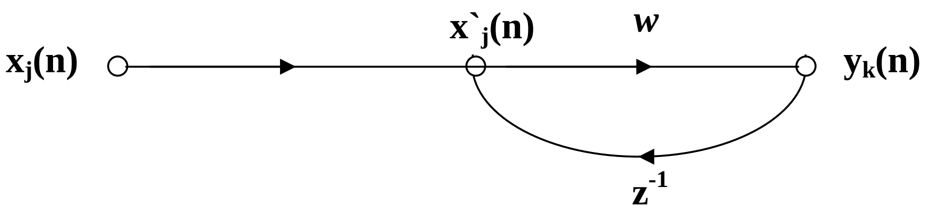
From the definition of $\mathbf{z}^{-1}$, we have

$$\mathbf{z}^{-l}[x_j(n)] = x_j(n-l) \quad \text{……………..(7)}$$

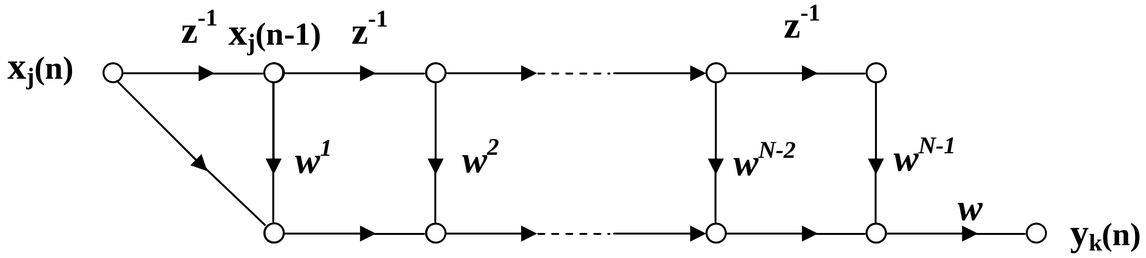
where $x_j(n-l)$ is a sample of the input signal delayed by $l$ time units. Accordingly, we may express the output signal $y_k(n)$ as an infinite weighted summation of present and past samples of the input signal $x_j(n)$**,** as shown by:

$$y_k(n) = \sum_{l=0}^{\infty} w^{l+1} x_j(n-l) \quad \text{…………(8)}$$

We now see clearly that the dynamic behavior of a feedback system represented by the signal-flow graph of fig. (10) is controlled by the weight ***w***.




(a)




(b)

Fig. (10) (a) signal-flow graph of first-order, infinite-duration impulse response IIR filter.

(b) Feedforward approximation of part (a) of the figure, obtained by truncating eq. (6)

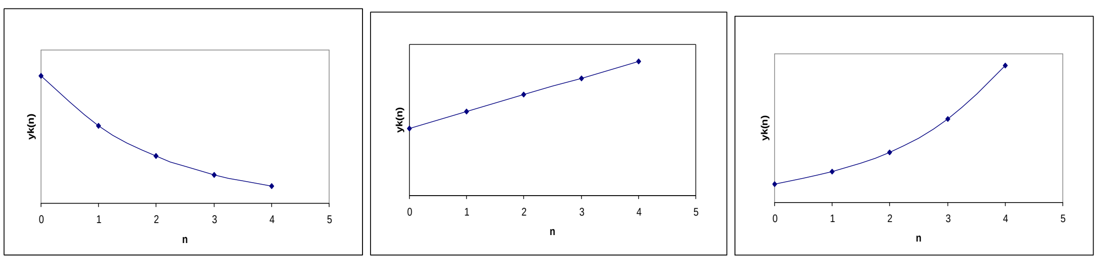
In particular two specific cases can be distinguished:

1- **$|w| < 1$**, the output signal $y_k(n)$ is exponentially convergent, that is the **system is stable**. This case is illustrated in fig. 11-a for a positive ***w***.

2- **$|w| \geq 1$,** the output signal $y_k(n)$ is divergent, that is, the **system is unstable**.

If **$|w| = 1$** the divergence is linear, as in fig. 11-b,

and if **$|w| > 1$** the divergence is exponential, as in fig. 11-c.




(a) stable |w|<1 (positive) (b) |w|=1(linear divergence) (c)|w|>1 exponential divergence

Fig. (11) the response of fig. (10) for 3 different values of ***w*.**

Now, suppose that, for some power **N**, |***w***| is small enough relative to the unity such that $w^N$ is negligible for all practical purpose. In such situation we may approximate the output $y_k$ by finite sum

$$y_k(n) \approx \sum_{l=0}^{N-1} w^{l+1} x_j(n-l)$$

$$= wx_j(n) + w^2x_j(n-1) + w^3x_j(n-2) + \dots + w^Nx_j(n-N+1)$$

## Network Architecture

In general, we may identify three fundamentally different classes of network architectures:

**1- Single layer feedforward networks:**

In a layered neural network, the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects directly onto an output layer of neurons, (computation nodes), but not vice versa. In other words, this network of a feedforward type. Fig. (12) illustrated four nodes in both the input and the output layers. Such a network is called a single-layer network. With the designation "single layer" referring to the output layer of computation nodes (neurons). We do not count the input layer of source nodes because no computation is performed there.

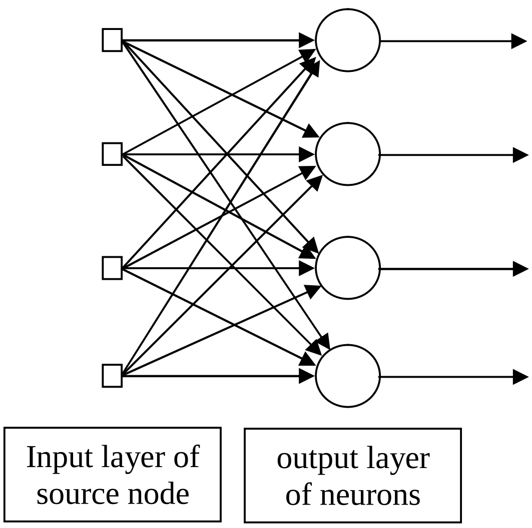



Fig. (12) Feedforward network with a single layer of neurons

Example-1: Design neural network represents basic Boolean function AND with two input variables and bias, using the threshold function as the output activation function.

Sol:

Let $x_1, x_2$ be the two input variables, $w_0$ the weight of the bias and $y$ the output of the N.N.

Since the output of the AND gate is 1 only when x1 and x2 are 1's, this will lead to the following:

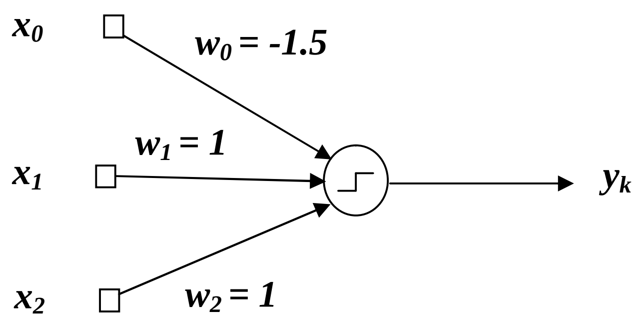
$w_1x_1 + w_2x_2 + w_0 \leq 0$, for $y = 0$

$w_1x_1 + w_2x_2 + w_0 > 0$, for $y = 1$

| $x_1$ | $x_2$ | $w_1x_1 + w_2x_2$ | $y$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $w_0 \leq 0$ |
| 1 | 0 | $w_1$ | 0 | $(w_1 + w_0) \leq 0$ |
| 0 | 1 | $w_2$ | 0 | $(w_2 + w_0) \leq 0$ |
| 1 | 1 | $w_1 + w_2$ | 1 | $(w_1 + w_2 + w_0) > 0$ |

From the table we can recognize that $w_0$ is negative value, and $(w_1 + w_2 + w_0)$ should be positive, thus,

$w_1 = w_2 = 1$, and $w_0 = -1.5$ will satisfy the required conditions.

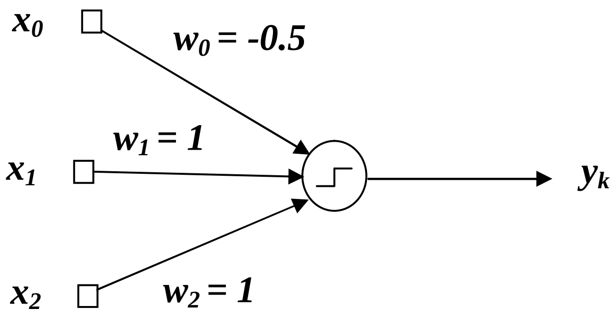



Example-2: As example-1 but for OR function

Sol: The output of OR function is 1 except for $x_1 = x_2 = 0$

| $x_1$ | $x_2$ | $w_1x_1 + w_2x_2$ | $y$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $w_0 \leq 0$ |
| 1 | 0 | $w_1$ | 1 | $(w_1 + w_0) > 0$ |
| 0 | 1 | $w_2$ | 1 | $(w_2 + w_0) > 0$ |
| 1 | 1 | $w_1 + w_2$ | 1 | $(w_1 + w_2 + w_0) > 0$ |

$w_1 = w_2 = 1$, and $w_0 = -0.5$ will satisfy the required conditions.

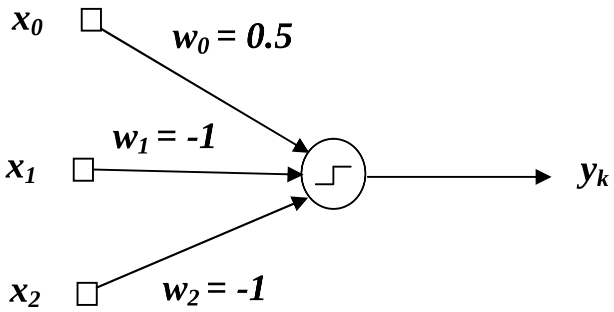



Example-3 : As example-2 but for NOR function

Sol:

| $x_1$ | $x_2$ | $w_1x_1 + w_2x_2$ | $y$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | $w_0 > 0$ |
| 1 | 0 | $w_1$ | 0 | $(w_1 + w_0) \leq 0 \rightarrow w_1 < 0$ |
| 0 | 1 | $w_2$ | 0 | $(w_2 + w_0) \leq 0 \rightarrow w_2 < 0$ |
| 1 | 1 | $w_1 + w_2$ | 0 | $(w_1 + w_2 + w_0) \leq 0$ |

Thus, $w_1 = w_2 = -1$, and $w_0 = 0.5$ will satisfy the required conditions




H.W: Design N.N that can perform the following logic unit, using threshold function as an activation function (a) with bias, (b) without bias

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

## 2- Multi-layer feedforward networks

The second class of a feedforward N.N distinguishes it self by the presence of one or more hidden layers.

- The source nodes in the input layer of the network constitute the input signals applied to the neurons in the 2nd. Layer (i.e., the first hidden layer). The output signals of the second layer are used as input to the 3rd. layer, and so on for the rest of the network.
- Figure (13) shows fully connected feedforward network with one hidden layer and one output layer.

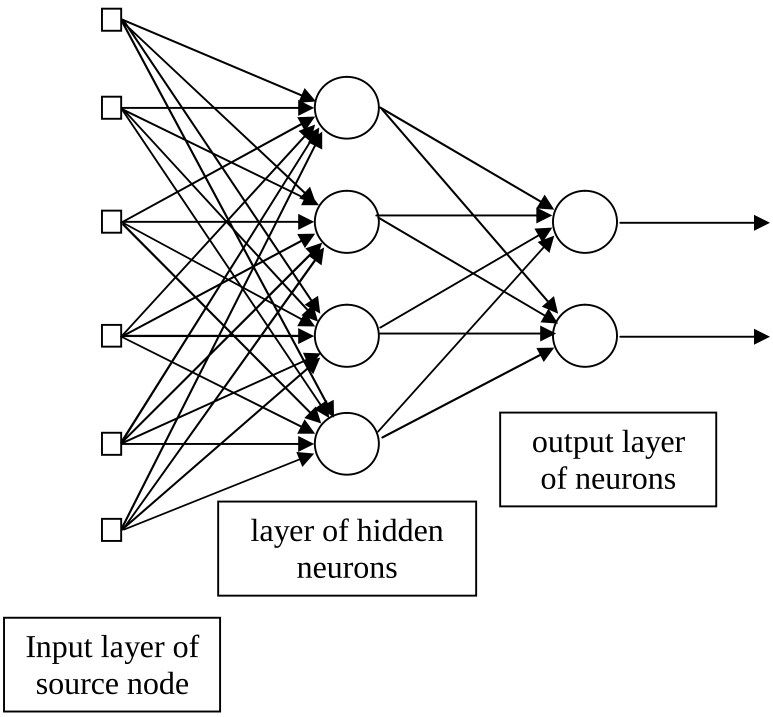
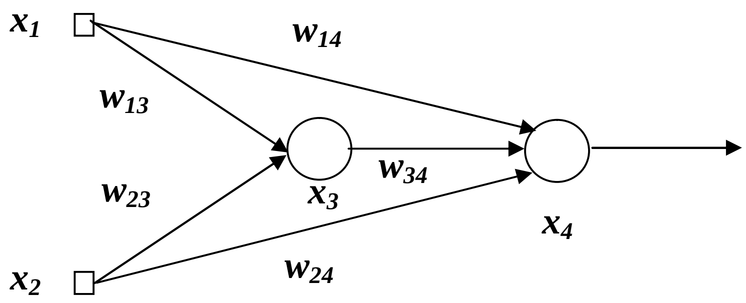



Figure (13) Fully connected multilayer feedforward network

- A feedforward network with: ***m*** source node, ***h1*** neurons in the first hidden layer, ***h2*** neurons in the second hidden layer, and ***q*** neurons in the output layer, is referred to as an ***m-h1-h2-q*** network.
- As an example, the network illustrated in fig. (13) is referred to as an ***6-4-2*** network.

Example-4: A simple example often given to illustrate the behavior of multilayer feedforward N.Ns is the one used to implement the XOR (exclusive OR) function. Such a function can be obtained in several ways, such as,




| $x_1$ | $x_2$ | $w_{13}x_1 + w_{23}x_2$ | $x_3$ | $y$ | $w_{14}x_1 + w_{24}x_2 + w_{34}x_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $0 + 0 + 0$ |
| 1 | 0 | $w_{13}$ | $sgn(w_{13})$ | 1 | $(w_{14} + w_{34}\, sgn(w_{13})) > 0$ |
| 0 | 1 | $w_{23}$ | $sgn(w_{23})$ | 1 | $(w_{24} + w_{34}\, sgn(w_{23})) > 0$ |
| 1 | 1 | $w_{13} + w_{23}$ | $sgn(w_{13} + w_{23})$ | 0 | $(w_{14} + w_{24} + w_{34}\, sgn(w_{13} + w_{23})) \leq 0$ |

Starting from last pattern, if $sgn(w_{13} + w_{23}) > 0$ then $x3=1$, and if $w_{34} > 0$, then $|w_{14} + w_{24}| > w_{34}$, and $(w_{14} + w_{24}) < 0$ ……..(*)

***Now let $w_{13} = w_{23} = 1$, adding the output of pattern 2 to pattern 3:***

$((w_{14} + w_{34}\, sgn(w_{13})) + (w_{24} + w_{34}\, sgn(w_{23}))) > 0$

$((w_{14} + w_{24}) + w_{34}(sgn(w_{13}) + sgn(w_{23}))) > 0 \rightarrow ((w_{14} + w_{24}) + w_{34}(1 + 1)) > 0$

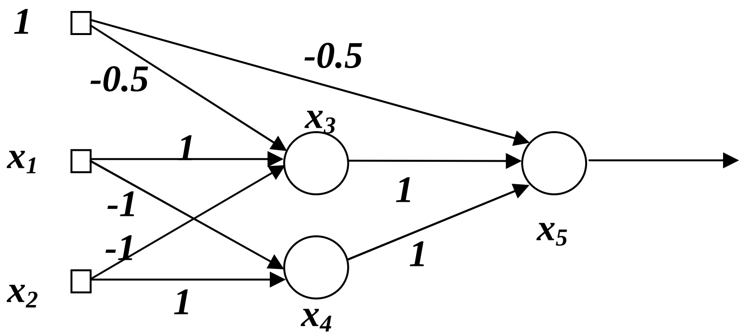
$((w_{14} + w_{24}) + 2w_{34}) > 0$

From eq. (*) we have $(w_{14} + w_{24}) < 0$

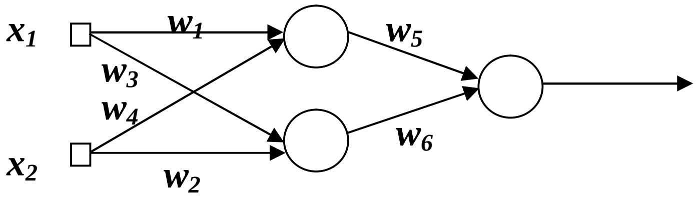
***Let*** $w_{14} = w_{24} = -1$,

**then** $2w_{34}$ ***should be*** $> 2$, ***i. e.,*** $w_{34} > 1$, ***say*** $1.1$

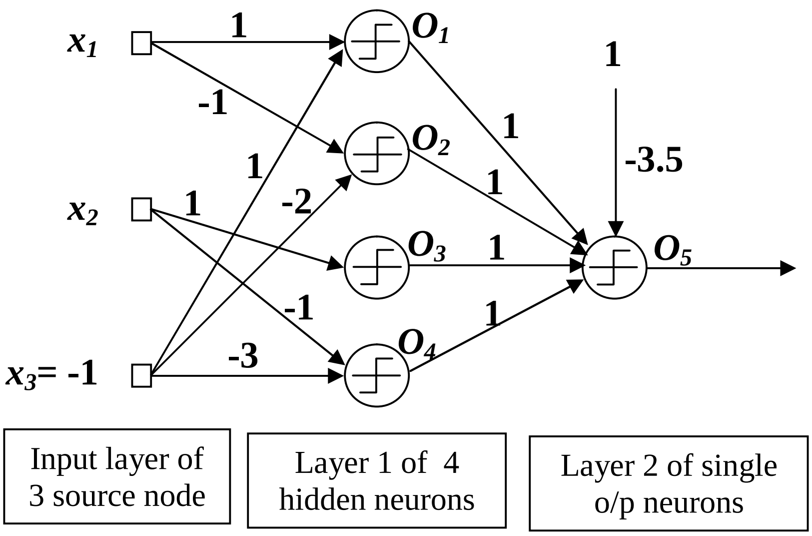
Another N.N example for XOR function is as illustrated in the figure:




H.W: Find the suitable weights to represent the XOR for (a) threshold = 1, (b) threshold = 2.




Example-5: This example presents the analysis of a two-layer feedforward N.N having the bipolar binary function (-1, +1) as an activation function. Find the output for the given network and the input pattern to each layer.




Sol: For the first layer, the input, output vectors and the weight matrix, respectively are:

$$\mathbf{X} = [\ x_1\ \ x_2\ \ -1]^T$$

$$\mathbf{O} = [\, \boldsymbol{o_1}\ \ \boldsymbol{o_2}\ \ \boldsymbol{o_3}\ \ \boldsymbol{o_4}]^T$$

$$\mathbf{W^1} = \begin{pmatrix} \mathbf{w11} & \mathbf{w21} & \mathbf{w31} \\ \mathbf{w12} & \mathbf{w22} & \mathbf{w32} \\ \mathbf{w13} & \mathbf{w23} & \mathbf{w33} \\ \mathbf{w14} & \mathbf{w24} & \mathbf{w34} \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} & \mathbf{-2} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{-1} & \mathbf{-3} \end{pmatrix}$$

similarly for the second layer:

$$\mathbf{X} = [\, \boldsymbol{o_1}\ \ \boldsymbol{o_2}\ \ \boldsymbol{o_3}\ \ \boldsymbol{o_4}\ \ \boldsymbol{1}]^T$$

$$\mathbf{O} = [\boldsymbol{o_5}]$$

$$\mathbf{W^{2T} = [1\ \ 1\ \ 1\ \ 1\ \ -3.5]}$$

The response of the first layer can be computed for bipolar binary activation function as:

$$\mathbf{O = W^1X} =$$

$$\begin{pmatrix} 1 & 0 & 1 \\ -1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & -1 & -3 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ -1 \\ \ \end{pmatrix} = \begin{pmatrix} \mathrm{sgn}(x_1-1) \\ \mathrm{sgn}(-x_1+2) \\ \mathrm{sgn}(x_2) \\ \mathrm{sgn}(-x_2+3) \end{pmatrix}$$

The response of the second layer can easily be obtained as:

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

Note that $\boldsymbol{y = o_5}$ (fifth neuron) responds **+1** if and only if $\mathbf{o_1=o_2=o_3=o_4=1}$

## 3- Recurrent Networks

A recurrent network distinguishes it self from a feedforward neural network in that it has at least one feedback loop.

Fig. (14) illustrates a recurrent neural network with hidden neurons. The feedback connections shown in the figure originate from the hidden neurons as well as from the output neurons.

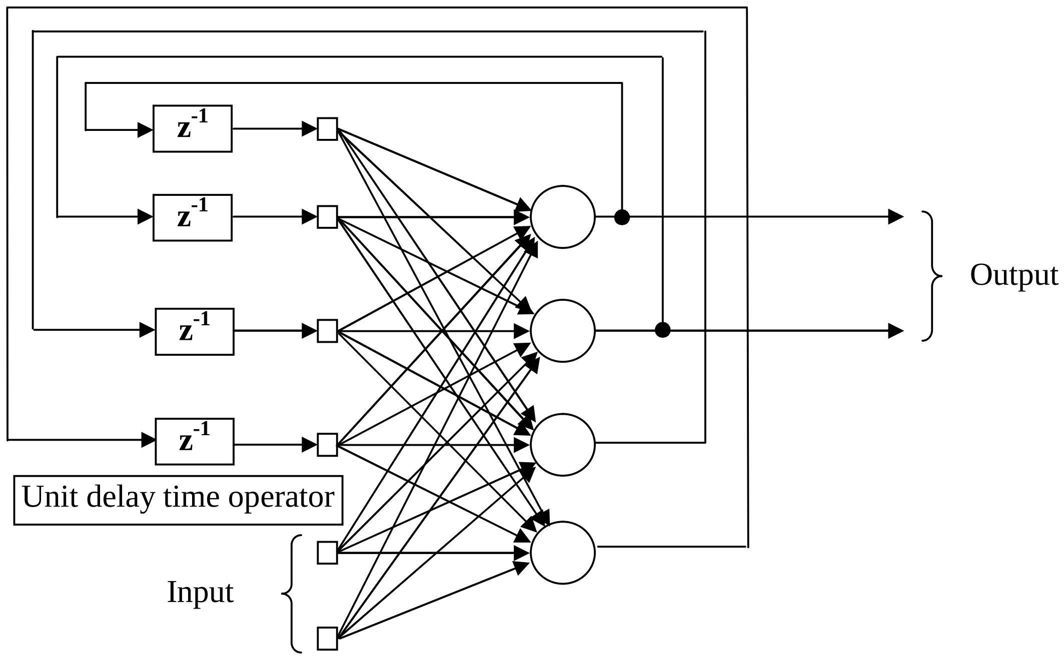



Figure (14) Recurrent network with hidden neurons