Môn học

NHẬP MÔN ĐIỀU KHIỂN THÔNG MINH

Giảng viên: PGS. TS. Huỳnh Thái Hoàng
Bộ môn Điều Khiển Tự Động
Khoa Điện – Điện Tử
Đại học Bách Khoa TP.HCM
Email: hthoang@hcmut.edu.vn

Chương 4

MẠNG THẦN KINH

Nội dung

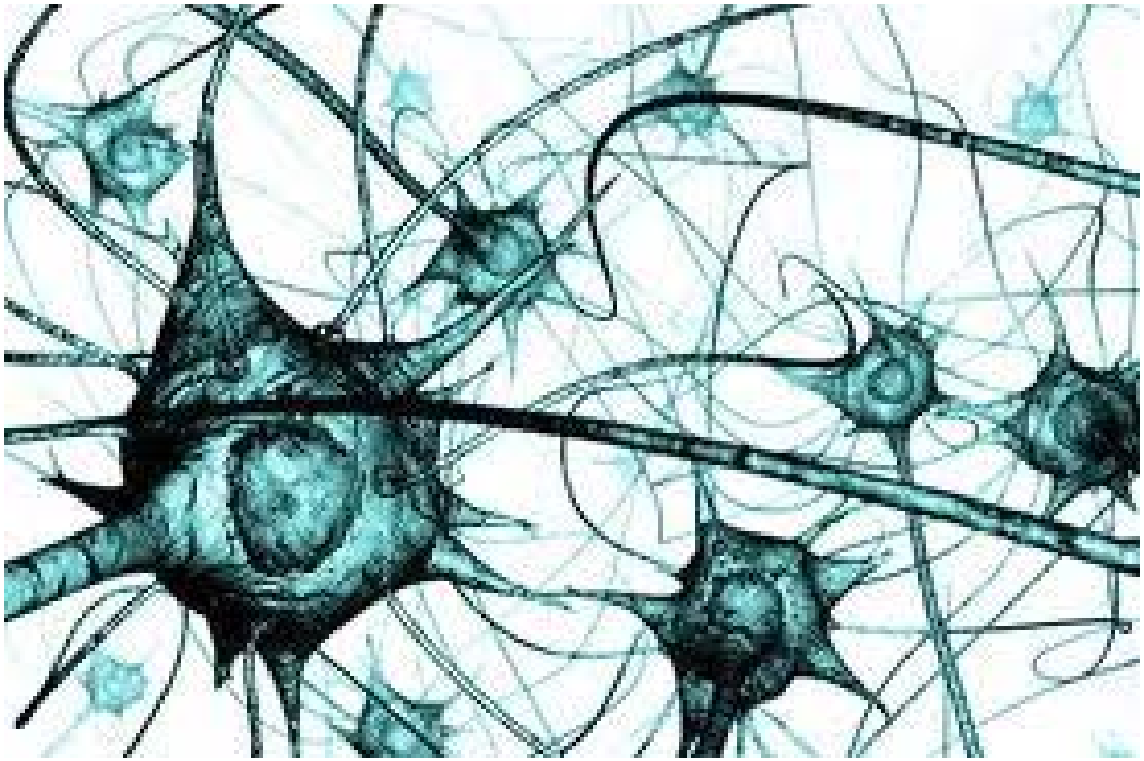
- Khái niệm về mạng thần kinh
- Các cấu trúc mạng và giải thuật huấn luyện
- Perceptron
- Adaline
- Mạng truyền thẳng nhiều lớp
- Giới thiệu Neural Network Toolbox của Matlab

Khái niệm về mạng thần kinh

Bộ não

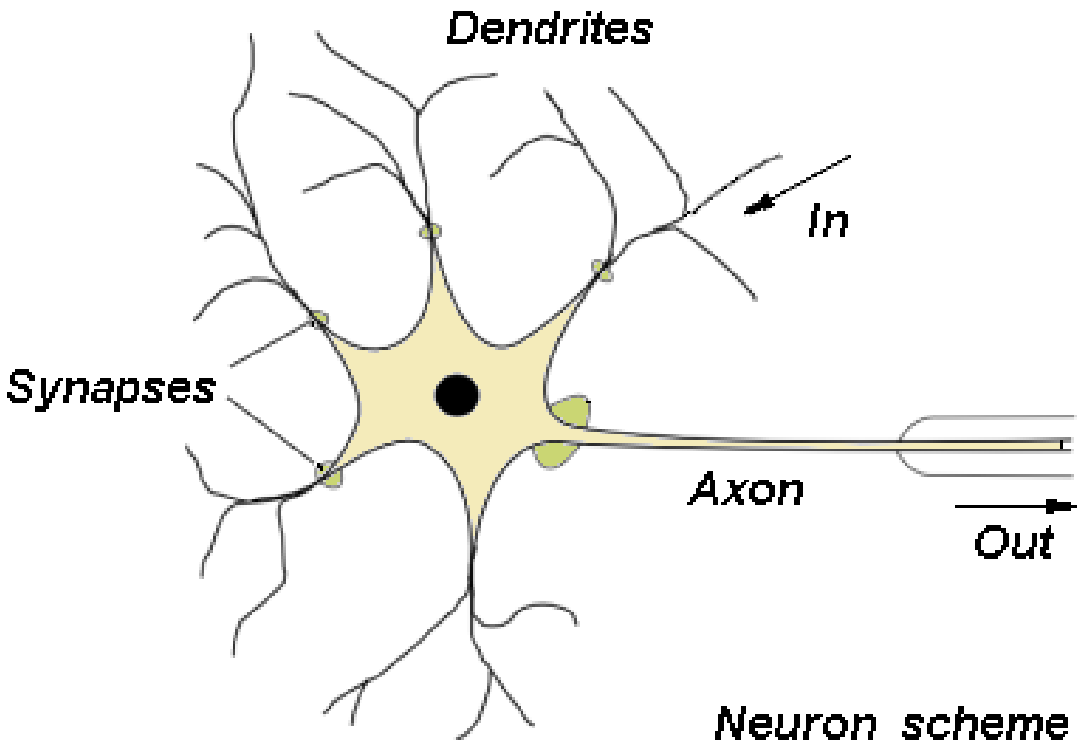
- Bộ não con người là hệ thống xử lý thông tin phức hợp, phi tuyến và song song có khả năng học, ghi nhớ, tổng quát hóa và xử lý lỗi.

Mạng thần kinh sinh học



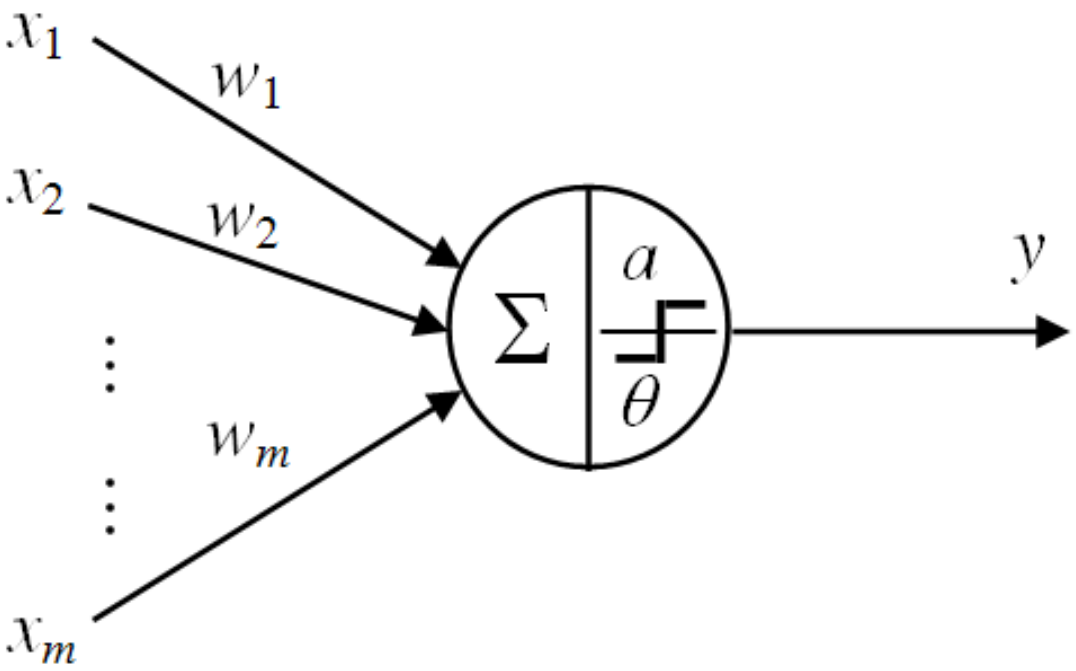
- Bộ não con người gồm khoảng 10^{11} tế bào thần kinh liên kết với nhau thành mạng

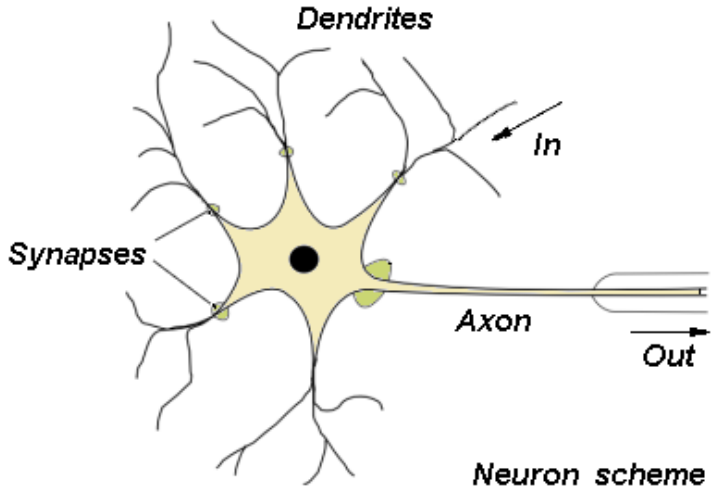
Tế bào thần kinh sinh học



- Tế bào thần kinh gồm: thân tế bào (soma), đầu dây thần kinh vào (dendrite), khớp nối (synapse), sợi trục (axon) và đầu dây thần kinh ra

Tế bào thần kinh





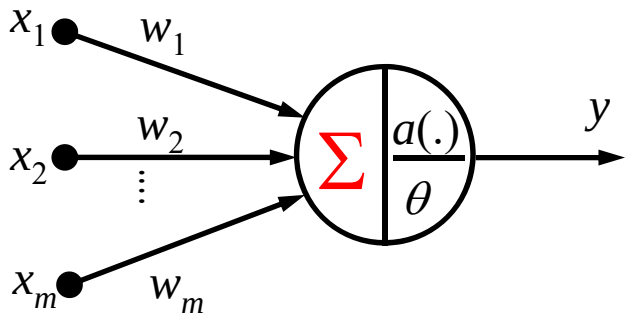
$\boldsymbol{x} = [x_1 \quad x_2 \quad \dots \quad x_m]^T$ vector tín hiệu vào tế bào thần kinh

$\boldsymbol{w} = [w_1 \quad w_2 \quad \dots \quad w_m]^T$ vector trọng số tế bào thần kinh

Hàm tích hợp ngõ vào tế bào thần kinh

- Hàm tuyến tính (linear function):

$$f = net = \left(\sum_{j=1}^{m} w_j x_j \right) - \theta = \boldsymbol{w}^T \boldsymbol{x} - \theta$$



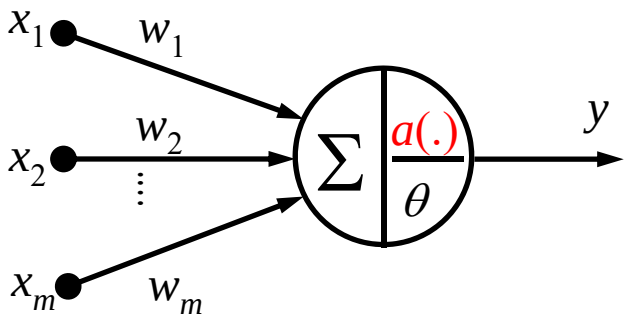
- Hàm toàn phương (quadratic function):

$$f = net = \left(\sum_{j=1}^{m} w_j x_j^2 \right) - \theta$$

- Hàm cầu (spherical function):

$$f = net = \left(\rho^{-2} \sum_{j=1}^{m} (x_j - w_j)^2 \right) - \theta = \rho^{-2} (\boldsymbol{x} - \boldsymbol{w})^T (\boldsymbol{x} - \boldsymbol{w}) - \theta$$

Hàm tác động



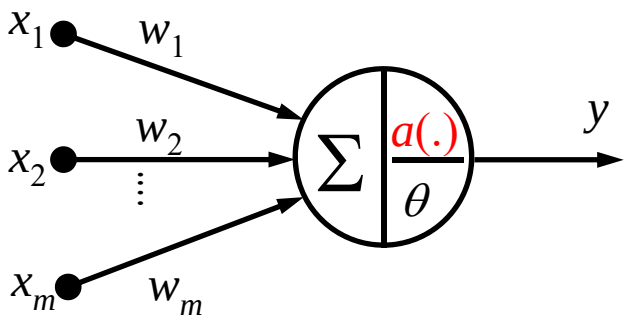
★ Hàm nấc (step):

$$a(f) = \begin{cases} 1 & if \quad f \geq 0 \\ 0 & if \quad f < 0 \end{cases}$$

★ Hàm dấu (sign):

$$a(f) = \begin{cases} 1 & if \quad f \geq 0 \\ -1 & if \quad f < 0 \end{cases}$$

Hàm tác động



★ Hàm dốc bão hòa:

$$a(f) = \begin{cases} 1 & \text{nếu } f > 1 \\ f & \text{nếu } 0 \le f \le 1 \\ 0 & \text{nếu } f < 0 \end{cases}$$

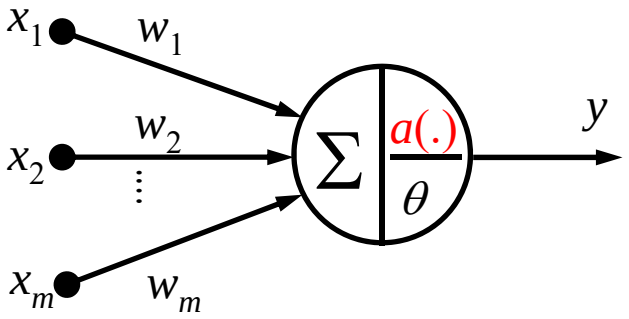
★ Hàm tuyến tính bão hòa:

$$a(f) = \begin{cases} 1 & \text{nếu } f > 1 \\ f & \text{nếu } 0 \le |f| \le 1 \\ -1 & \text{nếu } f < -1 \end{cases}$$

Hàm tác động

- Hàm tuyến tính: (purelin)

$$a(f) = f$$



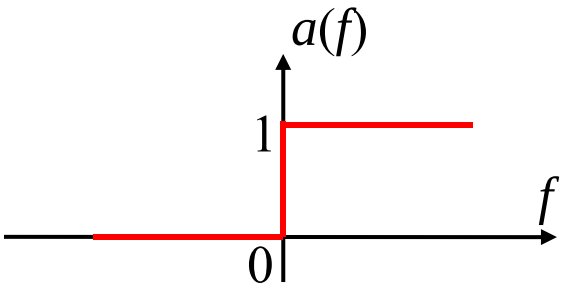
- Hàm dạng S đơn cực (logistic sigmoid function - logsig)

$$a(f) = \frac{1}{1+e^{-f}}$$

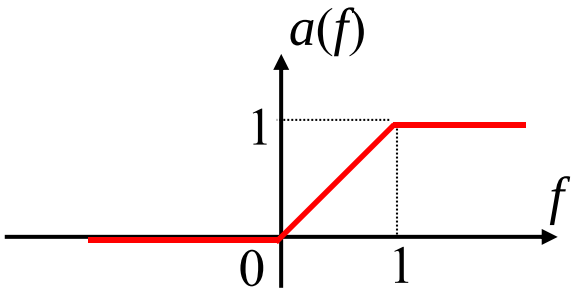
- Hàm dạng S lưỡng cực (hyperbolic tangent sigmoid function - tansig)

$$a(f) = \frac{1-e^{-2f}}{1+e^{-2f}}$$

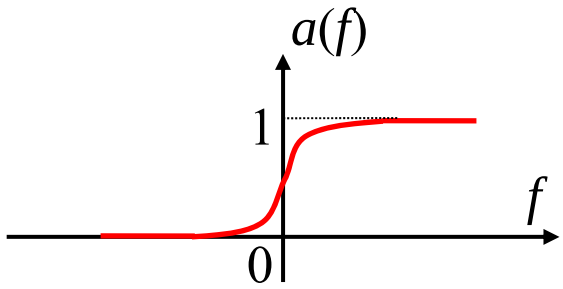
Các dạng hàm tác động



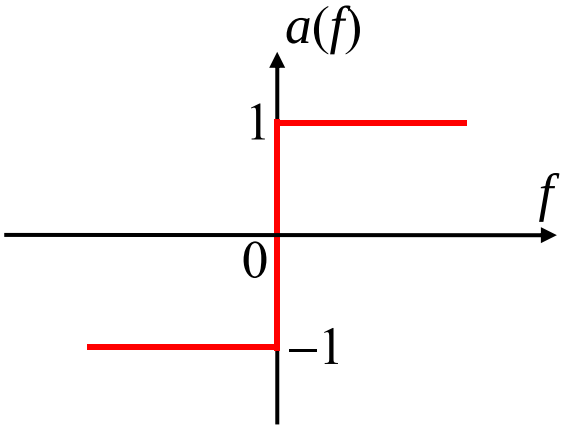
Hàm nấc



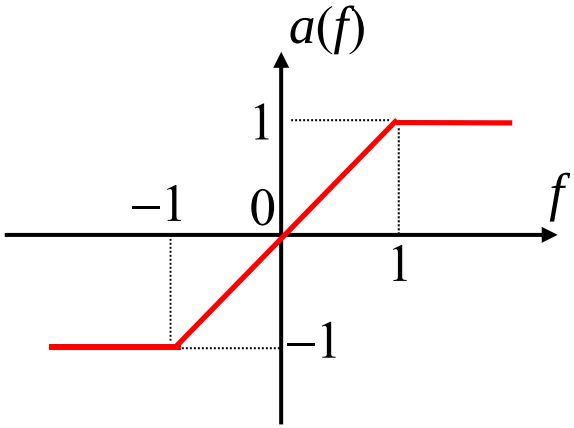
Hàm dốc bão hòa



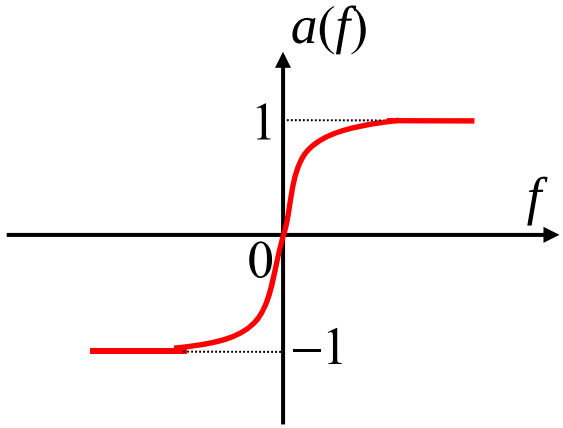
Hàm dạng S đơn cực



Hàm dấu

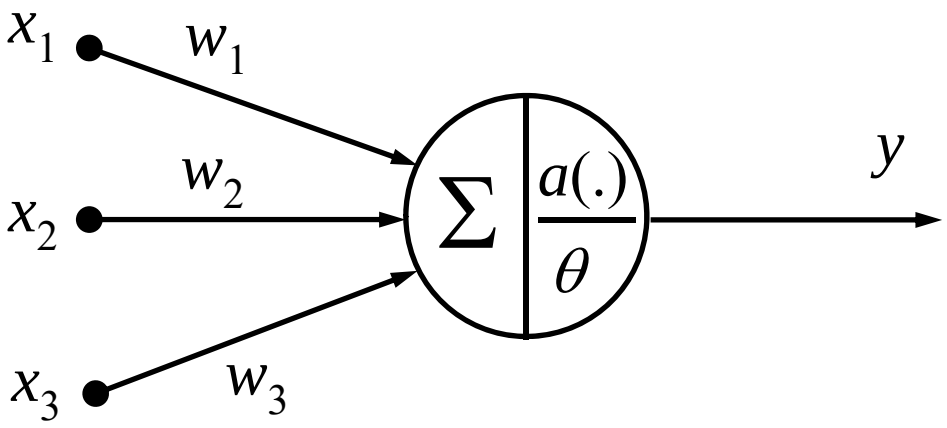


Hàm tuyến tính bão hòa



Hàm dạng S lưỡng cực

Ví dụ tính ngõ ra của tế bào thần kinh



- Tính ngõ ra của tế bào thần kinh, biết rằng hàm tổng ở ngõ vào là hàm tuyến tính, hàm tác động là sigmoid đơn cực:

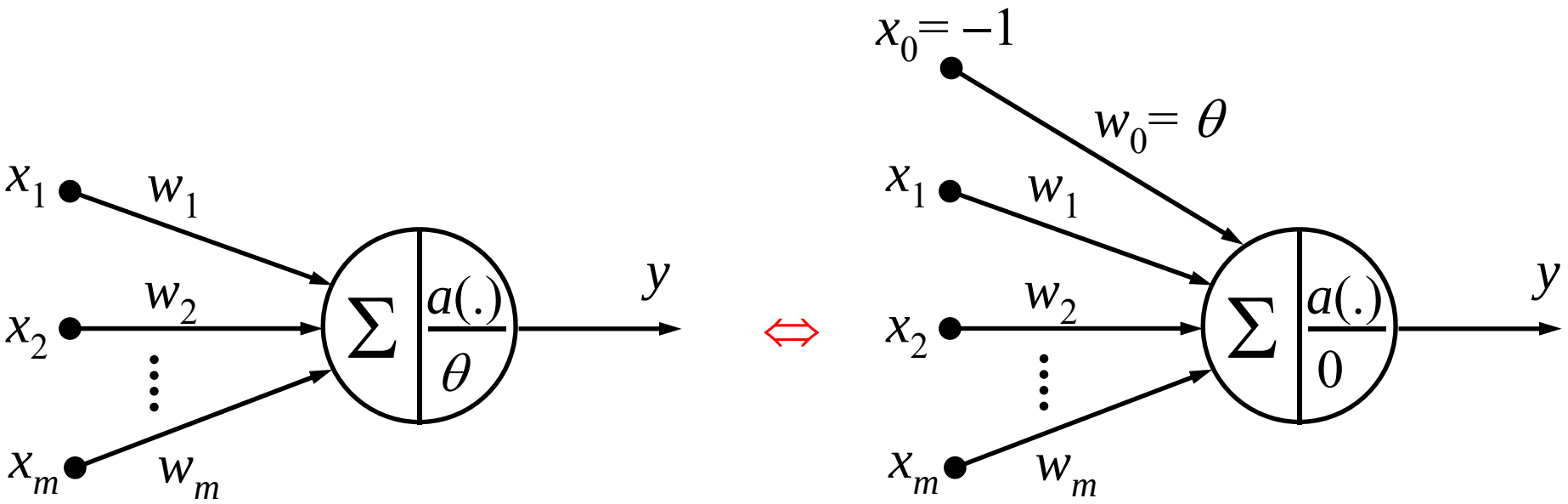
$$x_1 = 1; x_2 = 0.5; x_3 = -0.4;$$

$$w_1 = 0.3; w_2 = -0.8; w_3 = 0.4; \theta = 0.6$$

- **Giải:**

Chú ý

- Tế bào thần kinh có mức ngưỡng khác 0 có thể biến đổi tương đương thành tế bào thần kinh có mức ngưỡng bằng 0 như sau



$$net = w_1 x_1 + \ldots + w_m x_m - \theta$$

$$y = a(net)$$

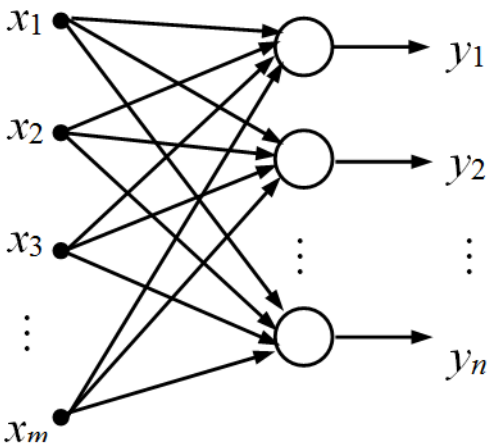
$$net = w_0 x_0 + w_1 x_1 + \ldots + w_m x_m$$

$$= -\theta + w_1 x_1 + \ldots + w_m x_m$$

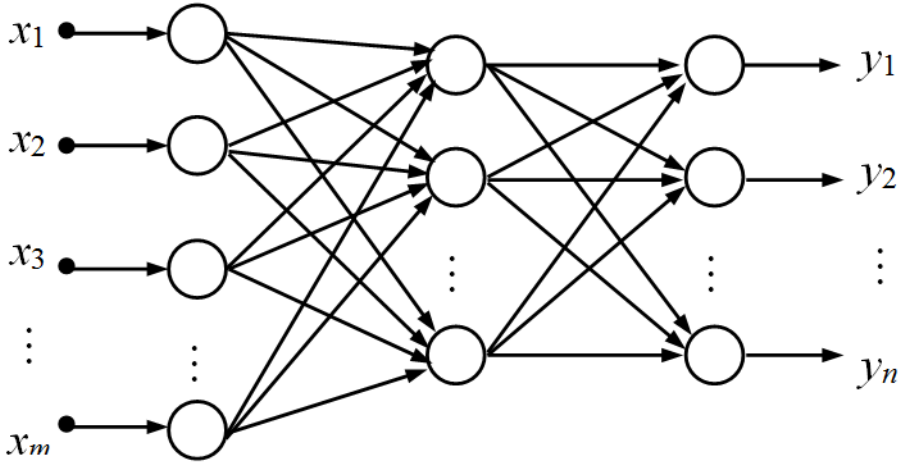
$$y = a(net)$$

Cấu trúc mạng và giải thuật huấn huyện mạng

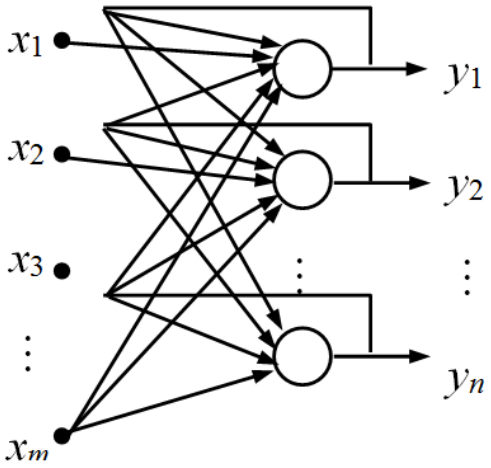
Cấu trúc mạng thần kinh



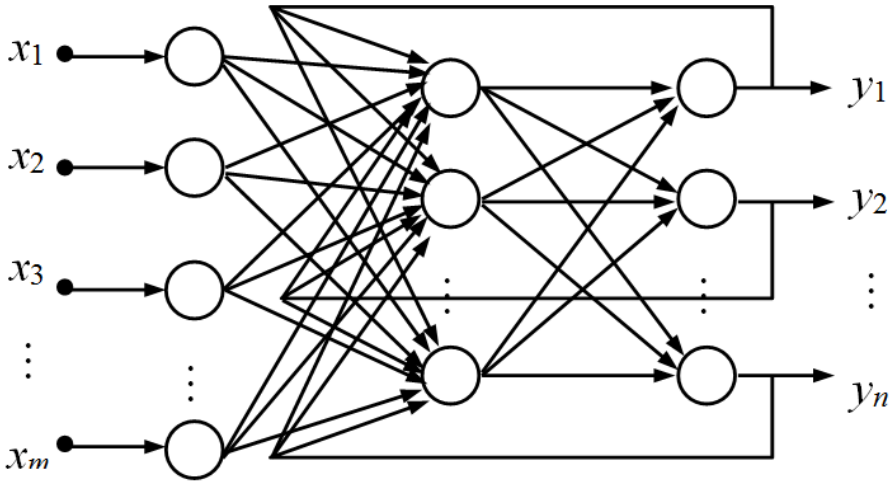
Mạng truyền thẳng 1 lớp



Mạng truyền thẳng nhiều lớp

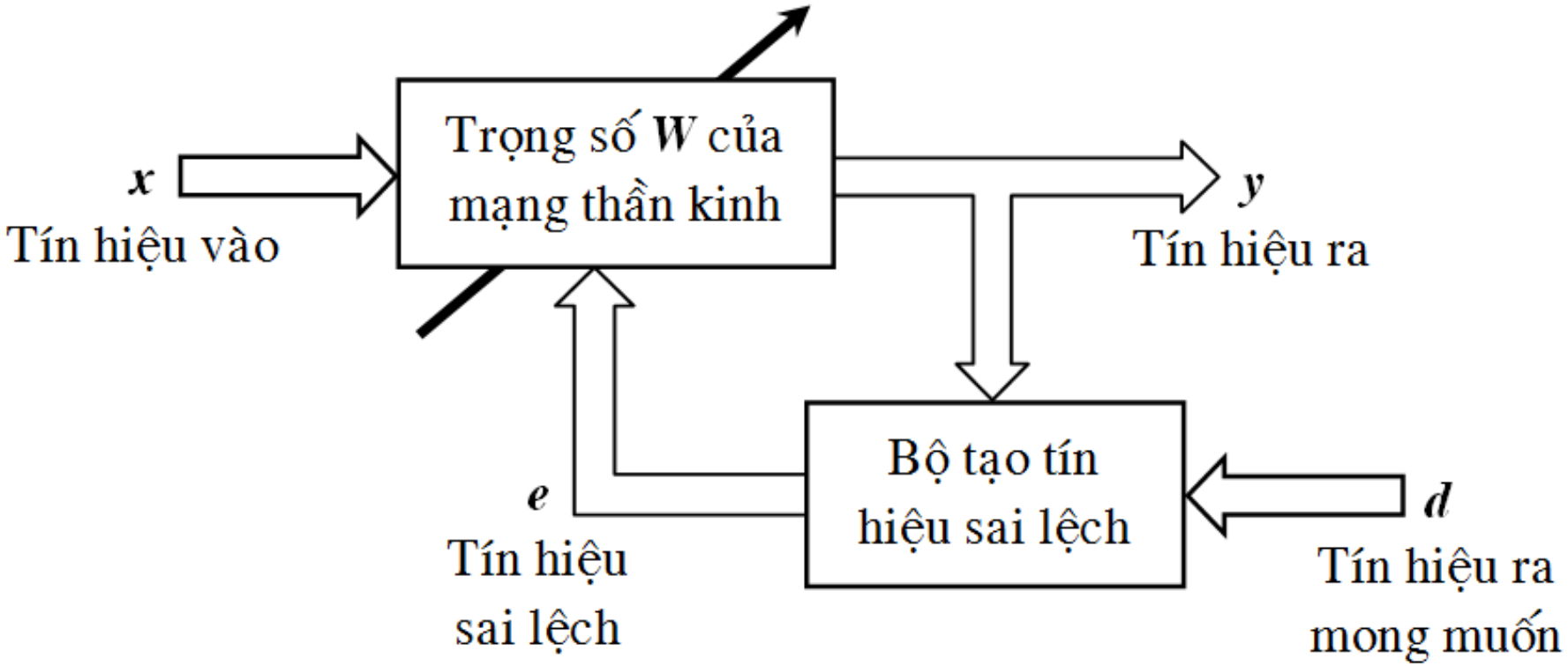


Mạng hồi qui lớp



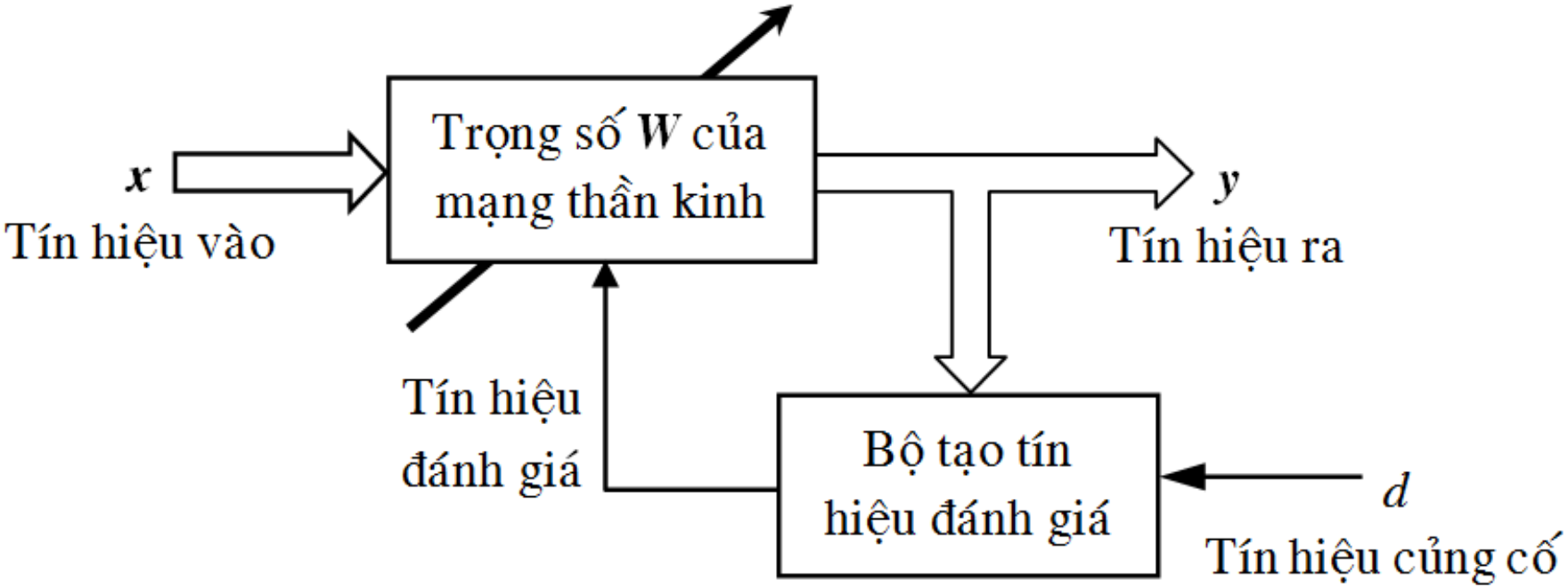
Mạng hồi qui nhiều lớp

Huấn luyện mạng dùng pp học có giám sát



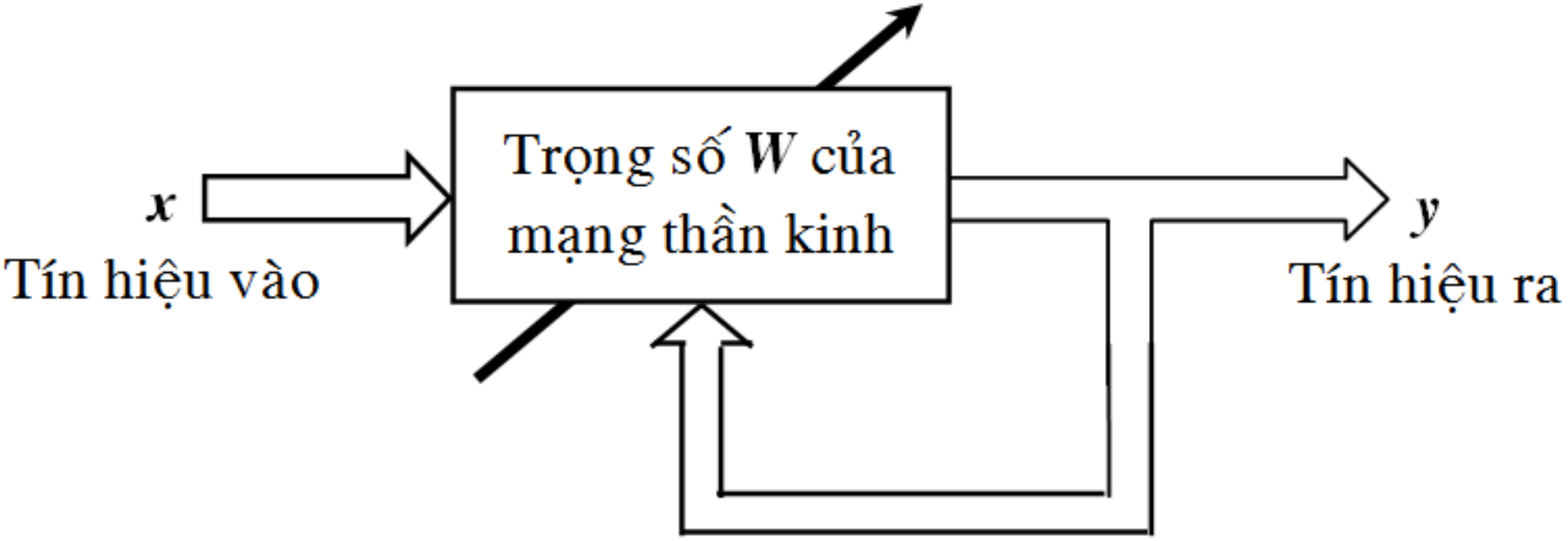
Học có giám sát

Huấn luyện mạng dùng pp học củng cố



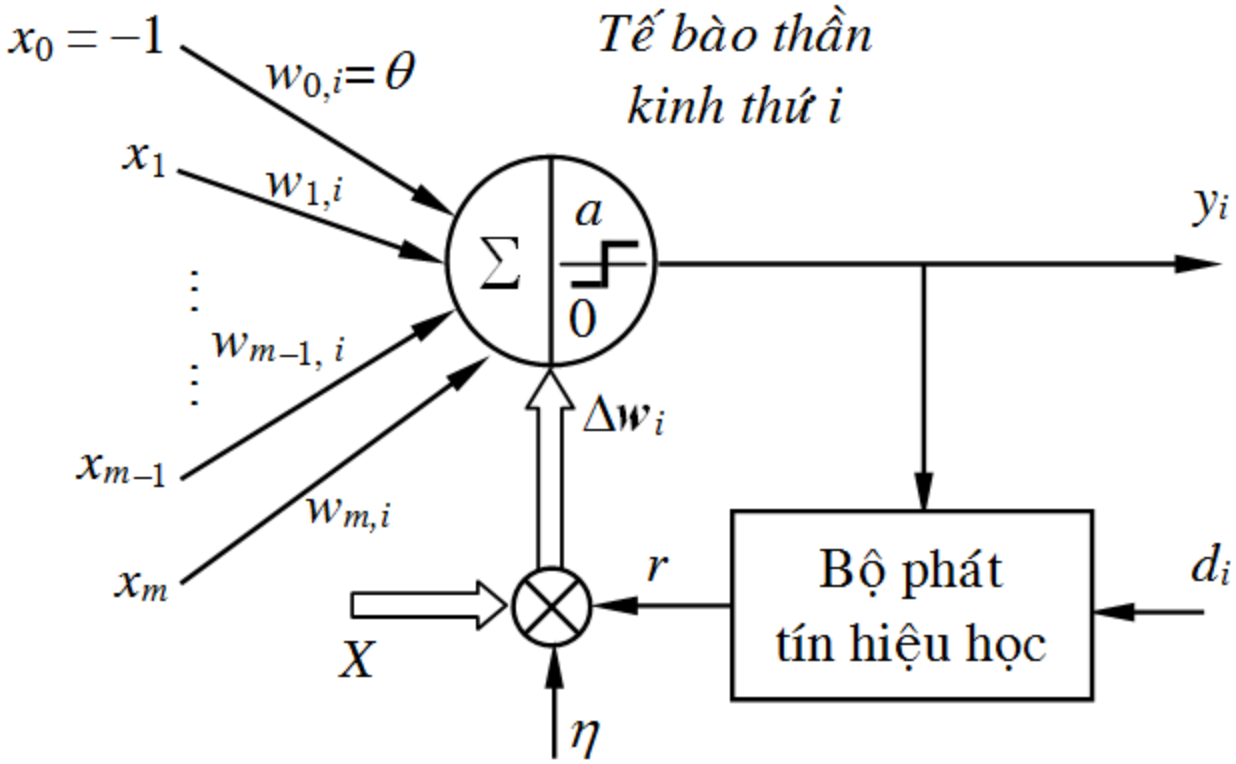
Học củng cố

Huấn luyện mạng dùng pp học không có giám sát



Học có không giám sát

Sơ đồ huấn luyện một tế bào thần kinh



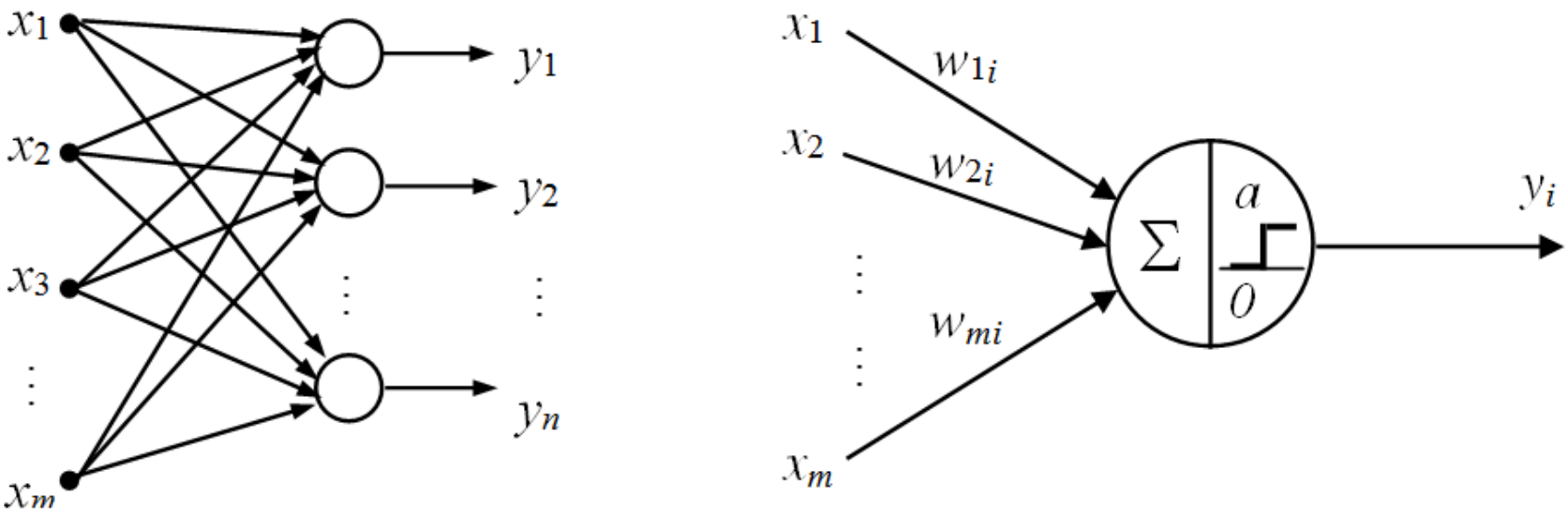
$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \Delta\boldsymbol{w}_i(k)$$

$$\Delta\boldsymbol{w}_i(k) = \eta r \boldsymbol{x}(k)$$

$$r = f_r\left(\boldsymbol{w}_i(k), \boldsymbol{x}(k), d_i\right)$$

Mạng Perceptron

Cấu trúc mạng perceptron



★ Mạng perceptron là mạng truyền thẳng 1 lớp, trong đó:

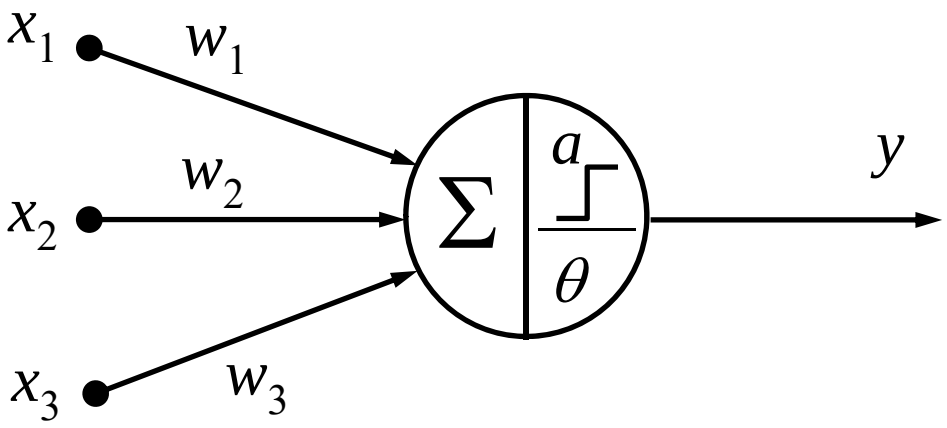
- Hàm xử lý ngõ vào là hàm tuyến tính

$$net_i = \sum_{j=1}^{m} w_{ji} x_j = \boldsymbol{w}_i^T \boldsymbol{x}$$

- Hàm kích hoạt ở ngõ ra là hàm nấc

$$y_i = a(net_i) = \text{step}(net_i) = \begin{cases} 1 & \text{nếu } net_i \geq 0 \\ 0 & \text{nếu } net_i < 0 \end{cases}$$

Ví dụ tính ngõ ra của Perceptron



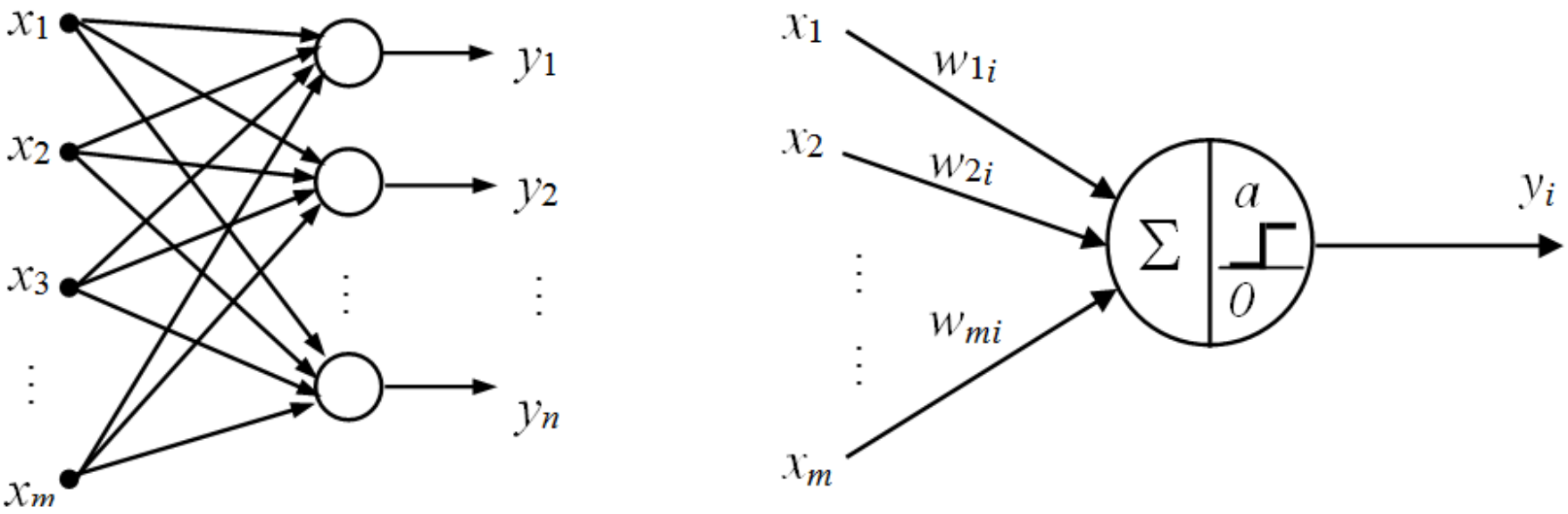
★ Tính ngõ ra của Perceptron ở trên, biết:

$$x_1 = 0.9; x_2 = 0.5; x_3 = -0.4;$$

$$w_1 = 0.5; w_2 = -0.8; w_3 = -0.6; \theta = 0.2$$

★ **Giải:**

Thuật toán học sửa sai huấn luyện perceptron



★ Thuật toán học sửa sai (Delta learning rule)

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \Delta\boldsymbol{w}_i(k)$$

$$\Delta\boldsymbol{w}_i(k) = \eta r \boldsymbol{x}(k)$$

$$r_i(k) = d_i(k) - y_i(k) = \delta_i(k)$$

$$\Rightarrow \quad \boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta(d_i(k) - y_i(k))\boldsymbol{x}(k)$$

Thuật toán học sửa sai huấn luyện perceptron

Giả sử tập dữ liệu huấn luyện gồm K mẫu $\{\boldsymbol{x}(k), \boldsymbol{d}(k)\}$ (k=1..K)

- ***Bước 1:*** Chọn η, gán k=1, E=0; khởi động ngẫu nhiên $\boldsymbol{w}_i(k)$
- ***Bước 2:*** Tính ngõ ra của mạng:

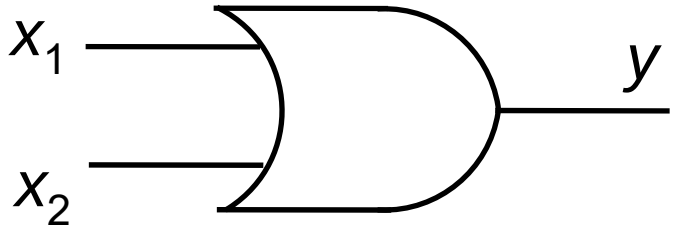
$$y_i(k) = \text{step}(\boldsymbol{w}_i^T(k)\boldsymbol{x}(k))$$

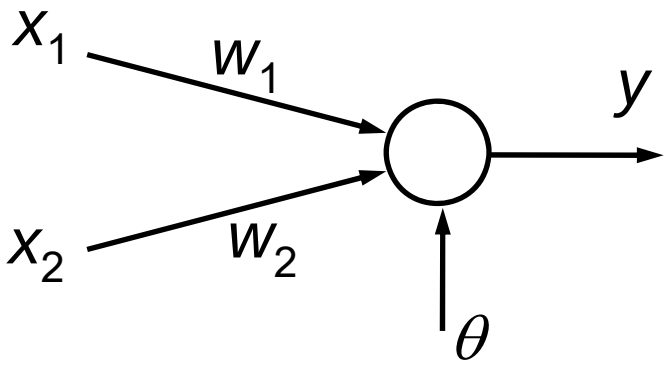
- ***Bước 3:*** Cập nhật trọng số:

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta(d_i(k) - y_i(k))\boldsymbol{x}(k)$$

- ***Bước 4:*** Tính sai số: $E = E + \frac{1}{2}\left\|\boldsymbol{d}(k) - \boldsymbol{y}(k)\right\|^2$
- ***Bước 5:*** Nếu $k<K$ thì gán $k = k+1$ và trở lại bước 2.
  Nếu $k=K$ thì tiếp tục bước 6
- ***Bước 6:*** Kết thúc chu kỳ huấn luyện. Nếu E=0 thì kết thúc. Nếu $E>0$ thì gán k =1, E =0 và trở lại bước 2

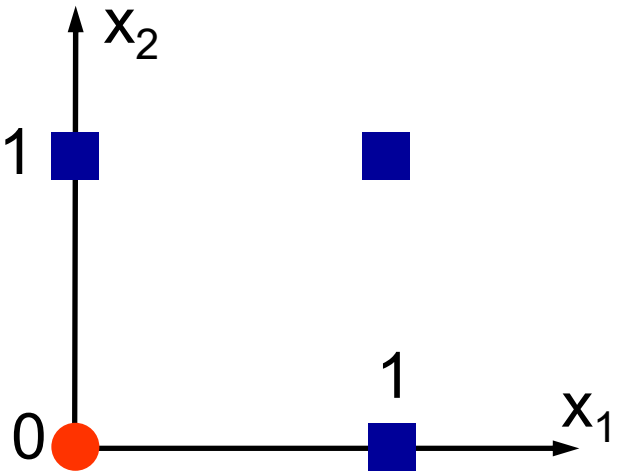
Ví dụ: Huấn luyện Perceptron thực hiện cổng OR





★ Dữ liệu huấn luyện Perceptron:

x_1	x_2	d
0	0	0
0	1	1
1	0	1
1	1	1



★ Tính trọng số của Perceptron sau 2 bước huấn luyện nếu:

$w_1(1) = 0.5$, $w_2(1) = -0.6$, $\theta(1) = -0.1$, hệ số học $\eta = 0.2$,

Ví dụ: Huấn luyện Perceptron thực hiện cổng OR

Tập dữ liệu huấn luyện Perceptron gồm 4 mẫu (K=4)

$$\boldsymbol{X} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix}$$

***k*=1:**

$$\boldsymbol{x}(1) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{w}(1) = \begin{bmatrix} \theta(1) \\ w_1(1) \\ w_2(1) \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.5 \\ -0.6 \end{bmatrix}$$

$$y(1) = \text{step}(\boldsymbol{w}^T(1)\boldsymbol{x}(1)) = \text{step}(0.1) = 1$$

$$\boldsymbol{w}(2) = \boldsymbol{w}(1) + \eta(d(1) - y(1))\boldsymbol{x}(1) = \begin{bmatrix} -0.1 \\ 0.5 \\ -0.6 \end{bmatrix} + 0.2 \times (0-1) \times \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.5 \\ -0.6 \end{bmatrix}$$

$$\boldsymbol{E} = \boldsymbol{E} + 0.5 \times (d(1) - y(1))^2 = 0.5$$

Ví dụ: Huấn luyện Perceptron thực hiện cổng OR

k=2:

$$\boldsymbol{x}(2)=\begin{bmatrix}-1\\0\\1\end{bmatrix} \qquad \boldsymbol{w}(2)=\begin{bmatrix}\theta(2)\\w_1(2)\\w_2(2)\end{bmatrix}=\begin{bmatrix}0.1\\0.5\\-0.6\end{bmatrix}$$

$$y(2)=\text{step}(\boldsymbol{w}^T(2)\boldsymbol{x}(2))=\text{step}(-0.7)=0$$

$$\boldsymbol{w}(3)=\boldsymbol{w}(2)+\eta(d(2)-y(2))\boldsymbol{x}(2)=\begin{bmatrix}0.1\\0.5\\-0.6\end{bmatrix}+0.2\times(1-0)\times\begin{bmatrix}-1\\0\\1\end{bmatrix}=\begin{bmatrix}-0.1\\0.5\\-0.4\end{bmatrix}$$

$$\boldsymbol{E}=\boldsymbol{E}+0.5\times(d(2)-y(2))^2=1.0$$

k=3:

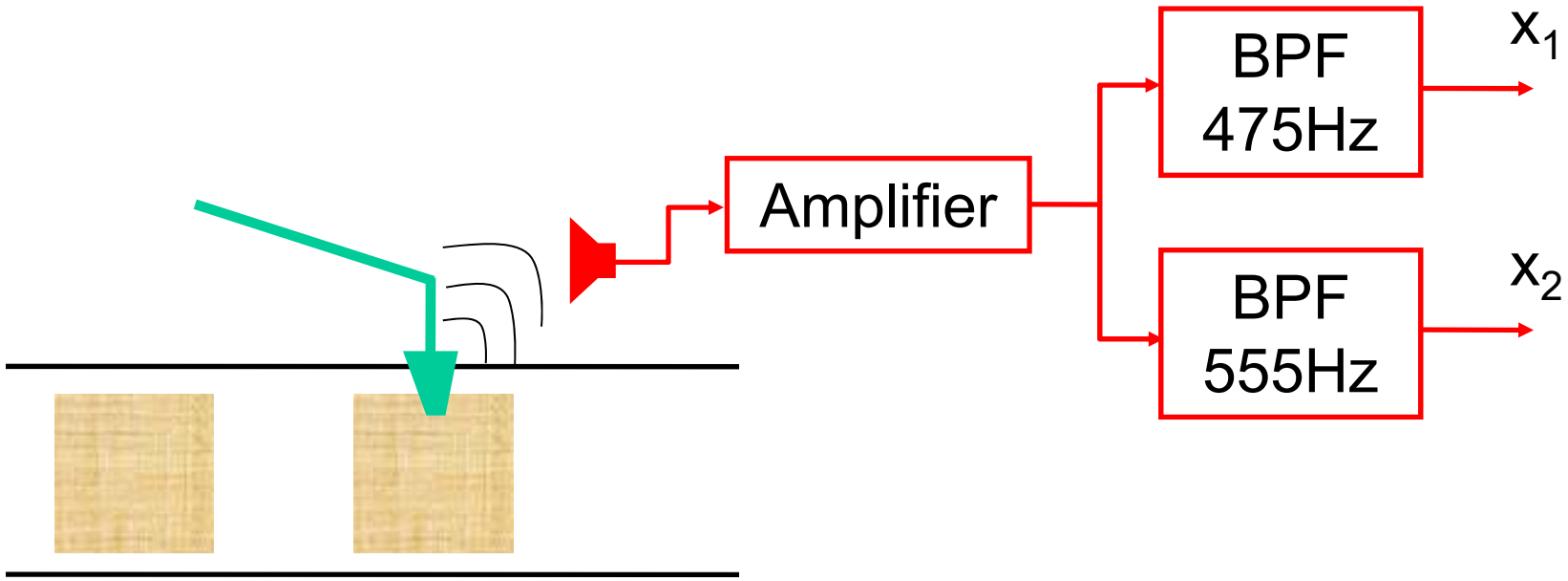
…

…

$$\Rightarrow \quad \boldsymbol{w}^*=\begin{bmatrix}0.1\\0.5\\0.2\end{bmatrix}$$

Thí dụ ứng dụng: Phát hiện viên gạch bị lỗi

✶ Sơ đồ thu thập dữ liệu:

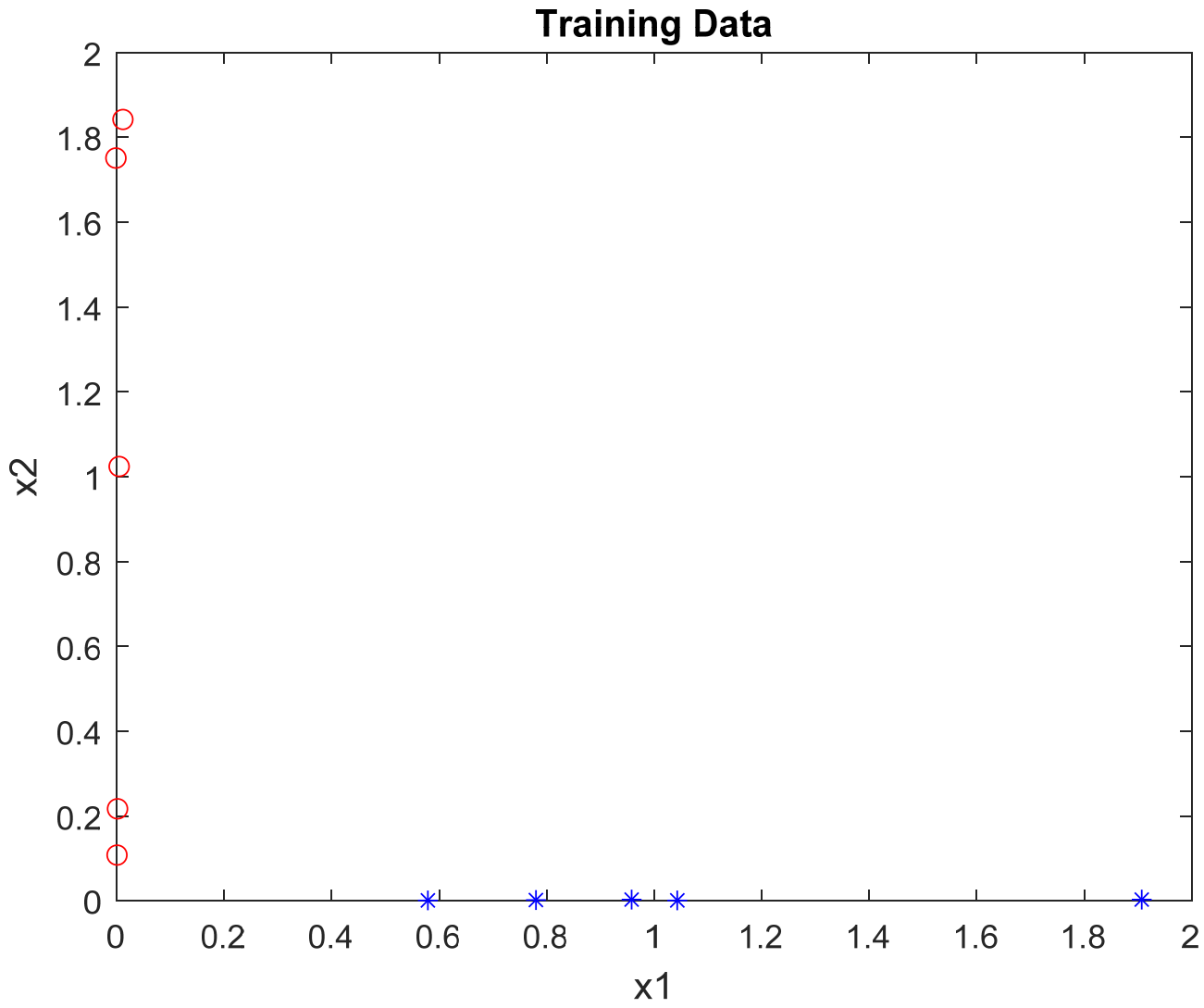


Thí dụ ứng dụng: Phát hiện viên gạch bị lỗi

★ Dữ liệu huấn luyện mạng:

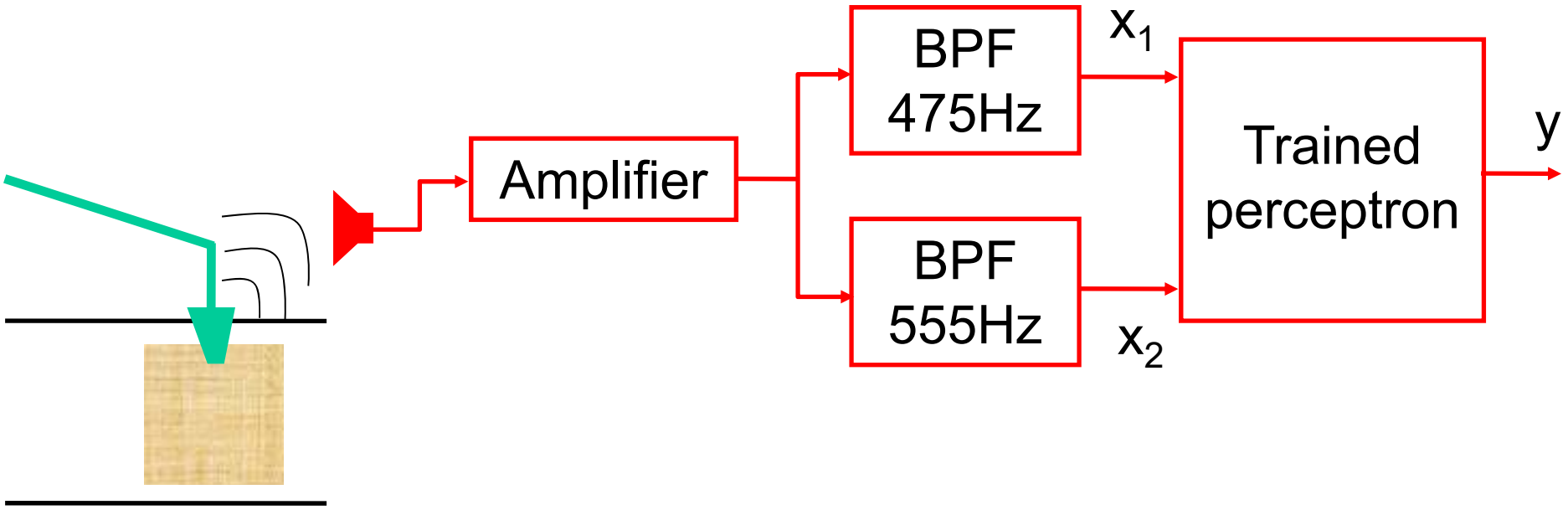
Biên độ tín hiệu 475Hz	Biên độ tín hiệu 555 Hz	Chất lượng viên gạch
0.958	0.003	Đạt
1.043	0.001	Đạt
1.907	0.003	Đạt
0.780	0.002	Đạt
0.579	0.001	Đạt
0.003	0.105	Không đạt
0.001	1.748	Không đạt
0.014	1.839	Không đạt
0.007	1.021	Không đạt
0.004	0.214	Không đạt

Thí dụ ứng dụng: Phát hiện viên gạch bị lỗi



Thí dụ ứng dụng: Phát hiện viên gạch bị lỗi

✶ Sơ đồ dùng Perceptron phát hiện viên gạch bị lỗi:



Khái niệm khả phân tuyến tính

- Xét trường hợp mạng chỉ có một Perceptron. Gọi tập hợp mẫu vector ngõ vào huấn luyện Perceptron là:

$$S = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \cdots, \boldsymbol{x}(K)\}$$

  - S_0 : tập hợp các vector vào có ngõ ra mong muốn là 0

$$S_0 = \{\boldsymbol{x}(k) \in S | d(k) = 0\}$$

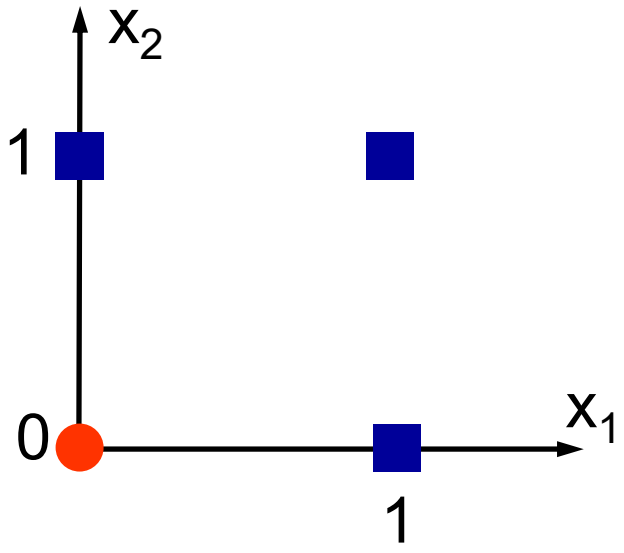
  - S_1 : tập hợp các vector vào có ngõ ra mong muốn là 1

$$S_1 = \{\boldsymbol{x}(k) \in S | d(k) = 1\}$$

- Nếu tồn tại một đường thẳng (trường hợp mạng có 2 ngõ vào), một mặt phẳng (trường hợp mạng có 3 ngõ vào) hay một mặt siêu phẳng (hyper-plane, trường hợp mạng có nhiều hơn 3 ngõ vào) phân chia không gian vector ngõ vào thành 2 miền riêng biệt: một miền chứa S_0, một miền chứa S_1 thì bài toán được gọi là khả phân tuyến tính.

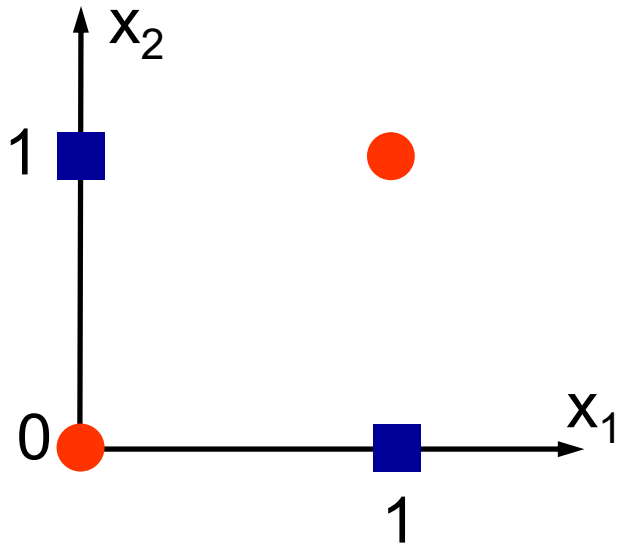
Ví dụ

x1	x2	d
0	0	0
0	1	1
1	0	1
1	1	1



Khả phân tuyến tính

x1	x2	d
0	0	0
0	1	1
1	0	1
1	1	0



Không khả phân tuyến tính

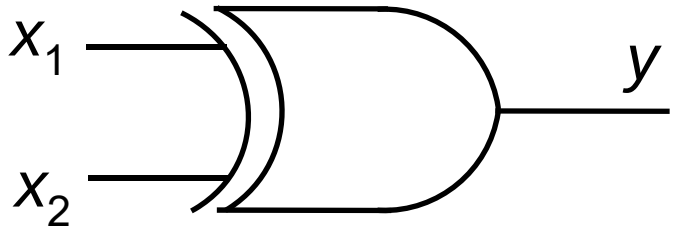
Tính hội tụ của giải thuật huấn luyện Perceptron

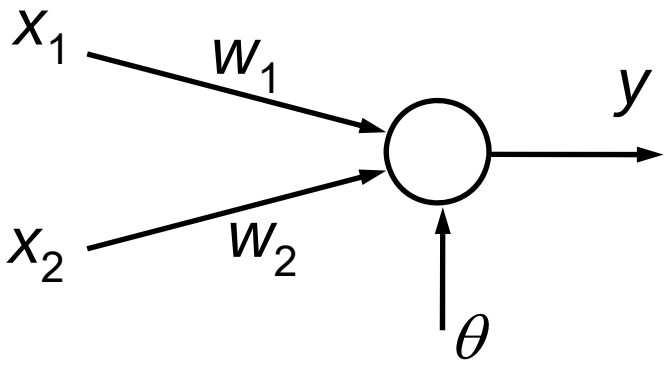
- Nếu tập dữ liệu huấn luyện Perceptron khả phân tuyến tính thì giải thuật huấn luyện Perceptron hội tụ.

- ***Nhận xét:***
  - Nếu bài toán phân nhóm khả phân tuyến tính thì có thể sử dụng 1 Perceptron
  - Nếu bài toán phân nhóm không khả phân tuyến tính, cần sử dụng mạng Perceptron nhiều lớp

Ví dụ: Huấn luyện Perceptron thực hiện cổng XOR

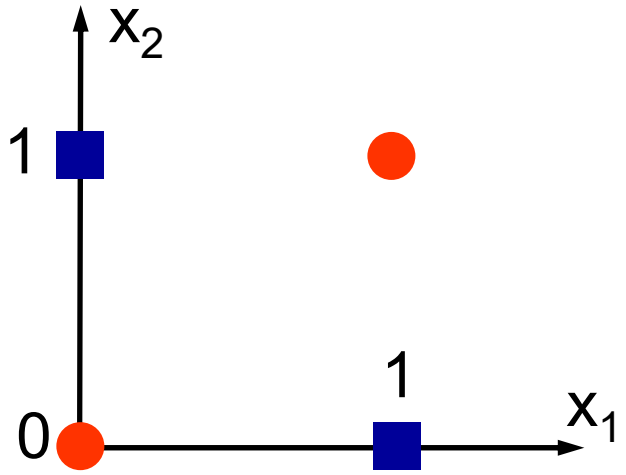
- Thiết kế mạng Perceptron và trình bày cách huấn luyện mạng để thực hiện chức năng cổng XOR:





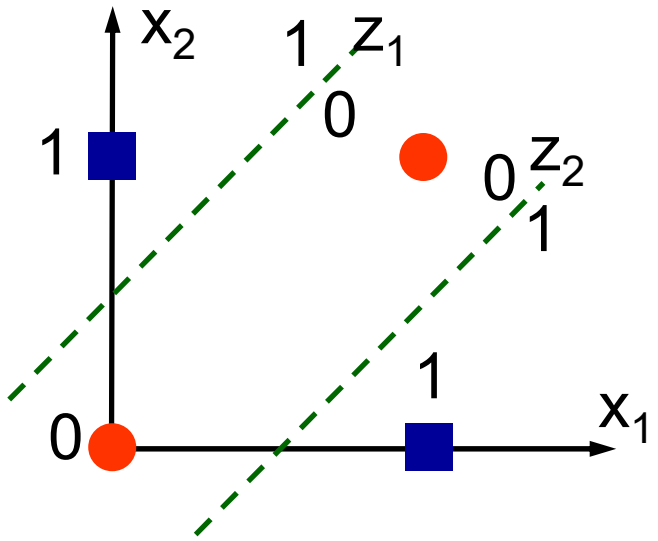
- Dữ liệu huấn luyện Perceptron:

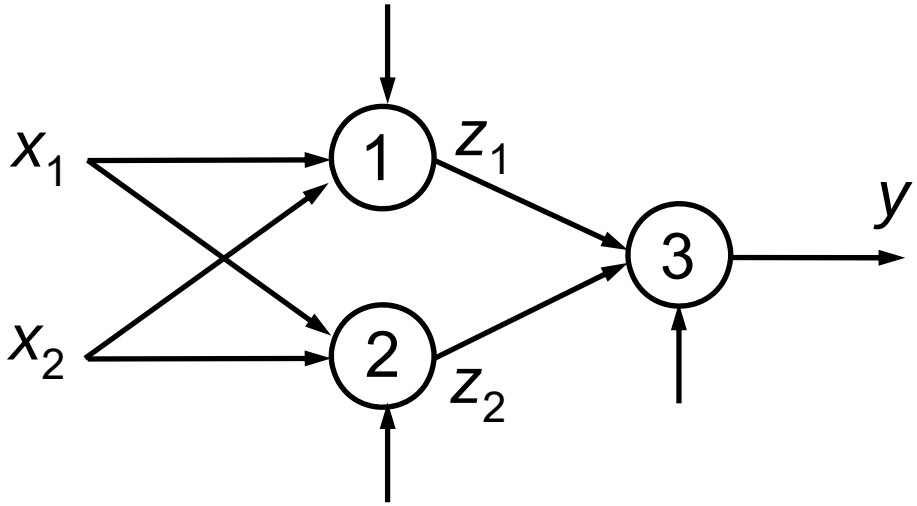
x_1	x_2	d
0	0	0
0	1	1
1	0	1
1	1	0



Ví dụ: Huấn luyện Perceptron thực hiện cổng XOR

★ Thiết kế cấu trúc mạng Perceptron:

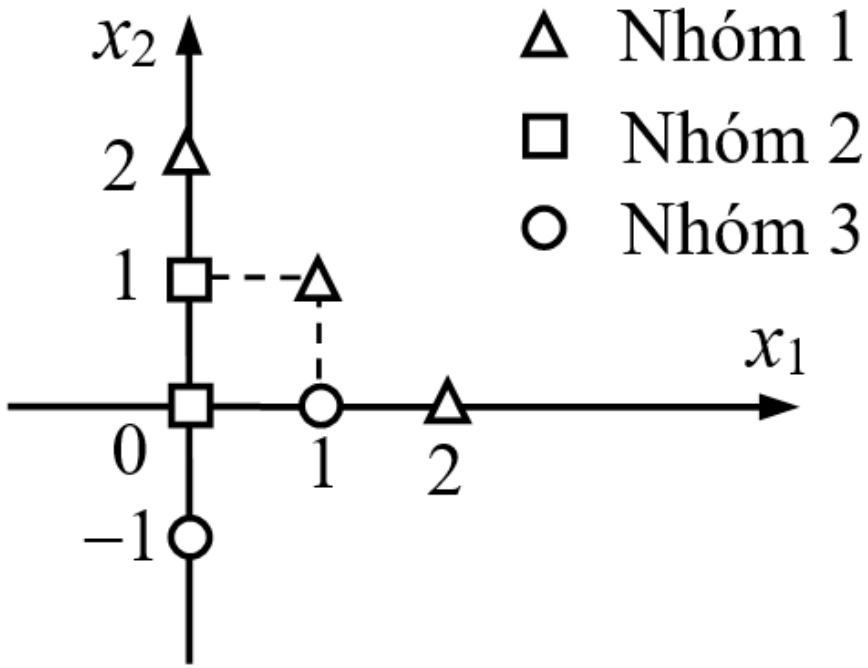




x_1	x_2	z_1	z_2	d
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

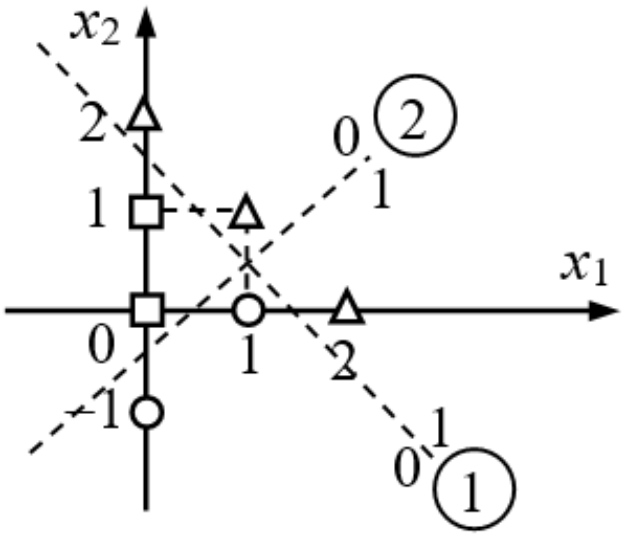
Ví dụ phân nhóm dữ liệu không khả phân tuyến tính

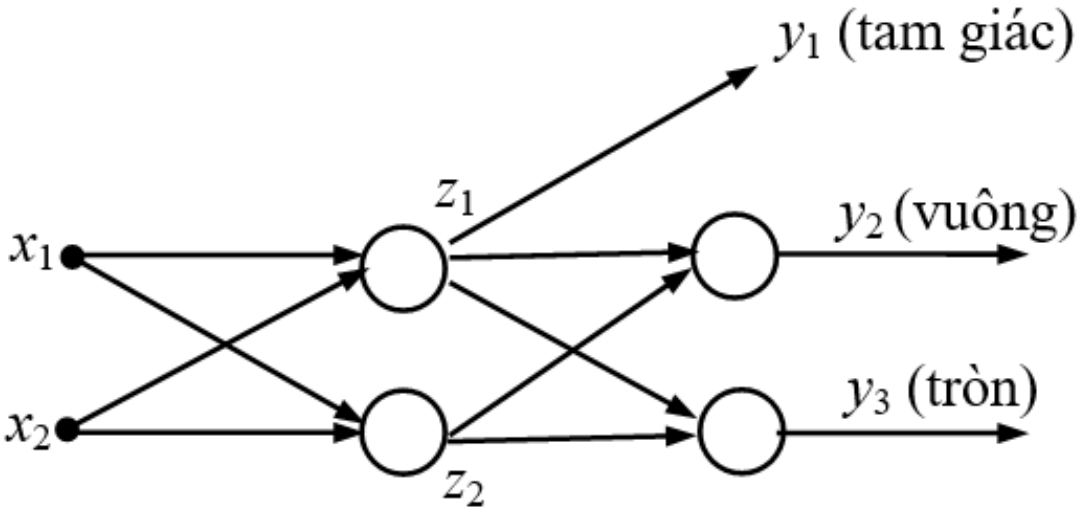
- Cho tập dữ liệu gồm 3 nhóm biểu diễn trên đồ thị ở hình bên. Hãy trình bày cấu trúc mạng và cách huấn luyện mạng Perceptron (nêu rõ tập dữ liệu huấn luyện từng Perceptron trong mạng) để phân tập dữ liệu thành 3 nhóm.



Lời giải

★ Cấu trúc mạng Perceptron



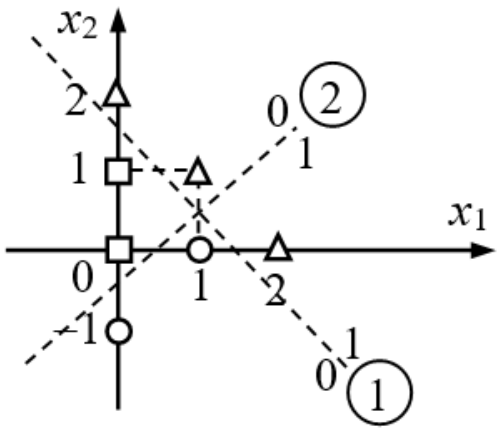


- Dữ liệu thuộc nhóm 1 (tam giác) nếu ngõ ra Perceptron z_1 bằng 1
- Dữ liệu thuộc nhóm 2 (vuông) nếu ngõ ra Perceptron z_1 bằng 0, và z_2 bằng 0
- Dữ liệu thuộc nhóm 3 nếu Perceptron z_1 bằng 0 và z_2 bằng 01

Lời giải

★ Quan hệ vào – ra của các Perceptron

x_1	x_2	z_1	z_2	y_1	y_2	y_3
2	0	1	1	1	0	0
1	1	1	0	1	0	0
0	2	1	0	1	0	0
0	0	0	0	0	1	0
0	1	0	0	0	1	0
1	0	0	1	0	0	1
0	−1	0	1	0	0	1



Lời giải

★ Ma trận dữ liệu huấn luyện z_1 và z_2

$$X_1 = X_2 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 & -1 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

★ Ma trận dữ liệu huấn luyện y_2 và y_3

$$X_3 = X_4 = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

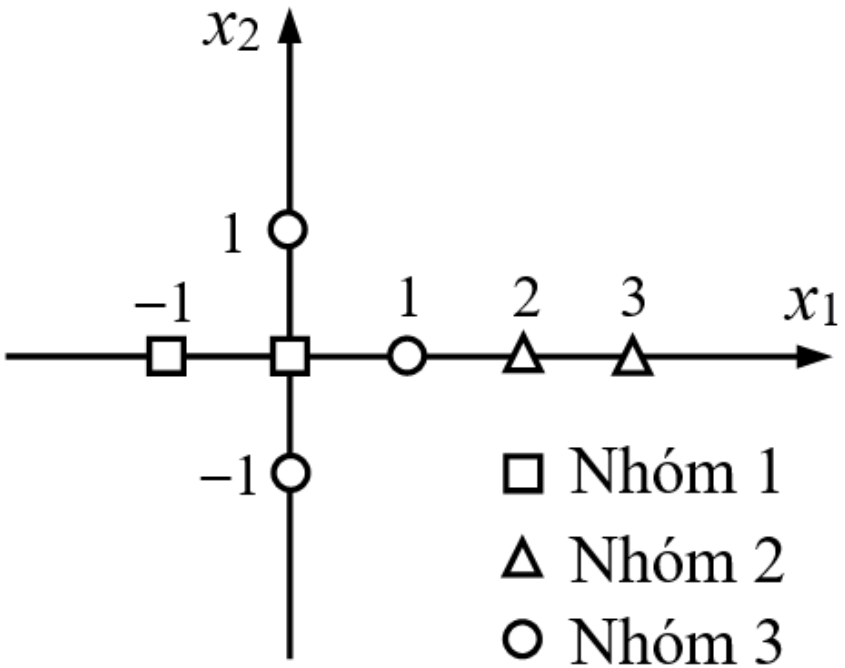
$$D_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

x_1	x_2	z_1	z_2	y_1	y_2	y_3
2	0	1	1	1	0	0
1	1	1	0	1	0	0
0	2	1	0	1	0	0
0	0	0	0	0	1	0
0	1	0	0	0	1	0
1	0	0	1	0	0	1
0	−1	0	1	0	0	1

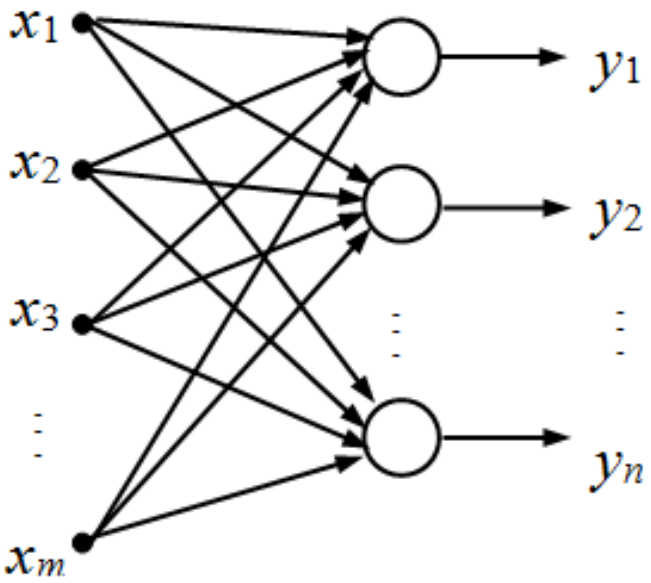
Bài tập

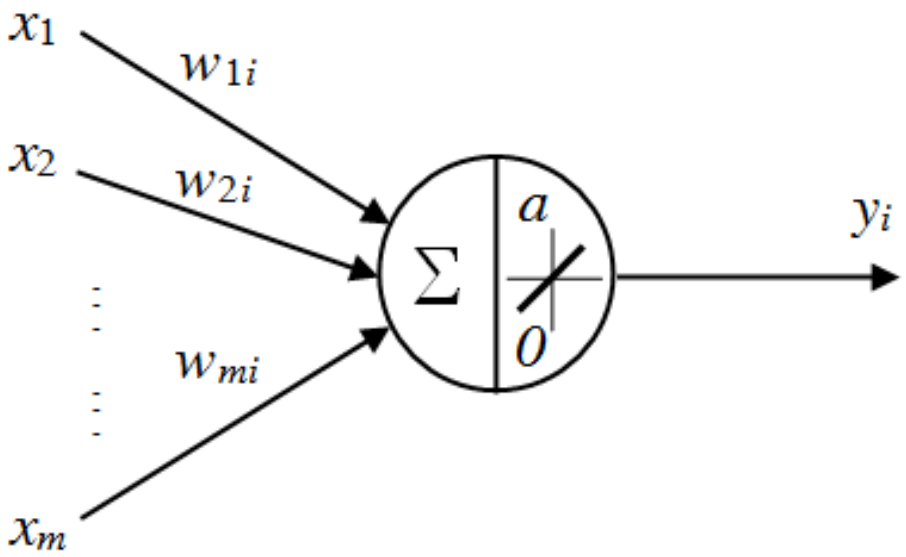
- Cho tập dữ liệu gồm 3 nhóm biểu diễn trên đồ thị ở hình bên. Hãy trình bày cấu trúc mạng và cách huấn luyện mạng Perceptron (nêu rõ tập dữ liệu huấn luyện từng Perceptron trong mạng) để phân tập dữ liệu thành 3 nhóm.



Adaline và mạng tuyến tính

Adaline và mạng tuyến tính

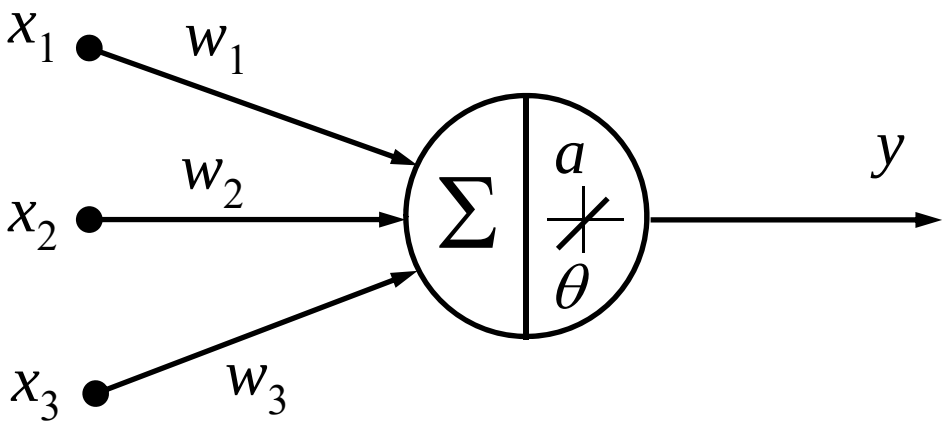




- Mạng tuyến tính (Linear Network) là mạng truyền thẳng một lớp gồm các phần tử thích nghi tuyến tính Adaline (Adaptive Linear Element).
- Ngõ ra của Adaline thứ *i* là: $y_i = net_i$

  với $$net_i = \sum_{j=1}^{m} w_{ji} x_j = \mathbf{w}_i^T \mathbf{x}$$

Ví dụ tính ngõ ra của Adaline



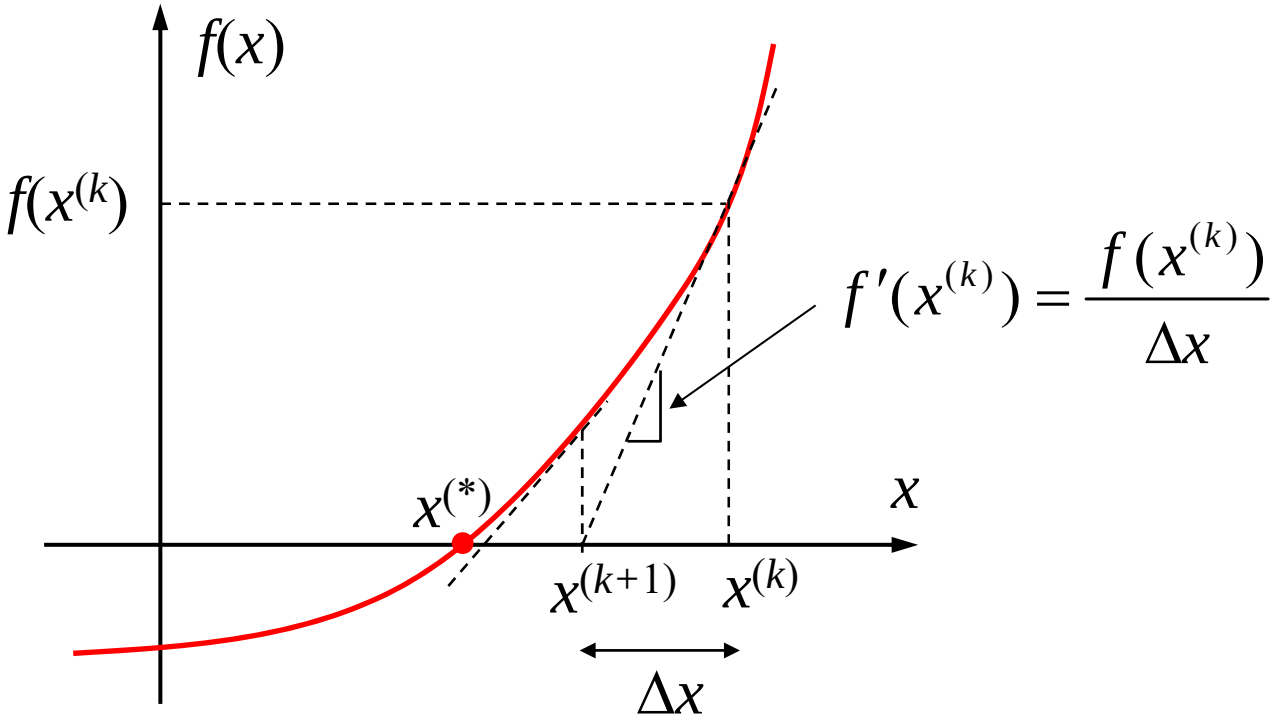
- Tính ngõ ra của Adaline, biết:

$$x_1 = -1; x_2 = -2; x_3 = 4;$$

$$w_1 = -0.5; w_2 = 0.7; w_3 = 0.3; \theta = 0.2$$

- **Giải:**

Nhắc lại giải thuật lặp Newton tìm nghiệm phương trình



$$x^{(k+1)} = x^{(k)} - \Delta x$$

$$\Delta x = \left[f'(x^{(k)}) \right]^{-1} f(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} - \left[f'(x^{(k)}) \right]^{-1} f(x^{(k)})$$

Huấn luyện mạng Adaline

★ Tập dữ liệu huấn luyện mạng gồm *K* mẫu:

$$\{(\boldsymbol{x}(1),\boldsymbol{d}(1));(\boldsymbol{x}(2),\boldsymbol{d}(2);\ldots;(\boldsymbol{x}(K),\boldsymbol{d}(K))\}$$

Trong đó: $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^T \quad \boldsymbol{d} = [d_1, d_2, \ldots, d_n]^T$

★ Hàm mục tiêu : $E(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{n}(d_i - y_i)^2 \rightarrow \min$

★ Thuật toán suy giảm độ dốc: $\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) - \eta\frac{\partial E(k)}{\partial \boldsymbol{w}_i}$

Do $\frac{\partial E(k)}{\partial \boldsymbol{w}_i} = \left[\frac{\partial E(k)}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_i}\right]\left[\frac{\partial net_i}{\partial \boldsymbol{w}_i}\right] = -[d_i(k) - y_i(k)].1.\boldsymbol{x}(k)$

$\Rightarrow$ $\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta(d_i(k) - y_i(k))\boldsymbol{x}(k)$

Thuật toán Widrow-Hoff huấn luyện Adaline

- ***Bước 1:*** Chọn η; gán k=1, E=0; khởi động ngẫu nhiên $\boldsymbol{w}_i(k)$
- ***Bước 2:*** Tính ngõ ra của mạng:

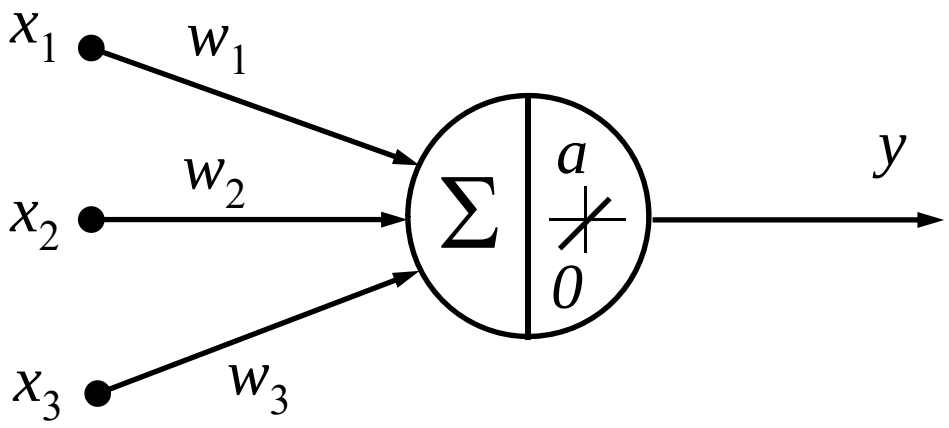
$$y_i(k) = \boldsymbol{w}_i^T(k)\boldsymbol{x}(k)$$

- ***Bước 3:*** Cập nhật trọng số:

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta(d_i(k) - y_i(k))\boldsymbol{x}(k)$$

- ***Bước 4:*** Tính sai số tích lũy $E = E + \frac{1}{2}\left\|\boldsymbol{d}(k) - \boldsymbol{y}(k)\right\|^2$
- ***Bước 5:*** Nếu $k<K$ thì gán $k = k+1$ và trở lại bước 2.
  Nếu $k=K$ thì tiếp tục bước 6
- ***Bước 6:*** Kết thúc chu kỳ huấn luyện. Nếu $E \le \varepsilon$ thì kết thúc. Nếu $E > \varepsilon$ thì gán gán k =1, E =0 và trở lại bước 2

Ví dụ huấn luyện Adaline



- Cho tập dữ liệu huấn luyện Adaline:

$$X = \begin{bmatrix} 0.3 & 0.5 & 0.6 \\ 0.7 & 0.1 & 0.2 \\ 1.0 & 0.8 & 0.3 \end{bmatrix} \qquad D = \begin{bmatrix} 0.5 & 0.2 & 0.7 \end{bmatrix}$$

- Tính trọng số Adaline sau 1 chu kỳ huấn luyện nếu η = 0.3 và trọng số ban đầu của Adaline là:

$$w_1(1) = -0.5; w_2(1) = 0.7; w_3(1) = 0.3$$

Ví dụ huấn luyện Adaline (tt)

Tập dữ liệu huấn luyện Adaline gồm 3 mẫu (K=3)

$$X = \begin{bmatrix} 0.3 & 0.5 & 0.6 \\ 0.7 & 0.1 & 0.2 \\ 1.0 & 0.8 & 0.3 \end{bmatrix} \qquad D = \begin{bmatrix} 0.5 & 0.2 & 0.7 \end{bmatrix}$$

***k*=1:**

$$\boldsymbol{x}(1) = \begin{bmatrix} 0.3 \\ 0.7 \\ 1.0 \end{bmatrix} \qquad \boldsymbol{w}(1) = \begin{bmatrix} w_1(1) \\ w_2(1) \\ w_3(1) \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.7 \\ 0.3 \end{bmatrix}$$

$$y(1) = \boldsymbol{w}^T(1)\boldsymbol{x}(1) = 0.64$$

$$\boldsymbol{w}(2) = \boldsymbol{w}(1) + \eta(d(1) - y(1))\boldsymbol{x}(1) = \begin{bmatrix} -0.5 \\ 0.7 \\ 0.3 \end{bmatrix} + 0.3 \times (0.5 - 0.64) \times \begin{bmatrix} 0.3 \\ 0.7 \\ 1.0 \end{bmatrix} = \begin{bmatrix} -0.513 \\ 0.671 \\ 0.258 \end{bmatrix}$$

$$\boldsymbol{E} = \boldsymbol{E} + 0.5 \times (d(1) - y(1))^2 = 0.0098$$

Ví dụ huấn luyện Adaline (tt)

***k*=2:**

$$\boldsymbol{x}(2)=\begin{bmatrix}0.5\\0.1\\0.8\end{bmatrix} \qquad \boldsymbol{w}(2)=\begin{bmatrix}w_1(2)\\w_2(2)\\w_3(2)\end{bmatrix}=\begin{bmatrix}-0.513\\0.671\\0.258\end{bmatrix}$$

$$y(2)=\boldsymbol{w}^T(2)\boldsymbol{x}(2)=0.017$$

$$\boldsymbol{w}(3)=\boldsymbol{w}(2)+\eta(d(2)-y(2))\boldsymbol{x}(2)=\begin{bmatrix}-0.513\\0.671\\0.258\end{bmatrix}+0.3\times(0.2-0.017)\times\begin{bmatrix}0.5\\0.1\\0.8\end{bmatrix}=\begin{bmatrix}-0.486\\0.677\\0.302\end{bmatrix}$$

$$\boldsymbol{E}=\boldsymbol{E}+0.5\times(d(2)-y(2))^2=0.0265$$

Ví dụ huấn luyện Adaline (tt)

***k*=3:**

$$\boldsymbol{x}(3)=\begin{bmatrix}0.6\\0.2\\0.3\end{bmatrix} \qquad \boldsymbol{w}(3)=\begin{bmatrix}w_1(3)\\w_2(3)\\w_3(3)\end{bmatrix}=\begin{bmatrix}-0.486\\0.677\\0.302\end{bmatrix}$$

$$y(3)=\boldsymbol{w}^T(3)\boldsymbol{x}(3)=-0.0656$$

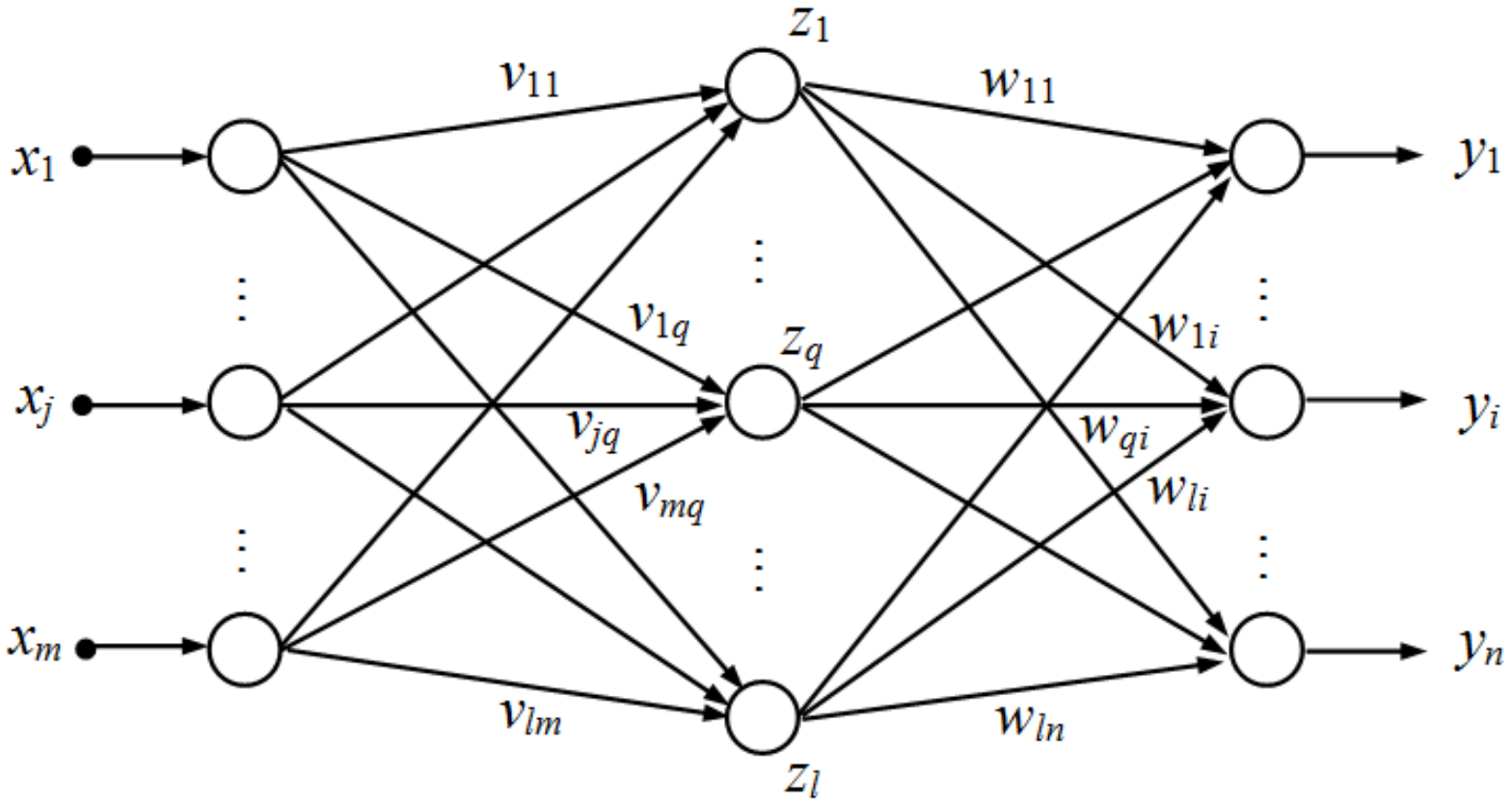
$$\boldsymbol{w}(4)=\boldsymbol{w}(3)+\eta(d(3)-y(3))\boldsymbol{x}(3)=\begin{bmatrix}-0.486\\0.677\\0.302\end{bmatrix}+0.3\times(0.7+0.0656)\times\begin{bmatrix}0.6\\0.2\\0.3\end{bmatrix}=\begin{bmatrix}-0.348\\0.723\\0.371\end{bmatrix}$$

$$\boldsymbol{E}=\boldsymbol{E}+0.5\times(d(2)-y(2))^2=0.3195$$

★ Sau 1 chu kỳ huấn luyện, trọng số của Adaline là $\boldsymbol{w}=\begin{bmatrix}-0.348\\0.723\\0.371\end{bmatrix}$

Mạng truyền thẳng nhiều lớp

Mạng truyền thẳng nhiều lớp (MLP)



$$net_{hq} = \mathbf{v}_q^T \mathbf{x} \qquad net_{oi} = \mathbf{w}_i^T \mathbf{z}$$

$$z_q = a_h(net_{hq}) \qquad y_i = a_o(net_{oi})$$

Biểu thức ngõ ra mạng truyền thẳng 3 lớp

- Tổng có trọng số tín hiệu vào tế bào thần kinh thứ *q* ở lớp ẩn:

$$net_{hq} = \boldsymbol{v}_q^T \boldsymbol{x} = \sum_{j=1}^{m} v_{jq} x_j$$

- Ngõ ra tế bào thần kinh thứ *q* ở lớp ẩn:

$$z_q = a_h\left(net_{hq}\right) = a_h\left(\boldsymbol{v}_q^T \boldsymbol{x}\right)$$

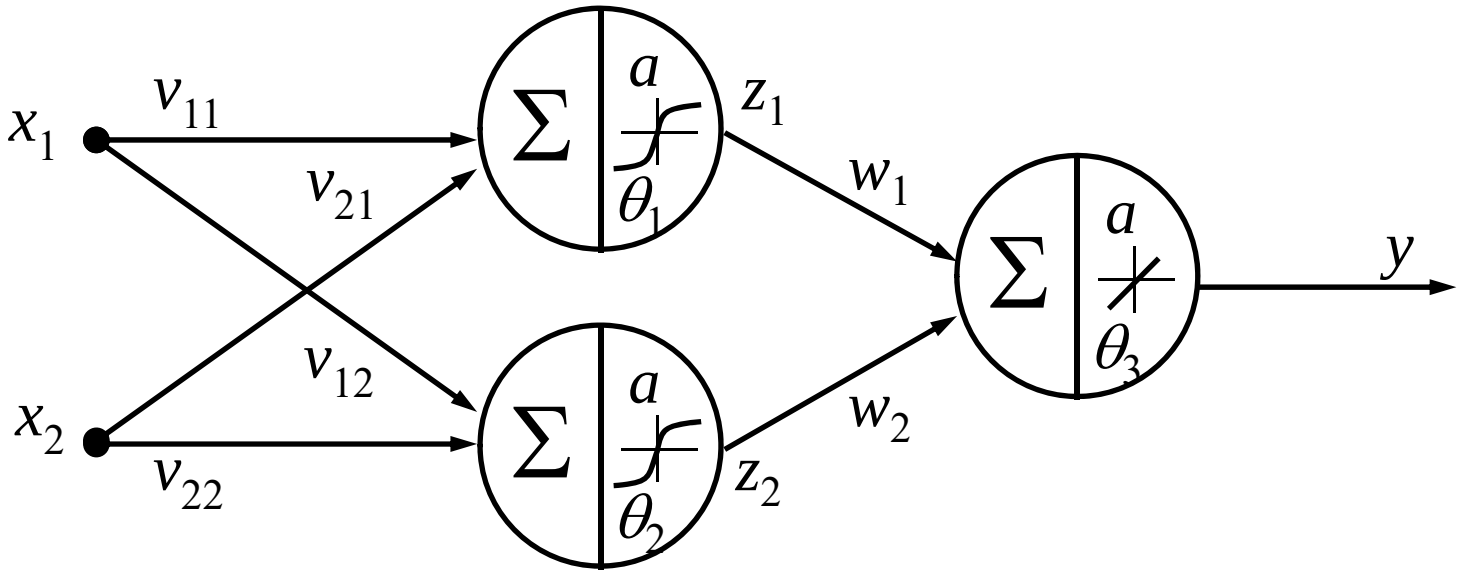
- Tổng có trọng số tín hiệu vào tế bào thần kinh thứ *i* ở lớp ra:

$$net_{oi} = \boldsymbol{w}_i^T \boldsymbol{z} = \sum_{q=1}^{l} w_{qi} z_q$$

- Ngõ ra tế bào thần kinh thứ *i* ở lớp ra:

$$y_i = a_o\left(net_{oi}\right) = a_o\left(\boldsymbol{w}_i^T \boldsymbol{z}\right)$$

Ví dụ tính ngõ ra của mạng MLP



- Tính ngõ ra của mạng truyền thẳng ở trên, biết hàm kích hoạt lớp ẩn là S lưỡng cực (tansig), hàm kích hoạt lớp ra là tuyến tính (purelin)

$$x_1 = -1; x_2 = 2; v_{11} = -0.5; v_{12} = 0.7; \theta_1 = 0.2$$

$$v_{21} = 0.3; v_{22} = 0.9; \theta_2 = 0.1$$

$$w_1 = 0.4; w_2 = -0.5; \theta_3 = -0.3$$

Ví dụ tính ngõ ra của mạng MLP

- **Giải:**
- Lớp ẩn:

$$net_{h1} = v_{11}x_1 + v_{21}x_2 - \theta_1 = (-0.5)\times(-1) + 0.3\times 2 - 0.2 = 0.9$$

$$net_{h2} = v_{12}x_1 + v_{22}x_2 - \theta_2 = (0.7)\times(-1) + 0.9\times 2 - 0.1 = 1.0$$

$$z_1 = \frac{1-e^{-2\times net_{h1}}}{1+e^{-2\times net_{h1}}} = \frac{1-e^{2\times 0.9}}{1+e^{2\times 0.9}} = 0.716$$

$$z_2 = \frac{1-e^{-2\times net_{h2}}}{1+e^{-2\times net_{h2}}} = \frac{1-e^{2\times 1.0}}{1+e^{2\times 1.0}} = 0.762$$

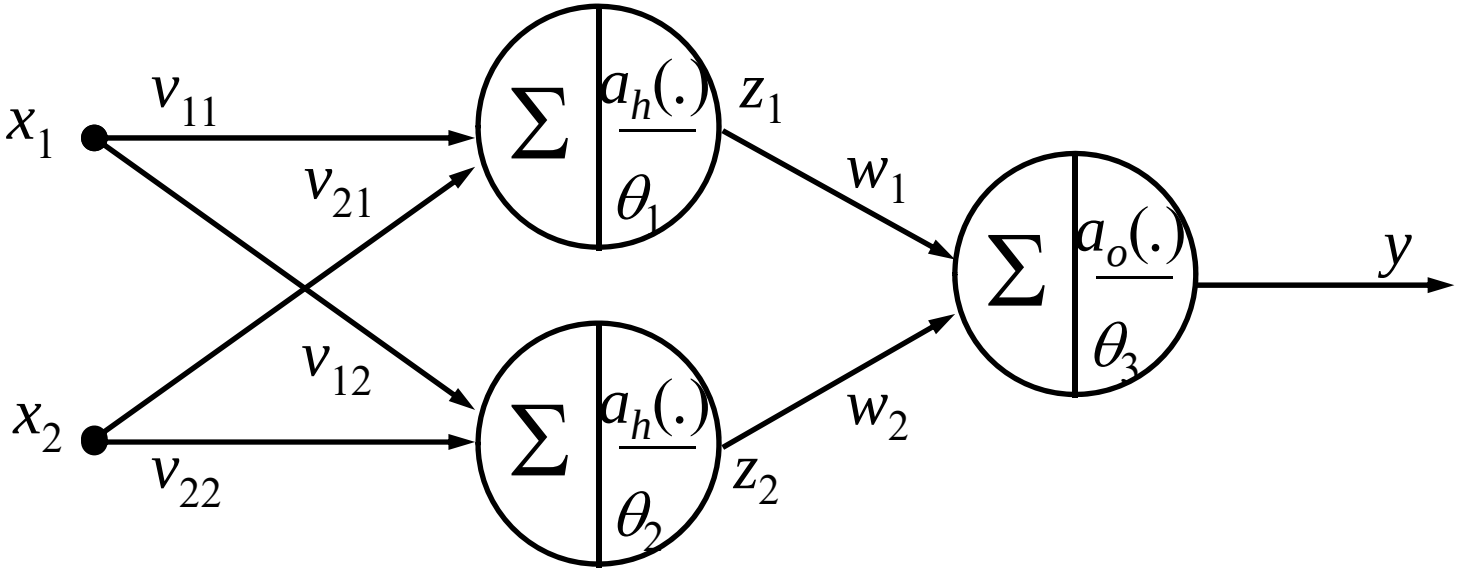
- Lớp ra:

$$net_o = w_1 z_1 + w_2 z_2 - \theta_3$$

$$= 0.4\times 0.716 + (-0.5)\times 0.762 + 0.3 = 0.205$$

$$y = net_o = 0.205$$

Bài tập tính ngõ ra của mạng MLP



- Tính ngõ ra của mạng truyền thẳng ở trên, biết hàm kích hoạt lớp ẩn là sigmoid đơn cực (logsig), hàm kích hoạt lớp ra là sigmoid lưỡng cực (tansig)

$$x_1 = 1.2; x_2 = 0.5; v_{11} = -0.2; v_{12} = -0.7; \theta_1 = -0.3$$

$$v_{21} = -0.2; v_{22} = 0.8; \theta_2 = 0.1$$

$$w_1 = 0.6; w_2 = -0.4; \theta_3 = -0.2$$

Huấn luyện mạng MLP: công thức tính sai số

- Sai số giữa ngõ ra của mạng và dữ liệu ra mong muốn là hàm phụ thuộc vào trọng số của mạng:

$$E(\boldsymbol{w},\boldsymbol{v}) = \frac{1}{2}\sum_{i=1}^{n}\left(d_i - y_i\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a_o(net_{oi})\right]^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a_o\left(\sum_{q=1}^{l} w_{qi} z_q\right)\right]^2 = \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a_o\left(\sum_{q=1}^{l} w_{qi} a_h(net_{hq})\right)\right]^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left[d_i - a_o\left(\sum_{q=1}^{l} w_{qi} a_h\left(\sum_{j=1}^{m} v_{jq} x_j\right)\right)\right]^2$$

Thuật toán suy giảm độ dốc cập nhật trọng số

- Biểu thức cập nhật trọng số lớp ra:

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) - \eta \frac{\partial E(k)}{\partial \boldsymbol{w}_i}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{w}_i} = \left[\frac{\partial E(k)}{\partial y_i}\right]\left[\frac{\partial y_i}{\partial net_{oi}}\right]\left[\frac{\partial net_{oi}}{\partial \boldsymbol{w}_i}\right]$$

$$= [-\underbrace{(d_i(k) - y_i(k))][a_o'(net_{oi}(k))}_{\delta_{oi}(k)}][\boldsymbol{z}(k)]$$

$$\Rightarrow \boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta \delta_{oi}(k)\boldsymbol{z}(k)$$

Thuật toán suy giảm độ dốc cập nhật trọng số

★ Biểu thức cập nhật trọng số lớp ẩn:

$$\boldsymbol{v}_q(k+1) = \boldsymbol{v}_q(k) - \eta \frac{\partial E(k)}{\partial \boldsymbol{v}_q}$$

$$\frac{\partial E(k)}{\partial \boldsymbol{v}_q} = \left[\frac{\partial E(k)}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial net_{hq}}\right]\left[\frac{\partial net_{hq}}{\partial \boldsymbol{v}_q}\right]$$

$$= \left[-\sum_{i=1}^{n} \underbrace{(d_i(k) - y_i(k))a_o'(net_{oi}(k))}_{\delta_{oi}(k)} w_{iq}(k)\right][a_h'(net_{hq}(k))][\boldsymbol{x}(k)]$$

$$\delta_{hq}(k)$$

$$\Rightarrow \boldsymbol{v}_q(k+1) = \boldsymbol{v}_q(k) + \eta\delta_{hq}(k)\boldsymbol{x}(k)$$

Thuật toán lan truyền ngược (Back propagation)

★ ***Bước 1:*** Chọn η; gán k=1, E=0; khởi động ngẫu nhiên trọng số lớp ẩn và lớp ra $\boldsymbol{v}_q(k)$, $\boldsymbol{w}_i(k)$ $(1 \le q \le l,\ 1 \le i \le n)$

★ ***Bước 2:*** (truyền thuận dữ liệu) Tính ngõ ra của mạng với tín hiệu vào $\boldsymbol{x}(k)$ $(1 \le j \le m)$:

Lớp ẩn: $$net_{hq}(k) = \boldsymbol{v}_q^T(k)\boldsymbol{x}(k) \qquad (1 \le q \le l)$$

$$z_q(k) = a_h\left(net_{hq}(k)\right)$$

Lớp ra: $$net_{oi}(k) = \boldsymbol{w}_i^T(k)\boldsymbol{z}(k) \qquad (1 \le i \le n)$$

$$y_i(k) = a_o\left(net_{oi}(k)\right)$$

Thuật toán lan truyền ngược (Back propagation)

★ ***Bước 3:*** (Lan truyền ngược sai số) Cập nhật trọng số:

Lớp ra: $\delta_{oi}(k) = [(d_i(k) - y_i(k))][a'_o(net_{oi}(k))] \quad (1 \le i \le n)$

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \eta\delta_{oi}(k)\boldsymbol{z}(k)$$

Lớp ẩn: $\delta_{hq}(k) = \left[\sum_{i=1}^{n} \delta_{oi}(k) w_{iq}(k)\right] a'_h(net_{hq}(k)) \quad (1 \le q \le l)$

$$\boldsymbol{v}_q(k+1) = \boldsymbol{v}_q(k) + \eta\delta_{hq}(k)\boldsymbol{x}(k)$$

★ ***Bước 4:*** Tính sai số tích lũy: $E = E + \frac{1}{2}\sum_{i=1}^{n}[d_i(k) - y_i(k)]^2$

★ ***Bước 5:*** Nếu $k<K$ thì gán $k = k+1$ và trở lại bước 2.
Nếu $k=K$ thì tiếp tục bước 6

★ ***Bước 6:*** Kết thúc chu kỳ huấn luyện. Nếu $E \le \varepsilon$ thì kết thúc. Nếu $E > \varepsilon$ thì gán gán $k = 1$, $E = 0$ và trở lại bước 2

Đạo hàm của hàm tác động tuyến tính

- Hàm tuyến tính (purelin):

$$a(f) = f$$

$$\Rightarrow \quad a'(f) = 1$$

Đạo hàm của hàm tác động dạng S

- Hàm sigmoid đơn cực (logsig):

$$a(f) = \frac{1}{1+e^{-f}}$$

$$\Rightarrow a'(f) = \frac{e^{-f}}{\left(1+e^{-f}\right)^2} = \frac{-1+1+e^{-f}}{\left(1+e^{-f}\right)^2}$$

$$\Rightarrow \boxed{a'(f) = a(f)\left[1-a(f)\right]}$$

Đạo hàm của hàm tác động dạng S

- Hàm sigmoid lưỡng cực (tansig):

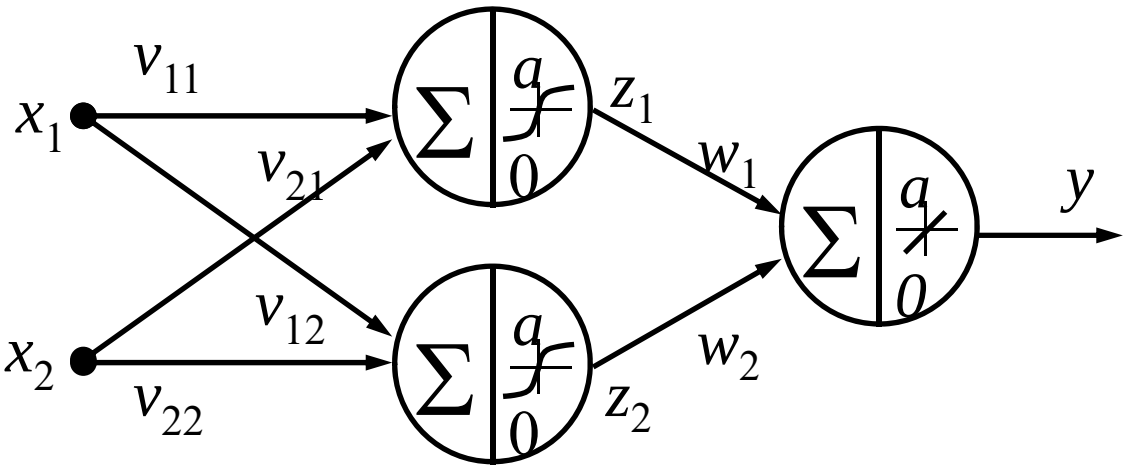
$$a(f) = \frac{2}{1+e^{-2f}} - 1 = \frac{1-e^{-2f}}{1+e^{-2f}}$$

$$\Rightarrow \quad a'(f) = \frac{4e^{-2f}}{(1+e^{-2f})^2} = \frac{(1+e^{-2f})^2 - (1-e^{-2f})^2}{(1+e^{-2f})^2}$$

$$\Rightarrow \quad a'(f) = 1 - a^2(f)$$

Ví dụ: giải thuật lan truyền ngược

- Cho mạng truyền thẳng ở trên, biết hàm kích hoạt lớp ẩn là S đơn cực (logsig), hàm kích hoạt lớp ra là tuyến tính (purelin). Cho trọng số ban đầu của mạng và tập dữ liệu như sau:



$$v_{11}(1)=0.5; v_{12}(1)=0.7; v_{21}(1)=-0.3;$$
$$v_{22}(1)=0.9; w_1(1)=0.4; w_2(1)=-0.5$$

$$X=\begin{bmatrix}-0.4 & 0.1 & 0.4\\ 0.6 & 0.3 & -0.2\end{bmatrix}; D=\begin{bmatrix}0.9 & -0.3 & 0.8\end{bmatrix}$$

- Biết hệ số học η=0.2. Tính trọng số của mạng sau 1 bước huấn luyện dùng giải thuật lan truyền ngược

Ví dụ: giải thuật lan truyền ngược (tt)

Tập dữ liệu huấn luyện mạng MLP gồm 3 mẫu (K=3)

$$X = \begin{bmatrix} -0.4 & 0.1 & 0.4 \\ 0.6 & 0.3 & -0.2 \end{bmatrix}; D = \begin{bmatrix} 0.9 & -0.3 & 0.8 \end{bmatrix}$$

Bước 1: (khởi động)

$$\boldsymbol{v}_1(1) = \begin{bmatrix} v_{11}(1) \\ v_{21}(1) \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix}$$

$$\boldsymbol{v}_2(1) = \begin{bmatrix} v_{12}(1) \\ v_{22}(1) \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix}$$

$$\boldsymbol{w}(1) = \begin{bmatrix} w_1(1) \\ w_2(1) \end{bmatrix} = \begin{bmatrix} 0.4 \\ -0.5 \end{bmatrix}$$

Ví dụ: giải thuật lan truyền ngược (tt)

Bước 2: truyền thuận dữ liệu

$$net_{h1}(1) = \mathbf{v}_1^T(1)\mathbf{x}(1) = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix}^T \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix} = -0.38$$

$$net_{h2}(1) = \mathbf{v}_2^T(1)\mathbf{x}(1) = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix}^T \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix} = 0.26$$

$$z_1(1) = \log\mathrm{sig}(net_{h1}(1)) = \frac{1}{1+\exp(0.38)} = 0.406$$

$$z_2(1) = \log\mathrm{sig}(net_{h2}(1)) = \frac{1}{1+\exp(-0.26)} = 0.565$$

$$net_o(1) = \mathbf{w}^T(1)\mathbf{z}(1) = \begin{bmatrix} 0.4 \\ -0.5 \end{bmatrix}^T \begin{bmatrix} 0.406 \\ 0.565 \end{bmatrix} = -0.12$$

$$y(1) = net_o(1) = -0.12$$

Ví dụ: giải thuật lan truyền ngược (tt)

Bước 3: cập nhật trọng số - lan truyền ngược sai số

Cập nhật trọng số lớp ra:

$a_o'(net_o) = 1$ (do $a_o(.)$ là hàm tuyến tính)

$$\delta_o(1) = [d(1) - y(1)] \times 1 = [0.9 - (-0.12)] = 1.02$$

$$\mathbf{w}(2) = \mathbf{w}(1) + \eta \delta_o(1) \mathbf{z}(1) = \begin{bmatrix} 0.4 \\ -0.5 \end{bmatrix} + 0.2 \times 1.02 \times \begin{bmatrix} 0.406 \\ 0.565 \end{bmatrix} = \begin{bmatrix} 0.483 \\ -0.385 \end{bmatrix}$$

Ví dụ: giải thuật lan truyền ngược (tt)

Bước 3: cập nhật trọng số - lan truyền ngược sai số

Cập nhật trọng số lớp ẩn:

$$a'_h(net_{hq}) = a_h(net_{hq}) \times \left[1 - a_h(net_{hq})\right] = z_q(1 - z_q) \quad \text{(do } a_h(.) \text{ là logsig)}$$

$$\delta_{h1}(1) = \delta_o(1).w_1(1)z_1(1).[1 - z_1(1)]$$

$$= 1.02 \times 0.4 \times 0.406 \times (1 - 0.406) = 0.098$$

$$\boldsymbol{v}_1(2) = \boldsymbol{v}_1(1) + \eta\delta_{h1}(1)\boldsymbol{x}(1) = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix} + 0.2 \times 0.098 \times \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix} = \begin{bmatrix} 0.492 \\ -0.288 \end{bmatrix}$$

$$\delta_{h2}(1) = \delta_o(1).w_2(1)z_2(1).[1 - z_2(1)]$$

$$= 1.02 \times (-0.5) \times 0.565 \times (1 - 0.565) = -0.125$$

$$\boldsymbol{v}_2(2) = \boldsymbol{v}_2(1) + \eta\delta_{h2}(1)\boldsymbol{x}(1) = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix} + 0.2 \times (-0.125) \times \begin{bmatrix} -0.4 \\ 0.6 \end{bmatrix} = \begin{bmatrix} 0.710 \\ 0.885 \end{bmatrix}$$

Ví dụ: Xấp xỉ hàm dùng mạng truyền thẳng nhiều lớp

★ Bài toán: Xấp xỉ hàm phi tuyến:

$$y = e^{-x}\sin(10x) \qquad (0 \le x \le 2)$$

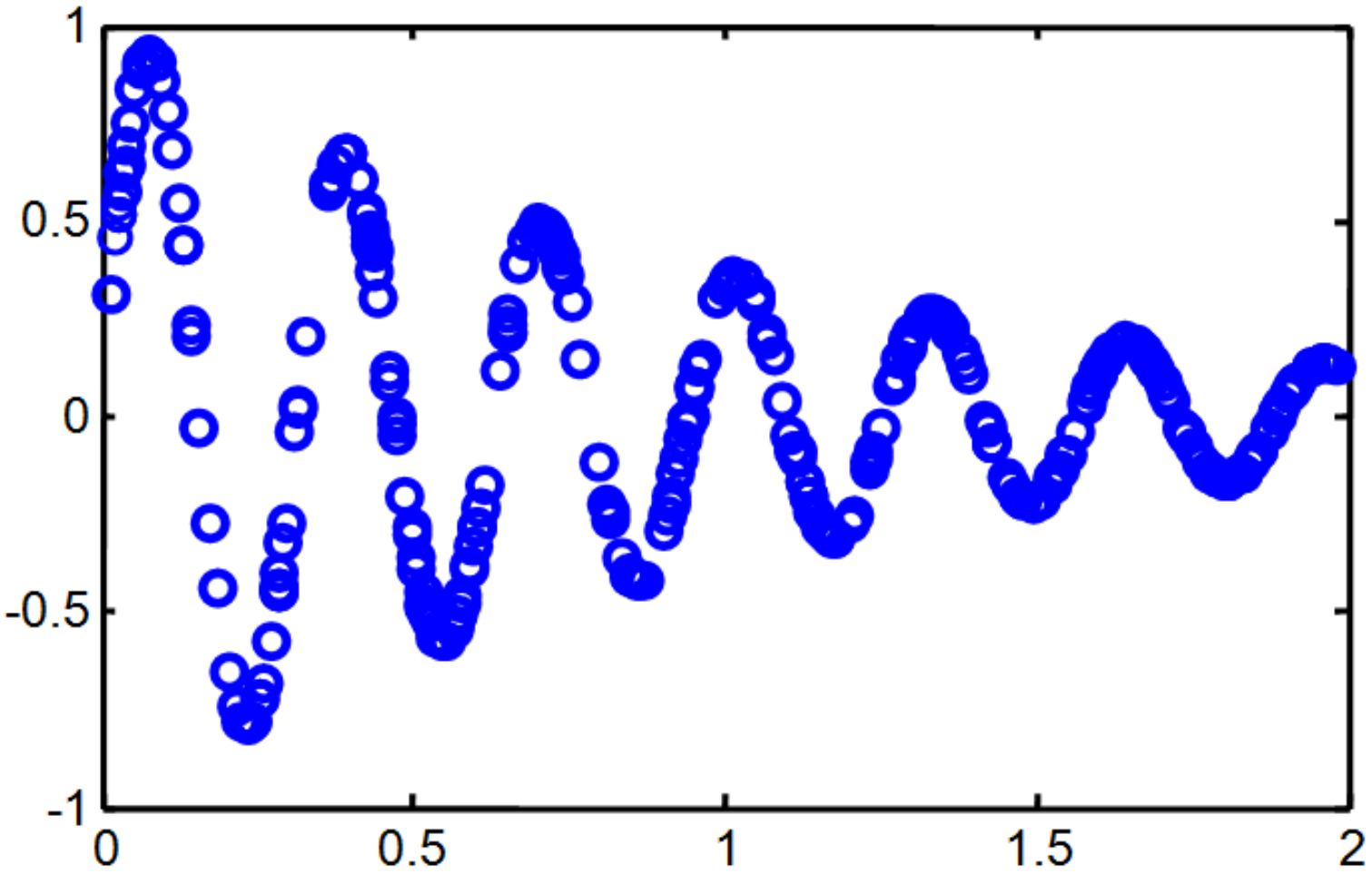
dùng mạng neuron

★ Cấu trúc mạng:

- 1 ngõ vào là *x*, 1 ngõ ra là *y*.
- Số tế bào thần kinh ở lớp ẩn là *N*. Thí dụ này khảo sát khả năng xấp xỉ của mạng trong 4 trường hợp *N* bằng 6, 8, 10, 12.
- Hàm kích hoạt ở lớp ẩn là tansig, ở lớp ra là tuyến tính.

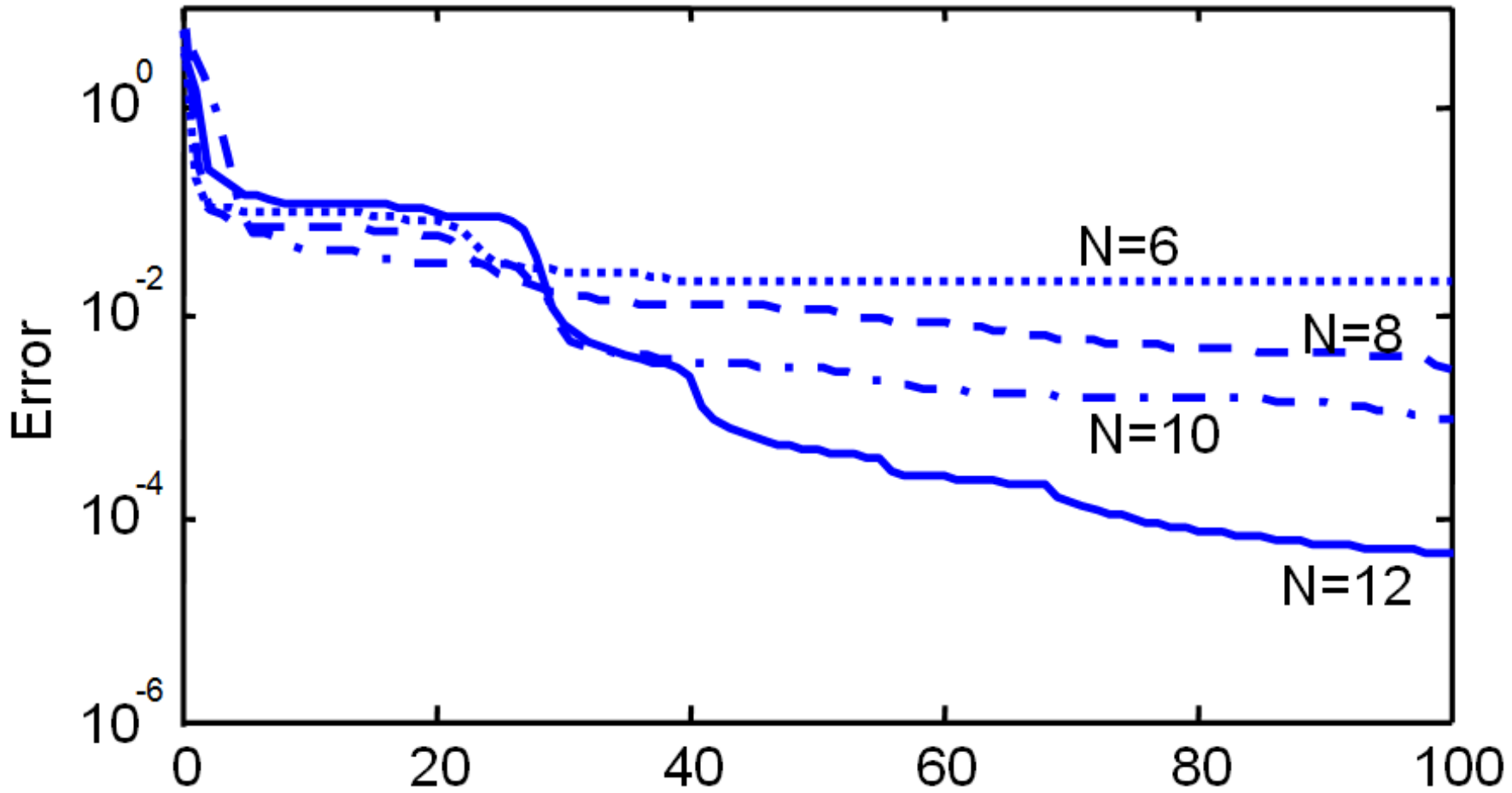
Ví dụ: Xấp xỉ hàm dùng mạng truyền thẳng nhiều lớp (tt)

- Tập dữ liệu huấn luyện mạng gồm 300 mẫu

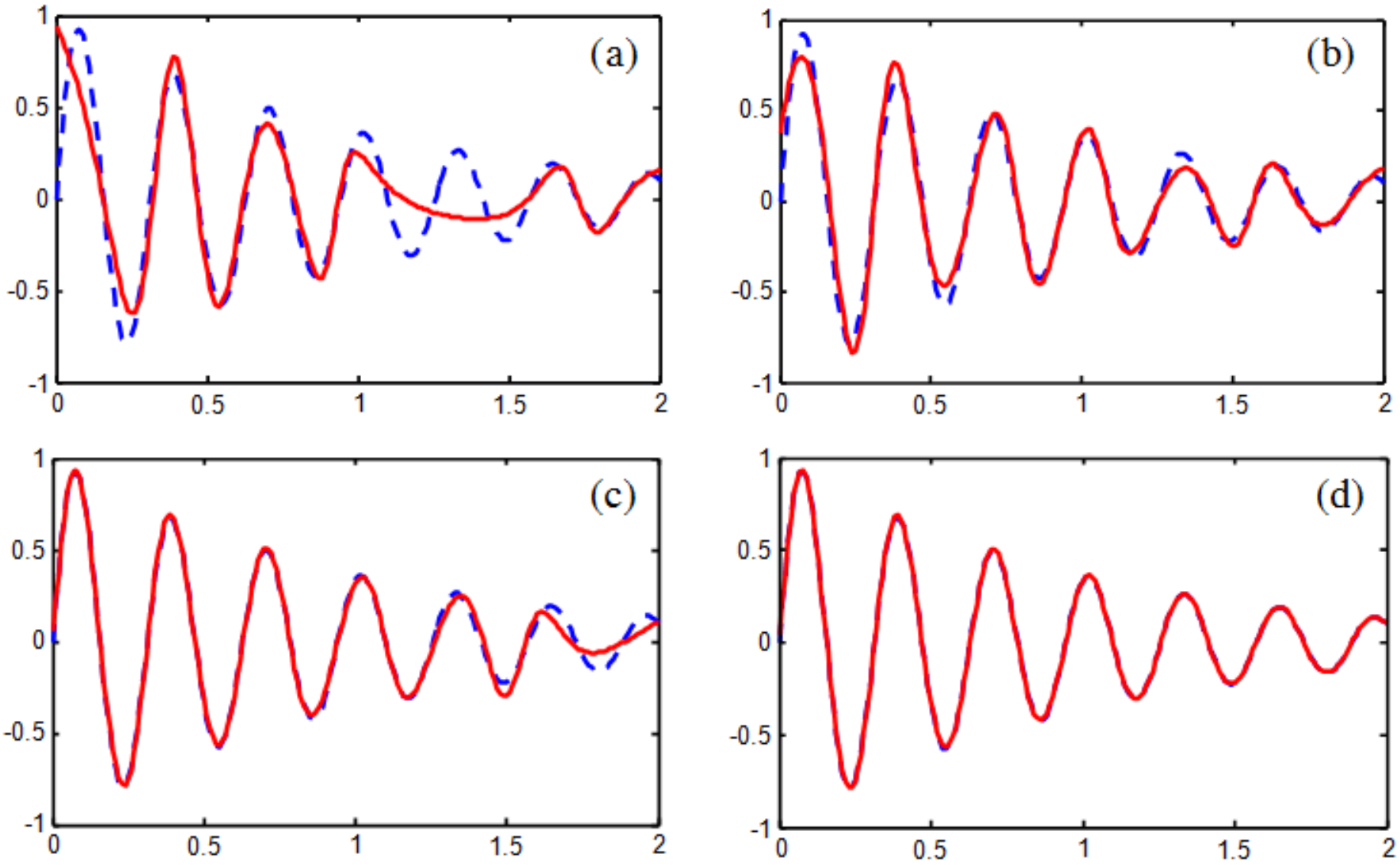


Ví dụ: Xấp xỉ hàm dùng mạng truyền thẳng nhiều lớp (tt)

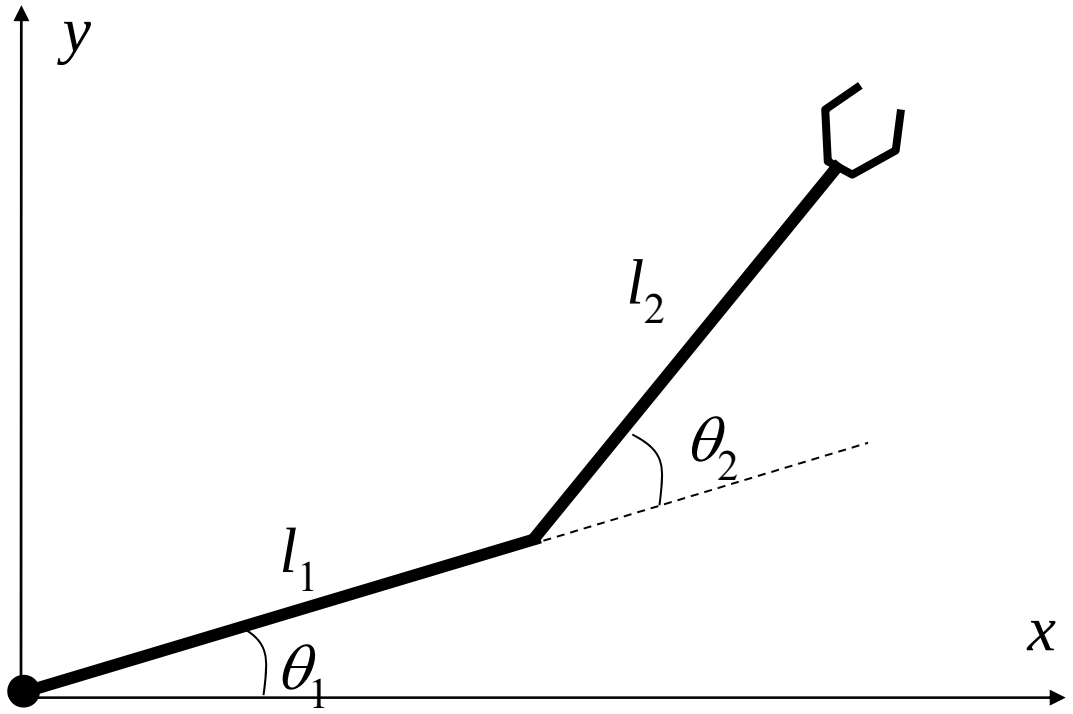
- Kết quả huấn luyện mạng



Ứng dụng xấp xỉ hàm



Bài tập



$$\begin{cases} x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{cases}$$

- Huấn luyện mạng nơ-ron giải bài toán động học thuận robot

GIỚI THIỆU
NEURAL NETWORKS TOOLBOX

Các hàm cơ bản của NN Toolbox

```
>> mynet=newff(X,D,N,{'tansig' 'purelin'});
   % khai báo mạng neuron 3 lớp
   % số ngõ vào là số hàng của X; số ngõ ra là số hàng của D
   % số nơ ron ở lớp ẩn là N
   % hàm tác động lớp ẩn là tansig, ở lớp ra là purelin
>> mynet=perceptron          %khai báo một perceptron
>> mynet=train(mynet,X,D);
   % huấn luyện mạng mynet với dữ liệu vào là X, dữ liệu ra là D
   % số cột của X = số cột của D = số mẫu dữ liệu huấn luyện
>> Y=sim(mynet,X);
   % tính ngõ ra của mạng mynet khi ngõ vào là X
   % số cột của X là số mẫu dữ liệu cần tính
>> gensim(mynet)        %tạo ra khối Simulink thực hiện mynet
>> help nnet    % xem đầy đủ các hàm của NN Toolbox
```

Chuẩn đầu ra chương 4

Sau khi học xong chương 4, SV phải có khả năng:

- Hiểu khái niệm mạng thần kinh (nhân tạo), cấu trúc mạng và các thuật toán huấn luyện
- Hiểu cấu trúc và cách huấn luyện Perceptron, sử dụng Perceptron để giải các bài toán phân nhóm
- Hiểu cấu trúc và cách huấn luyện mạng truyền thẳng nhiều lớp, sử dụng mạng truyền thẳng nhiều lớp để giải các bài toán xấp xỉ hàm
- Lập trình huấn luyện mạng thần kinh dung Matlab

Chuẩn đầu ra của môn học

- Hiểu khái niệm về hệ thống điều khiển thông minh
- Hiểu lý thuyết tập mờ, logic mờ, suy luận mờ và hệ mờ
- Phân tích và thiết kế bộ điều khiển mờ
- Hiểu khái niệm mạng thần kinh (nhân tạo), cấu trúc mạng và các thuật toán huấn luyện
- Sử dụng mạng thần kinh trong nhận dạng và điều khiển
- Phân tích các hệ thống điều khiển thông minh trong công nghiệp và dân dụng.
- Sử dụng kỷ năng làm việc nhóm và kỹ năng giao tiếp để giải các bài tập lớn, đồ án thiết kế;
- Sử dụng phần mềm Matlab trong mô phỏng và thiết kế hệ thống điều khiển thông minh