-Artificial Neural Network-
Chapter 2 Basic Model

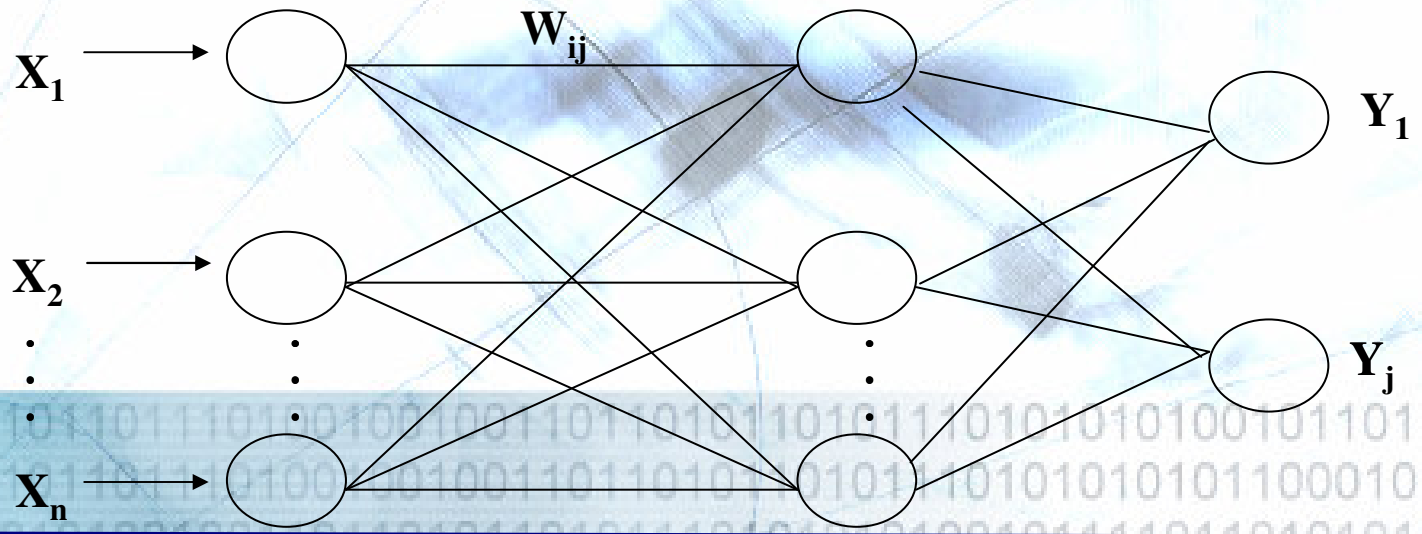
朝陽科技大學
資訊管理系
李麗華 教授

Introduction to ANN Basic Model

1. Input layer
2. Hidden layer
3. Output layer
4. Weights
5. Processing Element(PE)
6. Learning
7. Recalling
8. Energy function

ANN Components (1/4)

1. Input layer:$[X_1,X_2,.....X_n]^t$, where t means vector transpose.
2. Hidden layer: I_j => net_j => Y_j
3. Output layer:Y_j
   - Three ways of generating output: normalized, competitive output, competitive learning
4. Weights :W_{ij} means the connection value between layers



ANN Components (2/4)

5. Processing Element(PE)

(A)Summation Function: $I_j = \sum_i W_{ij} X_i$ **(supervised)**

or $I_j = \sum_i (X_i - W_{ij})^2$ **(unsupervised)**

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

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

(C)Transfer Function:

1. **Discrete type**
2. **Linear type**
3. **Non-linear type**

ANN Components (3/4)

6. Learning:

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

7. Recalling:

- **Based on the ANN model used, recalling is to apply the real data pattern to the trained network so that the outputs are generated and examined.**

ANN Components (4/4)

8. Energy function:

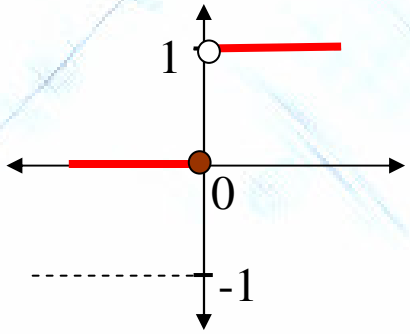
- **Energy function is a verification function which determines if the network energy has converged to its minimum. Whenever the energy function approaches to zero, the network approaches to its optimum solution.**

Transfer Functions (1/3)

- **Discrete type transfer function:**

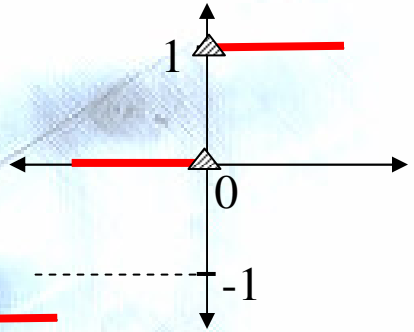
$$Y_j = \begin{cases} 1 & \text{if } net_j > 0 \\ 0 & \text{if } net_j <= 0 \end{cases}$$

Step function or perceptron fc.



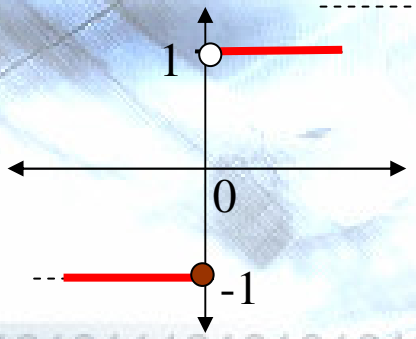
$$Y^n_j = \begin{cases} 1 & \text{if } net_j > 0 \\ Y^{n-1}_j & \text{if } net_j = 0 \\ 0 & \text{if } net_j < 0 \end{cases}$$

Hopfield-Tank fc.



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

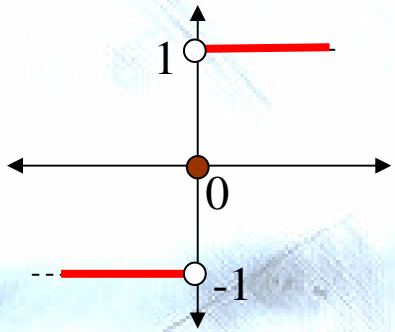
Signum fc.



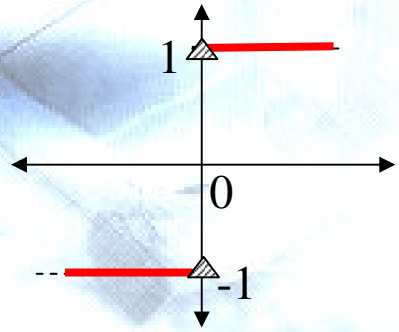
Transfer Functions (2/3)

- **Discrete type transfer function:**

$$Y_j = \begin{cases} 1 & net_j > 0 \\ 0 & \text{if } net_j = 0 \\ -1 & net_j < 0 \end{cases} \quad \text{Signum0 fc.}$$



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



Transfer Functions (3/3)

- Linear type:

$$Y_j = net_j$$

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

- Nonlinear type transfer function:

$$Y_j = \frac{1}{1 + \ell^{-net_j}} \quad \text{Sigmoid function}$$

$$Y_j = \frac{\ell^{net_j} - \ell^{-net_j}}{\ell^{net_j} + \ell^{-net_j}} \quad \text{Hyperbolic Tangent function}$$

Energy function

(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**

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

(b) The energy function for unsupervised network learning:

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

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