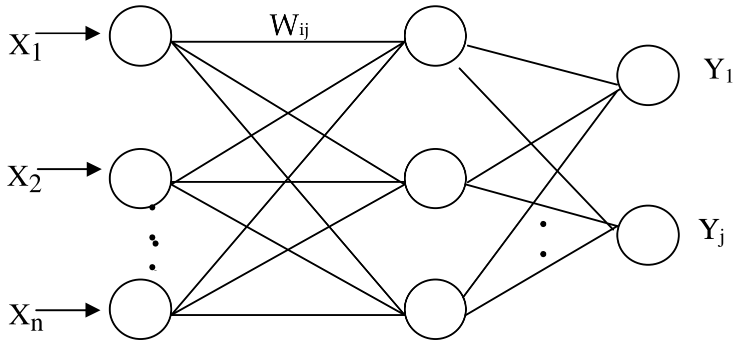
Introduction to ANN Basic Model



1. Input layer: $[X_1, X_2, \ldots.. X_n]^t$, where t means vector transpose

2. Hidden layer: $I_j \longrightarrow net_j \longrightarrow Y_j$

3. Output layer: Yj

   Three ways of generating output: normalized, competitive output, competitive learning

4. Weights: Wij means the value connecting between layers

5. Processing Element(PE)

   (A)Summation Function: $I_j = \sum_i W_{ij} X_i$ or $I_j = \sum_i (X_i - W_{ij})^2$

   (B)Activity Function: $net_j = I_j^{\,n}$ or $net_j^{\,n} = I_j^{\,n} + C \cdot net_j^{\,n-1}$

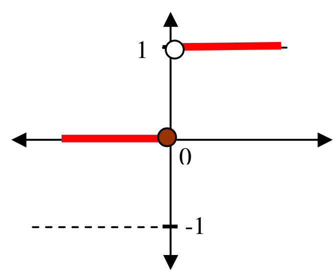
   or $net_j = I_j^{\,n} + C \cdot I_j^{\,n-1}$

   (C)Transfer Function:

   (1) Discrete

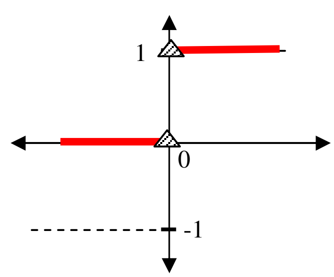
$$Y_j = \begin{cases} 1 & \text{if } net_j > 0 \\ 0 & net_j \le 0 \end{cases}$$

Step function or perceptron fc.



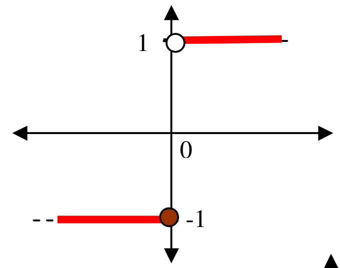
$$Y_j^n = \begin{cases} 1 & net_j > 0 \\ Y_j^{n-1} & \text{if } net_j = 0 \\ 0 & net_j < 0 \end{cases}$$

Hopfield-Tank



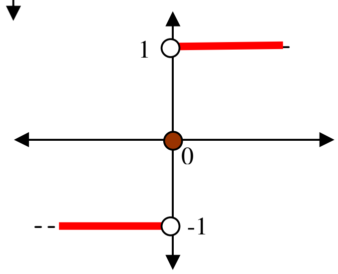
$$Y_j = \begin{cases} 1 & \\ & \text{if} \\ -1 & \end{cases} \quad \begin{matrix} net_j > 0 \\ \\ net_j <= 0 \end{matrix}$$

Signum fc.



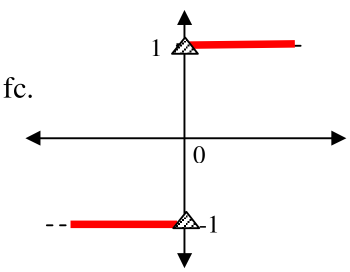
$$Y_j = \begin{cases} 1 & & net_j > 0 \\ 0 & \text{if} & net_j = 0 \\ -1 & & net_j < 0 \end{cases}$$

Signum0 fc.



$$Y^n_j = \begin{cases} 1 & & net_j > 0 \\ Y^{n-1}_j & \text{if} & net_j = 0 \\ -1 & & net_j < 0 \end{cases}$$

BAM fc.



(2)Linear:

$$Y_j = net_j$$

$$Y_j = \begin{cases} net_j & \\ & \text{if} \\ 0 & \end{cases} \quad \begin{matrix} net_j > 0 \\ \\ net_j \leq 0 \end{matrix}$$

(3)Non linear:

$$Y_j = \frac{1}{1 + \ell^{-net_j}}$$

Sigmoid fc.

$$Y_j \quad \frac{\ell^{net_j} - \ell^{-net_j}}{\ell^{net_j} + \ell^{-net_j}}$$

Hyperbolic Tangent fc.

6. Learning:

   Based on the ANN model used, learning is using a set of training pattern to adjust weights in the network.

7. Recalling:

   Based on the ANN model used, recalling is applying the real data pattern into the trained network to generate the output.

8. Energy function:

Energy function is a verification function which determines if the network energy has converge its minimum. Whenever the energy function is approach zero the network is approach its optimum solution.

(a) The energy function for supervised network learning:

$$E = \frac{1}{2}\sum_{j}\left(T_j - Y_j\right)^2$$, where E is the energy value

$$\Delta W_{ij} = -\eta \cdot \frac{\partial E}{\partial W_{ij}}$$; this is the value for adjusting weight W_{ij}

(b) The energy function for un-supervised network learning:

$$E = \frac{1}{2}\sum_{i}\left(X_i - W_{ij}\right)^2$$

$$\Delta W_{ij} = -\eta \cdot \frac{\partial E}{\partial W_{ij}}$$; this is the value for adjusting weight W_{ij}