

Nhận dạng mẫu (Pattern Recognition)

Mạng nơ-ron (neural network)

By Hoàng Hữu Việt

Email: viethh@vinhuni.edu.vn

Viện Kỹ thuật và Công nghệ, Đại học Vinh

Vinh, 5/2019

Thông tin môn học

■ Tài liệu chính

[1] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale.
Neural Network Design, 2nd.

link: hagan.okstate.edu/nnd.html.

■ Tài liệu khác

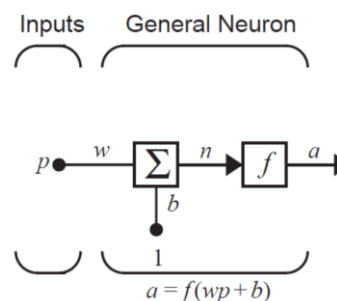
[2] Các nguồn từ internet

Nội dung

- Mô hình và kiến trúc mạng nơ-ron
- Mạng Perceptron
- Mạng Perceptron đa tầng (Multilayer Perceptrons)





Mô hình nơ-ron

- Mô hình nơ-ron một đầu vào (single-input neuron)
 - Đầu vào, p , là một số vô hướng.
 - Trọng số, w , là một số vô hướng.
 - Độ lệch, b , là một số vô hướng.
 - Đầu vào mạng, $n = wp + b$, là một số vô hướng.
 - Hàm truyền, f , tạo ra đầu ra của mạng $a = f(wp + b)$.
 - Ví dụ: $w = 3, p = 2, b = -1.5$
 $\Rightarrow f = f(3 \cdot 2 - 1.5) = f(4.5)$








Các hàm truyền (transfer functions)

- Có thể là hàm tuyến tính hoặc hàm phi tuyến.
- Một số hàm truyền:

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin

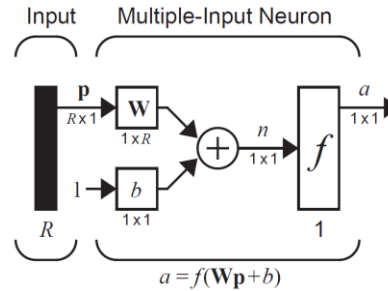
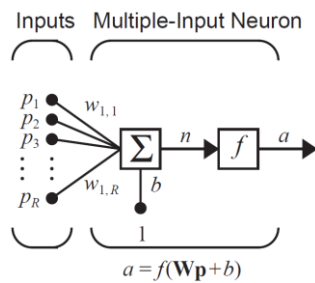
Các hàm truyền (transfer functions)

Name	Input/Output Relation	Icon	MATLAB Function
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

Mô hình nơ-ron

■ Mô hình nơ-ron nhiều đầu vào (multiple-input neuron)

- Đầu vào, $p = [p_1, p_2, \dots, p_R]^t$ là một vec-tơ cột.
- Trọng số, $W = [w_{1,1}, w_{1,2}, \dots, w_{1,R}]$ là một vec-tơ hàng.
- Độ lệch, b , là một số vô hướng.
- Đầu vào mạng, $n = w_{1,1}p_1 + \dots + w_{1,R}p_R + b = Wp + b$.
- Hàm truyền f tạo ra đầu ra của mạng $a = f(Wp + b)$.



Kiến trúc mạng nơ-ron

■ Mạng neuron một tầng (a layer of neurons)

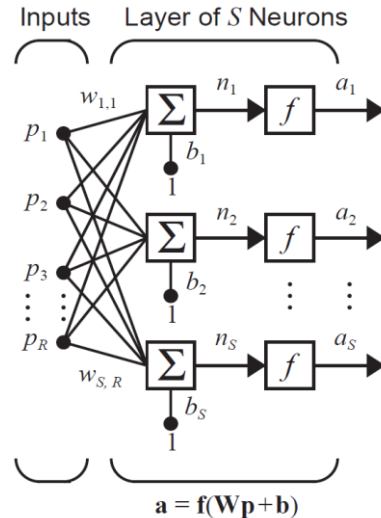
- Đầu vào: $p = (p_1, p_2, \dots, p_R)^t$.
- Ma trận trọng số $W_{S \times R}$

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

- Độ lệch: $b = (b_1, b_2, \dots, b_S)^t$.
- Đầu vào mạng:

$$n = (n_1, n_2, \dots, n_S)^t$$
- Đầu ra mạng:

$$a = (a_1, a_2, \dots, a_S)^t$$

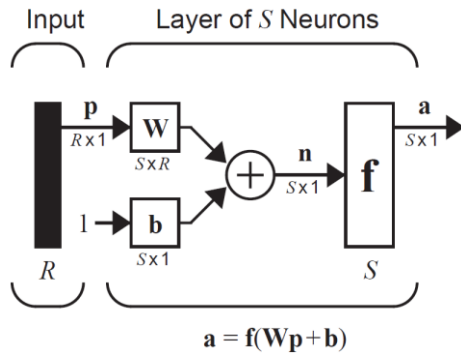


Kiến trúc mạng nơ-ron

- Mạng neuron một tầng – biểu diễn dạng ma trận
 - Đầu vào p , ma trận trọng $W_{S \times R}$, độ lệch b .
 - Đầu vào mạng n , hàm truyền f , đầu ra mạng $a = f(Wp + b)$.

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$



Kiến trúc mạng nơ-ron

- Ví dụ: mạng nơ-ron một tầng
 - Cho mạng một tầng với các tham số sau, f là hàm hardlim:

$$W = \begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

- Hãy vẽ mạng, và tính đầu ra của mạng biết các mẫu đầu vào là:

$$p_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, p_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, p_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, p_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix},$$

$$p_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, p_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, p_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, p_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}.$$

- Nhận xét gì về sự phân lớp các mẫu?

Kiến trúc mạng nơ-ron

■ Ví dụ: mạng nơ-ron một tầng

- Biểu diễn các mẫu đầu vào như sau:

$$p = [p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8] = \begin{bmatrix} 1 & 1 & 2 & 2 & -1 & -2 & -1 & -2 \\ 1 & 2 & -1 & 0 & 2 & 1 & -1 & -2 \end{bmatrix}$$

- Đầu vào của mạng: $n = Wp + b$

$$\begin{aligned} n &= \begin{bmatrix} -3 & -1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 & 2 & -1 & -2 & -1 & -2 \\ 1 & 2 & -1 & 0 & 2 & 1 & -1 & -2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -3 & -4 & -4 & -5 & 2 & 6 & 5 & 9 \\ -2 & -4 & 3 & 1 & -6 & -5 & 0 & 1 \end{bmatrix} \end{aligned}$$

- Đầu ra của mạng: $a = \text{hardlim}(n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$

Kiến trúc mạng nơ-ron

■ Ví dụ: mạng nơ-ron một tầng – tính bằng Python

- Khai báo ma trận w và b

```
w = np.array([[ -3, -1], [1, -2]])  
b = np.array([[1], [0]])
```

- Tính từng đầu vào

```
p1 = np.array([[1], [1]])  
n1 = np.matmul(w, p1) + b  
print(n1)
```

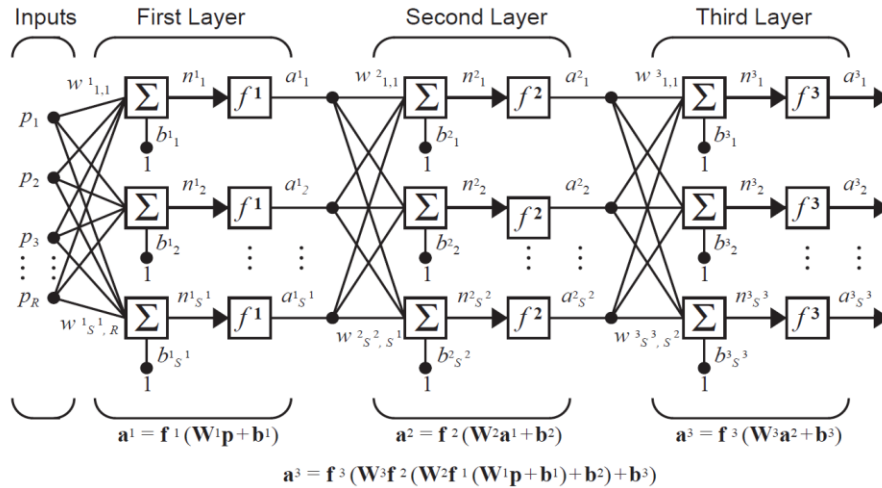
- Tính cho tất cả các đầu vào:

```
p = np.array([[1, 1, 2, 2, -1, -2, -1, -2],  
               [1, 2, -1, 0, 2, 1, -1, -2]])  
n = np.matmul(w, p) + b  
print(n)
```

Kiến trúc mạng nơ-ron

■ Mạng nơ-ron đa tầng (multilayer network)

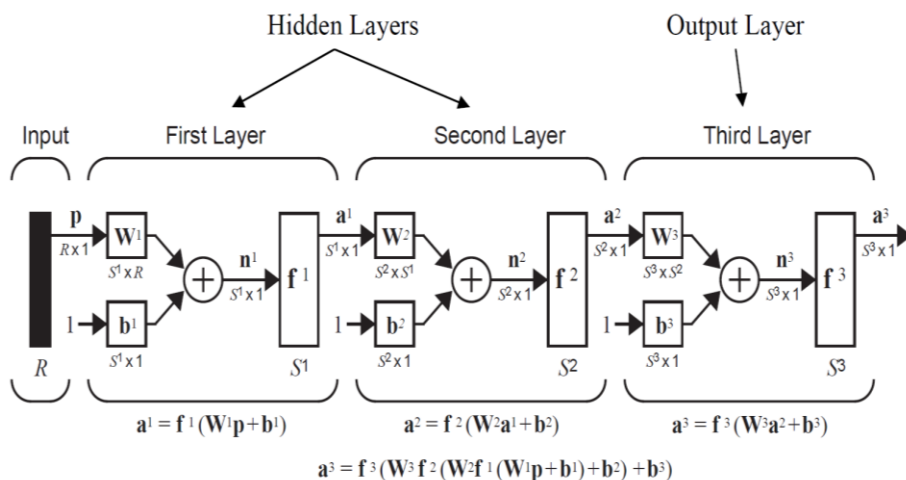
□ Mạng nơ-ron 3 tầng – ký hiệu chỉ số trên biểu diễn tầng mạng



Kiến trúc mạng nơ-ron

■ Mạng nơ-ron đa tầng (multilayer network)

□ Mạng nơ-ron 3 tầng – biểu diễn bằng ma trận



Kiến trúc mạng nơ-ron

■ Ví dụ mạng nơ-ron 2 tầng

- Cho mạng với các tham số sau, biết hàm truyền của các tầng là hàm hardlim:

$$W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}, b^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}, W^2 = [1 \ 1], b^2 = [-1.5]$$

- Hãy vẽ mạng, và tính đầu ra của mạng biết các mẫu đầu vào là:

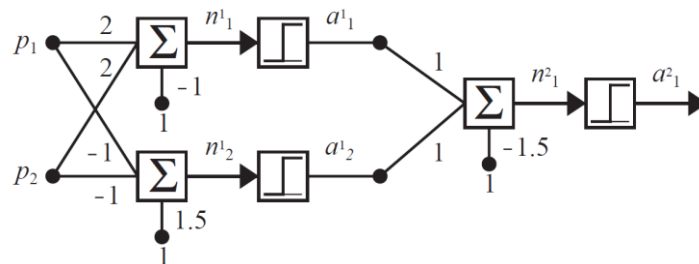
$$p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, p_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, p_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, p_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- Hãy đưa ra các nhận xét về sự phân lớp các tập mẫu.

Kiến trúc mạng nơ-ron

■ Ví dụ mạng nơ-ron 2 tầng

- Vẽ mạng



- Biểu diễn các đầu vào thành ma trận:

$$p = [p_1 p_2 p_3 p_4] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Kiến trúc mạng nơ-ron

■ Ví dụ mạng nơ-ron 2 tầng

- Đầu vào của tầng 1: $n^1 = W^1 p + b^1$

$$n^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1.5 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 3 \\ 1.5 & 0.5 & 0.5 & -0.5 \end{bmatrix}$$

- Đầu ra của tầng 1: $a^1 = \text{hardlim}(n^1)$

$$a^1 = \text{hardlim}(n^1) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

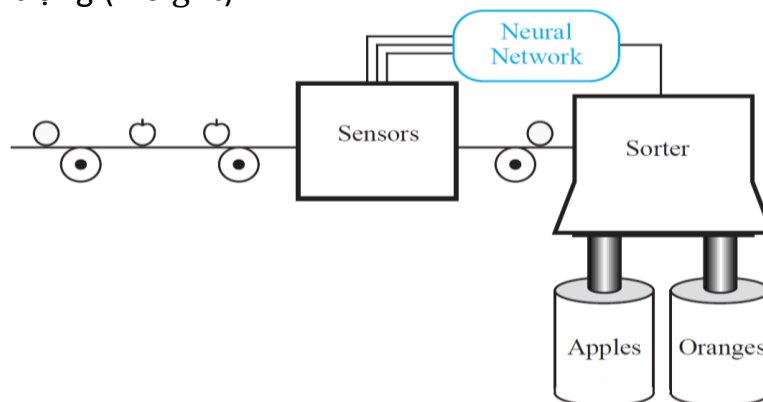
- Đầu vào của tầng 2: $n^2 = W^2 a^1 + b^2$

$$n^2 = [1 \ 1] \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} + [-1.5] = [-0.5 \ 0.5 \ 0.5 \ -0.5]$$

- Đầu ra của mạng: $a^2 = \text{hardlim}(n^2) = [0 \ 1 \ 1 \ 0]$

Ví dụ về ứng dụng của mạng nơ-ron

- Một băng chuyền để chuyển hoa quả.
- Hệ thống 3 sensors xác định 3 thuộc tính của mỗi đối tượng: hình dạng (shape), bề mặt (texture) và trọng lượng (weight).



Ví dụ về ứng dụng của mạng nơ-ron

- Mã hóa các thuộc tính của mỗi đối tượng thành một vec-tơ đầu vào.

$$\mathbf{p} = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix}$$

- $shape = \{1: round; -1: elliptical\}$
- $texture = \{1: smooth; -1: rough\}$
- $weight = \{1: \geq 1 \text{ pound}; -1: < 1 \text{ pound}\}$

- Ví dụ một quả cam được biểu diễn như p_1 và quả táo được biểu diễn như p_2 .

- Mạng nơ-ron phân lớp mẫu? $\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$

Ví dụ về ứng dụng của mạng nơ-ron

- Có nhiều mạng để phân lớp được các mẫu p_1 và p_2 .
- Ví dụ, chọn một mạng 1 tầng đơn giản như sau:

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, b = 0.$$

Orange:

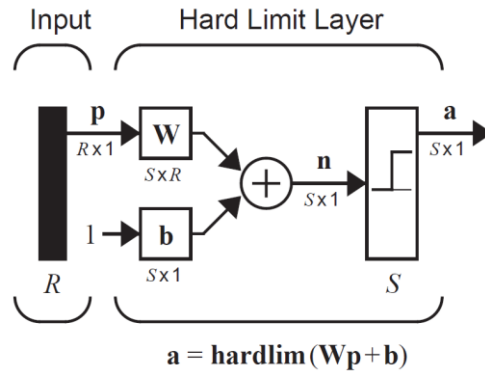
$$a = \text{hardlims} \left(\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0 \right) = -1(\text{orange}),$$

Apple:

$$a = \text{hardlims} \left(\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + 0 \right) = 1(\text{apple}).$$

Mạng Perceptron

■ Kiến trúc mạng



□ Đầu ra của mạng: $a = \text{hardlim}(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$

Mạng Perceptron

■ Thiết kế luật học cho mạng – huấn luyện mạng

■ Xét ma trận trọng số của mạng:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad \Rightarrow \quad \text{Ký hiệu dòng thứ } i \text{ của } \mathbf{W} \quad {}_i\mathbf{W} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

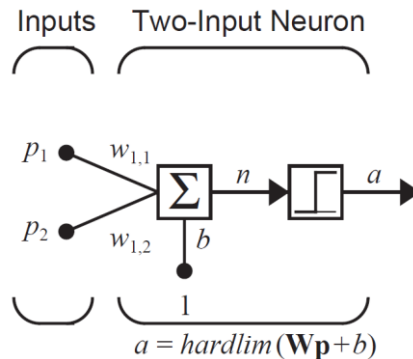
■ Ma trận \mathbf{W} được biểu diễn thành: $\mathbf{W} =$

$$\begin{bmatrix} {}_1\mathbf{W}^T \\ {}_2\mathbf{W}^T \\ \vdots \\ {}_S\mathbf{W}^T \end{bmatrix}$$

■ Đầu ra dòng thứ i : $a_i = \text{hardlim}(n_i) = \text{hardlim}({}_i\mathbf{W}^T \mathbf{p} + b_i)$

Mạng Perceptron

- Xét mạng Perceptron 1 nơ-ron



- Đầu ra của mạng:

$$a = \text{hardlim}(n) = \text{hardlim}(\mathbf{W}\mathbf{p} + b)$$

$$= \text{hardlim}({}_1\mathbf{w}^T \mathbf{p} + b) = \text{hardlim}(w_{1,1}p_1 + w_{1,2}p_2 + b)$$

Mạng Perceptron

- Đường biên quyết định (decision boundary)

$$n = {}_1\mathbf{w}^T \mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = 0$$

- Giả sử rằng $w_{1,1} = 1$, $w_{1,2} = 1$ và $b = 1$, đường biên quyết định là:

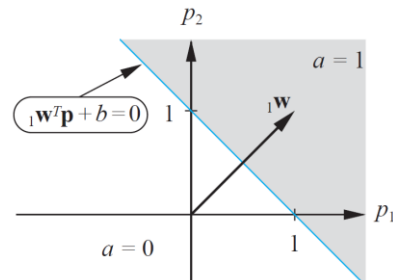
$$n = {}_1\mathbf{w}^T \mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = p_1 + p_2 - 1 = 0$$

- Tìm các miền giá trị phân

chia bởi đường biên:

□ Cho $\mathbf{p} = [0 \ 0]^T \rightarrow a = 0$

□ Miền còn lại là $a = 1$



- Đường biên quyết định vuông góc với véc tơ ${}_1\mathbf{w}$.

Mạng Perceptron

■ Tìm đường biên quyết định – phương pháp hình học

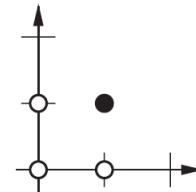
□ Đường biên quyết định: ${}_1w^T p + b = 0$, nghĩa là tích của tất cả các mẫu vào với véc-tơ ${}_1w^T$ có cùng giá trị $-b$.

□ Ví dụ thiết kế mạng để phân lớp các mẫu sau (cổng AND):

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t_1 = 0 \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t_2 = 0 \right\} \left\{ \mathbf{p}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t_3 = 0 \right\} \left\{ \mathbf{p}_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t_4 = 1 \right\}$$

□ Biểu diễn các điểm trên mặt phẳng:

- Điểm đen có đầu ra $t = 1$.
- Điểm trắng có đầu ra $t = 0$.



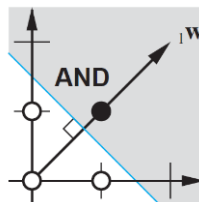
□ Chúng ta cần 1 đường thẳng để phân lớp các điểm đen và điểm trắng → có vô số đường.

Mạng Perceptron

■ Tìm đường biên quyết định – phương pháp hình học

1. Chọn đường biên phân chia 2 miền dữ liệu.

2. Chọn véc tơ trọng số vuông góc với đường biên, ví dụ:



$${}_1w = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

3. Tìm độ lệch b :

- Lấy một điểm trên đường biên quyết định, ví dụ: $p = [1.5 \ 0]^T$
- Tìm b : ${}_1w^T p + b = 0 \rightarrow b = -{}_1w^T p \rightarrow b = -[2 \ 2][1.5 \ 0]^T = -3$

□ Kiểm tra lại các giá trị đầu vào xem thỏa mãn đầu ra?

Mạng Perceptron

■ Luật học cho mạng Perceptron – huấn luyện mạng

- Phương pháp hình học không thể dùng để huấn luyện mạng nơ-ron.
- Luật học được xây dựng dựa trên học có hướng dẫn (supervised learning).
- Giả sử tập mẫu dùng để huấn luyện mạng như sau:

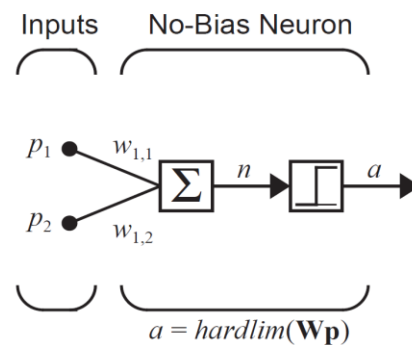
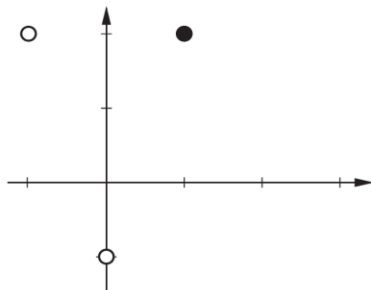
$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}, \text{ trong đó}$$

- p_q ($q = 1, 2, \dots, Q$) là mẫu đầu vào mạng và
- t_q là đầu ra thực tế tương ứng của mẫu p_q
- Luật học của mạng nhằm điều chỉnh tập trọng số và độ lệch để đầu ra của mạng đúng/gần đúng với đầu ra thực tế.

Mạng Perceptron

■ Ví dụ cho tập mẫu sau và giả sử $b = 0$

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1 \right\} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0 \right\} \quad \left\{ \mathbf{p}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0 \right\}$$



Mạng Perceptron

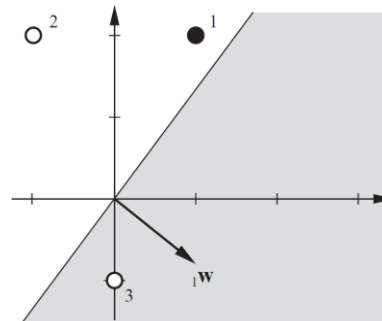
- Khởi tạo ngẫu nhiên véc-tơ trọng số: ${}_1\mathbf{w} = [1.0 \ -0.8]^T$
- Tính đầu ra của mạng cho mẫu p1:

$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p}_1) = \text{hardlim}\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$a = \text{hardlim}(-0.6) = 0.$$

Random initial weight:

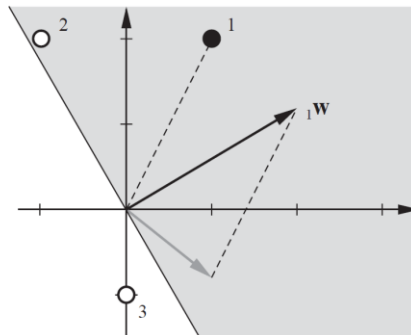
$${}_1\mathbf{w} = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$$



Mạng Perceptron

- Luật 1: if $t = 1$ and $a = 0$ then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$
- Áp dụng luật tính ${}_1\mathbf{w}$:

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$



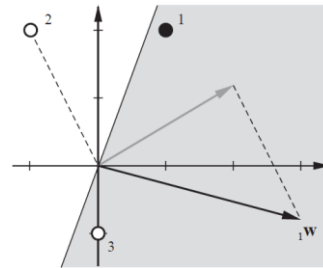
Mạng Perceptron

- Tính đầu ra của mạng cho mẫu p_2 :

$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p}_2) = \text{hardlim}\left(\begin{bmatrix} 2.0 & 1.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix}\right) \\ = \text{hardlim}(0.4) = 1 .$$

- Luật 2: If $t = 0$ and $a = 1$ then ${}_1\mathbf{w}^{\text{new}} = {}_1\mathbf{w}^{\text{old}} - \mathbf{p}$
- Áp dụng luật tính ${}_1\mathbf{w}$:

$${}_1\mathbf{w}^{\text{new}} = {}_1\mathbf{w}^{\text{old}} - \mathbf{p}_2 = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} - \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$



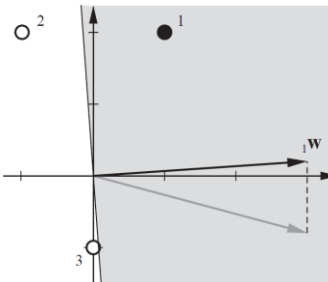
Mạng Perceptron

- Tính đầu ra của mạng cho mẫu p_3 :

$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p}_3) = \text{hardlim}\left(\begin{bmatrix} 3.0 & -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right) \\ = \text{hardlim}(0.8) = 1 .$$

- Áp dụng luật tính ${}_1\mathbf{w}$:

$${}_1\mathbf{w}^{\text{new}} = {}_1\mathbf{w}^{\text{old}} - \mathbf{p}_3 = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3.0 \\ 0.2 \end{bmatrix}$$



- Tất cả các mẫu phân lớp đúng.

- Luật 3: If $t = a$ then ${}_1\mathbf{w}^{\text{new}} = {}_1\mathbf{w}^{\text{old}}$.

Mạng Perceptron

■ Các luật:

- If $t = 1$ and $a = 0$ then ${}_1w^{new} = {}_1w^{old} + p$.
- If $t = 0$ and $a = 1$ then ${}_1w^{new} = {}_1w^{old} - p$.
- If $t = a$ then ${}_1w^{new} = {}_1w^{old}$.

■ Đặt lỗi $e = t - a$, các luật được viết lại:

- If $e = 1$ then ${}_1w^{new} = {}_1w^{old} + p$.
- If $e = -1$ then ${}_1w^{new} = {}_1w^{old} - p$.
- If $e = 0$ then ${}_1w^{new} = {}_1w^{old}$.

■ Luật tổng quát

- ${}_1w^{new} = {}_1w^{old} + ep = {}_1w^{old} + (t-a)p$.
- $b^{new} = b^{old} + e = b^{old} + (t-a)p$.

Mạng Perceptron

■ Luật học mạng nhiều nơ-ron – cập nhật dòng i

- ${}_iw^{new} = {}_iw^{old} + e_ip$
- $b_i^{new} = b_i^{old} + e_i$

■ Luật học viết theo dạng ma trận:

- $w^{new} = w^{old} + ep^T$
- $b^{new} = b^{old} + e$.

Mạng Perceptron

- Ví dụ (P4.4 – page 106): cho tập mẫu như sau:

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, t_1 = 0 \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, t_2 = 1 \right\}$$

$$\left\{ \mathbf{p}_3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, t_3 = 0 \right\} \left\{ \mathbf{p}_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, t_4 = 1 \right\}$$

- Tìm w và b , khởi tạo $w = [0 \ 0]^T$, $b = 0$.
 - 1) $a = \text{hardlim}(w p_1 + b) = \text{hardlim}(0) = 1 \rightarrow e = t_1 - a = -1$
 $w = w + e p_1^T = [-2 \ -2], b = b + e = -1$
 - 2) $a = \text{hardlim}(w p_2 + b) = \text{hardlim}(1) = 1 \rightarrow e = t_2 - a = 0$
 - 3) $a = \text{hardlim}(w p_3 + b) = \text{hardlim}(-1) = 0 \rightarrow e = t_3 - a = 0$

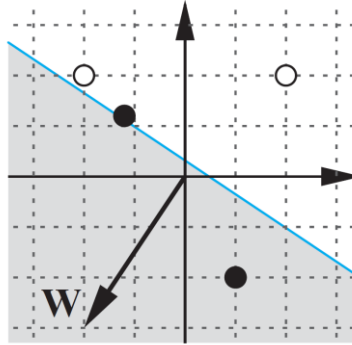
Mạng Perceptron

- Tìm w và b , khởi tạo $w = [0 \ 0]^T$, $b = 0$.
 - 4) $a = \text{hardlim}(w p_4 + b) = \text{hardlim}(-1) = 0 \rightarrow e = t_4 - a = 1$.
 $w = w + e p_4^T = [-3 \ -1], b = b + e = 1$.
 - 5) $a = \text{hardlim}(w p_1 + b) = \text{hardlim}(-8) = 0 \rightarrow e = t_1 - a = 0$.
 - 6) $a = \text{hardlim}(w p_2 + b) = \text{hardlim}(-1) = 0 \rightarrow e = t_2 - a = 1$
 $w = w + e p_2^T = [-2 \ -3], b = b + e = 1$.
 - 7) $a = \text{hardlim}(w p_3 + b) = 0 \rightarrow e = t_3 - a = 0$.
 - 8) $a = \text{hardlim}(w p_4 + b) = 1 \rightarrow e = t_4 - a = 0$.
 - 9) $a = \text{hardlim}(w p_1 + b) = 0 \rightarrow e = t_1 - a = 0$.
 - 10) $a = \text{hardlim}(w p_2 + b) = 1 \rightarrow e = t_2 - a = 0 \rightarrow \text{dừng}$.

Mạng Perceptron

- Khi thuật toán dừng: $w = [-2 \ -3]$, $b = 1$.

- Đường biên quyết định: $n = wp + b = -2p_1 - 3p_2 + 1 = 0$

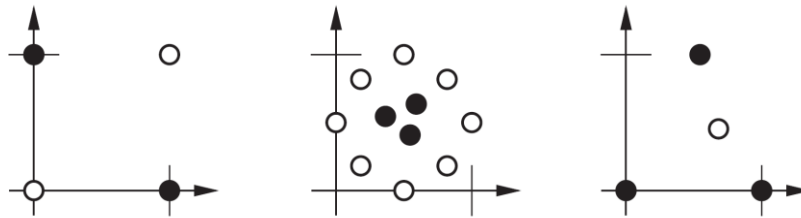


- Chú ý rằng đường biên quyết định đi qua một mẫu huấn luyện. Điều này là chấp nhận được vì hàm $\text{hardlim}(n)$ trả về 1 khi $n = 0$ và trả về 0 khi $n < 0$.

Mạng Perceptron

- Tổng kết

- Luật học Perceptron luôn hội tụ tới tập trọng số để phân lớp đúng nếu tập trọng số tồn tại.
- Luật học Perceptron chỉ tìm được đường biên quyết định tuyến tính: $w^T p + b = 0$.
- Luật học Perceptron không giải quyết được các bài toán mà các mẫu không tách tuyến tính.



Mạng Perceptron

- Bài tập 1 (P4.5 – trang 109): cho tập mẫu như sau:

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \left\{ \mathbf{p}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \left\{ \mathbf{p}_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \mathbf{t}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

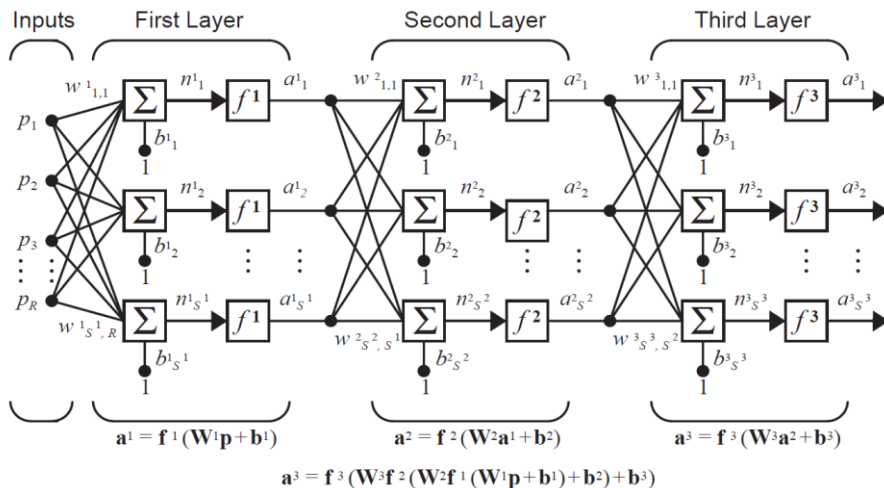
$$\left\{ \mathbf{p}_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \mathbf{t}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \left\{ \mathbf{p}_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \mathbf{t}_5 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\} \left\{ \mathbf{p}_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \mathbf{t}_6 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$$

$$\left\{ \mathbf{p}_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{t}_7 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \left\{ \mathbf{p}_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \mathbf{t}_8 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

- Sử dụng luật học tìm w, b .
- Vẽ mạng sau khi tìm được w và b .
- Vẽ tập mẫu và đường phân lớp của mạng.

Mạng Perceptron đa tầng

- Các mạng Perceptron có thể giải quyết được các bài toán phức tạp.

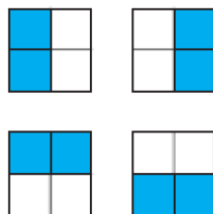


Mạng Perceptron đa tầng

- Ví dụ 1: Cho các tham số của mạng như sau:

$$W^1 = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix}; W^2 = [2 \ 2]; b^1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; b^2 = 1$$

- Hãy vẽ mạng và tính đầu ra của mạng với các mẫu đầu vào như sau:



- Mã hóa ảnh thành các véc-tơ dữ liệu: 1: điểm xanh, 2: điểm trắng, mỗi véc-tơ có 4 thành phần.

Mạng Perceptron đa tầng

- Ví dụ 2: bài toán phân lớp cổng XOR

$$\{p_1 = [0 \ 0]^t, t_1 = 0\}, \{p_2 = [0 \ 1]^t, t_2 = 1\}, \\ \{p_3 = [1 \ 0]^t, t_3 = 1\}, \{p_4 = [1 \ 1]^t, t_4 = 0\}$$

