XỬ LÝ ẢNH SỐ

GIỚI THIỆU

Trong phần này chúng ta sẽ tìm hiểu về một số cách tiếp cận trong phân loại mẫu ảnh:

- Phân loại bằng đối sánh nguyên mẫu
- Phân loại dựa trên công thức thống kê tối ưu
- Phân loại dựa trên mạng nơ-ron

Phân loại mẫu ảnh

- Phân loại bằng đối sánh nguyên mẫu và Phân loại dựa trên công thức thống kê tối ưu:
  - được sử dụng rộng rãi trong các ứng dụng mà bản chất của dữ liệu được hiểu rõ, dẫn đến việc ghép nối hiệu quả các đặc trưng và việc thiết kế bộ phân loại.
  - Phụ thuộc rất nhiều vào các kỹ thuật để xác định các đặc trưng và các thành phần của một bộ phân loại.
- Phân loại bằng mạng nơ-ron:
  - phụ thuộc ít hơn vào kiến thức của 2 cách tiếp cận trước mà hầu hết các đặc điểm lớp mẫu (hay đặc trưng) được học bởi hệ thống hơn là được chỉ định trước bởi một nhà thiết kế là con người.

Ứng dụng của nhận dạng mẫu

- Máy đọc mã vạch
- Kiểm tra ngân hàng
- Kiểm tra chất lượng sản phẩm
- Đọc dấu vân tay
- Sắp xếp thư
- Nhận dạng giọng nói

Mẫu và lớp mẫu

- Mẫu (pattern) như một sự sắp xếp không gian của các đặc điểm.
- Một lớp mẫu (pattern class) là một tập hợp các mẫu có chung một số thuộc tính.
- Nhận dạng mẫu bằng máy bao gồm các kỹ thuật để *tự động gán các mẫu cho các lớp tương ứng của chúng*. Đó là, đưa ra một mẫu hoặc tập hợp các mẫu mà lớp không xác định, công việc của hệ thống nhận dạng mẫu là gán nhãn lớp cho mỗi mẫu đầu vào của nó.

Các tập dữ liệu trong nhận dạng

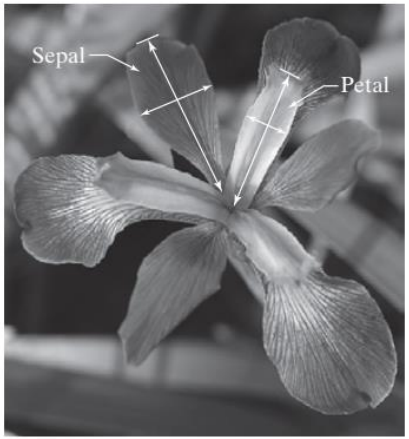
- Tập huấn luyện (training set: 50%):
  - được sử dụng để tạo các tham số bộ phân loại được gọi là huấn luyện.
  - mục tiêu là thực hiện điều chỉnh các tham số nếu bộ phân loại mắc lỗi trong việc xác định lớp của mẫu đã cho.
- Tập kiểm thử (validation set: 25%):
  - Được sử dụng khi kết thúc huấn luyện, chúng ta sử dụng tập kiểm thử (xác thực) để so sánh các thiết kế khác nhau để đạt được một mục tiêu hiệu suất mong muốn.
- Tập kiểm tra (test set: 25%):
  - Nhằm xác định một thiết kế đã được chọn có thực sự phù hợp. Chúng ta sử dụng bộ kiểm tra, bao gồm các mẫu mà hệ thống chưa từng "thấy" trước đây.
  - Kết quả huấn luyện/kiểm thử phải có hiệu suất gần với kết quả kiểm tra.

Vector mẫu

- Các vectơ mẫu được biểu thị bằng các chữ cái viết thường, chẳng hạn như **x, y** và **z** và có dạng:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{hoặc } x = (x_1, x_2, \ldots, x_n)^T$$

- Ví dụ:



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

x_1 = Petal width
x_2 = Petal length
x_3 = Sepal width
x_4 = Sepal length

Phân loại bằng đối sánh nguyên mẫu

- Đối sánh nguyên mẫu bao gồm việc *so sánh một mẫu không xác định* với một tập hợp các nguyên mẫu và gán cho mẫu chưa biết đó lớp của nguyên mẫu sao cho nó giống nhất với mẫu chưa biết đó.
- Mỗi nguyên mẫu đại diện cho một lớp mẫu duy nhất, nhưng có thể có nhiều hơn một nguyên mẫu cho mỗi lớp

Phân loại khoảng cách tối thiểu

- Đây là một trong những phương pháp đối sánh nguyên mẫu đơn giản nhất.
- Phương pháp này tính toán một đại lượng đo dựa trên *khoảng cách giữa một vectơ mẫu chưa biết với từng nguyên mẫu của lớp*. Sau đó nó gán mẫu chưa biết vào lớp nguyên mẫu gần nhất của nó.
- Vectơ nguyên mẫu của bộ phân loại khoảng cách tối thiểu thường là vectơ trung bình của các lớp mẫu khác nhau.

$$m_j = \frac{1}{n_j}\sum_{x \in c_j} \boldsymbol{x} \qquad j = 1,2,\dots,N_c$$

n_j: số vectơ mẫu được sử dụng để tính toán vectơ trung bình thứ j

c_j: là lớp mẫu thứ j và N_c là số lớp.

Phân loại khoảng cách tối thiểu

- Nếu sử dụng khoảng cách Euclidean để xác định độ tương tự, bộ phân loại khoảng cách tối thiểu sẽ tính toán khoảng cách:

$$D_j(\mathbf{x}) = \| \mathbf{x} - \mathbf{m}_j \| \qquad j = 1,2,\ldots, N_c$$

$$\| \mathbf{a} \| = (\mathbf{a}^T \mathbf{a})^{1/2}$$

- Bộ phân loại sau đó gán một mẫu chưa biết $\boldsymbol{x}$ vào lớp c_i nếu $D_i(\boldsymbol{x}) < D_j\ (\boldsymbol{x})\ với\ j\ =\ 1,2,\ldots,Nc$
- Điều này tương đương với: $d_i(\boldsymbol{x}) > d_j\ (\boldsymbol{x})$ với:

$$d_j(\mathbf{x}) = \mathbf{m}_j^T \mathbf{x} - \frac{1}{2}\mathbf{m}_j^T \mathbf{m}_j \qquad j = 1, 2, \ldots, N_c$$

Phân loại khoảng cách tối thiểu

- $d_i(\boldsymbol{x})$: hàm quyết định hay hàm phân biệt
- $d_i(\boldsymbol{x}) - d_j(\boldsymbol{x}) = 0$: biên quyết định phân tách lớp c_i và c_j
- Biên quyết định cho bộ phân loại khoảng cách tối thiểu:

$$d_{ij}(\boldsymbol{x}) = d_i(\boldsymbol{x}) - d_j(\boldsymbol{x})$$

$$= \left(m_i - m_j\right)^T x - (1/2)\left(m_i - m_j\right)^T \left(m_i + m_j\right) = 0$$

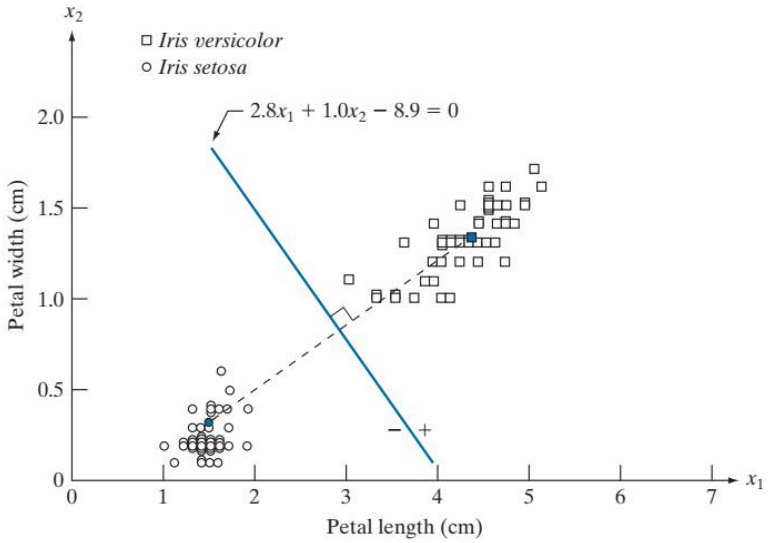
Ví dụ Phân loại hai lớp trong 2-D

- Vector mẫu gồm 2-D (chiều dài và chiều rộng cánh hoa)
- Versicolor: lớp c_1; Setosa: lớp c_2
- Trung bình hai lớp:

$$\mathbf{m}_1 = (4.3, 1.3)^T \qquad \mathbf{m}_2 = (1.5, 0.3)^T.$$

$$d_1(\mathbf{x}) = \mathbf{m}_1^T\mathbf{x} - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1$$
$$= 4.3x_1 + 1.3x_2 - 10.1$$

$$d_2(\mathbf{x}) = \mathbf{m}_2^T\mathbf{x} - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2$$
$$= 1.5x_1 + 0.3x_2 - 1.17$$



$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x})$$
$$= 2.8x_1 + 1.0x_2 - 8.9 = 0$$

$d_{12}(\mathbf{x}) > 0$ (c_1)
$d_{12}(\mathbf{x}) < 0$ (c_2)

Phân loại thống kê tối ưu (Bayes)

- Cách tiếp cận phân loại này tối ưu theo nghĩa là, về mặt trung bình, nó tạo ra xác suất mắc lỗi phân loại thấp nhất.
- $p(c_i|\boldsymbol{x})$: Xác suất để một vectơ mẫu $\boldsymbol{x}$ xuất phát từ lớp c_i
- L_{ij} : sự mất mát nếu bộ phân loại mẫu quyết định rằng $\boldsymbol{x}$ đến từ lớp c_j khi nó thực sự đến từ c_i.
- Vì mẫu $\boldsymbol{x}$ có thể thuộc về bất kỳ lớp nào trong số N_c lớp có thể, tổn thất trung bình xảy ra khi gán $\boldsymbol{x}$ cho lớp c_j là:

$$r_j(\mathbf{x}) = \sum_{k=1}^{N_c} L_{kj} p(c_k/\mathbf{x})$$

Phân loại thống kê tối ưu (Bayes)

- Bộ phân loại tối thiểu hóa tổng tổn thất trung bình được gọi là bộ phân loại Bayes.

$$r_j(\mathbf{x}) = \sum_{k=1}^{N_c} L_{kj} p(c_k/\mathbf{x})$$

- Bộ phân loại này gán một mẫu $\boldsymbol{x}$ không xác định cho lớp c_i nếu $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ với $j = 1,2,,,,N_c$.
- Hàm quyết định của bộ phân loại Bayes:

$$d_j(\mathbf{x}) = p(\mathbf{x}/c_j)P(c_j) \quad j = 1, 2, \ldots, N_c$$

- Bộ phân loại gán một mẫu cho lớp c_i nếu $d_i(x) > d_j(x)$ với mọi j ≠ i:

Ví dụ: Bộ phân loại Bayes cho mẫu 3-D

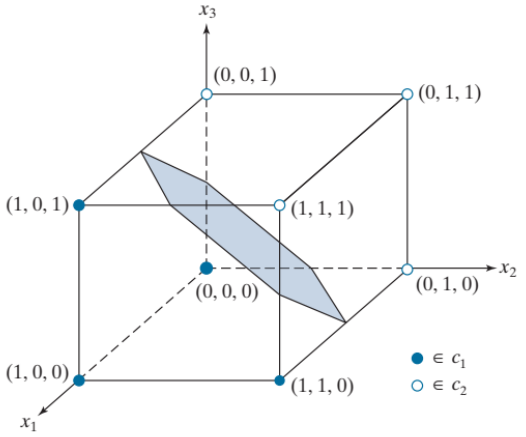
- Khi các lớp mẫu có phân bố Gauss với trung bình lớp mẫu thứ j là m_j.
- Khi đó hàm quyết định của bộ phân loại Bayes:

$$d_j(\mathbf{x}) = \ln P(c_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

- (C là ma trận hiệp phương sai và m là giá trị trung bình của lớp.

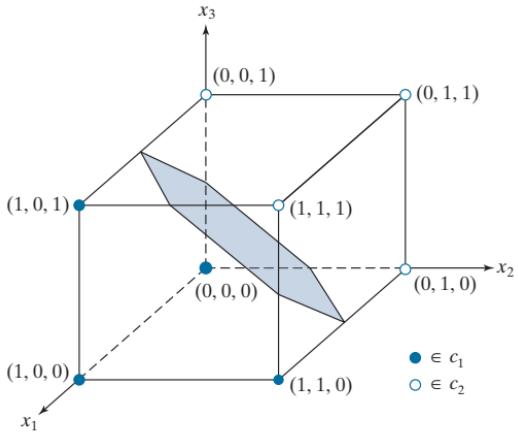
$$m_j = \frac{1}{n_j}\sum_{x \in c_j} \boldsymbol{x} \qquad j = 1,2,\ldots,\mathrm{Nc}$$

$$\mathbf{C}_j = \frac{1}{n_j}\sum_{\mathbf{x} \in c_j} \mathbf{x}\mathbf{x}^T - \mathbf{m}_j\mathbf{m}_j^T$$



Gỉa sử: các mẫu thuộc các lớp mẫu có phân bố Gauss và các lớp có khả năng xảy ra như nhau.

Ví dụ: Bộ phân loại Bayes cho mẫu 3-D



$$\mathbf{m}_1 = \frac{1}{3}\begin{bmatrix}3\\1\\1\end{bmatrix} \qquad \mathbf{m}_2 = \frac{1}{3}\begin{bmatrix}1\\3\\3\end{bmatrix}$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \frac{1}{16}\begin{bmatrix}3 & 1 & 1\\1 & 3 & -1\\1 & -1 & 3\end{bmatrix} \qquad \mathbf{C}_1^{-1} = \mathbf{C}_2^{-1} = \begin{bmatrix}8 & -4 & -4\\-4 & 8 & 4\\-4 & 4 & 8\end{bmatrix}$$

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

$$d_1(\mathbf{x}) = 4x_1 - 1.5$$

$$d_2(\mathbf{x}) = -4x_1 + 8x_2 + 8x_3 - 5.5$$

$$d_1(\mathbf{x}) - d_2(\mathbf{x}) = 8x_1 - 8x_2 - 8x_3 + 4 = 0$$

Các mạng nơ-ron

- Mạng nơ-ron đa lớp, được kết nối đầy đủ, có đầu vào là các vectơ mẫu
- Mạng nơ-ron tích chập, có khả năng chấp nhận hình ảnh làm đầu vào.
- Mặc dù mạng nơ-ron có thể sử dụng lan truyền ngược để tự động học các biểu diễn phù hợp để nhận dạng, bắt đầu bằng dữ liệu thô. Con người vẫn cần can thiệp để chỉ định các tham số như số lượng lớp, số lượng nơ-ron nhân tạo trên mỗi lớp và các hệ số khác.

Perceptron

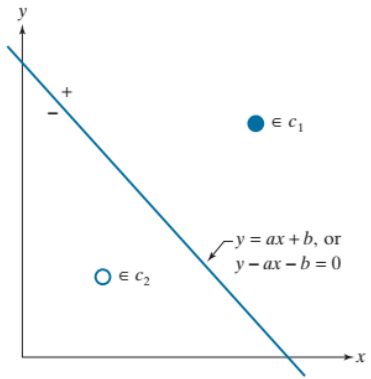
- Một đơn vị perceptron duy nhất học một ranh giới tuyến tính giữa hai lớp mẫu có thể phân tách tuyến tính.
- Ví dụ: phân tách hai lớp mẫu 2D,

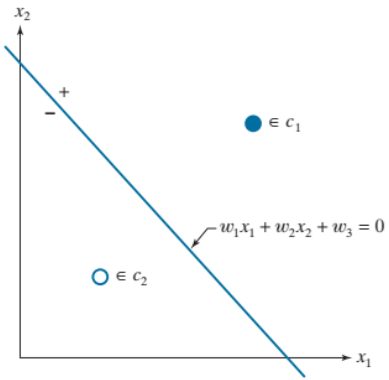
Mỗi lớp gồm 1 mẫu duy nhất.

$$y = ax + b$$

Tổng quát: $w_1 x_1 + w_2 x_2 + w_3 = 0$

Với điểm mẫu n chiều:



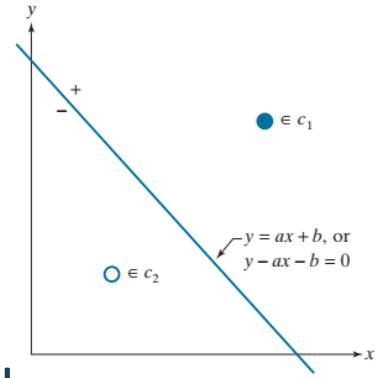


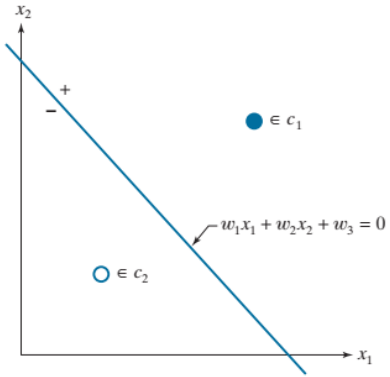
$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1} = 0$$

Perceptron

$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1} = 0$$

$$\boldsymbol{w}^T \mathbf{x} + w_{n+1} = 0$$





w: vector trọng số

w_{n+1}: bias (độ lệch)

Bài toán: tìm tập các trọng số sao cho:

$$\boldsymbol{w}^T \mathbf{x} + w_{n+1} = \begin{cases} > 0 & \text{if } \mathbf{x} \in c_1 \\ < 0 & \text{if } \mathbf{x} \in c_2 \end{cases}$$

Thuật toán Perceptron

- Đặt α > 0 biểu thị mức tăng hiệu chỉnh (tốc độ học), đặt **w**(1) là một vectơ có giá trị tùy ý và đặt $w_{n+1}(1)$ là một hằng số tùy ý.
- Thực hiện các bước sau cho k = 2, 3,… Đối với vectơ mẫu, **x**(k) , tại bước k:

1) If $\mathbf{x}(k) \in c_1$ and $\boldsymbol{w}^T(k)\mathbf{x}(k) + w_{n+1}(k) \leq 0$, let

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \alpha\mathbf{x}(k)$$
$$\omega_{n+1}(k+1) = \omega_{n+1}(k) + \alpha$$

2) If $\mathbf{x}(k) \in c_2$ and $\boldsymbol{w}^T(k)\mathbf{x}(k) + w_{n+1}(k) \geq 0$, let

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \alpha\mathbf{x}(k)$$
$$\omega_{n+1}(k+1) = \omega_{n+1}(k) - \alpha$$

3) Otherwise, let

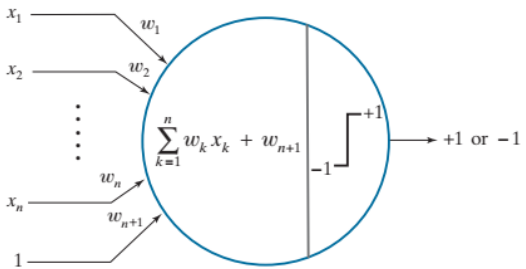
$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k)$$
$$\omega_{n+1}(k+1) = \omega_{n+1}(k)$$

Thuật toán Perceptron

$$\mathbf{x} \triangleq [x_1, x_2, \ldots, x_n, 1]^T$$

$$\boldsymbol{w} \triangleq [w_1, w_2, \ldots, w_n, w_{n+1}]^T$$

$$\boldsymbol{w}^T\mathbf{x} = \begin{cases} > 0 & \text{if } \mathbf{x} \in c_1 \\ < 0 & \text{if } \mathbf{x} \in c_2 \end{cases}$$



1′) If $\mathbf{x}(k) \in c_1$ and $\boldsymbol{w}^T(k)\mathbf{x}(k) \leq 0$, let

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \alpha\mathbf{x}(k)$$

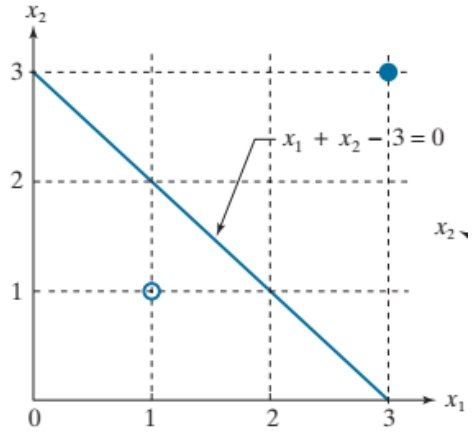
2′) If $\mathbf{x}(k) \in c_2$ and $\boldsymbol{w}^T(k)\mathbf{x}(k) \geq 0$, let

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \alpha\mathbf{x}(k)$$

3′) Otherwise, let

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

Ví dụ: Perceptron học ranh giới quyết định



- Hai vector mẫu: x = [3 3 1]^T và x = [1 1 1]^T thuộc lớp c_1 và c_2 tương ứng.
- Các mẫu được quay vòng qua perceptron theo thứ tự đó trong quá trình huấn luyện (**một lần lặp lại hoàn chỉnh thông qua tất cả các mẫu của quá trình huấn luyện được gọi là một epoch**)
- α= 1 và w(1)=0=[0 0 0]^T
- k = 1, **x**(1)= [3 3 1]^T ∈ c_1 và **w**(1)=[0 0 0]^T
- $\boldsymbol{w}^T(1)\mathbf{x}(1) = \begin{bmatrix}0 & 0 & 0\end{bmatrix}\begin{bmatrix}3\\3\\1\end{bmatrix} = 0$ bước 1’: $\boldsymbol{w}(2) = \boldsymbol{w}(1) + \alpha\mathbf{x}(1) = \begin{bmatrix}0\\0\\0\end{bmatrix} + (1)\begin{bmatrix}3\\3\\1\end{bmatrix} = \begin{bmatrix}3\\3\\1\end{bmatrix}$

Ví dụ: Perceptron học ranh giới quyết định

- Với k=2; **x**(2)= $[1\ 1\ 1]^T \in c_2$ và **w**(2)=$[3\ 3\ 1]^T$

$$\boldsymbol{w}^T(2)\mathbf{x}(2) = \begin{bmatrix}3 & 3 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = 7$$

- Kết quả là dương mà đáng lẽ nó phải âm vì vậy áp dụng bước 2':

$$\boldsymbol{w}(3) = \boldsymbol{w}(2) - \alpha\mathbf{x}(2) = \begin{bmatrix}3\\3\\1\end{bmatrix} - (1)\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}2\\2\\0\end{bmatrix}$$

- Chúng ta đã trải qua một giai đoạn (epoch) huấn luyện hoàn chỉnh với ít nhất một lần điều chỉnh, vì vậy chúng ta quay vòng lại cho tập huấn luyện.

Ví dụ: Perceptron học ranh giới quyết định

- Với k = 3, $\mathbf{x}(3)= [3\ 3\ 1]^T \in c_1$ và $\mathbf{w}(3)=[2\ 2\ 0]^T$. Tích vô hướng là dương (tức là 6) như nó cần phải vậy vì $\mathbf{x}(3) \in c_1$. Do đó, Bước 3' được áp dụng và vectơ trọng số không thay đổi:

$$\boldsymbol{w}(4) = \boldsymbol{w}(3) = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

- Đối với k=4, $\mathbf{x}(4)= [1\ 1\ 1]^T \in c_2$ và $\mathbf{w}(4)=[2\ 2\ 0]^T$. Tích vô hướng của nó là dương (tức là 4) và đáng lẽ nó phải âm, vì vậy Bước 2' được áp dụng:

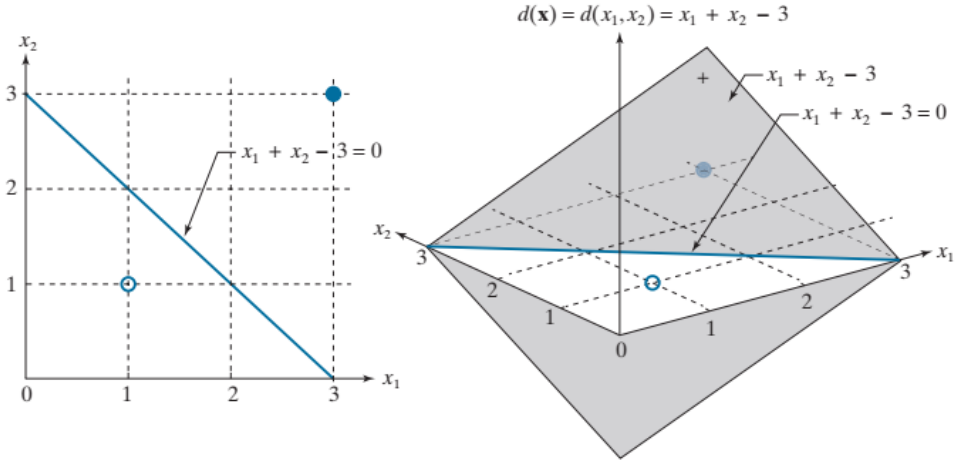
$$\boldsymbol{w}(5) = \boldsymbol{w}(4) - \alpha\mathbf{x}(4) = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} - (1)\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Ví dụ: Perceptron học ranh giới quyết định

- Thuật toán hội tụ với vectơ trọng số:

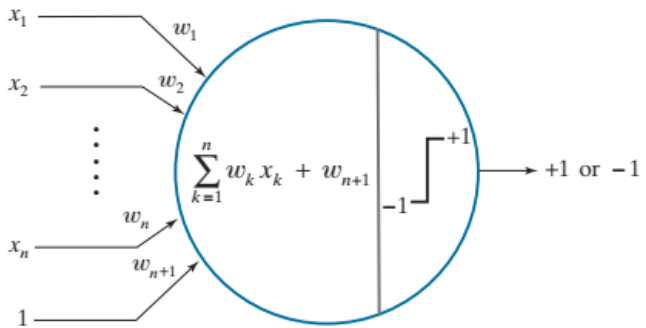
$$\boldsymbol{w} = \boldsymbol{w}(12) = \begin{bmatrix} 1 \\ 1 \\ -3 \end{bmatrix}$$

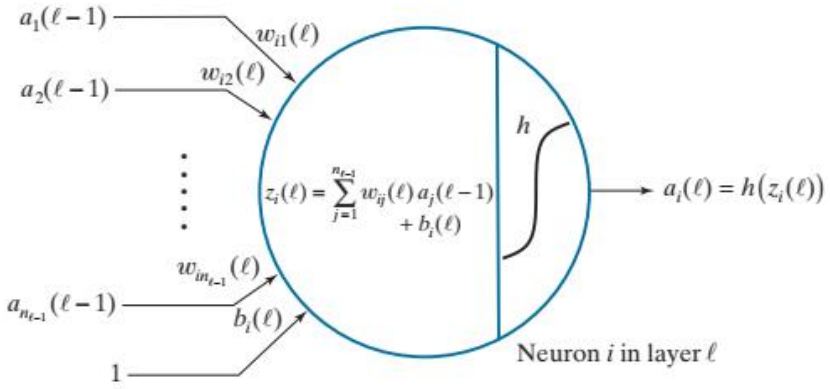
- Ranh giới quyết định: $x_1 + x_2 - 3 = 0$



Mô hình nơ-ron nhân tạo

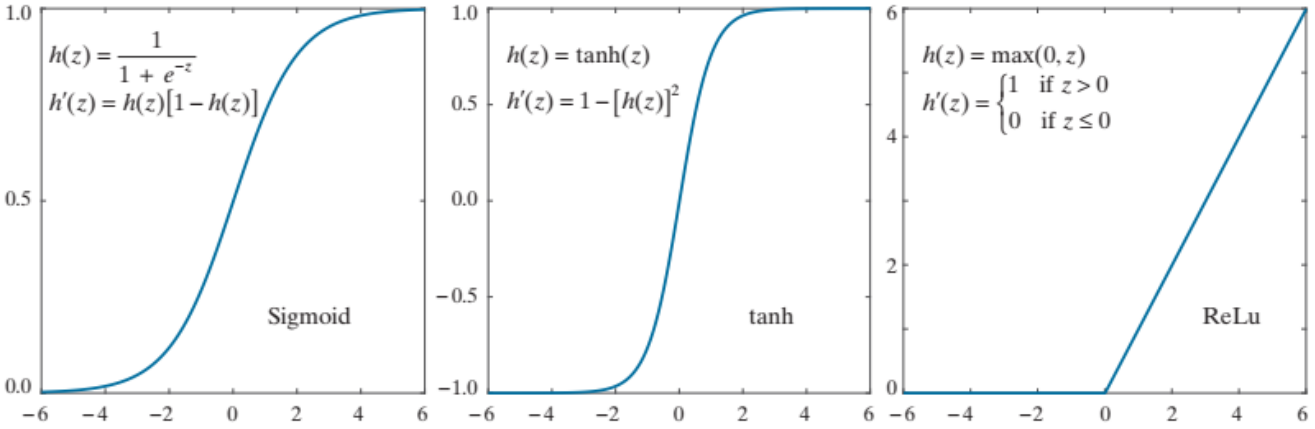
- Mạng lưới nơ-ron gồm các thành phần tính toán giống như perceptron được kết nối với nhau được gọi là mạng nơ-ron nhân tạo.
- Những nơ-ron này thực hiện các tính toán tương tự như perceptron, nhưng chúng khác ở cách chúng xử lý kết quả tính toán.





Một số hàm kích hoạt

- df



(a) Sigmoid (b) tanh (c) ReLu

Kết nối các nơ-ron để tạo thành một mạng nơ-ron được kết nối đầy đủ

- Lớp đầu vào có các nút là thành phần của vectơ mẫu đầu vào **x**. Do đó, đầu ra (giá trị kích hoạt) của lớp đầu tiên là các giá trị của các phần tử của **x**.
- Đầu ra của tất cả các nút khác là các giá trị kích hoạt của nơ-ron trong một lớp cụ thể.
- Mỗi lớp trong mạng có thể có số lượng nút khác nhau, nhưng mỗi nút có một đầu ra duy nhất.

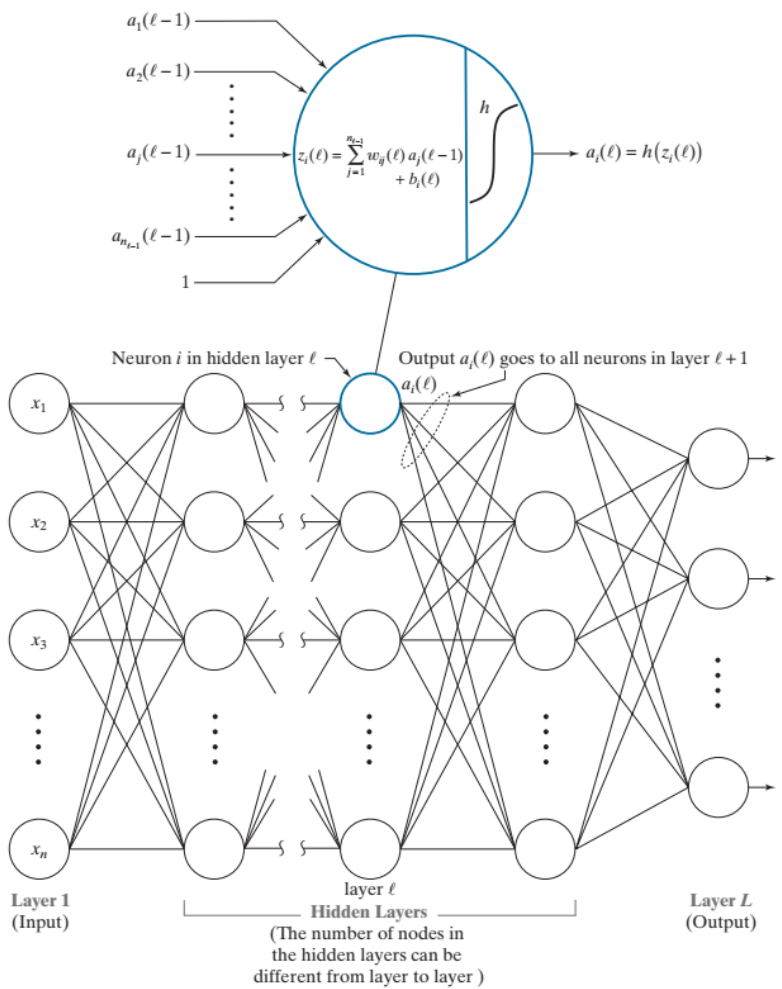


FIGURE 12.31 General model of a feedforward, fully connected neural net. The neuron is the same as in Fig. 12.29. Note how the output of each neuron goes to the input of all neurons in the following layer, hence the name *fully connected* for this type of architecture.

Ví dụ một mạng nơ-ron kết nối đầy đủ (hàm kích hoạt sigmoid)

$$z_1(2) = \sum_{j=1}^{3} w_{1j}(2)a_j(1) + b_1(2) = (0.1)(3) + (0.2)(0) + (0.6)(1) + 0.4 = 1.3$$

We obtain the output of this node using Eqs. (12-51) and (12-55):

$$a_1(2) = h(z_1(2)) = \frac{1}{1+e^{-1.3}} = 0.7858$$

A similar computation gives the value for the output of the second node in the second layer,

$$z_2(2) = \sum_{j=1}^{3} w_{2j}(2)a_j(1) + b_2(2) = (0.4)(3) + (0.3)(0) + (0.1)(1) + 0.2 = 1.5$$

and

$$a_2(2) = h(z_2(2)) = \frac{1}{1+e^{-1.5}} = 0.8176$$

We use the outputs of the nodes in layer 2 to obtain the net values of the neurons in layer 3:

$$z_1(3) = \sum_{j=1}^{2} w_{1j}(3)a_j(2) + b_1(3) = (0.2)(0.7858) + (0.1)(0.8176) + 0.6 = 0.8389$$

The output of this neuron is

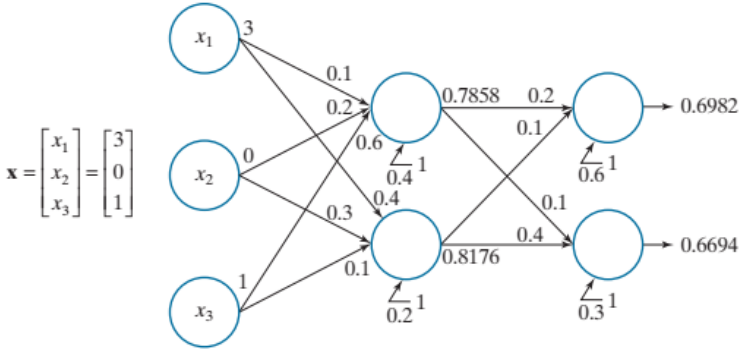
$$a_1(3) = h(z_1(3)) = \frac{1}{1+e^{-0.8389}} = 0.6982$$

Similarly,

$$z_2(3) = \sum_{j=1}^{2} w_{2j}(3)a_j(2) + b_2(3) = (0.1)(0.7858) + (0.4)(0.8176) + 0.3 = 0.7056$$

and

$$a_2(3) = h(z_2(2)) = \frac{1}{1+e^{-0.7056}} = 0.6694$$



Nếu chúng ta sử dụng mạng này để phân loại đầu vào, chúng ta sẽ nói rằng mẫu **x** thuộc về lớp c_1 bởi vì $a_1(L) > a_2(L)$ trong đó L = 3 và n_L = 2 trong trường hợp này.