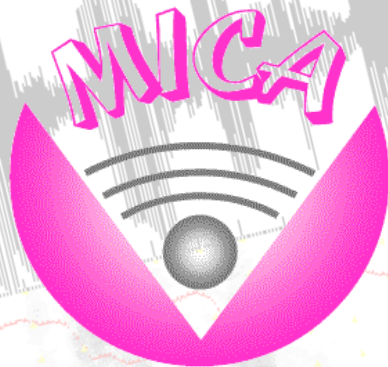




# Mạng neuron và ứng dụng trong xử lý tín hiệu



Giảng viên  
**Trần Thị Thanh Hải**

**International Research Institute MICA**  
Multimedia, Information, Communication & Applications  
UMI 2954

Hanoi University of Science and Technology  
1 Dai Co Viet - Hanoi - Vietnam

**Bài 2:**  
**Mô hình neuron**  
**và**  
**các kiến trúc mạng**



# Mục tiêu

- Mô hình toán học cơ bản của một neuron
- Giải thích tại sao các neuron nhân tạo có thể kết nối với nhau để tạo ra mạng neuron nhân tạo
- Minh họa thao tác cơ bản của mạng thông qua ví dụ
- Không đề cập đến các loại mạng song minh họa các khối cơ bản
- Các kiến trúc phức tạp sẽ được giới thiệu trong các bài sau



# Nội dung trình bày

---

- I. Mô hình neuron
- II. Neuron một đầu vào
- III. Neuron nhiều đầu vào
- IV. Các lớp neurons
- V. Mạng đa lớp
- VI. Một số ví dụ



# Nội dung trình bày

---

- I. Mô hình neuron**
- II. Neuron một đầu vào
- III. Neuron nhiều đầu vào
- IV. Các lớp neurons
- V. Mạng đa lớp

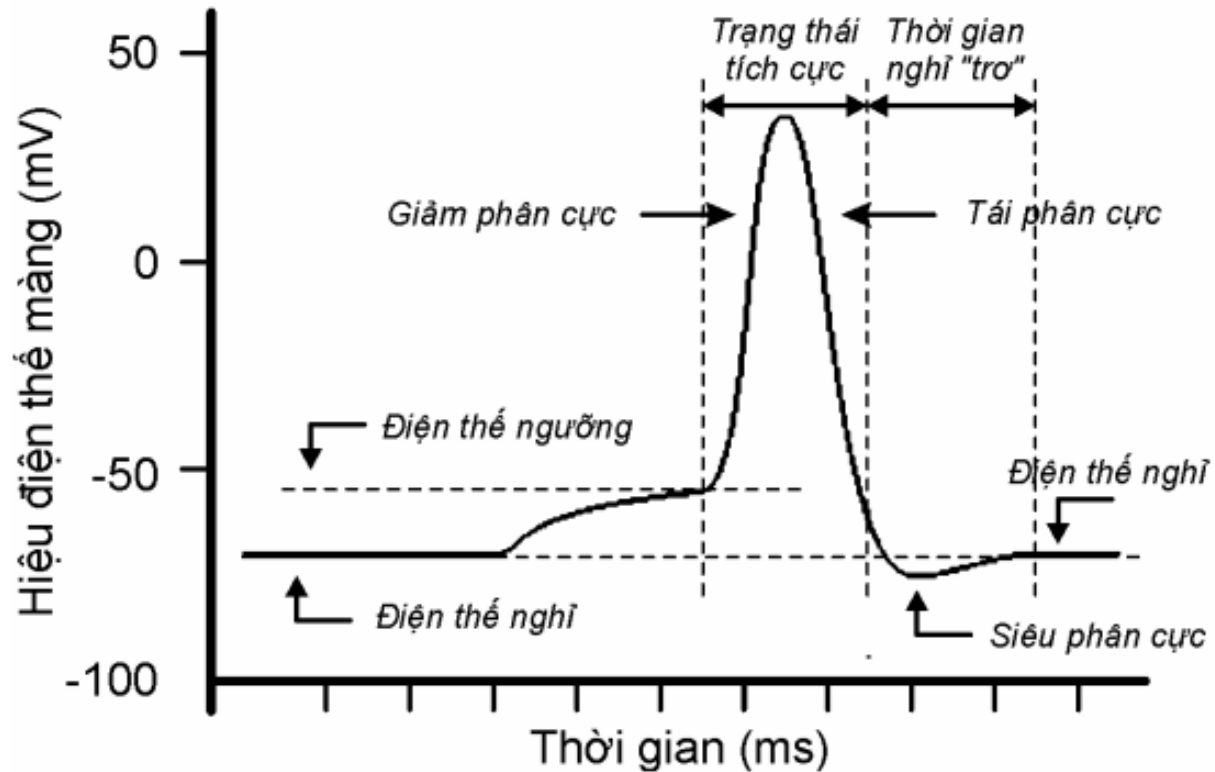


# I. Mô hình neuron

- **Mô hình đề xuất năm 1943 bởi McCulloch-Pitts**
- **Neuron sinh học có các đặc điểm sau**
  - ◆ Có một hoặc nhiều tín hiệu đầu vào
  - ◆ Mức độ phụ thuộc vào các tín hiệu đầu vào khác nhau
  - ◆ Khi tổng kích thích đầu vào vượt quá một ngưỡng neuron sẽ tạo ra một xung tín hiệu đầu ra



# Ví dụ một neuron khi bị kích thích



Hình 2.2: Ví dụ hình dạng một xung điện áp đầu ra của một no-rôn khi bị kích thích

# I. Mô hình neuron

## II.1 Các thành phần cơ bản

- **Mô hình mạng neuron gồm 3 thành phần**
  - ◆ Hệ thống ghép nối thần kinh (synapse)
  - ◆ Bộ cộng
  - ◆ Hàm kích hoạt





# II. Neuron một đầu vào

## II.1 Các thành phần cơ bản

### ■ Neuron một đầu vào

- ◆ Đầu vào là một số  $p$
- ◆ Trọng số  $w$
- ◆ Bias  $b$ 
  - ★  $b$  và  $w$  là các tham số có thể thay đổi theo một luật học để quan hệ giữa đầu vào và đầu ra đạt được một mục đích nào đó
- ◆ Hàm truyền đạt  $f$ 
  - ★ Được lựa chọn bởi người thiết kế mạng

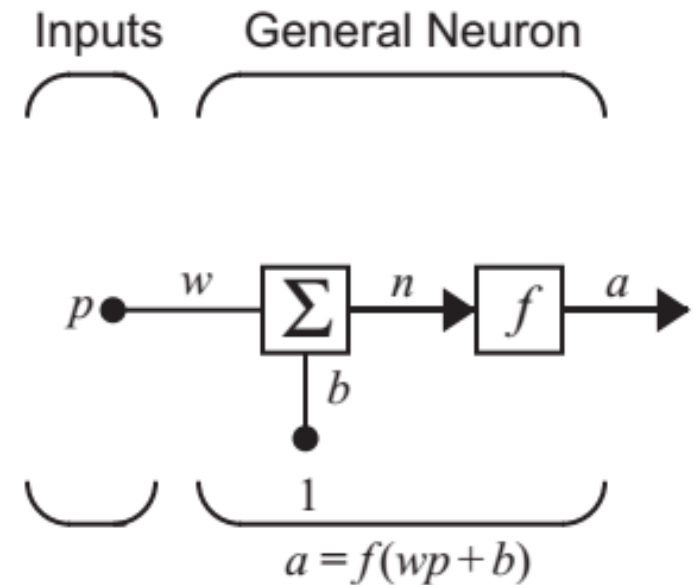


Figure 2.1 Single-Input Neuron

$$a = f(wp + b)$$

# II. Neuron một đầu vào

## II.2. Hàm truyền đạt

- Các neuron sử dụng hàm ngưỡng còn được gọi là neuron McCulloch-Pitts do các tác giả này đưa ra vào năm 1943
- Hàm ngưỡng
  - ◆ 😊 Đơn giản
  - ◆ ☹️ không tồn tại đạo hàm tại  $u = 1$  nên không thể sử dụng một số pp dựa trên Gradient



# II. Neuron một đầu vào

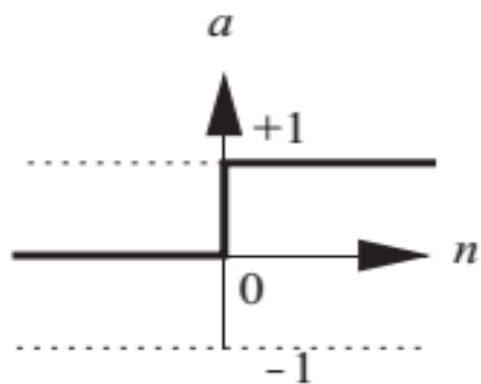
## II.2. Hàm truyền đạt

- Có thể là hàm tuyến tính hoặc phi tuyến
- Được lựa chọn để thỏa mãn một số đặc tính của bài toán
- Một số dạng hàm truyền đạt



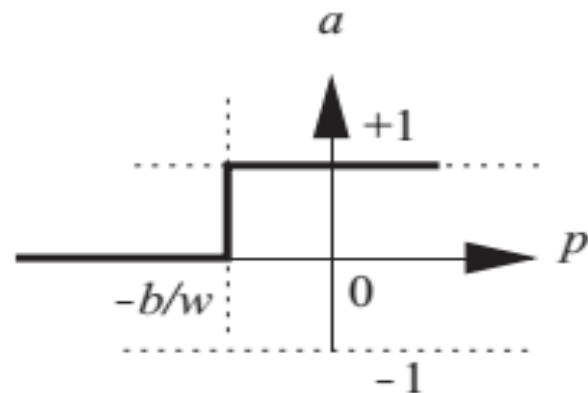
# II. Neuron một đầu vào

## II.2. Hàm truyền đạt



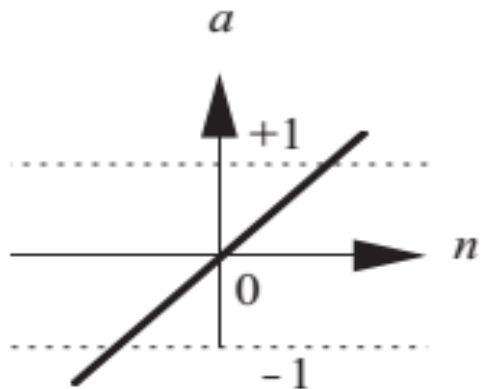
$a = \text{hardlim}(n)$

Hard Limit Transfer Function



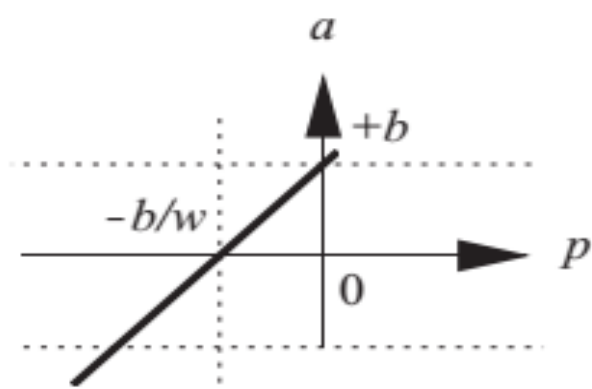
$a = \text{hardlim}(wp + b)$

Single-Input *hardlim* Neuron



$a = \text{purelin}(n)$

Linear Transfer Function



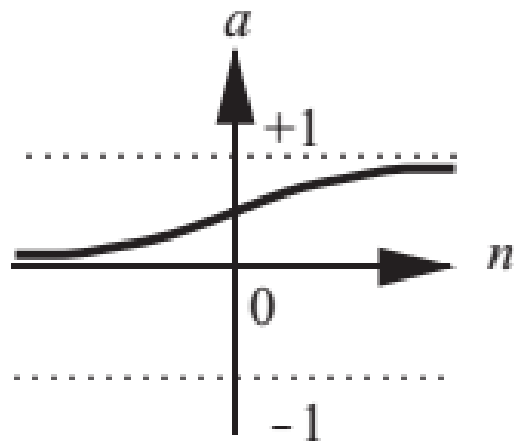
$a = \text{purelin}(wp + b)$

Single-Input *purelin* Neuron



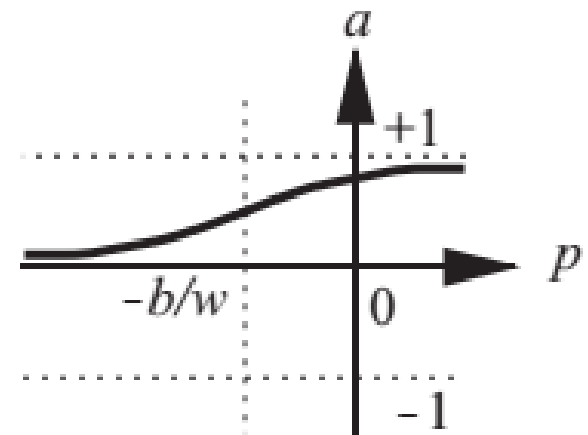
# II. Neuron một đầu vào

## II.2. Hàm truyền đạt



$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function



$$a = \text{logsig}(wp + b)$$

Single-Input  $\text{logsig}$  Neuron

# II. Neuron một đầu vào

## II.2. Hàm truyền đạt

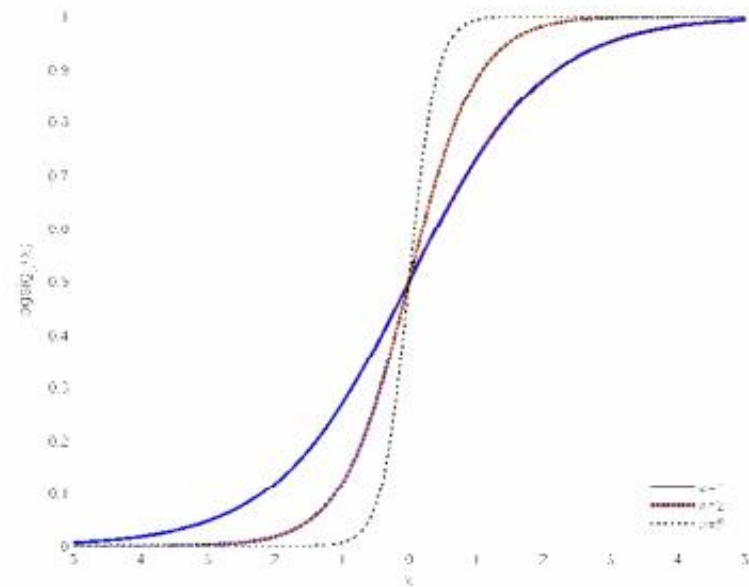
- Hàm logsig hoặc sigmoid là hàm truyền đạt được sử dụng nhiều nhất.
- Hàm có dạng đồng biến, có thể coi là trung gian giữa hàm tuyến tính và hàm ngưỡng
  - ◆ A: hệ số dốc
  - ◆ Khi  $a \Rightarrow$  vô cùng,  $f(u)$  tiến đến hàm ngưỡng
- Có ưu điểm so với hàm ngưỡng là tồn tại đạo hàm

$$\text{logsig}_a(u) = \frac{1}{1 + e^{-au}}$$

# II. Neuron một đầu vào

## II.2. Hàm truyền đạt: ví dụ logsig

$$a = \frac{1}{1 + e^{-n}}$$



Hình 2.6: Hàm truyền đạt logsig với các hệ số dốc a khác nhau

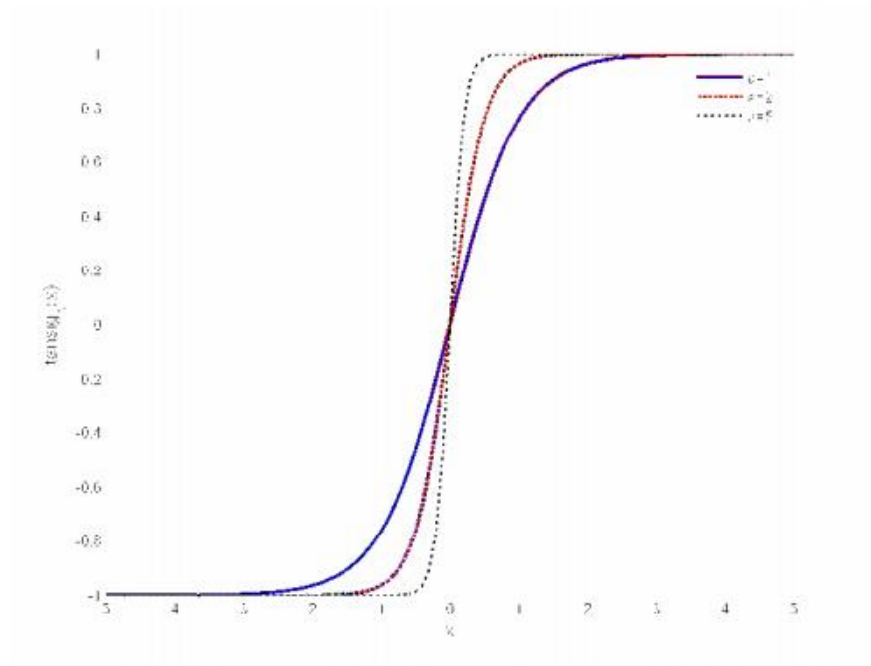


# II. Neuron một đầu vào

## II.2. Hàm truyền đạt

- Hàm logsig có giá trị đầu ra trong khoảng  $[0, 1]$
- Trong trường hợp cần đầu ra có giá trị âm, ta dùng tansig thuộc  $[-1, 1]$

$$f(u) = \tanh_a(u) = \frac{e^{au} - e^{-au}}{e^{au} + e^{-au}}$$
$$= 2 \cdot \text{logsig}_{2a}(u) - 1$$



Hình 2.7: Hàm truyền đạt tansig với các hệ số dốc  $a$  khác nhau



# II. Neuron một đầu vào

## II.2. Hàm truyền đạt

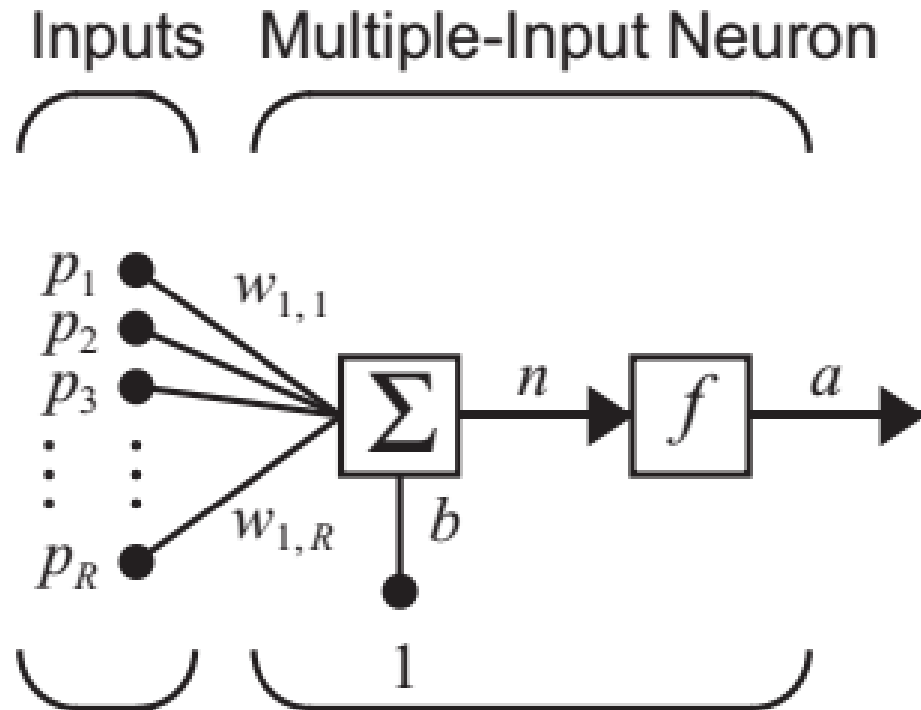
Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$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

Table 2.1 Transfer Functions

# III. Neuron nhiều đầu vào

## III.1 Mô hình chung

- Thông thường một neuron có nhiều hơn một đầu vào
- Một neuron có  $R$  đầu vào được biểu diễn như sau



◆ Kích thích  $p_1, p_2, \dots, p_R$

◆ Các trọng số:  $w_{1,1}, w_{1,2}, \dots, w_{1,R}$

◆ Bias:  $b$

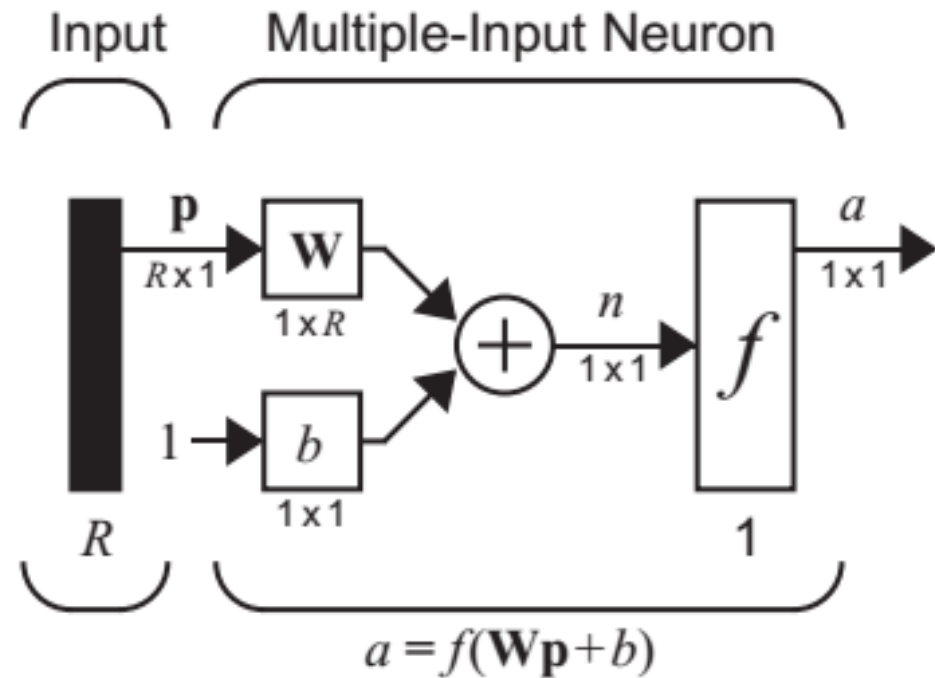
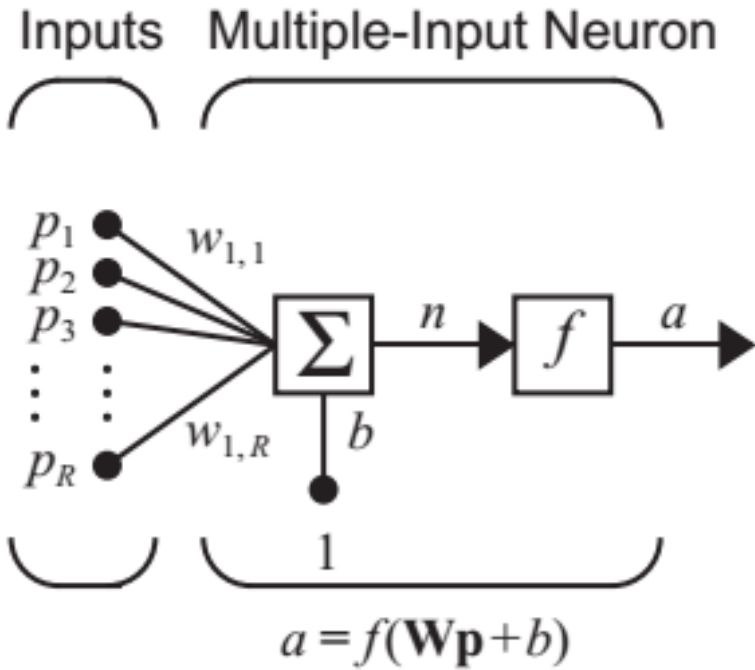
◆ Đầu vào của

neuron:  $n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$ .

$$n = \mathbf{Wp} + b$$

# III. Neuron nhiều đầu vào

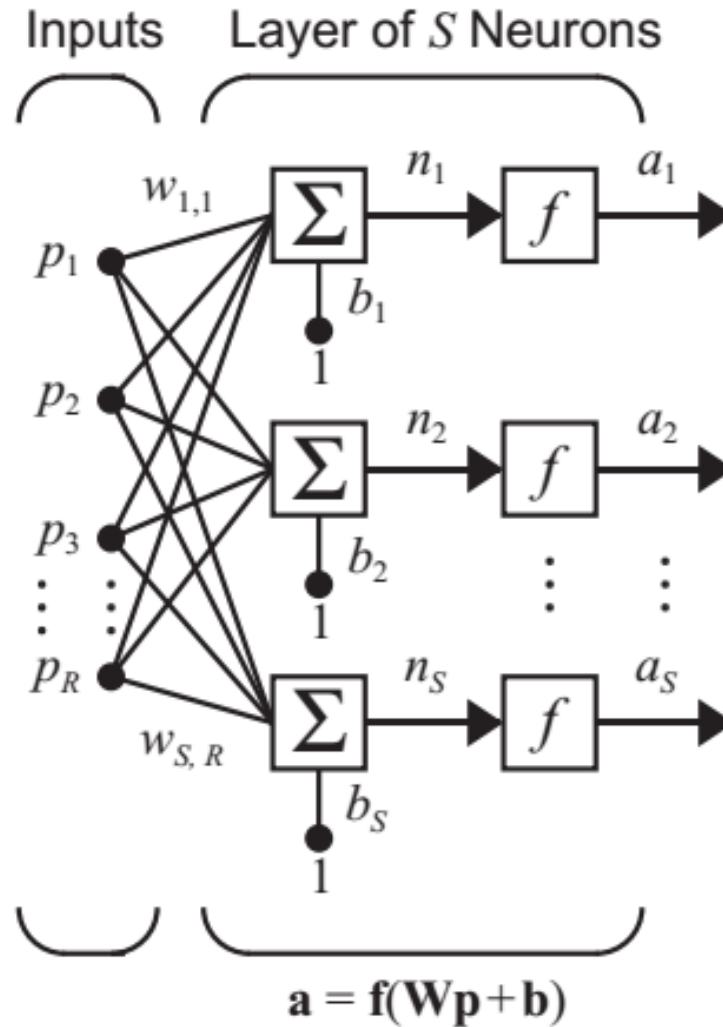
## III.1 Cách biểu diễn gọn



Abbreviated Notation

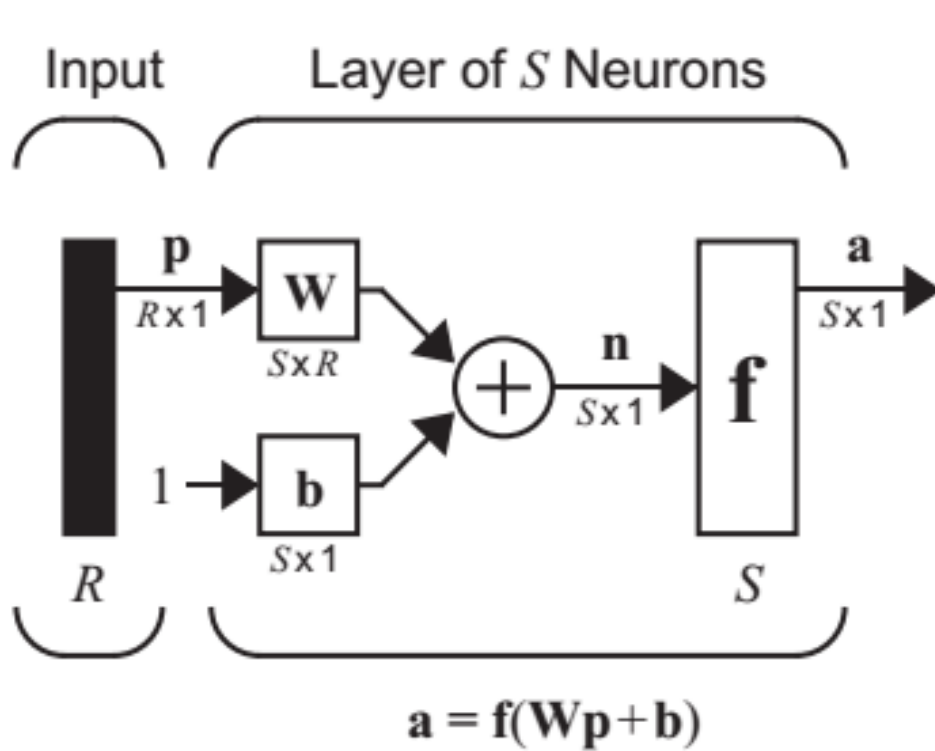
# IV. Kiến trúc mạng neuron

## IV.1 Lớp các neuron



# IV. Kiến trúc mạng neuron

## IV.2 Các ký pháp viết gọn

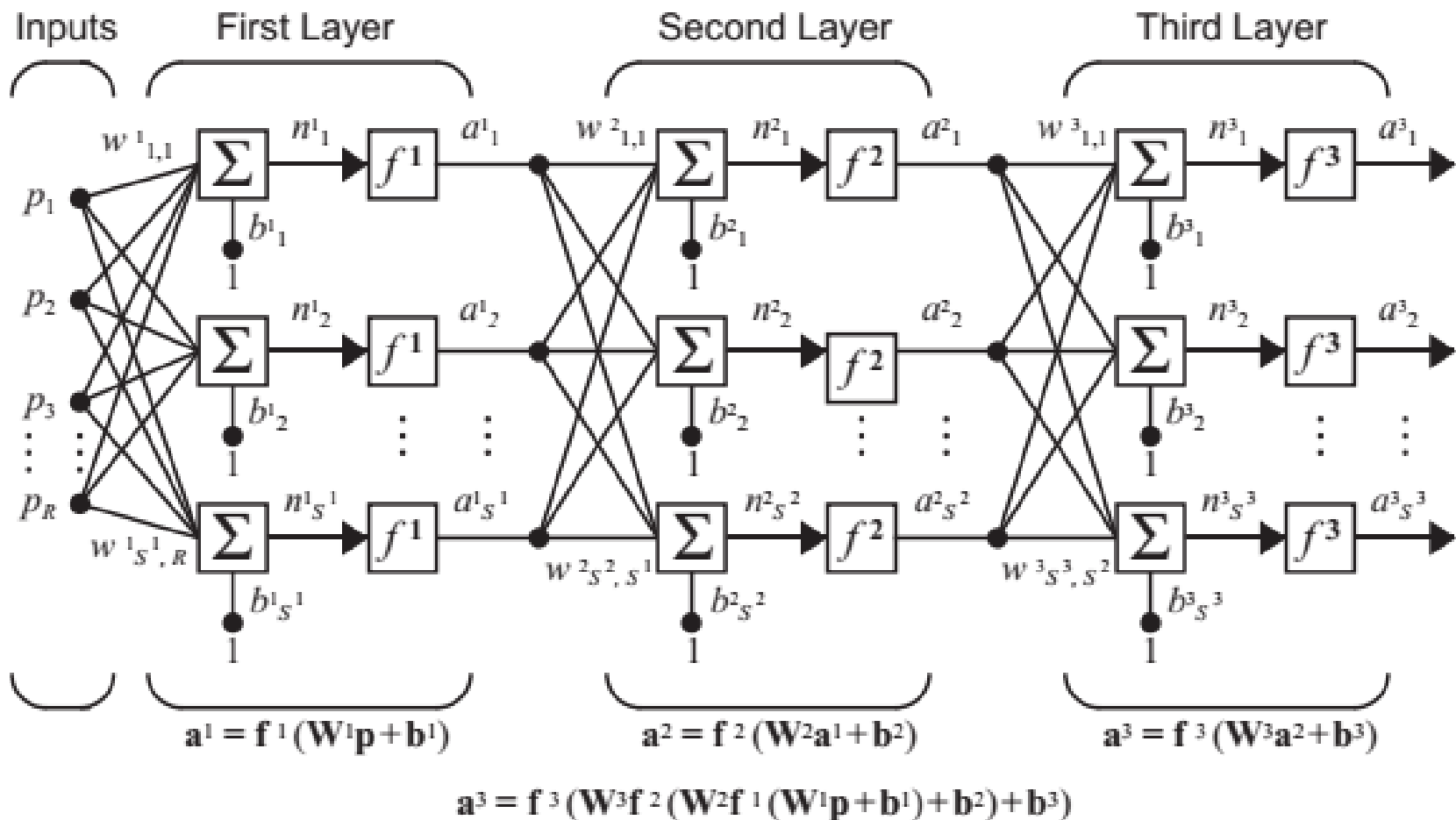


$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

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

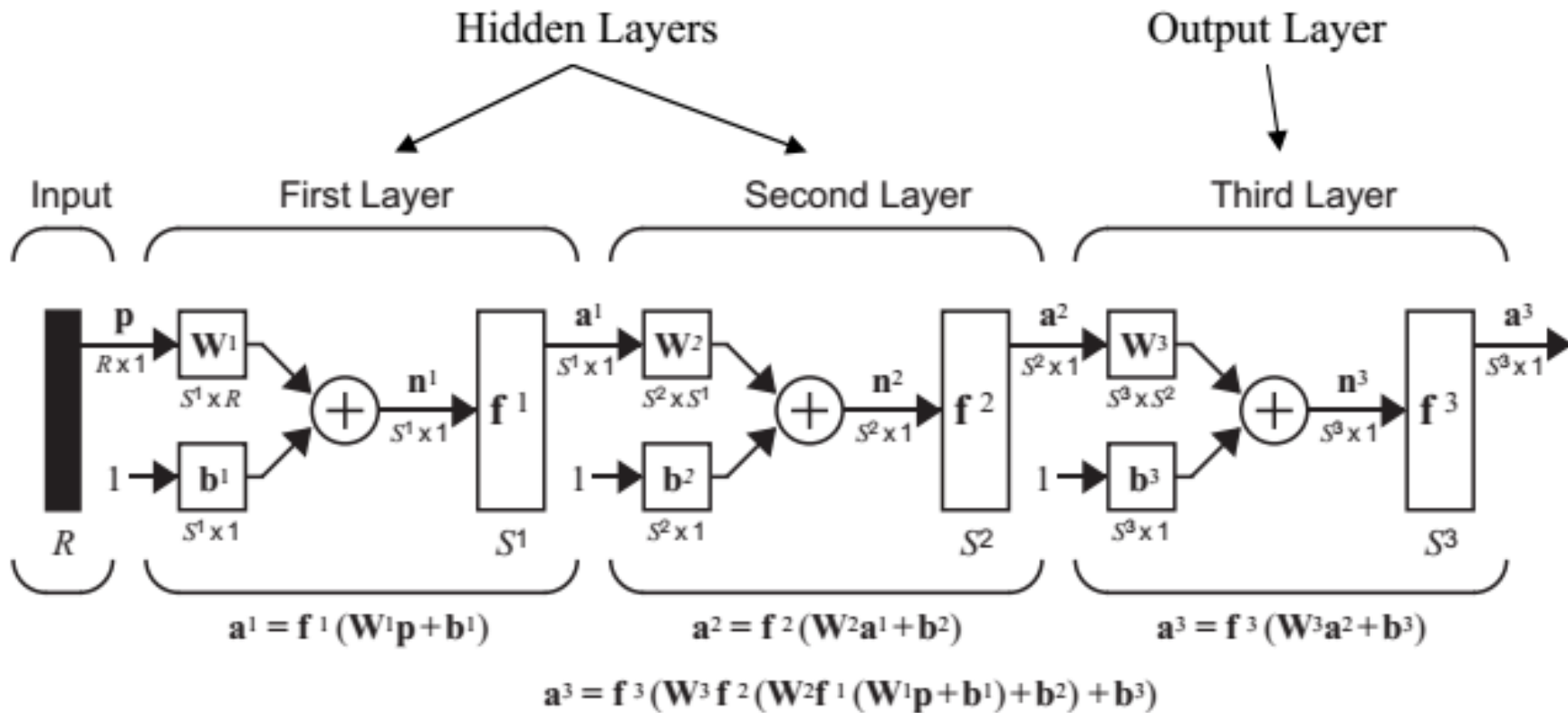
# IV. Kiến trúc mạng neuron

## IV.3 Mạng neuron đa lớp



# IV. Kiến trúc mạng neuron

## IV.3 Mạng neuron đa lớp – biểu diễn gọn



# Ví dụ 1:

- Mạng neuron có đầu vào là 2.0, trọng số là 2.3 và bias là -3

- ◆ Tính đầu vào của hàm truyền đạt ?
- ◆ Tính đầu ra của neuron ?

## ■ Trả lời

- ◆ Đầu vào của hàm truyền đạt

$$n = wp + b = (2.3)(2) + (-3) = 1.6$$

- ◆ Đầu ra: không xác định vì không định nghĩa hàm truyền đạt



# Ví dụ 1:

- Mạng neuron có đầu vào là 2.0, trọng số là 2.3 và bias là -3
  - ◆ Tính đầu vào của hàm truyền đạt ?
  - ◆ Tính đầu ra của neuron với các hàm truyền đạt

For the hard limit transfer function:

## ■ Trả lời

$$a = \text{hardlim}(1.6) = 1.0$$

. For the linear transfer function:

$$a = \text{purelin}(1.6) = 1.6$$

i. For the log-sigmoid transfer function:

$$a = \text{logsig}(1.6) = \frac{1}{1 + e^{-1.6}} = 0.8320$$

## Ví dụ 2:

- Mạng neuron có đầu vào là 2.0, trọng số là 2.3 và bias là -3
  - ◆ Tính đầu vào của hàm truyền đạt ?
  - ◆ Tính đầu ra của neuron với các hàm truyền đạt

For the hard limit transfer function:

- Trả lời

$$a = \text{hardlim}(1.6) = 1.0$$

. For the linear transfer function:

$$a = \text{purelin}(1.6) = 1.6$$

i. For the log-sigmoid transfer function:

$$a = \text{logsig}(1.6) = \frac{1}{1 + e^{-1.6}} = 0.8320$$

## Ví dụ 3

- Cho Neuron hai đầu vào  $b = 1.2$ ;  $W = [3 \ 2]$ ;  $p = [5, 6]^T$
- Tính đầu ra của neuron với các hàm truyền đạt sau

$$n = \mathbf{W}\mathbf{p} + b = [3 \ 2] \begin{bmatrix} -5 \\ 6 \end{bmatrix} + (1.2) = -1.8 .$$

**i.**  $a = \text{hardlims}(-1.8) = -1$

**ii.**  $a = \text{satlin}(-1.8) = 0$

**iii.**  $a = \text{tansig}(-1.8) = -0.9468$

# Ví dụ 4

- Một mạng neuron có 6 đầu vào, 2 đầu ra
- Đầu ra nằm trong khoảng  $[0, 1]$  và là các biến liên tục
- Hãy xác định kiến trúc mạng
  - ◆ Bao nhiêu neuron
  - ◆ Số chiều của ma trận trọng số
  - ◆ Kiểu hàm truyền đạt
  - ◆ Bias ?



# Ví dụ 4

## ■ Trả lời

- ◆ Hai neuron: mỗi neuron cho một đầu ra
- ◆ Ma trận trọng số có hai hàng tương ứng với hai neuron, 6 cột tương ứng với 6 đầu vào
- ◆ Hàm truyền đạt là hàm liên tục, nên phù hợp nhất là logsig
- ◆ Không đủ thông tin để xác định bias

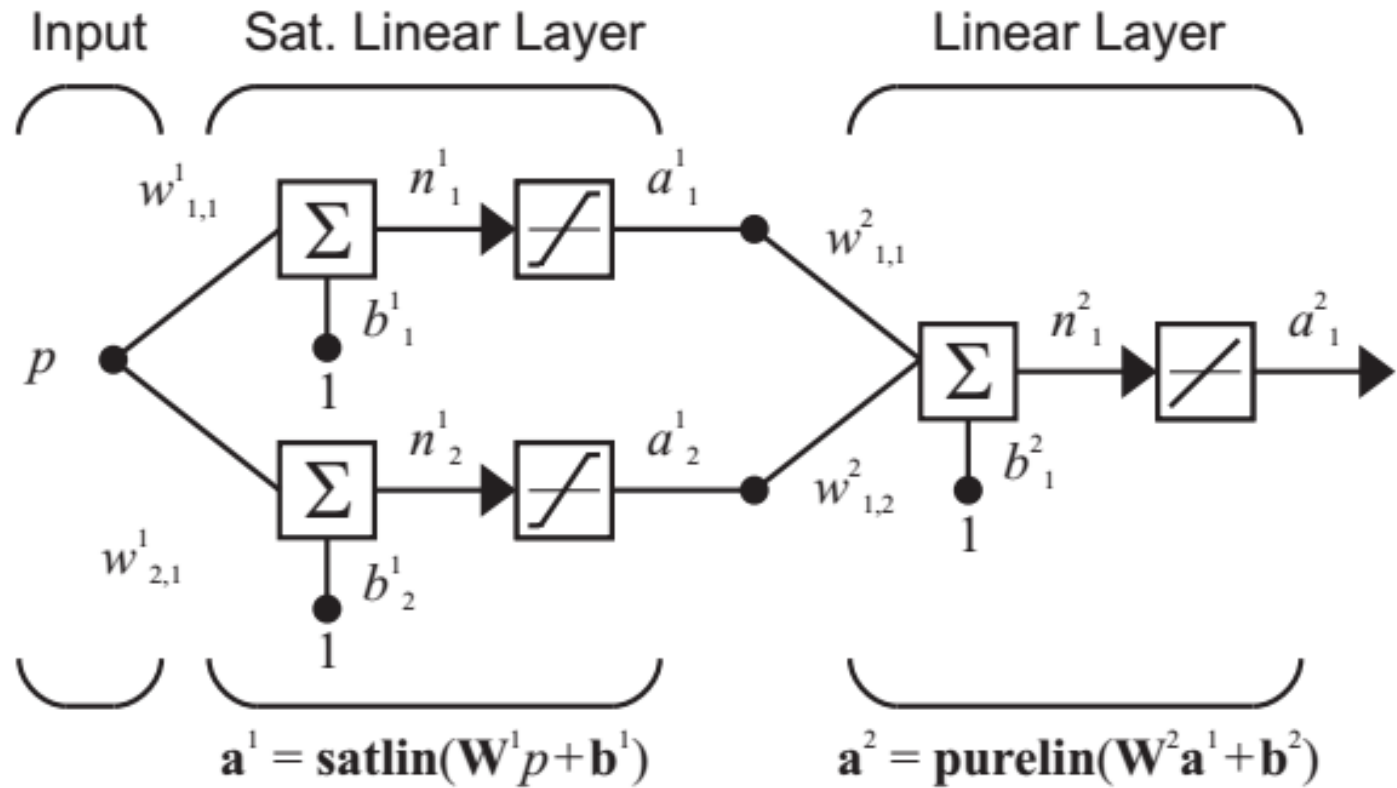


# Giới thiệu công cụ Matlab Toolbox NN

---



# Ví dụ 5



$$w_{1,1}^1 = 2, w_{2,1}^1 = 1, b_1^1 = 2, b_2^1 = -1, w_{1,1}^2 = 1, w_{1,2}^2 = -1, b_1^2 = 0$$