# Cơ sở hệ mờ và mạng nơ ron

Ts. Nguyễn H. Nam
Bộ môn Điều khiển tự động – Viện Điện – BKHN

https://sites.google.com/view/n2c

# Tài liệu tham khảo


- [1]. Lý thuyết điều khiển mờ, Phan Xuân Minh và Nguyễn Doãn Phước.
- [2]. Fuzzy Logic Toolbox User's Guide, R2014b.
- [3]. Fuzzy set theory and its applications, H.-J. Zimmermann.
- [4]. Neural network design, Martin Hagan.
- [5]. Neural Network Toolbox User's Guide.
- [6]. Giáo trình Điều khiển mờ và mạng nơ-ron

# Nội dung

- Chương 1: Tập mờ
- Chương 2: Hệ suy luận mờ
- Chương 3: Một số ứng dụng của hệ mờ
- Chương 4: Mạng nơ-ron nhân tạo
- Chương 5: Mạng nhiều lớp và ứng dụng

# Chương 1 Tập mờ

- 1.1 Tập hợp
- 1.2 Tập mờ và hàm liên thuộc
- 1.3 Các phép toán trên tập mờ

# 1.1 Tập hợp

- 1.1.1 Lịch sử
- 1.1.2 Tập hợp và hàm liên thuộc
- 1.1.3 Các phép toán

# 1.1.1 Lịch sử

- Lý thuyết tập mờ được phát triển từ năm 1965 bởi Lotfi A. Zadeh [1].
- Bộ điều khiển mờ ra đời năm 1975 bởi E. H. Mamdani [2].

[1] http://www.cs.berkeley.edu/~zadeh/ [2] http://www.iis.ee.ic.ac.uk/~amamdani/

# 1.1.2 Tập hợp

Ví dụ 1.1: Tập hợp các bạn nữ trong lớp $A = \{Hoa, Mai, Phượng\ \ldots, Thanh\}$

Ví dụ 1.2: Tập hợp các sai số dương và nhỏ hơn 10%

$$B = \{e | 0 < e < 0{,}1\}$$

- Phương pháp mô tả tập hợp:
  - Liệt kê các phần tử
  - Chỉ ra tính chất đặc trưng cho các phần tử
- Giả sử A là một tập hợp đã cho, nếu một phần tử a thuộc tập hợp A ta ký hiệu là $\boldsymbol{a \in A}$, nếu không thuộc tập hợp A ta ký hiệu là $\boldsymbol{a \notin A}$.

# 1.1.2 Tập hợp

- **Tập hợp rỗng:** $\Phi$

  Là tập hợp không chứa phần tử nào.

Ví dụ 1.3: Tập hợp các bạn sinh viên trong lớp có tuổi nhỏ hơn 12

- **Tập hợp con:** $A \subset B \Leftrightarrow \forall a, a \in A \Longrightarrow a \in B$

  \- Tập hợp A được gọi là tập hợp con của tập hợp B nếu như mọi phần tử thuộc tập hợp A cũng thuộc tập hợp B.

  \- *Nếu A không phải là tập hợp con của B, ta ký hiệu là:* $A \not\subset B$

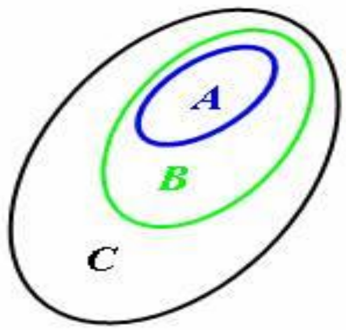
  \- $A = B \Leftrightarrow A \subset B$ *và* $B \subset A$ (hai tập hợp bằng nhau)

Ví dụ 1.4:

Tập hợp các số nguyên là tập hợp con của tập hợp các số thực

# 1.1.2 Tập hợp

- **Các tính chất:**
  - ✓ *$A \subset A$ với mọi tập hợp A.*
  - ✓ Nếu $A \subset B$ và $B \subset C$ thì $A \subset C$
  - ✓ $\Phi \subset A$ với mọi tập hợp $A$.




**Hình 1.1 Tính chất của tập hợp con**

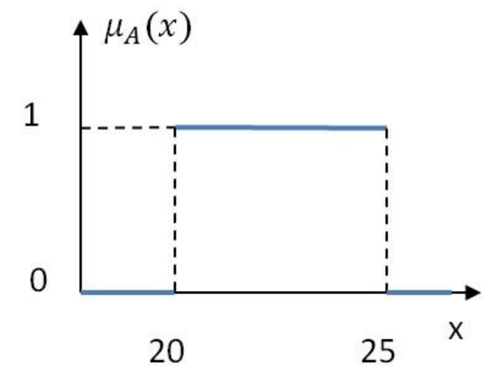
- **Hàm liên thuộc** *(phụ thuộc):*

$$\mu_A(x) = \begin{cases} 1, nếu\ x \in A \\ 0, nếu\ x \notin A \end{cases}$$

<u>Ví dụ 1.5:</u> Tập hợp nhiệt độ mát

$$A = \{x | 20\ ^oC \le x \le 25\ ^oC\}$$

$$\mu_A(x) = \begin{cases} 1, nếu\ x \in A \\ 0, nếu\ x \notin A \end{cases}$$




**Hình 1.2 Hàm liên thuộc của tập A**

# 1.1.3 Các phép toán

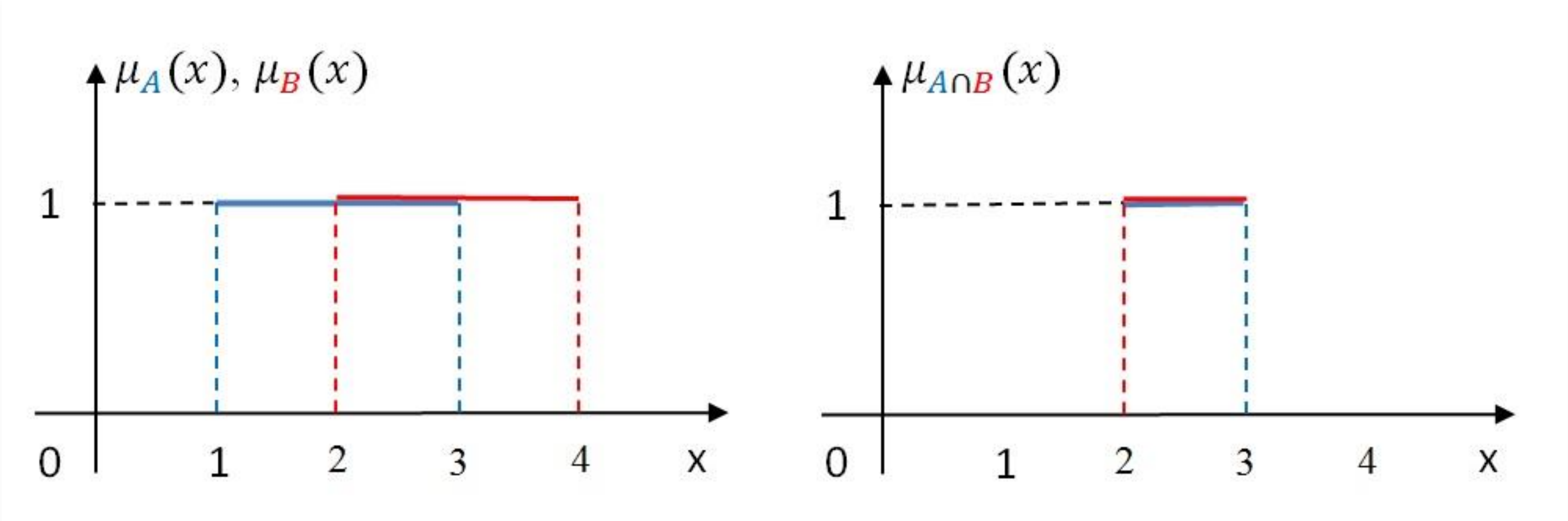
- **A) Phép giao của hai tập hợp**

$$A \cap B = \{x | x \in A \text{ và } x \in B\}$$

$$\mu_{A \cap B}(x) = \mu_A(x) \, . \mu_B(x) = \min(\mu_A(x), \mu_B(x))$$

Ví dụ 1.6: $A = \{x | 1 < x < 3\}$ và $B = \{x | 2 < x < 4\}$

$$A \cap B = \{x | 2 < x < 3\}$$




**Hình 1.3 Hàm liên thuộc cho ví dụ 1.6**

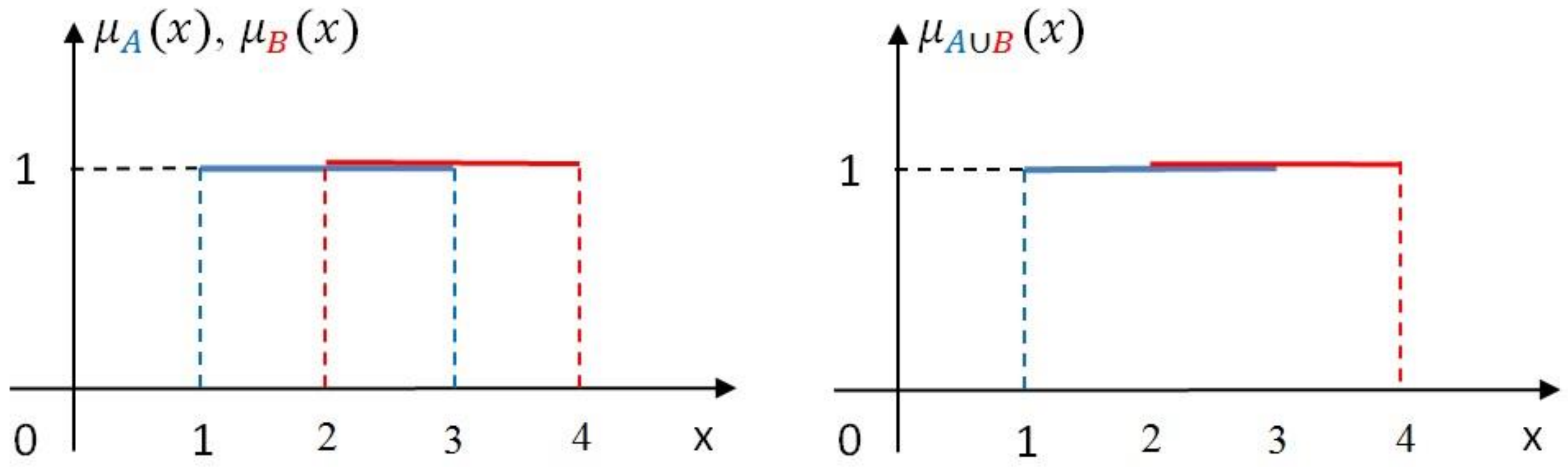
# 1.1.3 Các phép toán

- **B) Phép hợp của hai tập hợp**

$$A \cup B = \{x | x \in A \text{ hoặc } x \in B\}$$

$$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x) = \max(\mu_A(x), \mu_B(x))$$

<u>Ví dụ 1.7:</u> (tiếp theo Ví dụ 1.6) $\mu_{A\cup B}(x) = \{x | 1 < x < 4\}$




**Hình 1.4 Hàm liên thuộc cho ví dụ 1.7**

# 1.1.3 Các phép toán

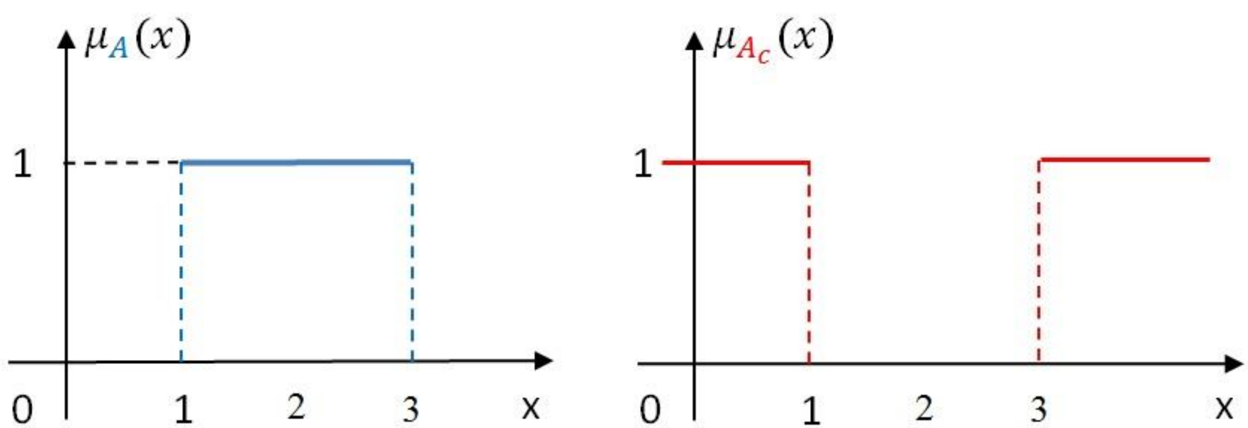
- **C) Phép bù của một tập hợp**

Giả sử tập hợp $A \subset U$, tập hợp bù của $A$:

$$A_c = \{x|x \in U \text{ và } x \notin A\} \text{ (c: complement)}$$

$$\mu_{A_c}(x) = \begin{cases} 1, nếu\ x \notin A \\ 0, nếu\ x \in A \end{cases} = 1 - \mu_A(x)$$

<u>Ví dụ 1.8:</u> $A_c = \{x|x < 1\ hoặc\ x > 3\}$ (tiếp theo ví dụ 1.6)




**Hình 1.5 Hàm liên thuộc cho ví dụ 1.8**

# 1.1.3 Các phép toán

- **Các tính chất của phép giao**
  - ✓ Tính giao hoán: $A \cap B = B \cap A$
  - ✓ Tính kết hợp: $A \cap (B \cap C) = (A \cap B) \cap C$
  - ✓ Nếu $A \subset B$ thì $A \cap B = A$ và $A \cap C \subset B \cap C$
  - ✓ $(A \cap B)_c = A_c \cup B_c$ (*De Morgan*)
- **Các tính chất của phép hợp**
  - ✓ Tính giao hoán: $A \cup B = B \cup A$
  - ✓ Tính kết hợp: $A \cup (B \cup C) = (A \cup B) \cup C$
  - ✓ Nếu $A \subset B$ thì $A \cup C \subset B \cup C$
  - ✓ $(A \cup B)_c = A_c \cap B_c$

# 1.2 Tập mờ và hàm liên thuộc

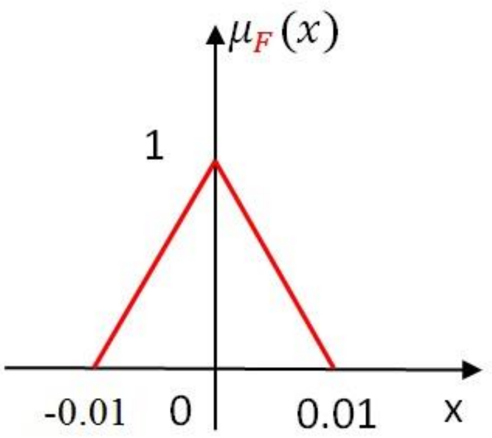
- **Định nghĩa:**

Cho tập hợp vũ trụ U, tập hợp mờ F được mô tả bởi hàm liên thuộc:

$$\mu_F(x): U \rightarrow [0;1], x \in U$$

- <u>Ví dụ 1.9:</u> Tập hợp các sai số xấp xỉ bằng không F với hàm liên thuộc như hình 1.6.

$$\mu_F(x) = \begin{cases} -100x + 1, nếu\ 0 \le x < 0.01 \\ 100x + 1, nếu\ -0.01 < x < 0 \\ 0, các\ trường\ hợp\ khác \end{cases}$$



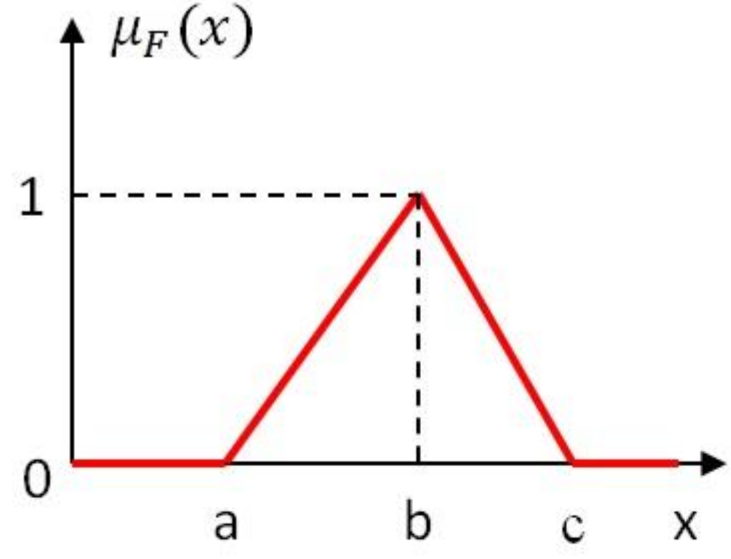

**Hình 1.6 Hàm liên thuộc cho F**

# 1.2 Tập mờ và hàm liên thuộc

- **Các dạng hàm liên thuộc:**

**Trimf** (TRIangular Membership Function) – tam giác

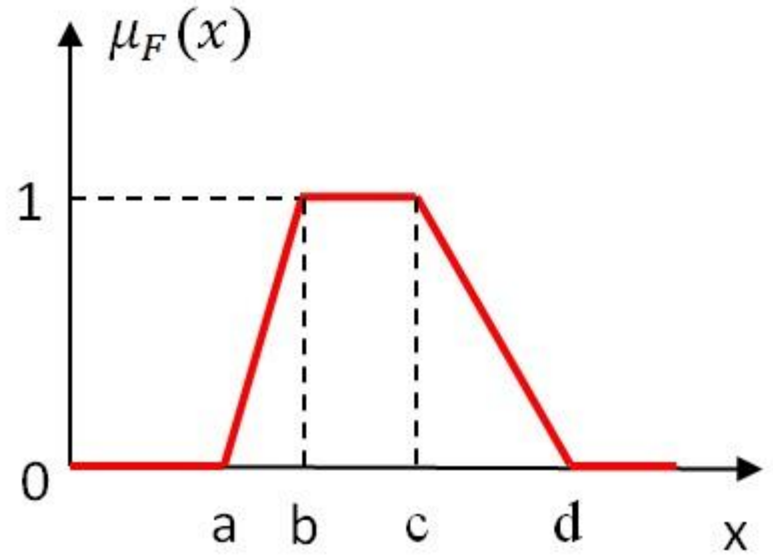
$$\mu_F(x) = \begin{cases} 0, nếu\ x \le a \\ \dfrac{x-a}{b-a}, nếu\ a \le x < b \\ \dfrac{c-x}{c-b}, nếu\ b \le x < c \\ 0, nếu\ c \le x \end{cases}$$



# 1.2 Tập mờ và hàm liên thuộc

- **Trapmf** (TRAPezoidal Membership Function) – hình thang

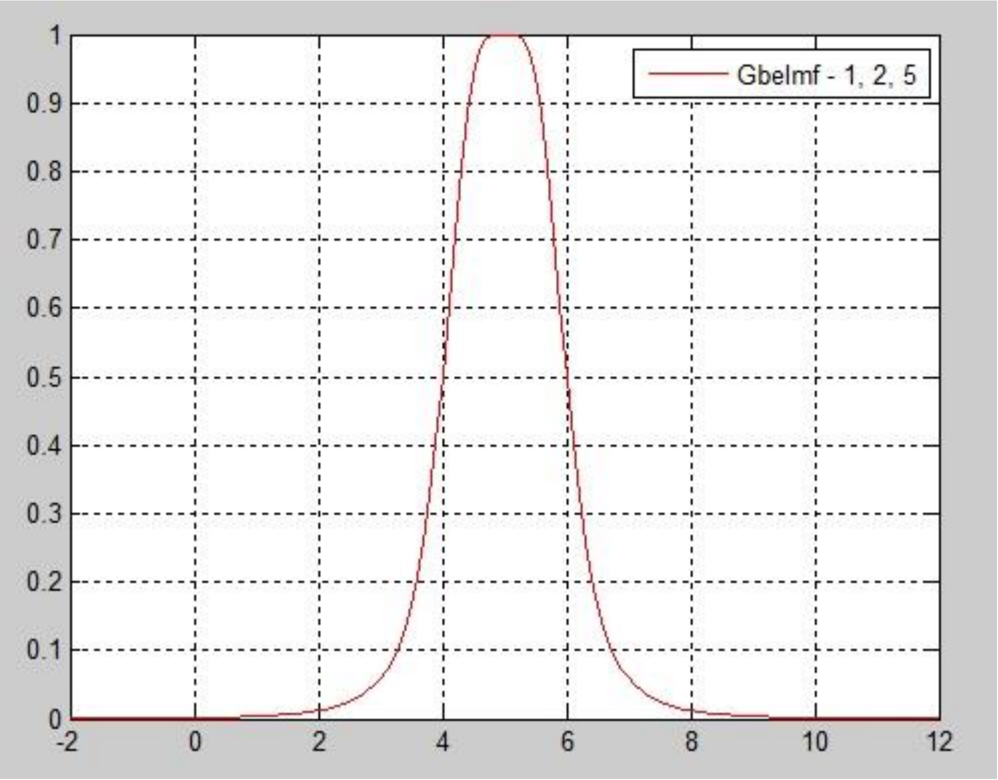
$$\mu_F(x) = \begin{cases} 0, nếu\ x \le a \\ \dfrac{x-a}{b-a}, nếu\ a \le x < b \\ 1, nếu\ b \le x \le c \\ \dfrac{d-x}{d-c}, nếu\ c \le x < d \\ 0, nếu\ d \le x \end{cases}$$



# 1.2. Tập mờ và hàm liên thuộc

**Gbellmf** (Generalized BELL Membership Function) – hình chuông

$$\mu_F(x) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}}$$




a – độ rộng

$b > 0$ độ dốc

c – tâm

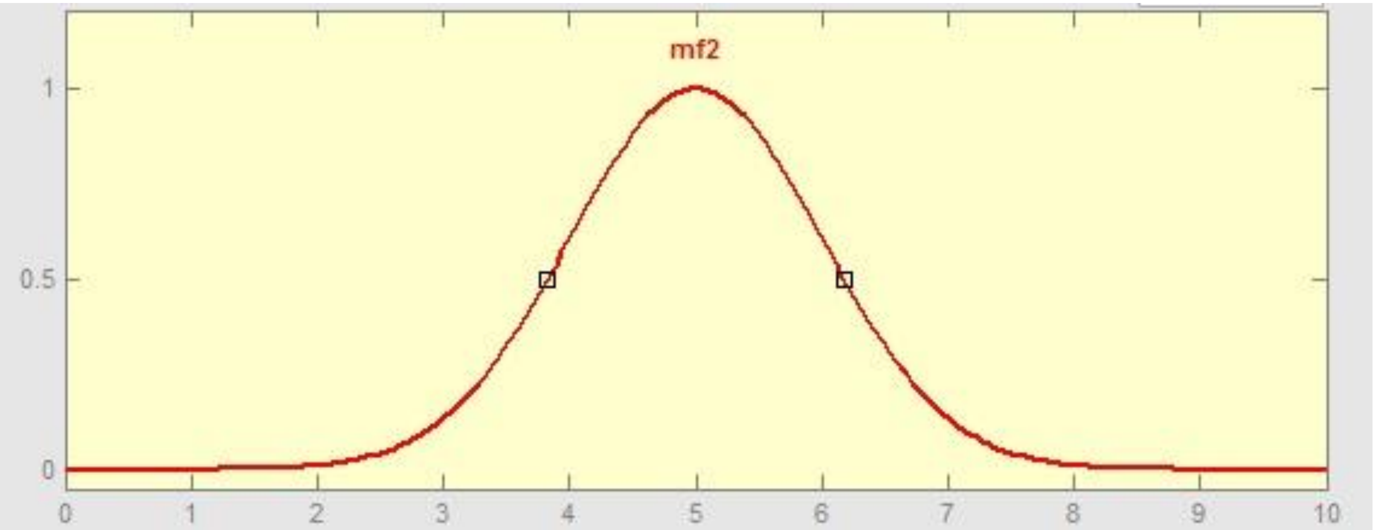
# 1.2. Tập mờ và hàm liên thuộc

- **Gaussmf** (GAUSSian Membership Function)

$$\mu_F(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$

c – là tâm của hàm liên thuộc

$\sigma$ – độ rộng



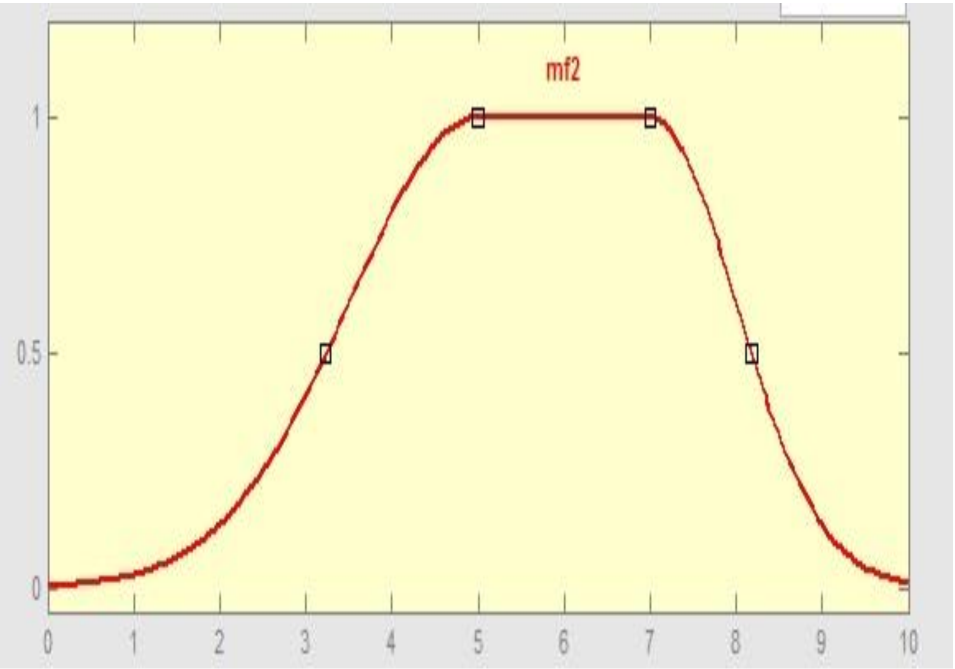
# 1.2. Tập mờ và hàm liên thuộc

- **Gauss2mf**

$$\mu_1(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-c_1}{\sigma_1}\right)^2}, \text{nếu } x \le c_1 \\ 1, \text{nếu } x > c_1 \end{cases}$$

$$\mu_2(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-c_2}{\sigma_2}\right)^2}, \text{nếu } x \ge c_2 \\ 1, \text{nếu } x < c_2 \end{cases}$$

$$\mu_F(x) = \mu_1(x) * \mu_2(x)$$

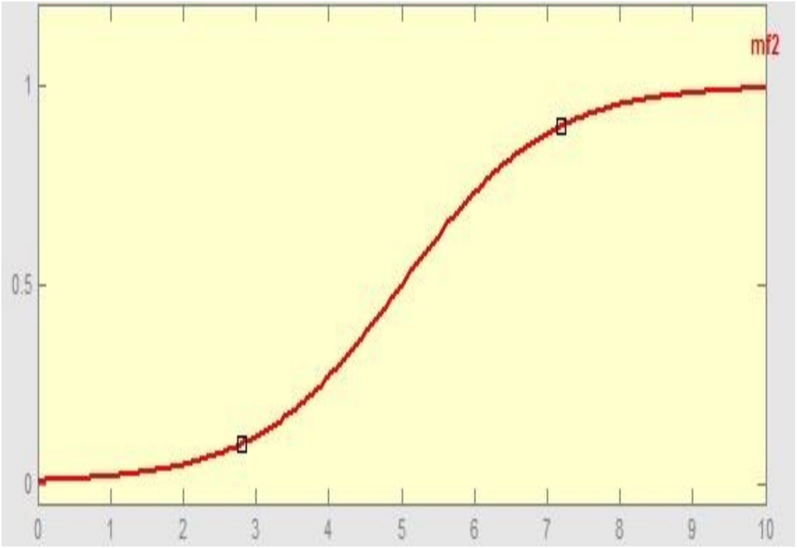


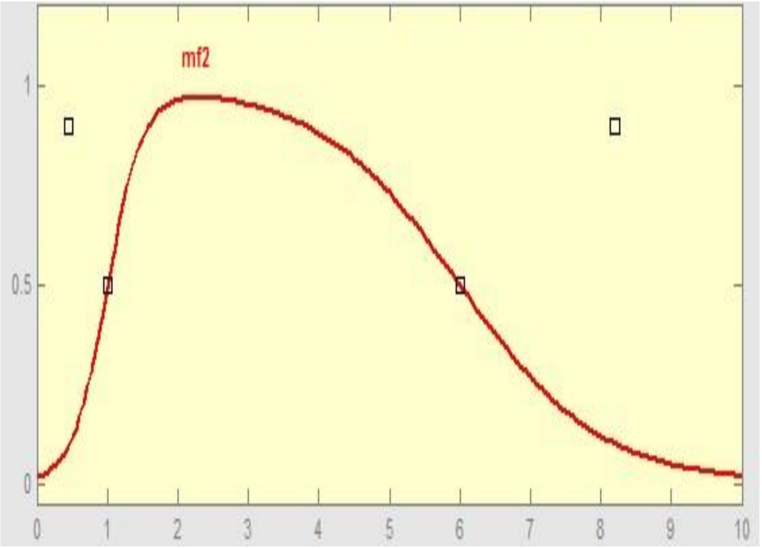
# 1.2. Tập mờ và hàm liên thuộc

- **Sigmf** (SIGmoid Membership Function)

$\mu_F(x) = \frac{1}{1+e^{-a(x-c)}} = sig(x, a, c)$, a – độ dốc tại tâm c

- **Dsigmf :** $\mu_F(x) = |sig(x, a_1, c_1) - sig(x, a_2, c_2)|$
- **Psigmf :** $\mu_F(x) = sig(x, a_1, c_1) * sig(x, a_2, c_2)|$



# 1.2. Tập mờ và hàm liên thuộc

- **Zmf** (Z-shaped curve MF)

$$\mu_F(x) = \begin{cases} 1, nếu\ x \le x_1 \\ 1 - 2\left(\dfrac{x - x_1}{x_0 - x_1}\right)^2, nếu\ x_1 \le x < \dfrac{x_1 + x_0}{2} \\ 2\left(\dfrac{x - x_1}{x_0 - x_1}\right)^2, nếu \dfrac{x_1 + x_0}{2} \le x \le x_0 \\ 0, nếu\ x_0 \le x \end{cases} = zmf(x_1, x_0)$$

- **Smf** (**S-s**haped curve MF) $\mu_F(x) = 1 - zmf(x_1, x_0) = smf(x_1, x_0)$
- **Pmf** (Pi-shaped curve MF) $\mu_F(x) = smf(x_1, x_2) * zmf(x_3, x_4)$

# Điều khiển mờ và mạng nơ ron

Ts. Nguyễn H. Nam
Bộ môn Điều khiển tự động – Viện Điện – BKHN

https://sites.google.com/view/n2c

# 1.2 Các phép tính cơ bản của tập mờ

- **Phép bù của một tập mờ**

Một toán tử $\boldsymbol{\eta}$ được định nghĩa là phép bù của tập mờ A nếu thỏa mãn các điều kiện sau:

❑ $\eta: [0,1] \rightarrow [0,1]$;

❑ $\eta(0) = 1$ và $\eta(1) = 0$; (điều kiện biên)

❑ $\mu_A(x) < \mu_A(y) \Leftrightarrow \eta(\mu_A(x)) \geq \eta(\mu_A(y))$ (tính không tăng)

❑ $\forall x, \eta(\eta(\mu_A(x))) = \mu_A(x)$

# 1.2 Các phép tính cơ bản của tập mờ

- **Một số toán tử bù thông dụng:**

➢ $\eta(\mu_A(x)) = \mu_{A_c}(x)$

$= 1 - \mu_A(x)$ (Zadeh, chuẩn)

➢ $\mu_{A_c}(x) = \left[1 - \left(\mu_A(x)\right)^v\right]^{1/v}$ (Yager)

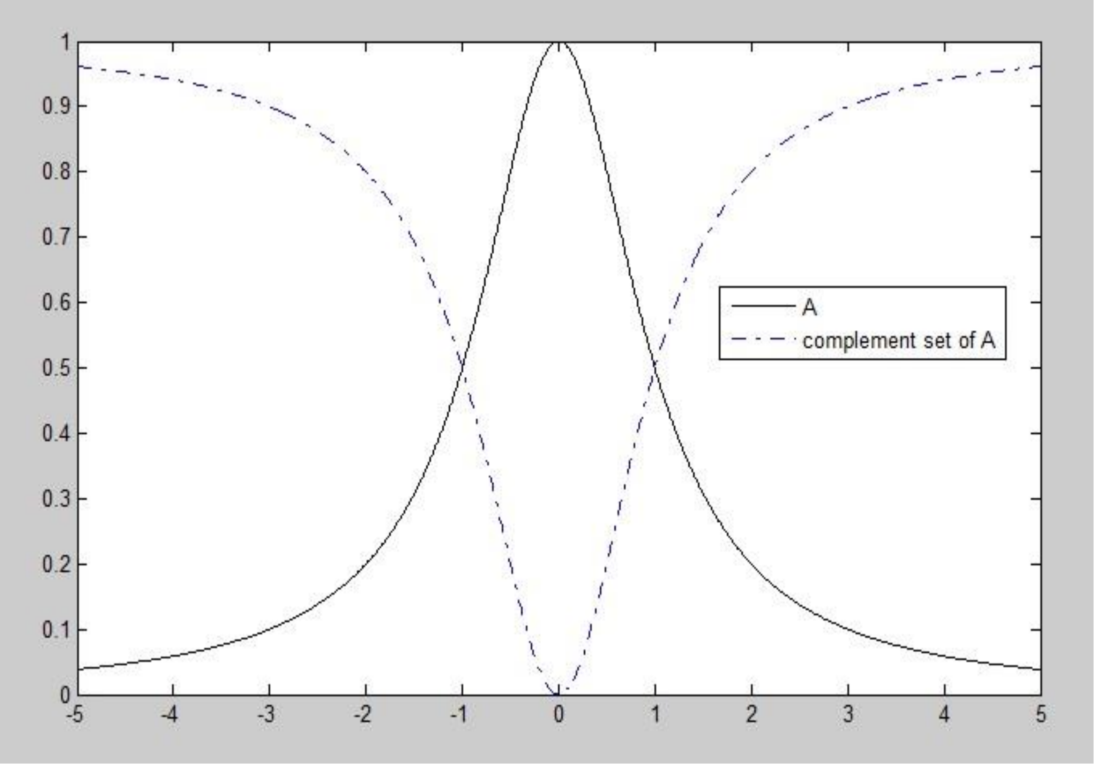
# 1.2 Các phép tính cơ bản của tập mờ

- Ví dụ 1.10:

$$\mu_A(x) = \frac{1}{1 + x^2}$$

(Gbellmf – 1, 1, 0)

$$\mu_{A_c}(x) = \frac{x^2}{1 + x^2}$$



# 1.2 Các phép tính cơ bản của tập mờ

- **Phép hợp của hai tập hợp mờ**

Một toán tử U được định nghĩa là phép hợp của hai tập hợp mờ A và B nếu thỏa mãn các điều kiện sau:

- ❑ U: $[0,1]\times[0,1] \rightarrow [0,1]$;
- ❑ $\mu_A(x) U \mu_B(x) = \mu_B(x) U \mu_A(x)$;
- ❑ $[\mu_A(x) U \mu_B(x)]\, U \mu_C(x) = \mu_A(x) U [\mu_B(x)\, U \mu_C(x)]$;
- ❑ $\mu_A(x) \geq \alpha,\ \mu_B(x) \geq \beta \Leftrightarrow$

$$\mu_A(x) U \mu_B(x) \geq \alpha U \mu_B(x),$$

$$\mu_A(x) U \mu_B(x) \geq \mu_A(x) U \beta;$$

- ❑ $\mu_A(x) U 1 = 1,\ \mu_A(x) U 0 = \mu_A(x)$

# 1.2 Các phép tính cơ bản của tập mờ

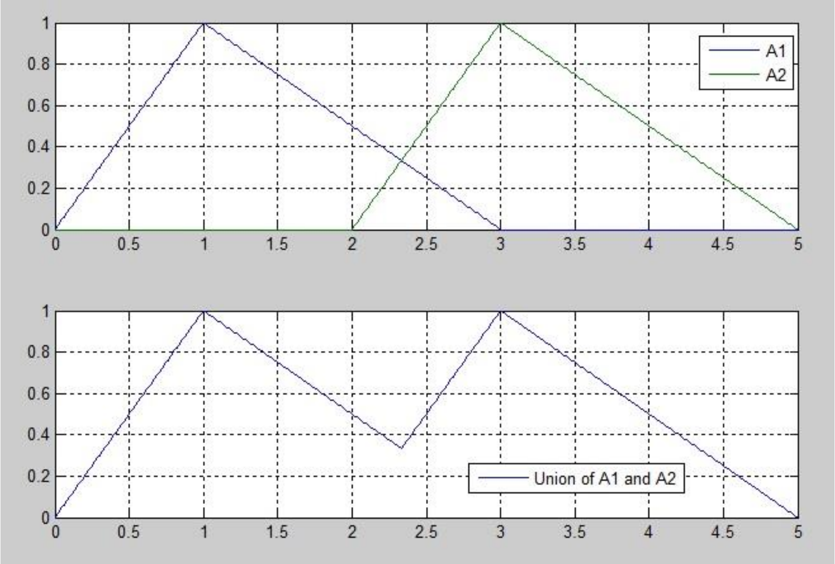
- Một số toán tử hợp thông dụng

$$\mu_{AUB}(x) = \max[\mu_A(x), \mu_B(x)]\,, \text{Zadeh}$$

$$\mu_{AUB}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\,\mu_B(x)$$

$$\mu_{AUB}(x) = \min\big(1, \mu_A(x) + \mu_B(x)\big)\,, \text{Lukasiewicz}$$

<u>Ví dụ 1.11:</u>



# 1.2 Các phép tính cơ bản của tập mờ

- **Phép giao của hai tập mờ**

Toán tử $\cap$ được định nghĩa là phép giao của hai tập hợp mờ nếu thỏa mãn các điều kiện sau:

- ❑ $\cap$: $[0,1]\times[0,1] \rightarrow [0,1]$;
- ❑ $\forall x,\ \mu_A(x)\cap\mu_B(x) = \mu_B(x)\cap\mu_A(x)$;
- ❑ $\forall C, x,\ (\mu_A(x)\cap\mu_B(x))\cap\mu_C(x) = \mu_A(x)\cap(\mu_B(x)\cap\mu_C(x))$;
- ❑ $\forall \alpha, \beta, x,\ \ \mu_A(x) \geq \alpha,\ \mu_B(x) \geq \beta \Leftrightarrow$

$$\mu_A(x)\cap\mu_B(x) \geq \alpha\cap\mu_B(x),$$
$$\mu_A(x)\cap\mu_B(x) \geq \mu_A(x)\cap\beta;$$

- ❑ $\mu_A(x)\cap 1 = \mu_A(x),\ \mu_A(x)\cap 0 = 0$

# 1.2 Các phép tính cơ bản của tập mờ

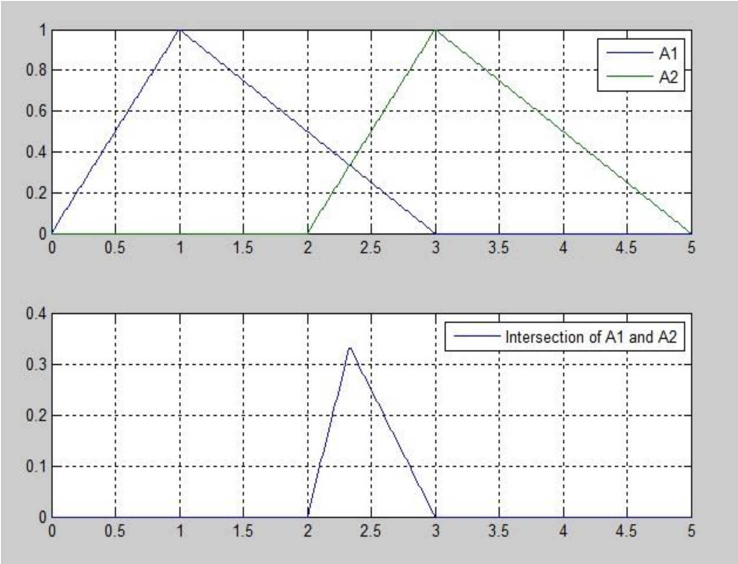
- **Một số toán tử giao thông dụng**

$\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)], \forall x \in U,$ Zadeh

$\mu_{A\cap B}(x) = \mu_A(x).\mu_B(x), \forall x \in U$

$\mu_{A\cap B}(x) = \max(\mu_A(x) + \mu_B(x) - 1, 0),$ Lukasiewicz

<u>Ví dụ 1.12:</u>



# Điều khiển mờ và mạng nơ ron

Ts. Nguyễn H. Nam
Bộ môn Điều khiển tự động – Viện Điện – BKHN

https://sites.google.com/view/n2c

# 1.3 Biến ngôn ngữ, luật hợp thành

- **Biến ngôn ngữ**

Biến ngôn ngữ gồm 5 thành phần như sau:

$$Biến\ ngôn\ ngữ = (y, T(y), U, Gr, Ru)$$

- y: tên biến ngôn ngữ
- T(y): tập hợp các thuật ngữ, là giá trị của biến
- U: tập vũ trụ
- Gr: Qui tắc cú pháp, tạo ra các thuật ngữ
- Ru: Qui tắc ngữ nghĩa, gắn các thuật ngữ với các tập mờ được định nghĩa trong U.

# 1.3 Biến ngôn ngữ, luật hợp thành

- <u>Ví dụ 1.13:</u>
- y: sai số
- T(y): very big, big, small, very small, very very small
- U: [0; 10]
- Gr: very
- Ru:

$$\mu_{very\ big}(x) = trimf(x, [7{,}5\ 10\ 12])$$
$$\mu_{big}(x) = trimf(x, [5\ 7{,}5\ 10])$$
$$\mu_{small}(x) = trimf(x, [2{,}5\ 5\ 7{,}5])$$
$$\mu_{very\ small}(x) = trimf(x, [0\ 2{,}5\ 5])$$
$$\mu_{very\ very\ small}(x) = trimf(x, [-2\ 0\ 2{,}5])$$

# 1.3 Biến ngôn ngữ, luật hợp thành

- **Mệnh đề hợp thành mờ**

R: NẾU $\chi$ là A THÌ $\gamma$ là B

$\chi$ và $\gamma$ là các biến ngôn ngữ.

A và B là các giá trị của biến ngôn ngữ

$\chi$ là A: mệnh đề điều kiện, hay tiên đề

$\gamma$ là B: mệnh đề kết luận, hay kết quả

Mệnh đề hợp thành có thể viết tắt như sau:

$$R: A \longrightarrow B$$

# 1.3 Biến ngôn ngữ, luật hợp thành

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y))$$

$f$: hàm suy luận mờ

$$f\big(\mu_A(x), \mu_B(y)\big) = \min\big(\mu_A(x), \mu_B(y)\big) \quad Mamdani$$

*Hoặc*

$$f\big(\mu_A(x), \mu_B(y)\big) = \mu_A(x).\mu_B(y)$$ Larsen - PROD

# 1.3 Biến ngôn ngữ, luật hợp thành

Trường hợp 1: Đầu vào rõ $x = x_0$

Đặt $\alpha_0 = \mu_A(x_0)$ và $\mu_{B'}(y) = \mu_R(x_0, y)$.

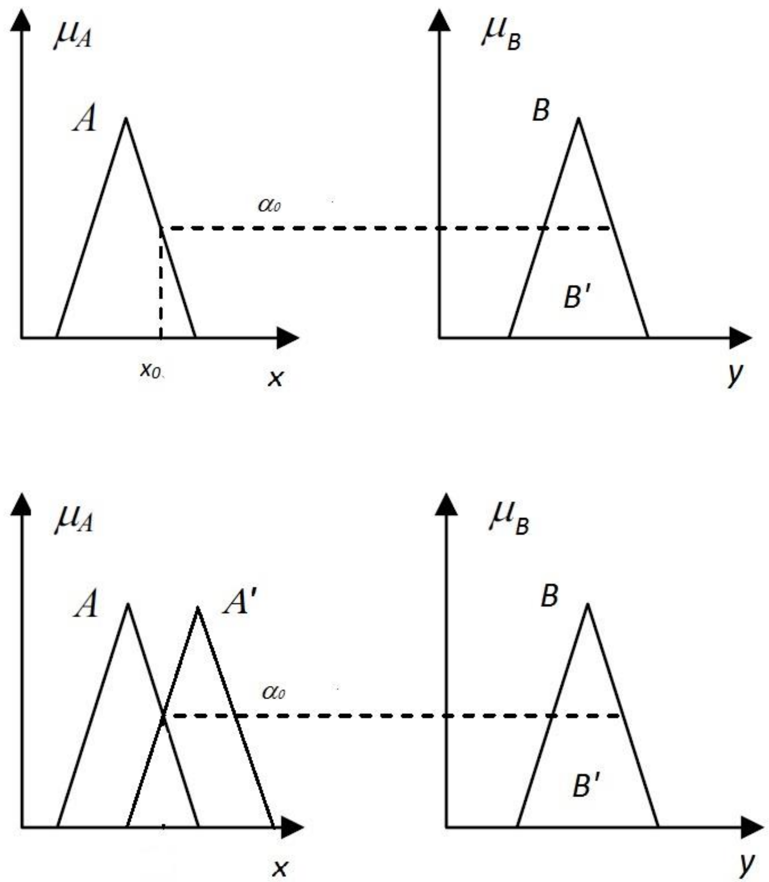
$$\mu_{B'}(y) = \min(\alpha_0, \mu_B(y))$$

Trường hợp 2: Đầu vào là tập mờ A'

$$\alpha_0 = \max(\min(\mu_A(x), \mu_{A'}(x)))$$

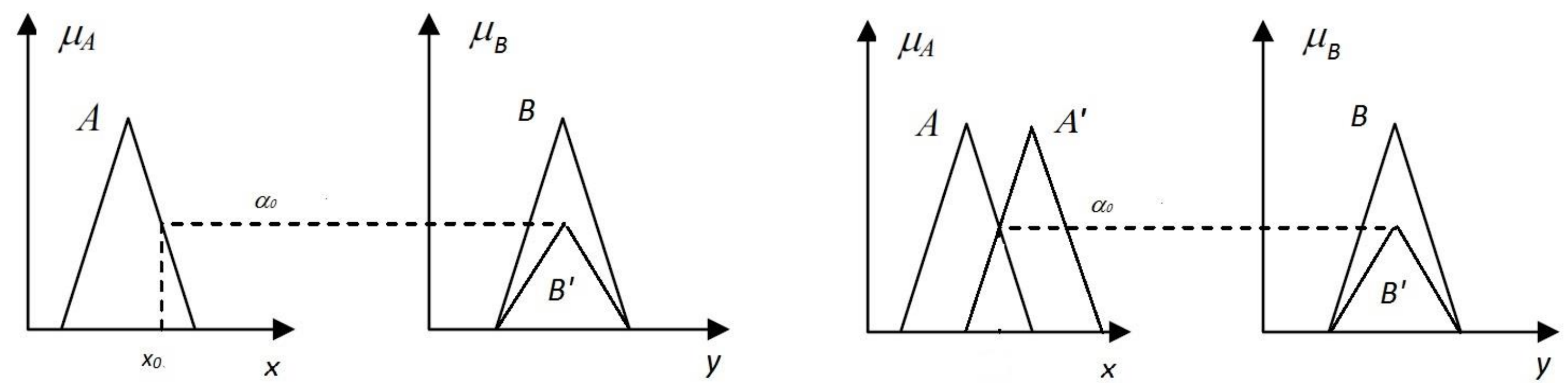
$$\mu_{B'}(y) = \min(\alpha_0 , \mu_B(y))$$

# 1.3 Biến ngôn ngữ, luật hợp thành



# 1.3 Biến ngôn ngữ, luật hợp thành

- $\alpha_0$: độ thỏa mãn mệnh đề điều kiện, độ thỏa mãn
- Khi hàm suy luận mờ là phép nhân (PROD) của Larsen.



# 1.3 Biến ngôn ngữ, luật hợp thành

- **Luật hợp thành mờ**

Xét hệ gồm nhiều mệnh đề hợp:

$R_1$: NẾU $\chi$ là $A_1$ THÌ $\gamma$ là $B_1$ hoặc

$R_2$: NẾU $\chi$ là $A_2$ THÌ $\gamma$ là $B_2$ hoặc

…

$R_n$: NẾU $\chi$ là $A_n$ THÌ $\gamma$ là $B_n$

Khi đó hàm liên thuộc của luật hợp thành mờ R là:

$$\mu_R(x, y) = \bigcup_{i=1}^{n} \mu_{R_i}(x, y)$$

# 1.3 Biến ngôn ngữ, luật hợp thành

Với $\mu_{R_i}(x, y) = f(\mu_{A_i}(x), \mu_{B_i}(y))$, f – hàm suy luận mờ

Phép hợp ở đây có thể là hàm MAX hoặc SUM (phép hợp của Lukasiewicz).

Như vậy, các phương pháp thực hiện luật hợp thành:

1) MAX – MIN (Phép hợp – hàm suy luận mờ)
2) MAX – PROD
3) SUM – MIN
4) SUM - PROD

# 1.3 Biến ngôn ngữ, luật hợp thành

- Ví dụ 1.14: Cho một hệ mờ với các tập mờ

$\mu_{A_1}(x) = trimf(x, [1\ 2\ 3])$,

$\mu_{A_2}(x) = trimf(x, [2\ 3\ 4])$,

$\mu_{B_1}(y) = trimf(y, [1\ 3\ 5])$,

$\mu_{B_2}(y) = trimf(y, [3\ 5\ 7])$,

- Và luật hợp thành

$R_1$: NẾU $\chi$ là $A_1$ THÌ $\gamma$ là $B_1$ hoặc

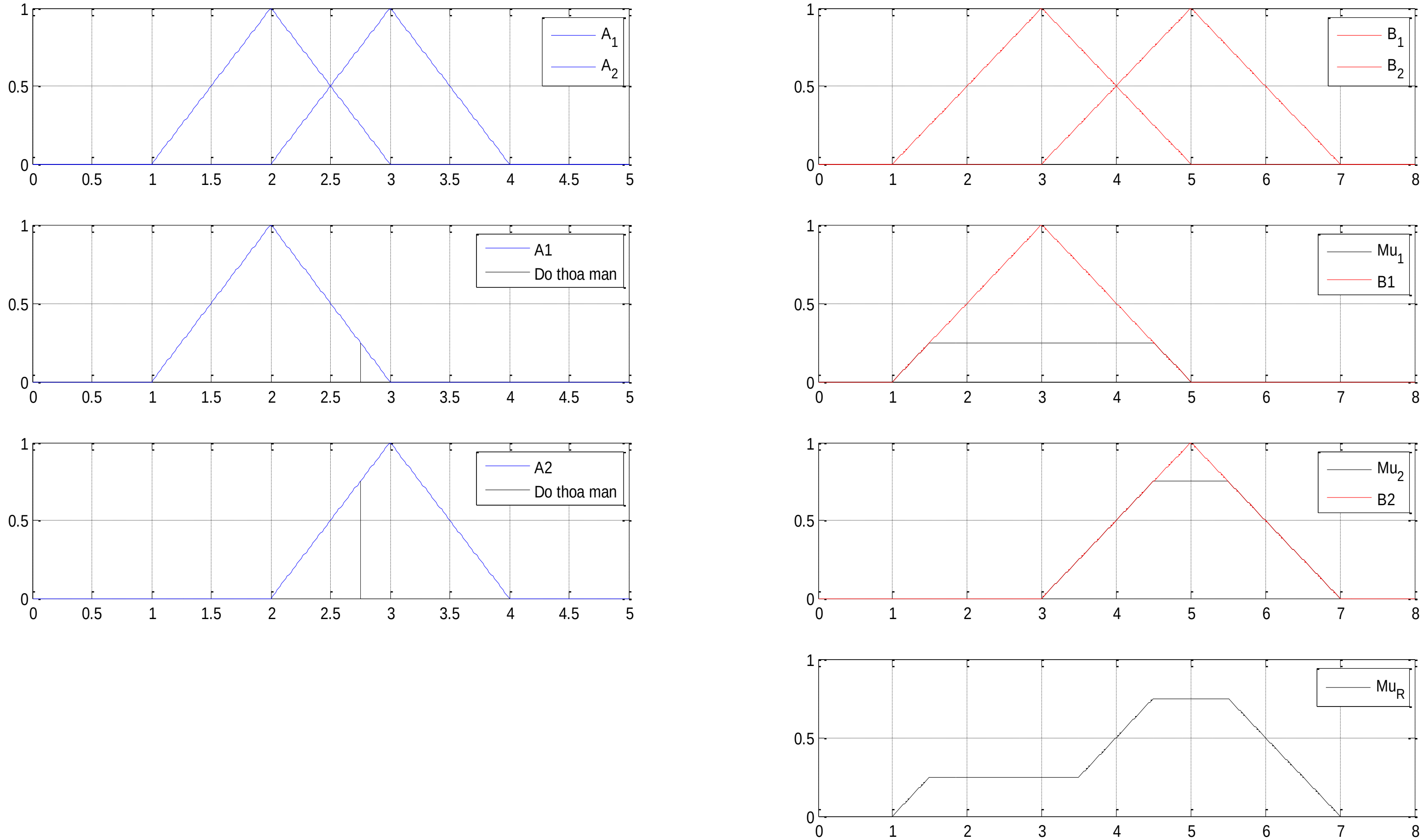
$R_2$: NẾU $\chi$ là $A_2$ THÌ $\gamma$ là $B_2$

- Tìm $\mu_R(x_0, y)$ sử dụng hàm suy luận MIN và phép HOẶC là MAX biết $x_0 = 2{,}75$.

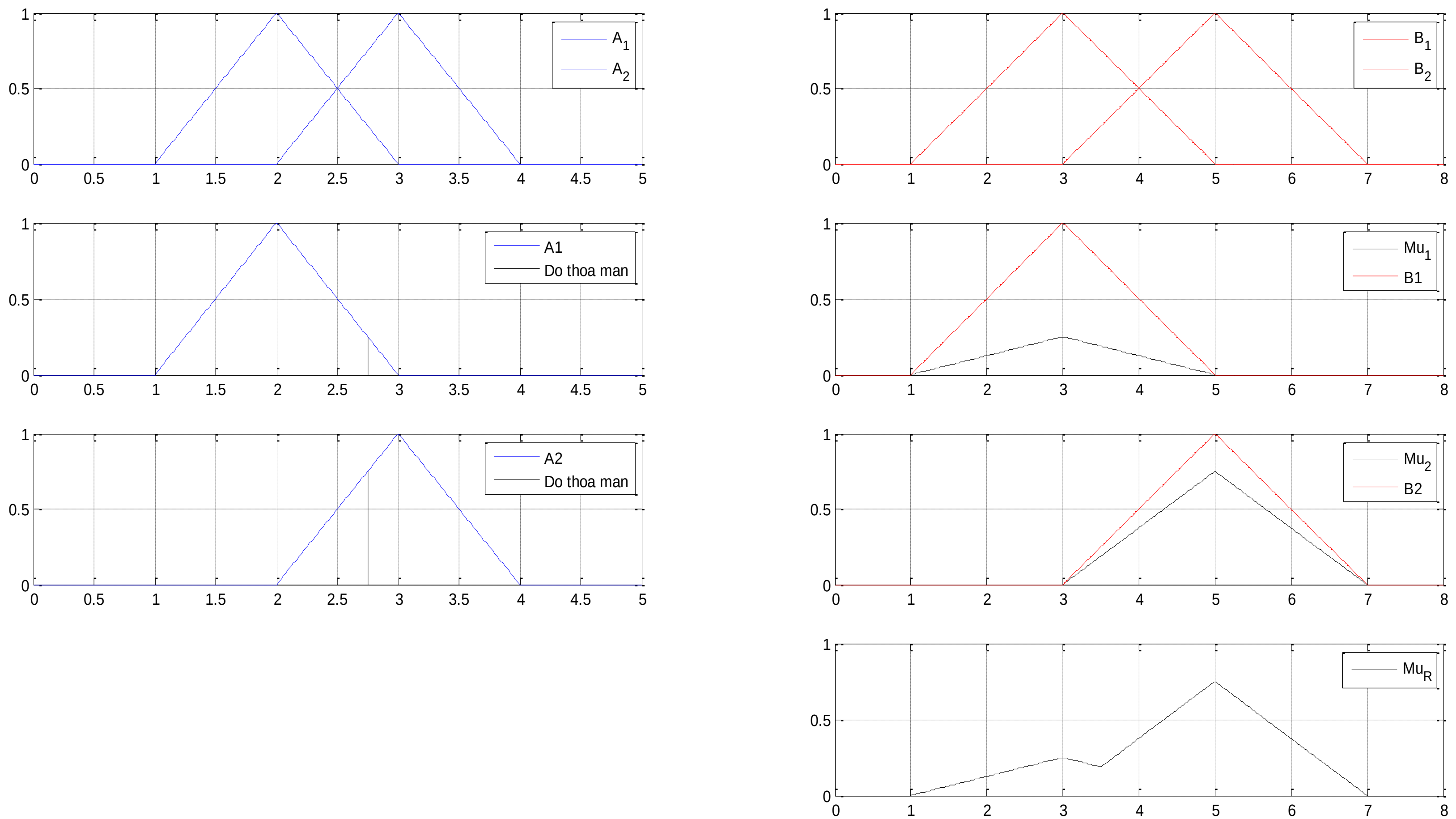
# 1.3 Biến ngôn ngữ, luật hợp thành

- $\alpha_1 = \mu_{A_1}(x_0) = 3 - 2{,}75 = 0{,}25$
- $\mu_{R_1}(2{,}75, y) = \min(0{,}25, \mu_{B_1}(y))$
- $\alpha_2 = \mu_{A_2}(x_0) = 2{,}75 - 2 = 0{,}75$
- $\mu_{R_2}(2{,}75, y) = \min(0{,}75, \mu_{B_2}(y))$
- $\mu_R(x, y) =$
  $$MAX(\min(0{,}25, \mu_{B_1}(y)), \min(0{,}75, \mu_{B_2}(y)))$$
- Code: problem_fuzzy_rule

# 1.3 Biến ngôn ngữ, luật hợp thành



# 1.3 Biến ngôn ngữ, luật hợp thành



# Điều khiển mờ và mạng nơ ron

Ts. Nguyễn H. Nam
Bộ môn Điều khiển tự động – Viện Điện – BKHN

https://sites.google.com/view/n2c

# 1.3 Biến ngôn ngữ, luật hợp thành

- **Mệnh đề hợp thành mờ hai đầu vào và một đầu ra.**

$R_1$: NẾU $\chi_1$ là $A_{1,1}$ VÀ $\chi_2$ là $A_{2,1}$ THÌ $\gamma$ là $B_1$

Tương đương với

$R_{1,1}$: NẾU $\chi_1$ là $A_{1,1}$ THÌ $\gamma$ là $B_1$ VÀ

$R_{1,2}$: $\chi_2$ là $A_{2,1}$ THÌ $\gamma$ là $B_1$

Trong đó: **VÀ** là phép giao được thực hiện bởi hàm MIN

Cho nên:

$$\mu_{R_1}(x_1, x_2, y) = MIN(\mu_{R_{1,1}}(x_1, y), \mu_{R_{1,2}}(x_2, y))$$

Đặt $\mu_{B_1'}(y) = \mu_{R_1}(x_1, x_2, y) = \mu_{R_1}(y)$

# 1.3 Biến ngôn ngữ, luật hợp thành

- Khi đầu vào rõ: $x_1 = x_{1,0}$ $và$ $x_2 = x_{2,0}$

Đặt $\alpha_{1,1} = \mu_{A_{1,1}}(x_{1,0})$ $và$ $\mu_{B'_{1,1}}(y) = \mu_{R_{1,1}}(x_{1,0}, y)$.

$$\mu_{B'_{1,1}}(y) = \min(\alpha_{1,1}, \mu_{B_1}(y)) \text{ ---- Zadeh}$$

Đặt $\alpha_{1,2} = \mu_{A_{2,1}}(x_{2,0})$ $và$ $\mu_{B'_{1,2}}(y) = \mu_{R_{1,2}}(x_{2,0}, y)$.

$$\mu_{B'_{1,2}}(y) = \min(\alpha_{1,2}, \mu_{B_1}(y)) \text{ ---- Zadeh}$$

$$\mu_{B'_1} = MIN(\mu_{B'_{1,1}}(y), \mu_{B'_{1,2}}(y))$$

# 1.3 Biến ngôn ngữ, luật hợp thành

$$\mu_{B_1'} = MIN[\min(\alpha_{1,1}, \mu_{B_1}(y)), \min(\alpha_{1,2}, \mu_{B_1}(y))]$$
$$= MIN[min(\alpha_{1,1}, \alpha_{1,2}), \mu_{B_1}(y)] \text{ ------ Zadeh}$$

- Kết luận:

✓ Độ thỏa mãn của mệnh đề điều kiện cho hệ mờ MISO

$$\alpha_1 = MIN(\alpha_{1,1}, \alpha_{1,2}, \ldots \alpha_{1,i}, \ldots) \qquad (1.1)$$

$\alpha_{1,i}$: độ thỏa mãn của đầu vào thứ $i$

✓ Trường hợp các đầu vào là mờ, độ thỏa mãn vẫn được tính theo CT (1.1).

# 1.3 Biến ngôn ngữ, luật hợp thành

- **Luật hợp thành nhiều đầu vào và một đầu ra**

$$R_1: \text{NẾU } \chi_1 \text{ là } A_{1,1} \text{ VÀ } \chi_2 \text{ là } A_{2,1} \text{ VÀ } \dots \text{ VÀ } \chi_m \text{ là } A_{m,1}$$

$$\text{THÌ } \gamma \text{ là } B_1 \text{ HOẶC}$$

$$R_2: \text{NẾU } \chi_1 \text{ là } A_{1,2} \text{ VÀ } \chi_2 \text{ là } A_{2,2} \text{ VÀ } \dots \text{ VÀ } \chi_m \text{ là } A_{m,2}$$

$$\text{THÌ } \gamma \text{ là } B_2 \text{ HOẶC } \dots$$

$$R_n: \text{NẾU } \chi_1 \text{ là } A_{1,n} \text{ VÀ } \chi_2 \text{ là } A_{2,n} \text{ VÀ } \dots \text{ VÀ } \chi_m \text{ là } A_{m,n}$$

$$\text{THÌ } \gamma \text{ là } B_n$$

$$\mu_R(\boldsymbol{x}, y) = \bigcup_{i=1}^{n} \mu_{R_i}(y)$$

$$\boldsymbol{x} = [\mathrm{x}_1 \quad \mathrm{x}_2 \dots \quad \mathrm{x}_m]$$

# 1.3 Biến ngôn ngữ, luật hợp thành

- $\mu_{R_i}(y) = \min\left(\alpha_i, \mu_{B_i}(y)\right)$ --- *Zadeh*

- $\alpha_i = MIN(\alpha_{i,1}, \alpha_{i,2}, \ldots \alpha_{i,m})$ – độ thỏa mãn của $R_i$
- $\alpha_{i,j}$ - độ thỏa mãn đối với đầu vào thứ j của $R_i$
- $\mu_R(\boldsymbol{x}, y) = MAX(\mu_{R_1}(y), \mu_{R_2}(y), \ldots, \mu_{R_m}(y))$

# 1.3 Biến ngôn ngữ, luật hợp thành

- Ví dụ 1.15: Cho một hệ mờ với

$\mu_{A_{1,1}}(x_1) = trimf(x_1, [1; 2; 3])$,

$\mu_{A_{1,2}}(x_1) = trimf(x_1, [2; 3; 4])$,

$\mu_{A_{2,1}}(x_2) = trimf(x_2, [-3; -2; -1])$,

$\mu_{A_{2,2}}(x_2) = trimf(x_2, [-4; -3; -2])$,

$\mu_{B_1}(y) = trimf(y, [1; 3; 5])$, $\mu_{B_2}(y) = trimf(y, [3; 5; 7])$,

$R_1$: NẾU $\chi_1$ là $A_{1,1}$ VÀ $\chi_2$ là $A_{2,1}$ THÌ $\gamma$ là $B_1$ hoặc

$R_2$: NẾU $\chi_1$ là $A_{1,2}$ VÀ $\chi_2$ là $A_{2,2}$ THÌ $\gamma$ là $B_2$

Tìm $\mu_R(\boldsymbol{x}, y)$ sử dụng hàm suy luận MIN và phép HOẶC là MAX biết $\boldsymbol{x_0} = [2{,}75; -2{,}5]$

# 1.4 Các phương pháp giải mờ

- Quá trình xác định giá trị rõ y' từ $\mu_R(\boldsymbol{x}, y)$ được gọi là quá trình giải mờ.
- Có ba phương pháp giải mờ chính:
  - ✓Phương pháp cực đại (Maximum)
  - ✓Phương pháp điểm trọng tâm (Centroid)
  - ✓Phương pháp đường phân tích (Bisector)
- **Phương pháp cực đại**

$$H = \max(\mu_R(\boldsymbol{x}, y)) \text{ --- độ cao của tập mờ}$$

$$G = \{y \subset Y | (\mu_R(y) = H\}$$

y' được xác định từ G với giá trị có thể chấp nhận được

Đặt $y_1 = \min(G)$ và $y_2 = max(G)$.

# 1.4 Các phương pháp giải mờ

- **Có 3 phương pháp xác định y' từ G:**
  - Giá trị trung bình (**MOM**) – **m**ean value **o**f **m**aximum

$$y' = \frac{y_1 + y_2}{2}$$

  - Giá trị nhỏ nhất (SOM)

$$y' = y_1$$

  - Giá trị lớn nhất (LOM)

$$y' = y_2$$

# 1.4 Các phương pháp giải mờ

- **Phương pháp điểm trọng tâm**

Đặt $S = \{y \in Y | \mu_R(y) > 0\}$

$$y' = \frac{\int_S \; y\mu_R(y)dy}{\int_S \; \mu_R(y)dy}$$

- **Phương pháp đường phân tích BISECTOR** (đường chia đôi diện tích)

$y'$ thỏa mãn điều kiện sau

$$\int_a^{y'} \mu_R(y)dy = \int_{y'}^{b} \mu_R(y)dy$$

# 1.4 Các phương pháp giải mờ

- Ví dụ 1.16: Cho một hệ mờ với

$\mu_{A_{1,1}}(x_1) = trimf(x_1, [1\ 2\ 3])$,

$\mu_{A_{1,2}}(x_1) = trimf(x_1, [2\ 3\ 4])$,

$\mu_{A_{2,1}}(x_2) = trimf(x_2, [-3\ \ -2\ \ -1])$,

$\mu_{A_{2,2}}(x_2) = trimf(x_2, [-4\ \ -3\ \ -2])$,

$\mu_{B_1}(y) = trimf(y, [1\ 3\ 5])$, $\mu_{B_2}(y) = trimf(y, [3\ 5\ 7])$,

$R_1$: NẾU $\chi_1$ là $A_{1,1}$ VÀ $\chi_2$ là $A_{2,1}$ THÌ $\gamma$ là $B_1$ hoặc

$R_2$: NẾU $\chi_1$ là $A_{1,2}$ VÀ $\chi_2$ là $A_{2,2}$ THÌ $\gamma$ là $B_2$

Tìm y' sử dụng hàm suy luận MIN và phép HOẶC là MAX biết

$\boldsymbol{x_0} = [2{,}75\ \ -2{,}5]$

# 1.4 Các phương pháp giải mờ

- $\mu_{A_{1,1}}(x_1) = 3 - x_1 với\ 2 < x_1 < 3$
- $x_{1,0} = 2{,}75$
- $\alpha_{1,1} = \mu_{A_{1,1}}(x_{1,0}) = 3 - 2{,}75 = 0{,}25$
- $\mu_{A_{2,1}}(x_2) = x_2 + 3\ với\ -3 < x_2 < -2$
- $x_{2,0} = -2{,}5$
- $\alpha_{1,2} = \mu_{A_{2,1}}(x_{2,0}) = -2.5 + 3 = 0{,}5$
- $\alpha_1 = \min(\alpha_{1,1}, \alpha_{1,2}) = \min(0{,}25; 0{,}5) = 0{,}25$
- $\mu_{R_1}(\boldsymbol{x_0}, y) = \min(0{,}25, \mu_{B_1}(y))$

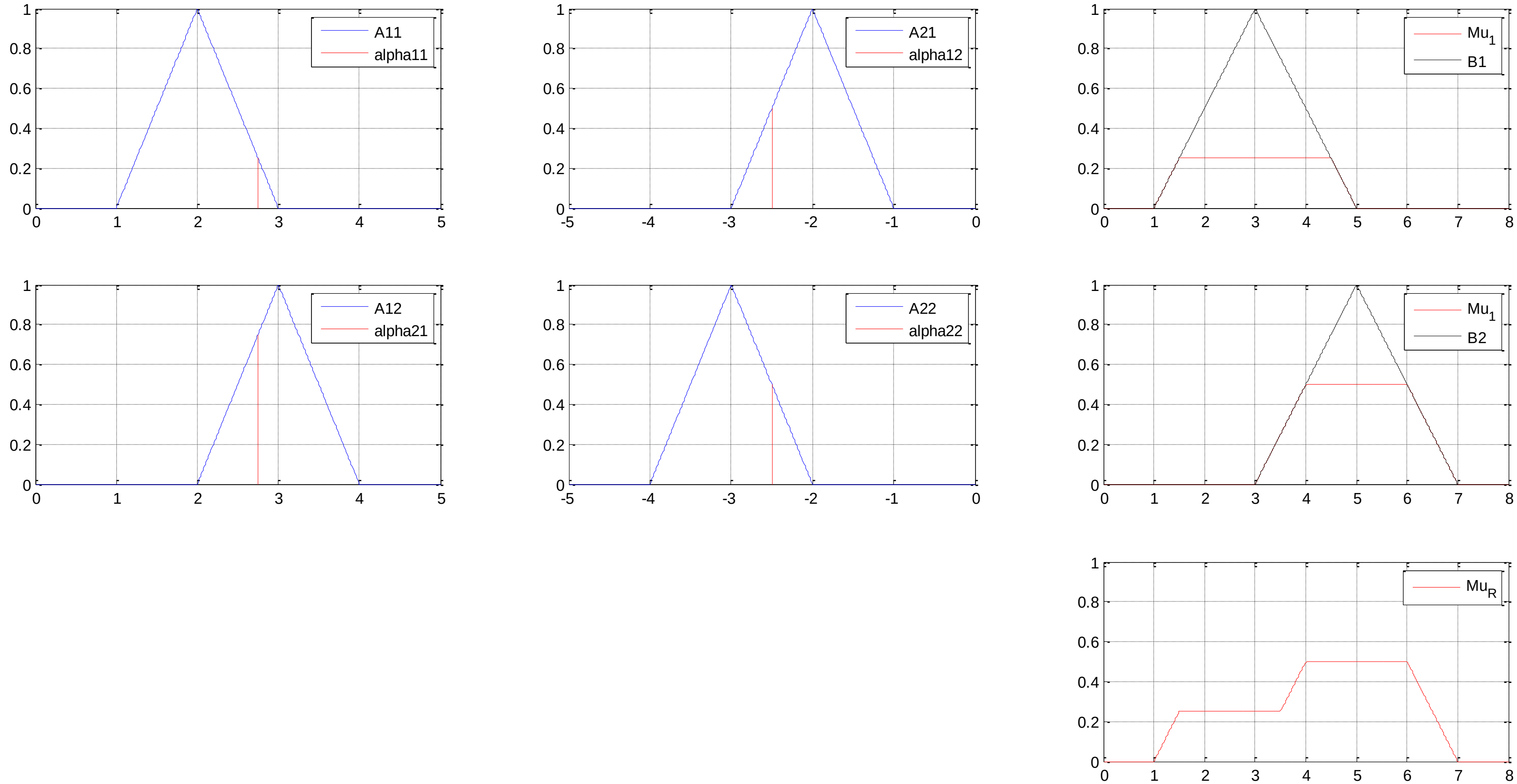
# 1.4 Các phương pháp giải mờ

- $\mu_{A_{1,2}}(x_1) = x_1 - 2 \text{ với } 2 < x < 3$
- $x_{1,0} = 2{,}75$
- $\alpha_{2,1} = \mu_{A_{1,2}}(x_{1,0}) = 2{,}75 - 2 = 0{,}75$
- $\mu_{A_{2,2}}(x_2) = -2 - x_2 \text{ với } -3 < x_2 < -2$
- $x_{2,0} = -2{,}5$
- $\alpha_{2,2} = \mu_{A_{2,2}}(x_{2,0}) = -2 - (-2.5) = 0{,}5$
- $\alpha_2 = \min(\alpha_{2,1}, \alpha_{2,2}) = \min(0{,}5; 0{,}75) = 0{,}5$
- $\mu_{R_2}(\boldsymbol{x_0}, y) = \min(0{,}5; \mu_{B_2}(y))$

# 1.4 Các phương pháp giải mờ

- $\mu_R(x_0, y) =$

$$MAX\left(\min\left(0{,}25; \mu_{B_1}(y)\right); \min(0{,}5; \mu_{B_2}(y))\right)$$



# 1.4 Các phương pháp giải mờ

**Phương pháp cực đại**

$$H = \max(\mu_R(\boldsymbol{x_0}, y)) = 0{,}5$$
$$G = \{y \subset Y | (\mu_R(y) = H\} = \{y \subset Y | 4 \leq y \leq 6\}$$
$$y_1 = \min(G) = 4$$
$$y_2 = max(G) = 6$$

Giá trị rõ y’ có thể được chọn:

y’ = 4, hoặc

y’ = 6, hoặc

y’ = 5

# 1.4 Các phương pháp giải mờ

- $S = \{y \in Y | \mu_R(y) > 0\} = \{y \in Y | 1 < y < 7\}$
- $Sd = \int_S \mu_R(y)dy = \int_1^7 \mu_R(\boldsymbol{x_0}, y)dy$
- $= 0{,}5 * 0{,}5 * 0{,}25 + 2 * 0{,}25 + 0{,}5 * (0{,}25 + 0{,}5) * 0{,}5 + 2 * 0{,}5 + 0{,}5 * 0{,}5 * 0{,}5 = 2$
- $Sn = \int_S y\mu_R(y)dy = \int_1^7 y\mu_R(\boldsymbol{x_0}, y)dy = \int_1^{1,5} y.0{,}5.(y-1)dy + \int_{1,5}^{3,5} y.0{,}25dy + \int_{3,5}^{4} y.0{,}5.(y-3)dy + \int_4^6 y.0{,}5dy + \int_6^7 y.0{,}5.(7-y)dy$
- $= 8.625$
- ***y' = Sn/Sd = 4.3125***

# Điều khiển mờ và mạng nơ ron

Ts. Nguyễn H. Nam
Bộ môn Điều khiển tự động – Viện Điện – BKHN

https://sites.google.com/view/n2c

# Mô hình mờ Sugeno (TSK)

- **Hệ nhiều đầu vào và một đầu ra**

$R_1$: NẾU $\chi_1$ là $A_{1,1}$ VÀ $\chi_2$ là $A_{2,1}$ VÀ … VÀ $\chi_m$ là $A_{m,1}$

THÌ y $= g_1(x_1, x_2, \dots, x_m)$ HOẶC

$R_2$: NẾU $\chi_1$ là $A_{1,2}$ VÀ $\chi_2$ là $A_{2,2}$ VÀ … VÀ $\chi_m$ là $A_{m,2}$

THÌ y $= g_2(x_1, x_2, \dots, x_m)$ HOẶC …

$R_n$: NẾU $\chi_1$ là $A_{1,n}$ VÀ $\chi_2$ là $A_{2,n}$ VÀ … VÀ $\chi_m$ là $A_{m,n}$

THÌ y $= g_n(x_1, x_2, \dots, x_m)$

(**TSK**) T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 1, pp. 116-132, Jan.-Feb. 1985

# Mô hình mờ Sugeno

- Đầu ra rõ:

$$y' = \frac{\alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n}{\alpha_1 + \alpha_2 + \cdots + \alpha_n}$$

**Weighted Average – wtaver**

$$y_i = g_i(x_1, x_2, \dots, x_m)$$

- $y_i$ đầu ra của mệnh đề hợp thành thứ *i*
- $\alpha_i = MIN(\alpha_{i,1}, \alpha_{i,2}, \ldots \alpha_{i,m})$ – độ thỏa mãn của $R_i$
- $\alpha_{i,j}$ - độ thỏa mãn đối với đầu vào thứ j của $R_i$

# Mô hình mờ Sugeno

$$y' = \alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n$$

**Weighted Sum – wtsum**

Khi $y_i$ là hằng số, ký hiệu:

$$\theta = [y_1\ y_2\ \ldots\ y_n]^T$$

và

$$\varphi(x) = [\alpha_1 \alpha_2 \ldots \alpha_n]^T$$

- Ta có:
  - $y' = \theta^T \varphi(x)$

# Mô hình mờ Sugeno

- Ví dụ:

$\mu_{A_{1,1}}(x_1) = trimf(x_1, [1\ 2\ 3]),$

$\mu_{A_{1,2}}(x_1) = trimf(x_1, [2\ 3\ 4]),$

$\mu_{A_{2,1}}(x_2) = trimf(x_2, [-3\ -2\ -1]),$

$\mu_{A_{2,2}}(x_2) = trimf(x_2, [-4\ -3\ -2]),$

$g_1(x_1, x_2) = 3,\ g_2(x_1, x_2) = 5$

$R_1$: NẾU $\chi_1$ là $A_{1,1}$ VÀ $\chi_2$ là $A_{2,1}$ THÌ $y = 3$ HOẶC

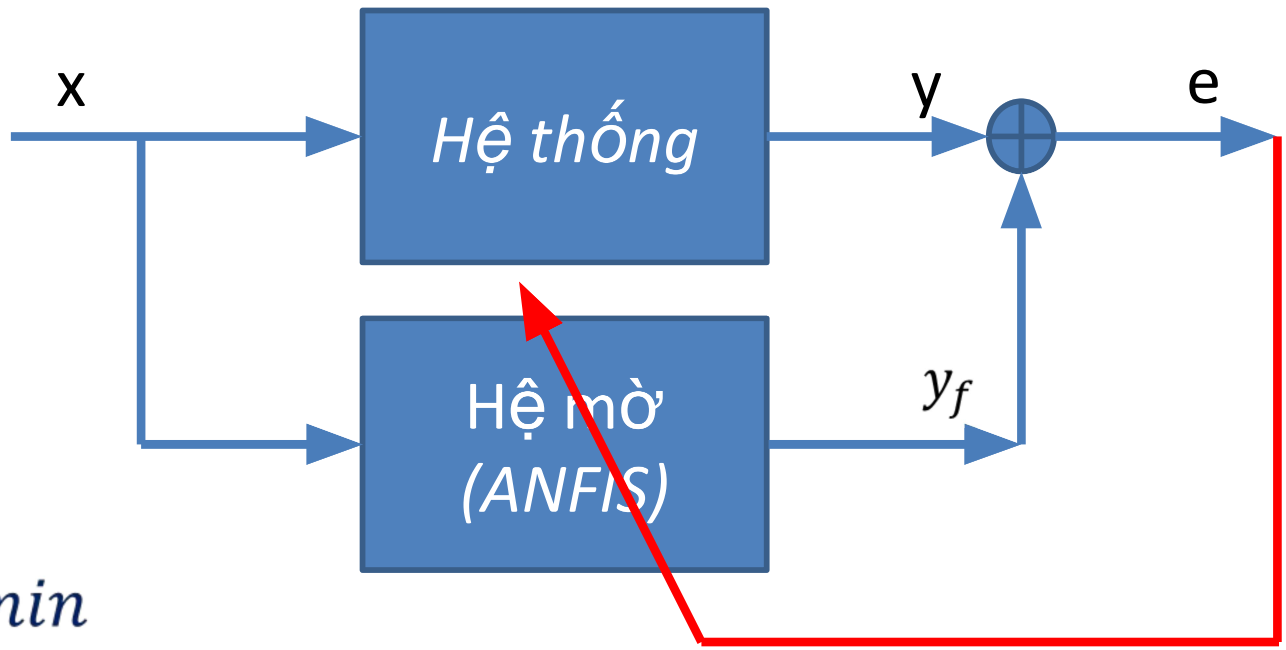
$R_2$: NẾU $\chi_1$ là $A_{1,2}$ VÀ $\chi_2$ là $A_{2,2}$ THÌ $y = 5$

Tìm y' sử dụng hàm suy luận MIN, biết $\boldsymbol{x_0} = [2{,}75\ -2{,}5]$.

# Mô hình mờ Sugeno

- $\alpha_1 = 0{,}25$
- $\alpha_2 = 0{,}5$
- $y' = \frac{\alpha_1 y_1 + \alpha_2 y_2}{\alpha_1 + \alpha_2} = \frac{0{,}25*3 \quad + \quad 0{,}5*5}{0{,}25+0{,}5} = \frac{3{,}25}{0{,}75} = 4{,}33$ – wtaver
- $y' = \alpha_1 y_1 + \alpha_2 y_2 = 3{,}25$ - wtsum

# Xấp xỉ hàm




$$J = \sum_{k=1}^{n} e^2(k) \rightarrow min$$

$$e(k) = y(k) - y_f(k)$$

Tham số của hàm liên thuộc: $a_{i,j}$

Tham số của hàm rõ: $b_{i,j}$

$$a_{i,j}(d+1) = a_{i,j}(d) - \lambda \frac{\partial J}{\partial a_{i,j}}$$

$$b_{i,j}(d+1) = b_{i,j}(d) - \lambda \frac{\partial J}{\partial b_{i,j}}$$

λ: tốc độ số học

d: số lần học (0, 1, 2, …)

# Xấp xỉ hàm

- **ANFIS** (Adaptive Neuro Fuzzy Inference System)

$$R_1\text{: NẾU } \chi_1 \text{ là } A_1 \text{ THÌ } y = b_{1,1}x + b_{1,2} \text{ HOẶC}$$
$$R_2\text{: NẾU } \chi_1 \text{ là } A_2 \text{ THÌ } y = b_{2,1}x + b_{2,2} \text{ HOẶC}$$
$$\dots$$
$$R_n\text{: NẾU } \chi_1 \text{ là } A_n \text{ THÌ } y = b_{n,1}x + b_{n,2}$$

$$\mu_{A_i}(x) = gbellmf(x, a_{i,1}, a_{i,2}, a_{i,3})$$
$$y_f = \frac{\alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n}{\alpha_1 + \alpha_2 + \cdots + \alpha_n}$$
$$y_i = b_{i,1}x + b_{i,2}$$
$$\alpha_i = \mu_{A_i}(x)$$

# Xấp xỉ hàm

- Cho một hàm tĩnh *y=f(x)* bất kỳ.
- Hệ mờ có thể xấp xỉ hàm *f* với độ chính xác bất kỳ.
- Ví dụ 1.20:
- x = (0:0.1:10)';
- y = sin(2*x).*exp(x/5);
- epoch_n = 20;
- in_fis = genfis1([x y],5,'gbellmf');
- out_fis = anfis([x y],in_fis,epoch_n);
- plot(x,y,x,evalfis(x,out_fis));
- legend('Training Data','anfis Output');

# Cơ sở hệ mờ và mạng nơ-ron

Nguyễn Hoài Nam

Bộ môn Điều khiển tự động, Viện Điện, Trường Đại học Bách Khoa Hà Nội

*Email: nam.nguyenhoai@hust.edu.vn*

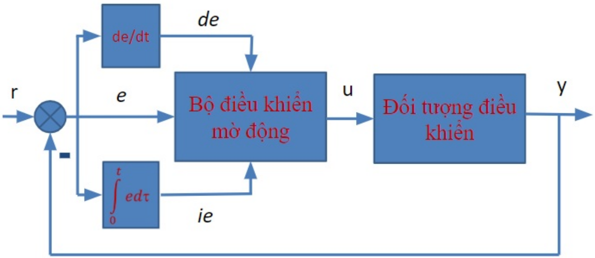
Ngày 8 tháng 4 năm 2021

# Overview

1. Thiết kế bộ điều khiển mờ PID
2. Bộ điều khiển mờ tỉ lệ
3. Bộ điều khiển mờ PD

# Thiết kế bộ điều khiển mờ PID

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1)$$




Hình 1: Sơ đồ cấu trúc hệ thống điều khiển mờP ID

# Bộ điều khiển mờ tỉ lệ

- Tín hiệu đầu vào thỏa mãn điều kiện $u_{min} \leq u \leq u_{max}$.
- Tín hiệu đầu ra thỏa mãn điều kiện $y_{min} \leq y \leq y_{max}$.
- Định nghĩa sai lệch giữa giá trị đặt $r$ và đầu ra của đối tượng $y$ là : $e = r - y$
- Dải đầu vào của hệ mờ (tập vũ trụ cho đầu vào): $[-\delta; \delta]$, trong đó $\delta = y_{max} - y_{min}$.
- Dải đầu ra của hệ mờ: $[u_{min}; u_{max}]$. Không mất tính tổng quát, ta có thể chuyển dải đầu ra về dạng $[-\lambda; \lambda]$, với $\lambda > 0$.

# Bộ điều khiển mờ tỉ lệ

Mờ hóa đầu vào:

$$\begin{aligned} \mu_{negative}(e) &= trimf(e, [-\delta; -\delta; 0]), \\ \mu_{zero}(e) &= trimf(e, [-\delta; 0; \delta]), \\ \mu_{positive}(e) &= trimf(e, [0; \delta; \delta]). \end{aligned} \tag{2}$$

Mờ hóa đầu ra:

$$\begin{aligned} \mu_{negative}(u) &= trimf(u, [-\lambda; -\lambda; 0]), \\ \mu_{zero}(u) &= trimf(u, [-\lambda; 0; \lambda]), \\ \mu_{positive}(u) &= trimf(u, [0; \lambda; \lambda]). \end{aligned} \tag{3}$$

# Bộ điều khiển mờ tỉ lệ

Nếu sai lệch là negative Thì tín hiệu điều khiển là negative Hoặc
Nếu sai lệch là zero Thì tín hiệu điều khiển là zero Hoặc
Nếu sai lệch là positive Thì tín hiệu điều khiển là positive

$$\begin{aligned} G(s) &= \frac{Y(s)}{U(s)} \\ &= \frac{5}{s + 0.02}. \end{aligned} \tag{4}$$

# Bộ điều khiển mờ PD

Mờ hóa đầu vào *de*:

$$\begin{aligned} \mu_{negative}(de) &= trimf(de, [-\epsilon; -\epsilon; 0]), \\ \mu_{zero}(de) &= trimf(de, [-\epsilon; 0; \epsilon]), \\ \mu_{positive}(de) &= trimf(de, [0; \epsilon; \epsilon]). \end{aligned} \tag{5}$$

Mờ hóa đầu ra:

$$\begin{aligned} \mu_{negative}(u) &= trimf(u, [-\lambda; -\lambda; -\lambda/2]), \\ \mu_{small-negative}(u) &= trimf(u, [-\lambda; -\lambda/2; 0]), \\ \mu_{zero}(u) &= trimf(u, [-\lambda/2; 0; \lambda/2]), \\ \mu_{small-positive}(u) &= trimf(u, [0; \lambda/2; \lambda]), \\ \mu_{positive}(u) &= trimf(u, [\lambda/2; \lambda; \lambda]). \end{aligned} \tag{6}$$

# Bộ điều khiển mờ PD

Các mệnh đề:

$R_1$ : Nếu E là negative Và DE là negative Thì U là negative Hoặc,
$R_2$ : Nếu E là negative Và DE là zero Thì U là negative Hoặc,
$R_3$ :Nếu E là negative Và DE là positive Thì U là small negative Hoặc,
$R_4$ : Nếu E là zero Và DE là negative Thì U là zero Hoặc,
$R_5$ : Nếu E là zero Và DE là zero Thì U là zero Hoặc.
$R_6$ : Nếu E là zero Và DE là positive Thì U là zero Hoặc,
$R_7$ : Nếu E là positive Và DE là negative Thì U là small positive Hoặc.
$R_8$ : Nếu E là positive Và DE là zero Thì U là positive Hoặc,
$R_9$ : Nếu E là positive Và DE là positive Thì U là positive.

# Mạng nơ-ron

Ts. Nguyễn Hoài Nam

# Mạng nơ-ron

- 1. Tổng quan về mạng nơ-ron
- 2. Phương pháp huấn luyện mạng nơron.
- 3. Sử dụng công cụ Matlab huấn luyện mạng nơron.
- 4. Ví dụ ứng dụng mạng nơron cho nhận dạng.

# 1 Tổng quan về mạng nơ-ron

- 1.1 Sự phát triển của mạng nơ-ron
- 1.2 Ứng dụng của mạng nơ-ron
- 1.3 Mô hình nơ-ron
- 1.4 Các loại mạng nơ-ron
- 1.5 Quan hệ giữa mờ và nơ-ron

# 1.1.1 Sự phát triển của mạng nơ-ron

- **Warren McCulloch** và **Walter Pitts** [1943] đã chỉ ra rằng mạng nơ ron nhân tạo có thể thực hiện bất kỳ hàm lô gíc hoặc hàm số nào.
- **Donald Hebb** [1949] đã đề xuất ra một cơ chế học của các nơ-ron sinh học.
- **Rosenblatt** [1958] và các đồng nghiệp đã xây dựng mạng **Perceptron** cùng với luật học tương ứng và chứng tỏ khả năng thực hiện việc phân loại mẫu.
- **Widrow** và **Hoff** đã đưa ra thuật toán huấn luyện mới và dùng nó để huấn luyện mạng tuyến tính thích nghi, mạng này có cùng cấu trúc và khả năng như mạng perceptron.

# 1.1.1 Sự phát triển của mạng nơ-ron

- Mạng của **Rosenblatt** và **Widrow** đều có chung những hạn chế. Các tác giả này đã đưa ra mạng mới khắc phục được các hạn chế đó, tuy nhiên họ đã không cải tạo được luật học để huấn luyện mạng phức tạp hơn.
- **Teuvo Kohonen** [1972] và **James Anderson** [1972] đã phát triển mạng nơ-ron có thể hoạt động như bộ nhớ.
- **Stephen Grossberg** [1976] đã nghiên cứu mạng nơ ron tự tổ chức.
- **John Hopfield** [1982] đã đưa ra mạng nơ-ron hồi qui.
- Một số tác giả (**David Rumelhart** và **James McClelland**) đã đưa ra thuật toán lan truyền ngược để huấn luyện mạng Perceptron nhiều lớp.

# 1.1.1 Sự phát triển của mạng nơ-ron

- Mạng tích chập CNN ra đời 1990s
- **Martin Hagan** [1994] đưa ra phương pháp huấn luyện mạng sử dụng thuật toán Levenberg (1944)– Marquardt (1963).
- 2013- nay: Mạng sâu (deep learning) phát triển mạnh trong lĩnh vực AI và học máy.

# 1.1.2 Ứng dụng của mạng nơ-ron

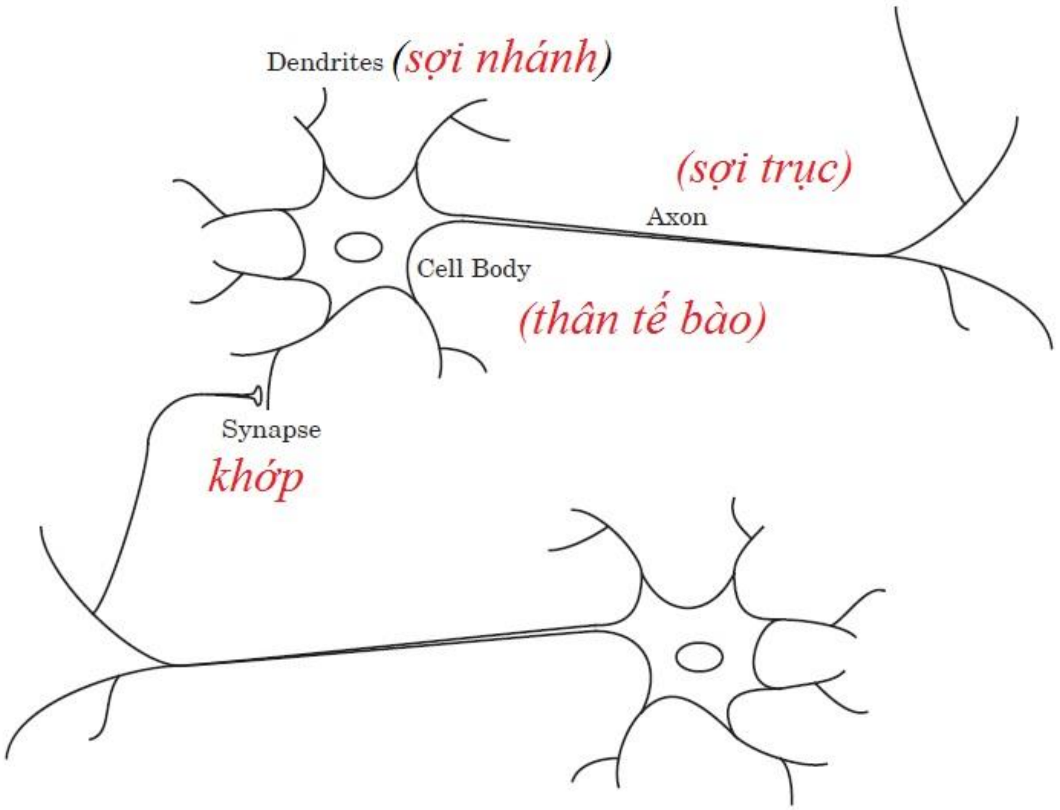
- **Điện tử:** Sự bố trí chíp IC, điều khiển quá trình, phân tích lỗi chíp, thị lực máy, tổng hợp tiếng nói, mô hình hóa phi tuyến.
- **Rô bốt:** Điều khiển quỹ đạo, xe nâng hàng, các bộ điều khiển tay máy,, các hệ thống thị giác, xe tự hành
- **Ô tô:** Các hệ thống dẫn hướng tự động, điều khiển bơm nhiên liệu, các hệ thống chống bó phanh, các cảm biến dò sự phát khí ảo.

# 1.1.2 Ứng dụng của mạng nơ-ron

- **Sản xuất:** Điều khiển quá trình sản xuất, phân tích và thiết kế sản phẩm, chuẩn đoán máy và quá trình, nhận dạng hạt thời gian thực, các hệ thống kiểm tra chất lượng, thử bia, phân tích chất lượng hàn, dự đoán chất lượng giấy, phân tích chất lượng chíp máy tính, phân tích các hoạt động nghiền, phân tích thiết kế sản phẩm hóa học, phân tích bảo dưỡng máy, đấu thầu dự án, quản lý và kế hoạch hóa, mô hình động của các quá trình hóa học.
- Vũ trụ, ngân hàng, quốc phòng, giải trí, tài chính, bảo hiểm, y tế, dầu khí, an ninh, giao thông và truyền thông.

# 1.1.3 Mô hình nơ-ron

- Nơ-ron sinh học: Sợi nhánh, sợi trục và khớp
- Não: $10^{11}$ nơ –ron
- Nơ-ron: $10^{4}$ liên kết
- Chức năng của mạng nơ-ron được hình thành bởi sự sắp xếp các nơ-ron và độ lớn của các khớp thần kinh, gây ra bởi một quá trình hóa học phức tạp.



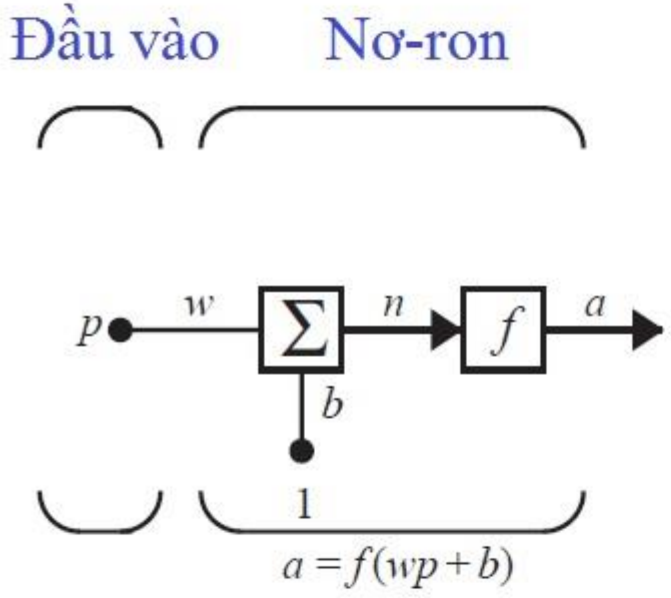

Mạng nơ-ron sinh học

# 1.1.3 Mô hình nơ-ron

- Một phần của bộ não được hình thành khi sinh ra.
- Những phần khác sẽ phát triển thông qua học, do có những liên kết mới được hình thành và các liên cũ mất đi.
- Cấu trúc mạng nơ-ron tiếp tục thay đổi theo thời gian trong đời. Những thay đổi sau nay có xu hướng chủ yếu làm mạnh hoặc yếu đi các khớp thần kinh.
- Có 2 sự giống nhau giữa mạng nơ-ron sinh học và mạng nơ-ron nhân tạo:
  - ✓ Các khối tạo ra mạng nơ-ron là những thiết bị tính toán đơn giản.
  - ✓ Các liên kết giữa các nơ-ron tạo ra chức năng của mạng.

# Mô hình nơ-ron nhân tạo

- Nơ ron một đầu vào
  - ❑ $p$ – đầu vào của nơ-ron
  - ❑ $w$ – trọng số
  - ❑ $b$ – ngưỡng (bias)
  - ❑ $\sum$ – bộ tổng
  - ❑ $n$ – đầu vào net (net input)
  - ❑ $a$ – đầu ra của nơ-ron
  - ❑ $f$ – hàm truyền



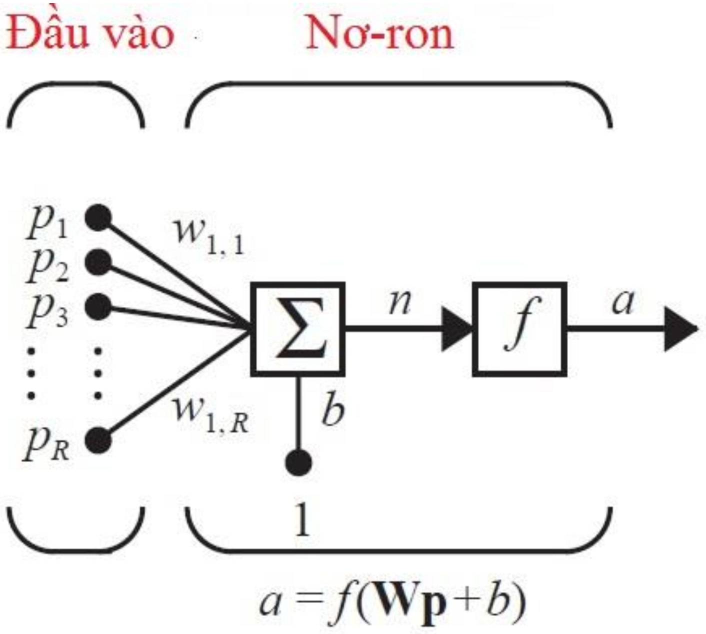

- <u>Ví dụ:</u> (nnd2n1) $p = 0{,}4; w = 1; b = 0{,}6; f$ *là hàm tuyến tính*.
- $n = wp + b = 1 * 0{,}4 + 0{,}6 = 1; a = f(n) = 1$

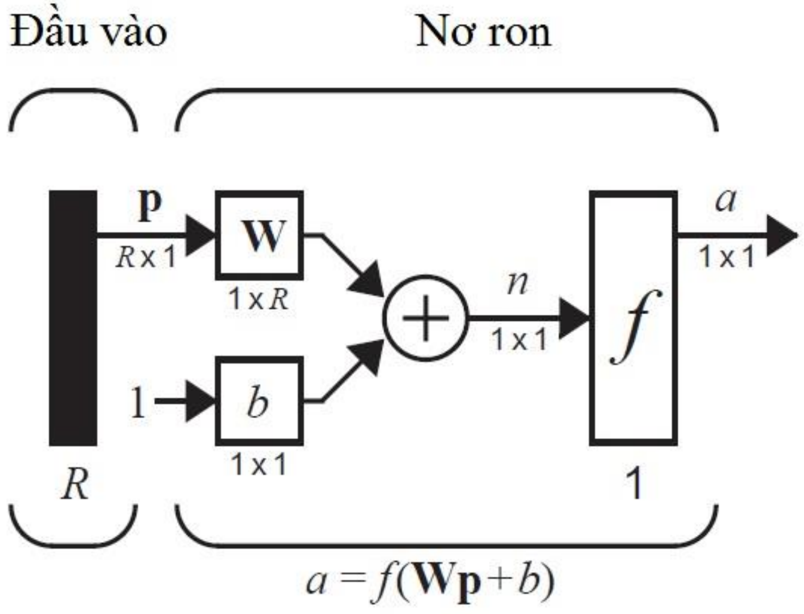
# Hàm truyền

| Tên hàm | Mô tả toán học | Ứng dụng |
|---|---|---|
| $a = \text{hardlim(n)}$ | $a = \begin{cases} 0, nếu\ n < 0 \\ 1, nếu\ 0 \le n \end{cases}$ | Mạng perceptron |
| $a = purelin(n)$ | $a = n$ | Mạng Adaline |
| $a = logsig(n)$ | $a = \frac{1}{1+e^{-n}}$ | Mạng nhiều lớp, thuật toán lan truyền ngược |
| $a = tansig(n)$ | $a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$ | |
| $a = poslin(n)$ | $a = \begin{cases} 0, nếu\ n < 0 \\ n, nếu\ 0 \le n \end{cases}$ | Mạng Hamming |
| $a$=satlins(n) | $a = \begin{cases} -1, n < -1 \\ n, -1 \le n \le 1 \\ 1, n > 1 \end{cases}$ | Mạng Hopfield |

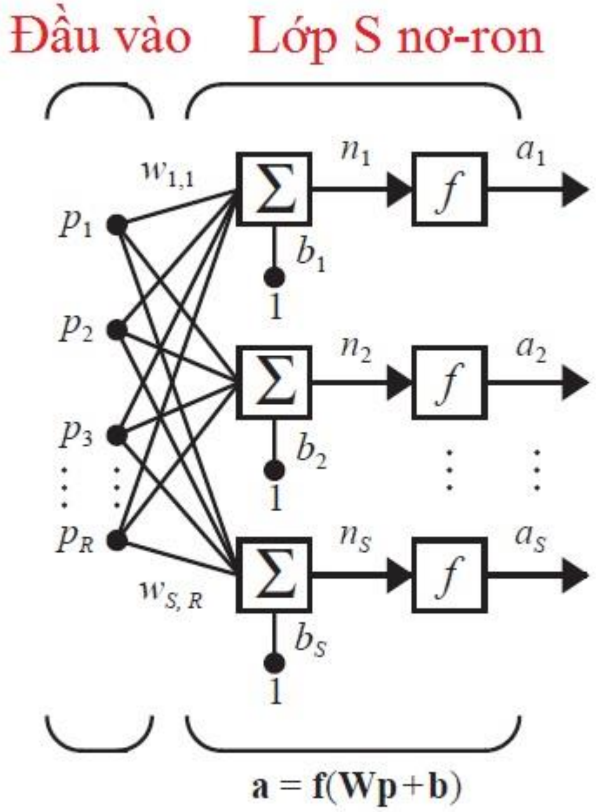
# Nơ-ron nhiều đầu vào

- $\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix}; \boldsymbol{W} = \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,R} \end{bmatrix}^T; n = b + \sum_{i=1}^{R} w_{1,i} p_i = \boldsymbol{W}\boldsymbol{p} + b$



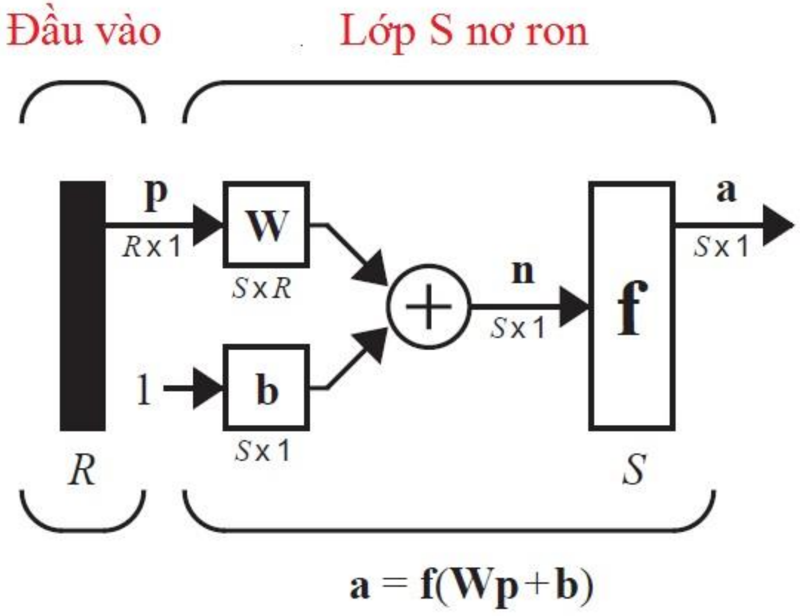



# Lớp nơ-ron

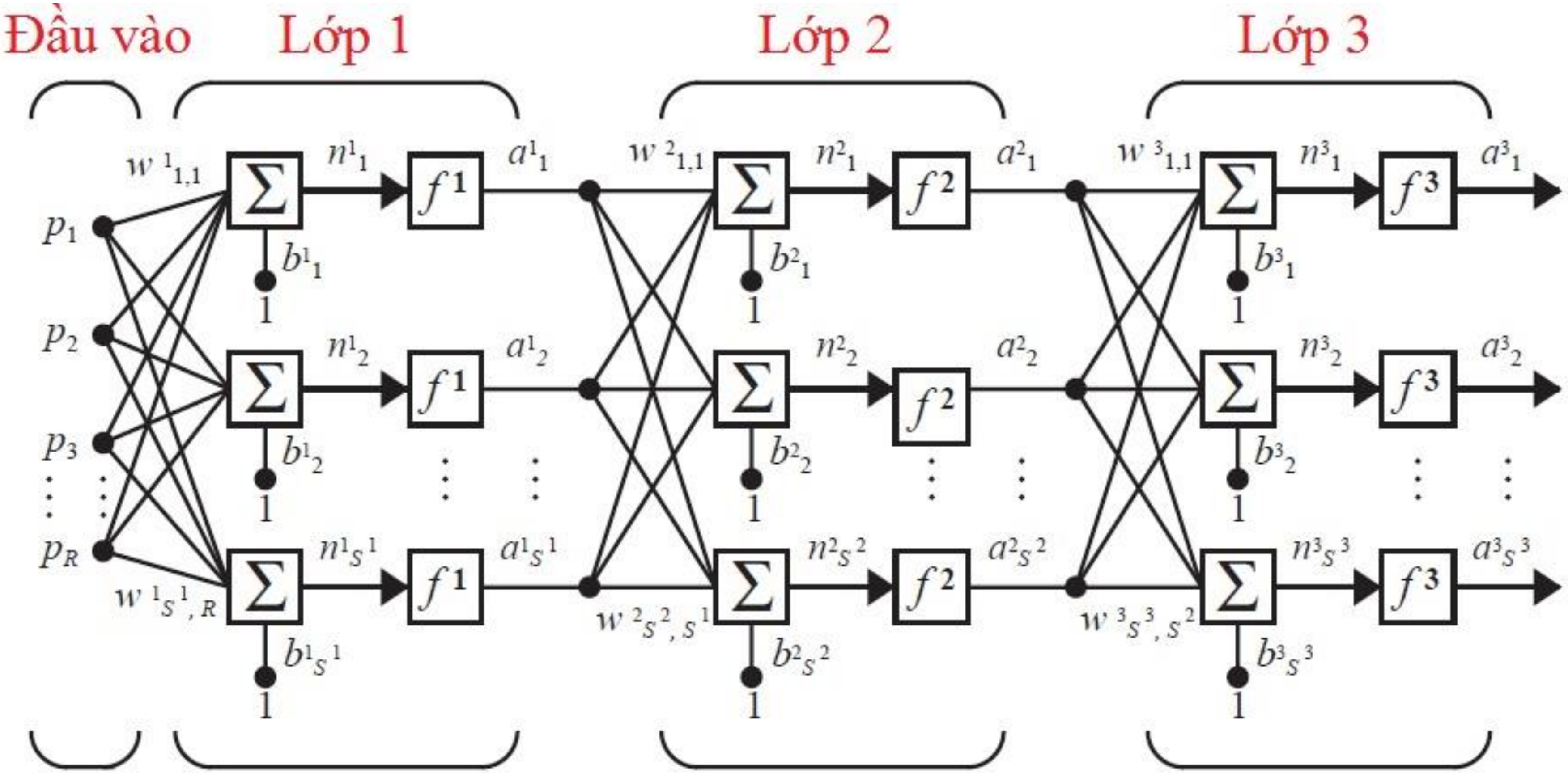



$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}.$$

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ . \\ \vdots \\ b_S \end{bmatrix}$$



# Mạng nhiều lớp



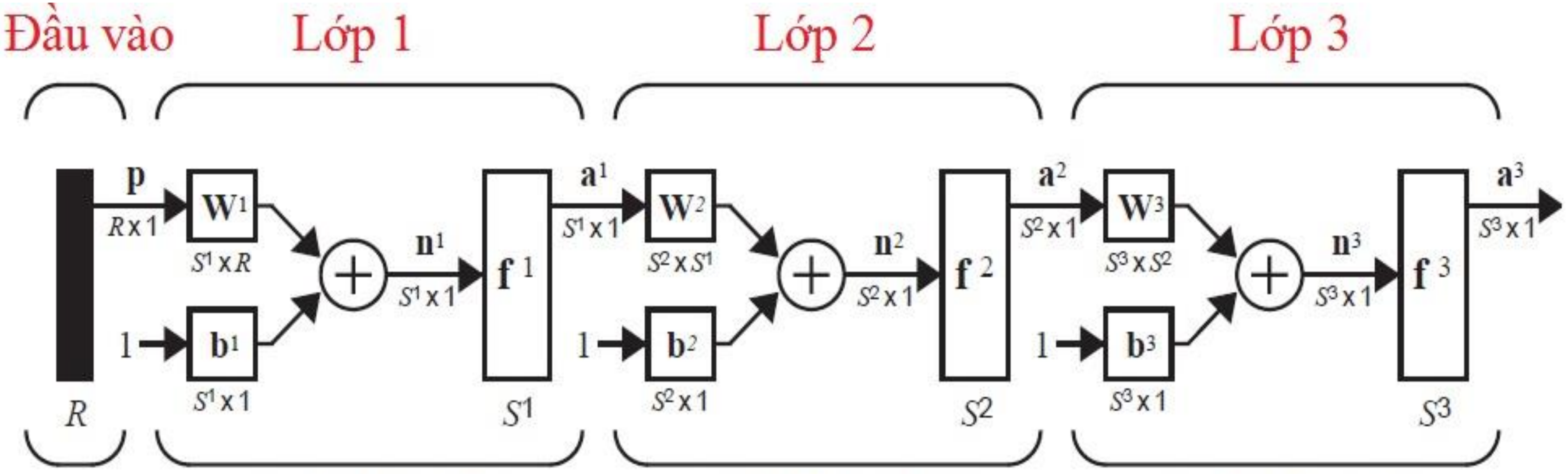

$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)$$

$$\mathbf{a}^2 = \mathbf{f}^2(\mathbf{W}^2\mathbf{a}^1+\mathbf{b}^2)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{a}^2+\mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

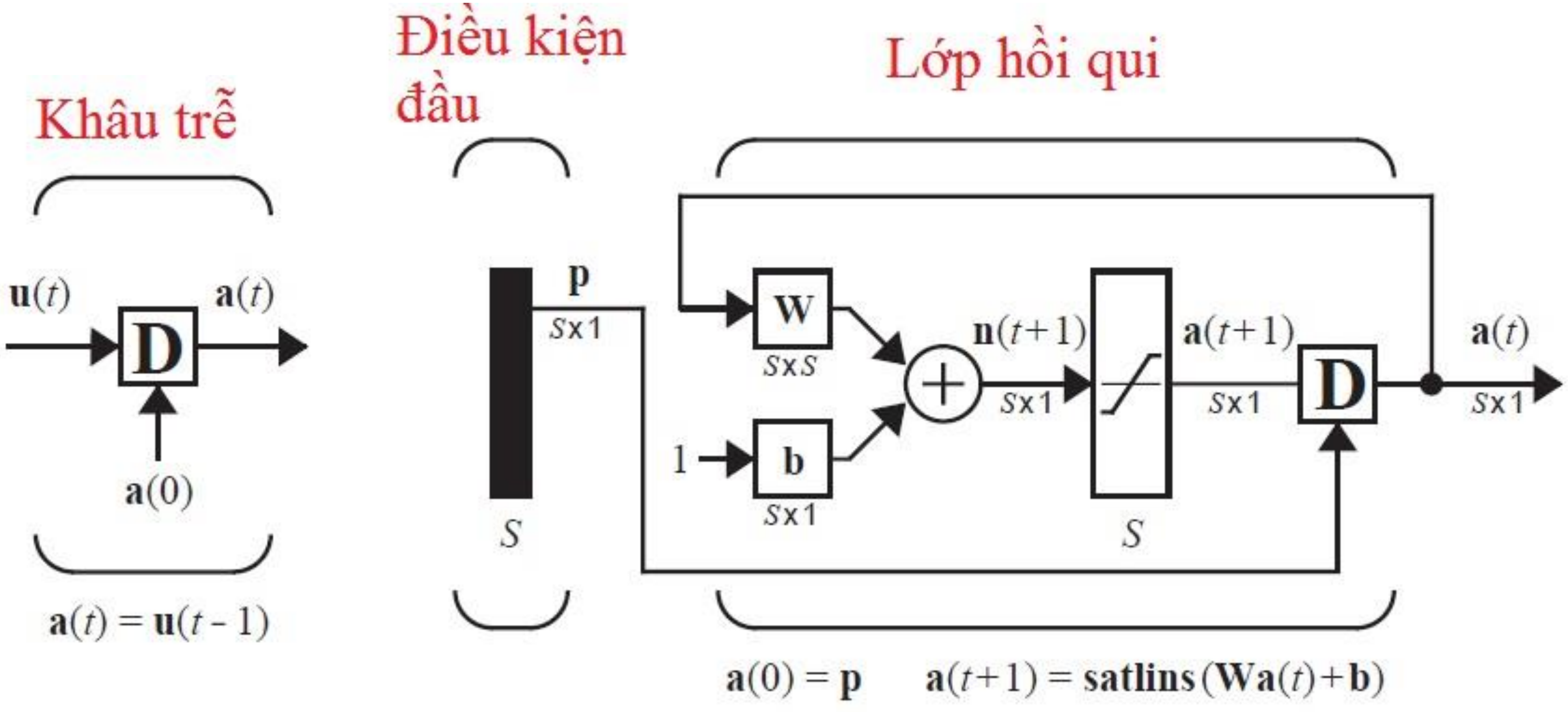
# Mạng nhiều lớp




$$a^1 = f^1(W^1p+b^1) \qquad a^2 = f^2(W^2a^1+b^2) \qquad a^3 = f^3(W^3a^2+b^3)$$

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1p+b^1)+b^2)+b^3)$$

# Mạng hồi qui



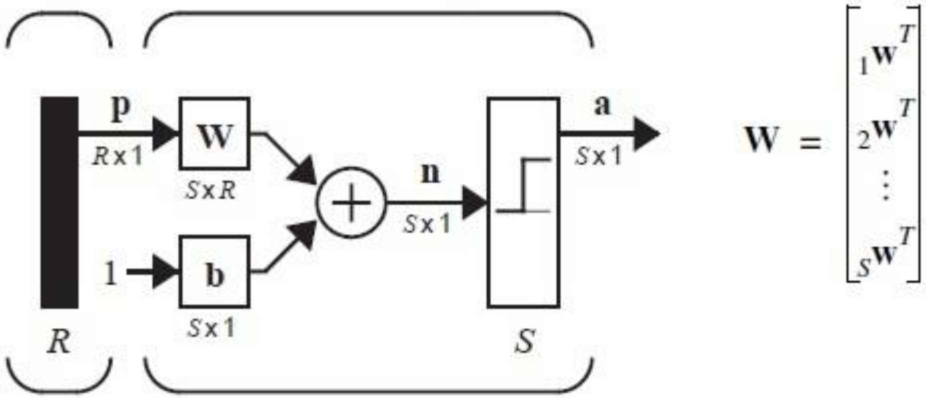
Khâu trễ
Điều kiện đầu
Lớp hồi qui
u(t)
a(t)
D
a(0)
a(t) = u(t - 1)
p
Sx1
S
W
SxS
b
Sx1
1
n(t+1)
Sx1
a(t+1)
Sx1
S
D
a(t)
Sx1
a(0) = p
a(t+1) = satlins (Wa(t)+b)

# Luật học

- **Luật học**: Là một thủ tục dùng để thay đổi các trọng số và bias nhằm mục đích huấn luyện mạng để thực hiện một số nhiệm vụ.
- **Có 3 loại luật học**:
  - ❑ Học có giám sát
  - ❑ Học không có giám sát
  - ❑ Học gia cố
- **Học có giám sát**:
  - ❑ Mẫu huấn luyện $\{p_1; t_1\}, \{p_2; t_2\}, \dots, \{p_Q; t_Q\}$
  - ❑ Luật học sẽ điều chỉnh các trọng số và bias sao cho đầu ra của mạng nơ-ron bám lấy đầu ra mẫu.

# 1.1.4 Các loại mạng nơ-ron

- Mạng perceptron
- Ứng dụng: Có thể giải quyết các bài toán phân loại với đường biên tuyến tính.

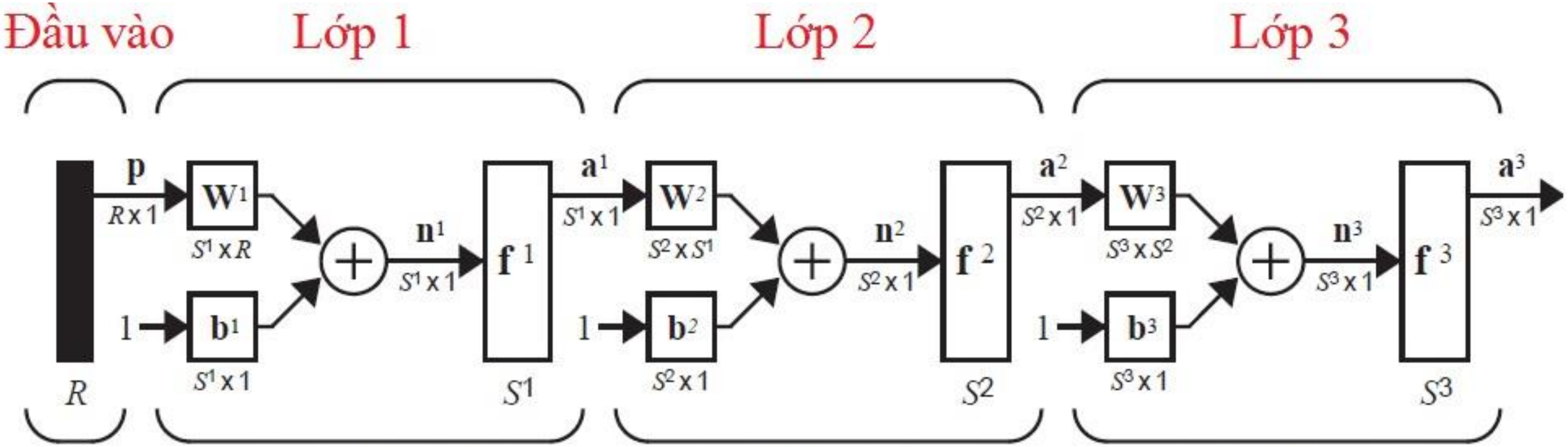



$$\mathbf{a} = \mathbf{hardlim}(\mathbf{Wp}+\mathbf{b})$$

$$a_i = hardlim(n_i) = hardlim({}_i\mathbf{w}^T\mathbf{p} + b_i)$$

# 1.1.4 Các loại mạng nơ-ron

- Mạng nhiều lớp (mở rộng của mạng perceptron)
- Ứng dụng:
  - ❑ Có thể giải quyết bài toán phân loại bất kỳ
  - ❑ Có thể xấp xỉ một hàm phi tuyến bất kỳ

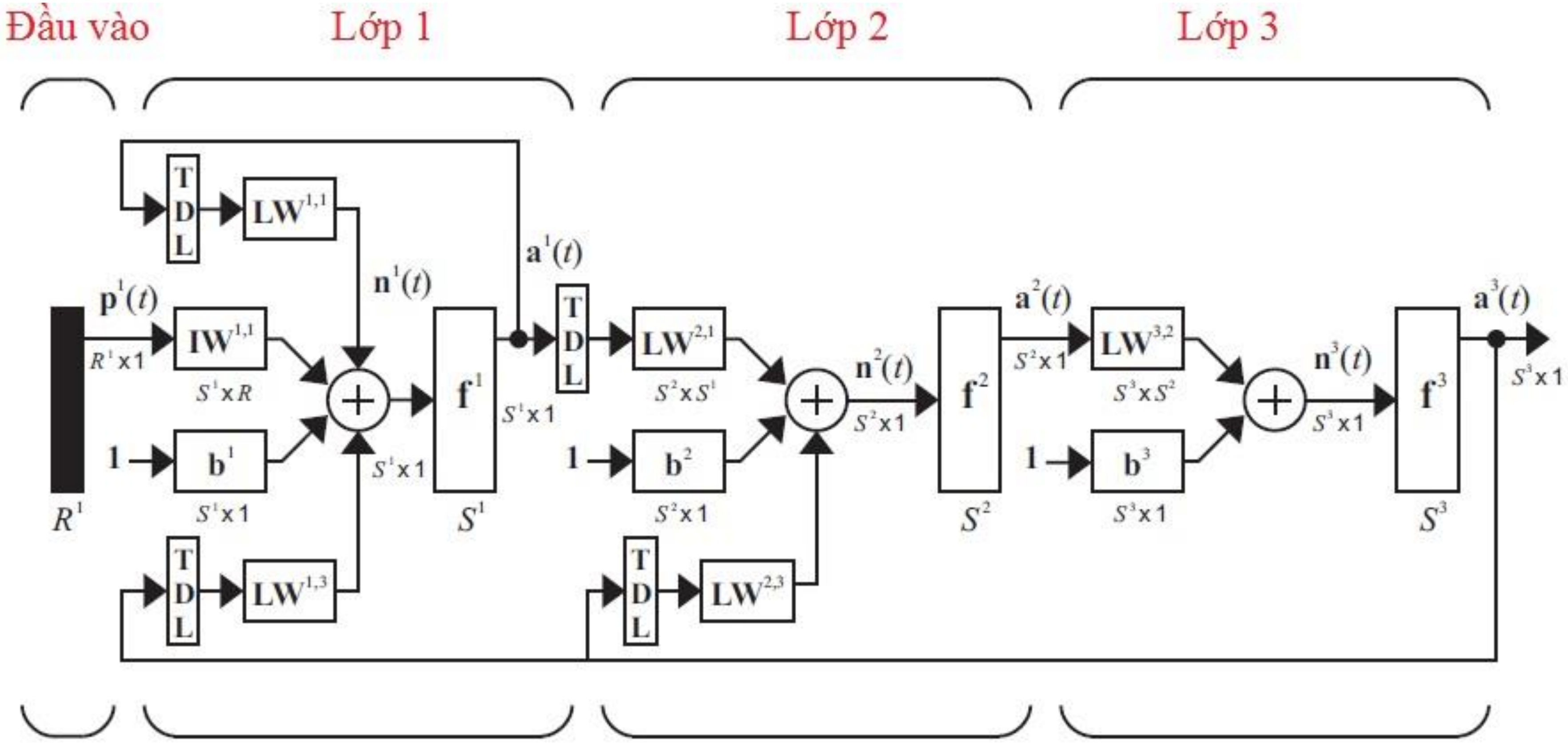



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{W}^2\mathbf{a}^1+\mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{a}^2+\mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

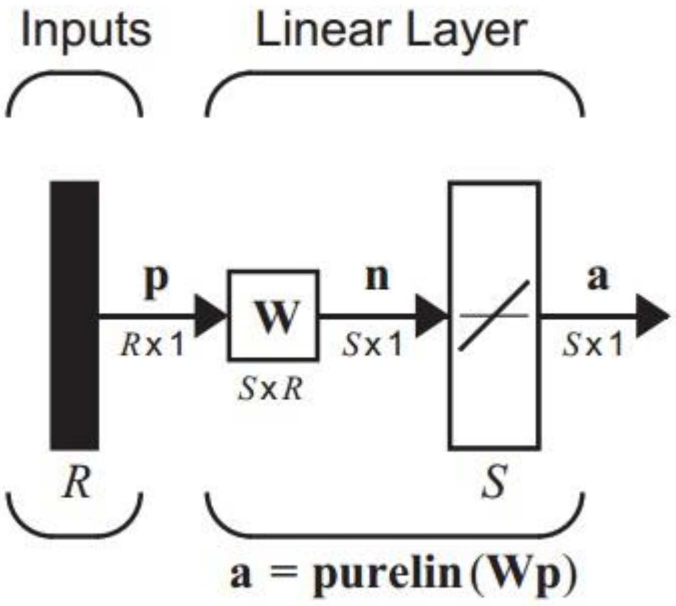
# 1.1.4 Các loại mạng nơ-ron

- Mạng nơ-ron động
- Ứng dụng: Nhận dạng và điều khiển các hệ thống động học và phi tuyến



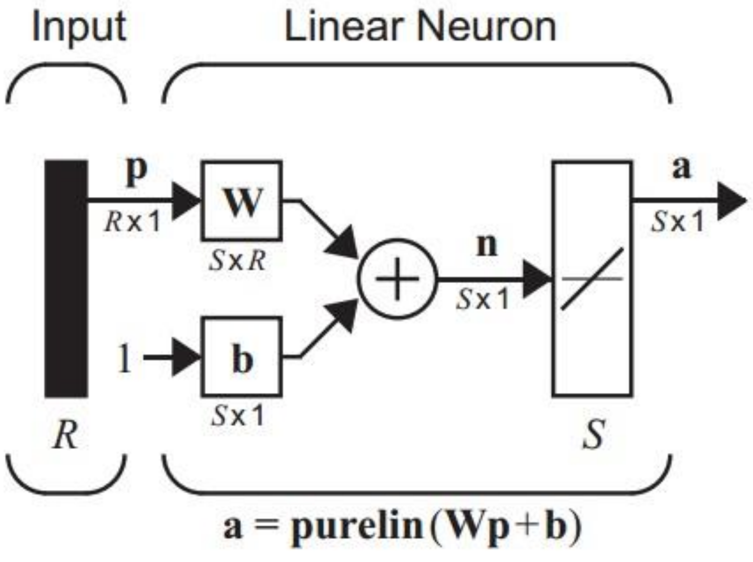
# 1.1.4 Các loại mạng nơ-ron

- Mạng nhớ:
  - ❑ Ứng dụng để phân loại mẫu (chữ viết, ảnh)
  - ❑ Dùng luật học Hebbian

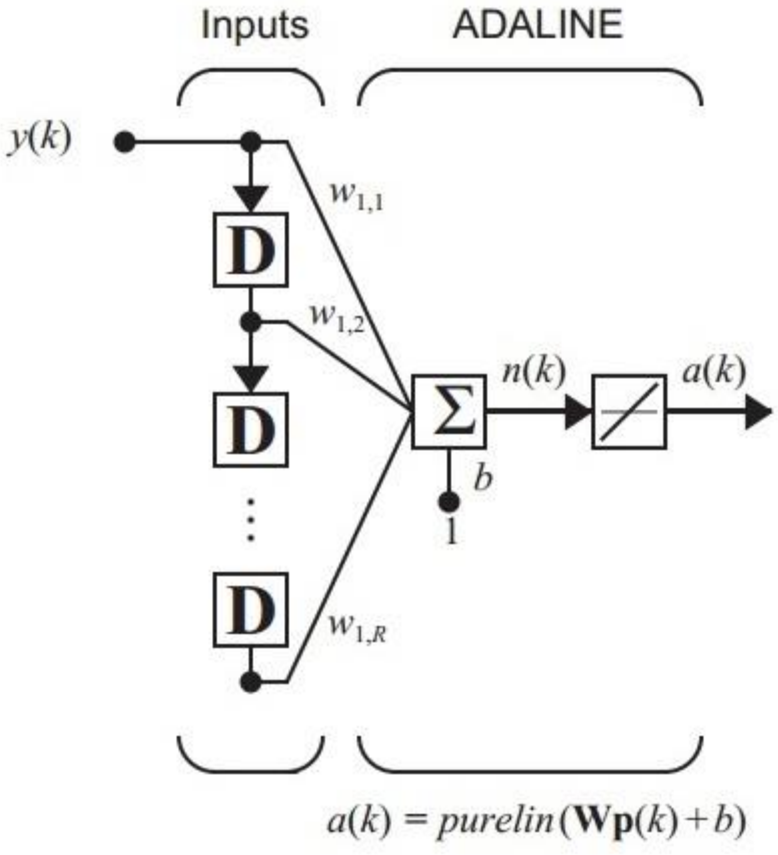


# 1.1.4 Các loại mạng nơ-ron

- Mạng ADALINE :
  - ❑Bộ lọc thích nghi.
  - ❑Dùng luật học Widrow – Hoff



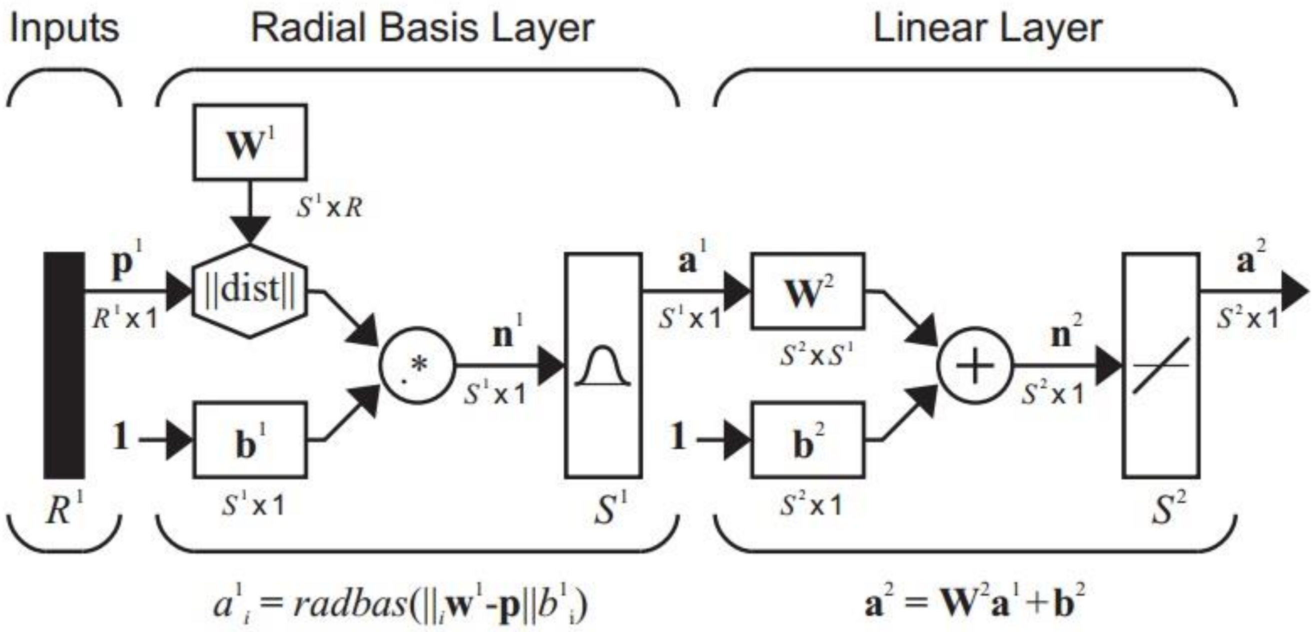

ADALINE Network




Adaptive Filter ADALINE

# 1.1.4 Các loại mạng nơ-ron

- Mạng RBF (Radial Basis Function): $a = e^{-n^2}$
  - ❑ Ứng dụng xấp xỉ hàm, phân loại mẫu
  - ❑ Dùng phương pháp bình phương nhỏ nhất




Radial Basis Network

# Ví dụ

- Phân loại táo và cam dùng mạng perceptron (nnd3pc)
- Các cảm biến:
  - ❑ Hình dáng: 1 nếu gần như tròn, -1 nếu gần như e líp
  - ❑ Bề mặt: 1 nếu như trơn, -1 nếu gồ ghề
  - ❑ Khối lượng: 1 nếu khối lượng lớn hơn 1 pound (0,45kg), -1 nếu nhỏ hơn 1 pound
- Ta có thể chọn đầu ra của mạng perceptron là 1 khi đầu vào là quả táo và đầu ra là -1 khi đầu vào là quả cam.
- Vì chỉ phân loại 2 sản phẩm, ta có thể dùng mạng perceptron có 1 nơ-ron. Ở đây dùng 3 cảm biến, do đó sẽ có 3 đầu vào.

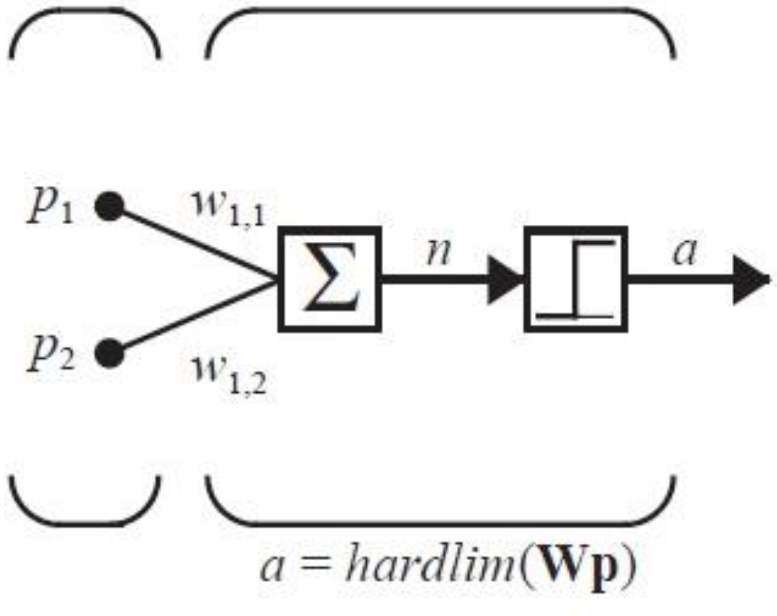
# 1.2. Phương pháp huấn luyện mạng nơron.

- 1.2.1 Ví dụ
- 1.2.2 Luật học perceptron
- 1.2.3 Thuật toán lan truyền ngược

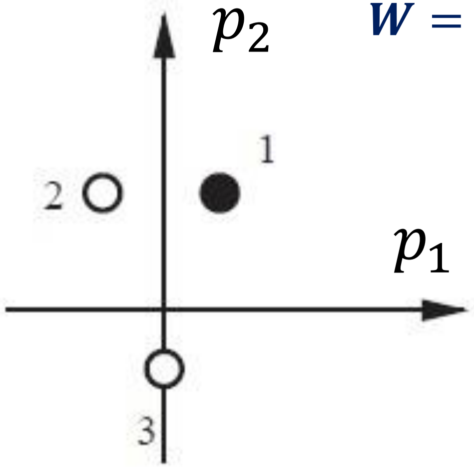
# 1.2.1 Ví dụ

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

Đầu vào　　Nơ-ron không có bias




$\boldsymbol{W} = [1 \quad -0{,}8]$




$$\boldsymbol{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \boldsymbol{W} = [w_{1,1} \quad w_{1,2}]$$

$$a = hardlim(n) = \begin{cases} 0, \text{nếu } n < 0 \\ 1, \text{nếu } 0 \le n \end{cases}$$

$$n = \boldsymbol{W}\boldsymbol{p}$$

# 1.2.1 Ví dụ

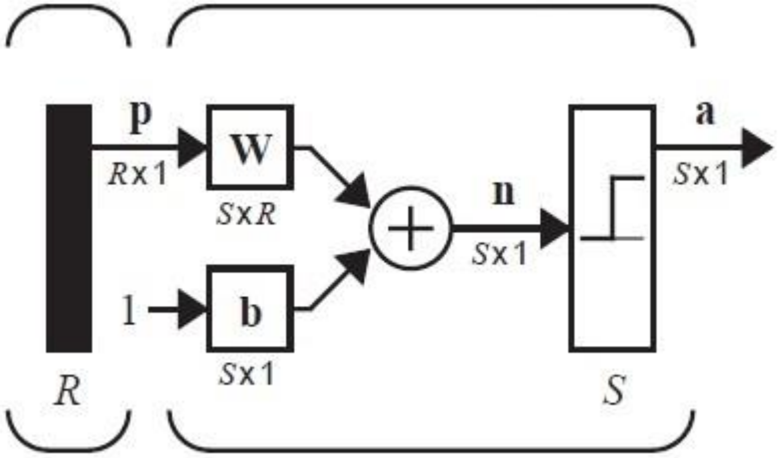
- $e = t - a$
- t: đầu ra mẫu tương ứng với đầu vào mẫu
- a: đầu ra của nơ-ron ứng với đầu vào mẫu
- Nếu e = 1, thì $\boldsymbol{w}_{\boldsymbol{m}ớ\boldsymbol{i}} = \boldsymbol{w}_{\boldsymbol{c}ũ} + \boldsymbol{p}^{\boldsymbol{T}}$
- Nếu e = −1, thì $\boldsymbol{w}_{\boldsymbol{m}ớ\boldsymbol{i}} = \boldsymbol{w}_{\boldsymbol{c}ũ} - \boldsymbol{p}^{\boldsymbol{T}}$
- Nếu e = 0, thì $\boldsymbol{w}_{\boldsymbol{m}ớ\boldsymbol{i}} = \boldsymbol{w}_{\boldsymbol{c}ũ}$
  - $\boldsymbol{w}_{\boldsymbol{m}ớ\boldsymbol{i}} = \boldsymbol{w}_{\boldsymbol{c}ũ} + e\boldsymbol{p}^{\boldsymbol{T}}$
    - $b_{\boldsymbol{m}ớ\boldsymbol{i}} = b_{\boldsymbol{c}ũ} + e$

# 1.2.2 Luật học perceptron

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

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}. \qquad \mathbf{W} = \begin{bmatrix} {}_1\mathbf{w}^T \\ {}_2\mathbf{w}^T \\ \vdots \\ {}_S\mathbf{w}^T \end{bmatrix}.$$

Đầu vào Lớp nơ-ron




$\mathbf{a} = \mathbf{hardlim}(\mathbf{Wp}+\mathbf{b})$

$${}_i\boldsymbol{w}_{\boldsymbol{m\acute{ơ}i}} = {}_i\boldsymbol{w}_{\boldsymbol{c\tilde{u}}} + e_i\boldsymbol{p}$$

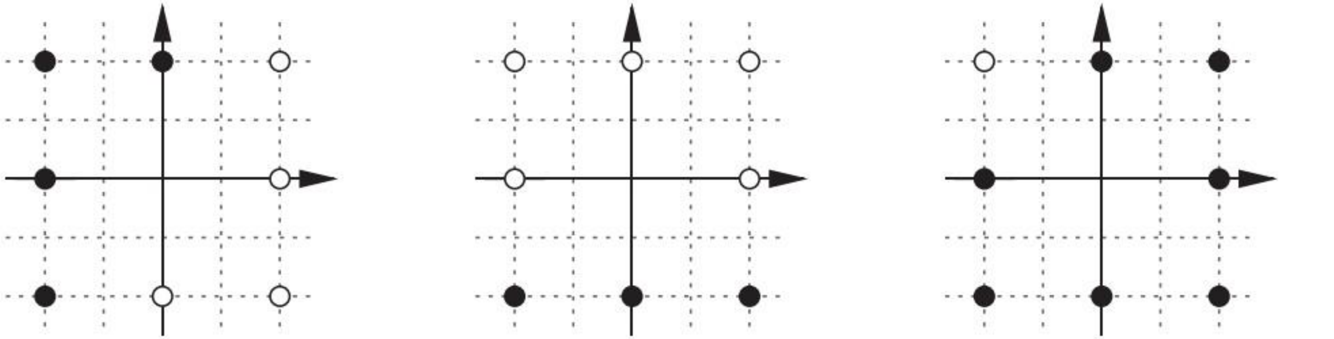
$${}_ib_{\boldsymbol{m\acute{ơ}i}} = {}_ib_{\boldsymbol{c\tilde{u}}} + e_i$$

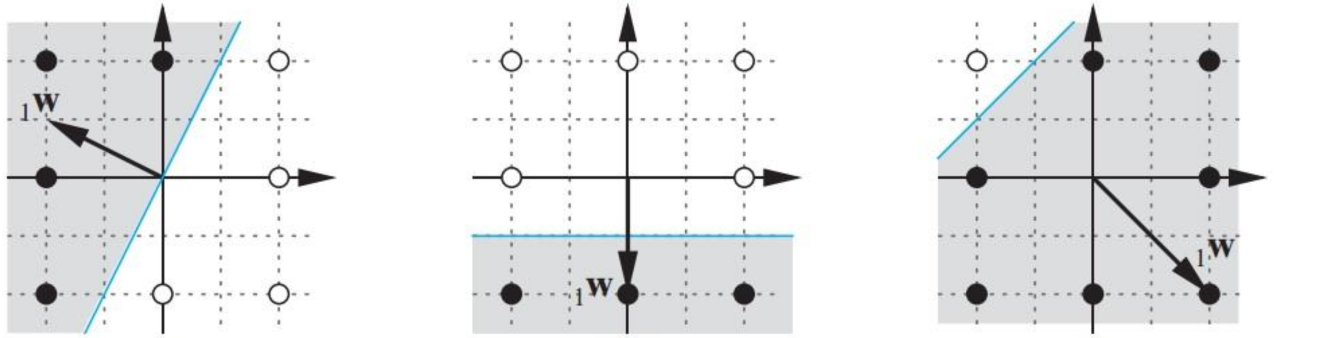
$$\boldsymbol{W}_{\boldsymbol{m\acute{ơ}i}} = \boldsymbol{W}_{\boldsymbol{c\tilde{u}}} + \boldsymbol{e}\boldsymbol{p}^{\boldsymbol{T}}$$

$$\boldsymbol{b}_{\boldsymbol{m\acute{ơ}i}} = \boldsymbol{b}_{\boldsymbol{c\tilde{u}}} + \boldsymbol{e}$$

# Bài tập

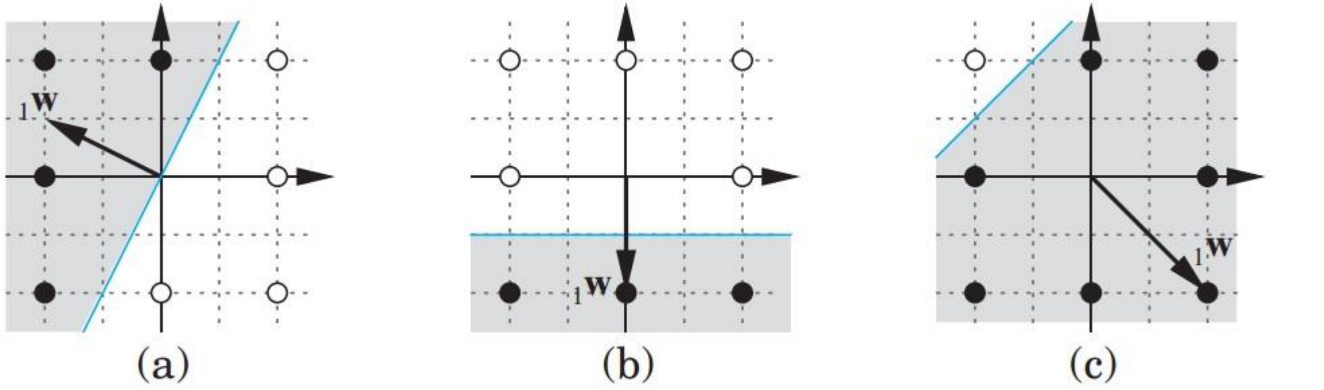
- 1. Giải các bài toán phân loại sau bằng cách vẽ các đường biên và tìm các trọng số và bias.


(a) (b) (c)



(a) (b) (c)

# Bài tập



$$(a)\ {}_1\mathbf{w}^T = \begin{bmatrix} -2 & 1 \end{bmatrix},\quad (b)\ {}_1\mathbf{w}^T = \begin{bmatrix} 0 & -2 \end{bmatrix},\quad (c)\ {}_1\mathbf{w}^T = \begin{bmatrix} 2 & -2 \end{bmatrix}$$

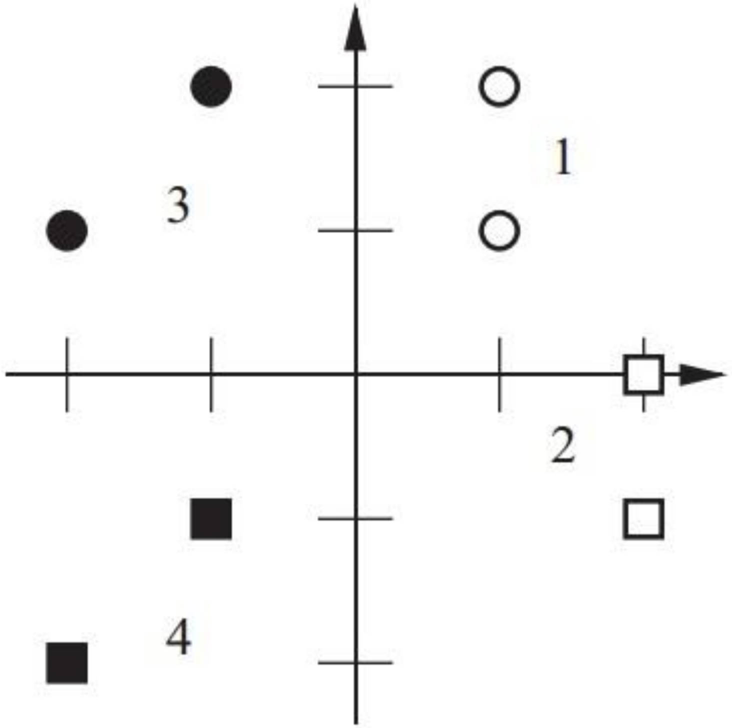
$$\begin{aligned} {}_1\mathbf{w}^T\mathbf{p} + b &= 0 \\ b &= -{}_1\mathbf{w}^T\mathbf{p} \end{aligned}$$

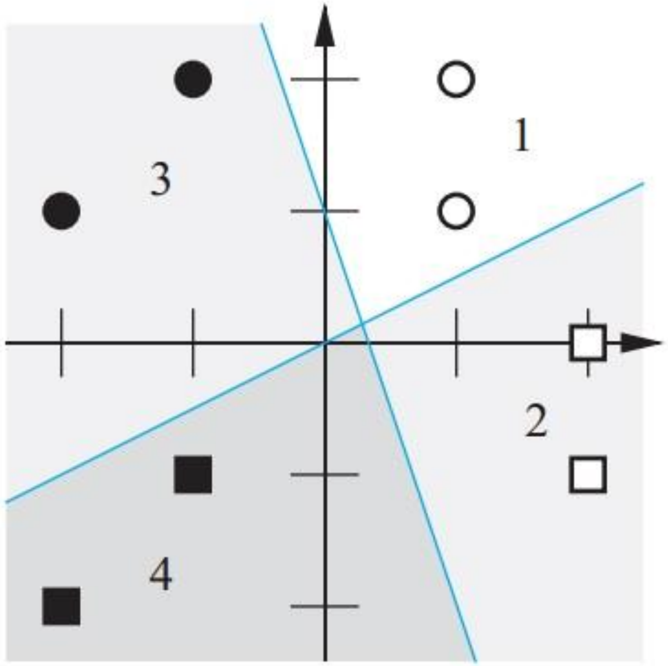
$$(a)\ b = -\begin{bmatrix} -2 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0,\ (b)\ b = -\begin{bmatrix} 0 & -2 \end{bmatrix}\begin{bmatrix} 0 \\ -1 \end{bmatrix} = -2,\ (c)\ b = -\begin{bmatrix} 2 & -2 \end{bmatrix}\begin{bmatrix} -2 \\ 1 \end{bmatrix} = 6$$

# Bài tập

- 2. Thiết kế mạng nơ-ron để phân loại các mẫu được biểu diễn như hình sau:

Mạng có 2 noron=> 2 đường biên



# Bài tập

- 3. Cho một bài toán phân loại như sau:

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

$$\left\{\mathbf{p}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t_4 = 0\right\} \left\{\mathbf{p}_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t_5 = 0\right\}.$$

- a, Vẽ sơ đồ mạng perceptron 1 nơ-ron mà bạn sẽ sử dụng để giải bài toán này. Cần có bao nhiêu đầu vào?
- b, Biểu diễn các mẫu trên mặt phẳng có hệ trục tọa độ là $Op_1p_2$. Với mạng đã được thiết kế ở ý (a) bài toán trên có lời giải hay không? Tại sao?

# Cơ sở hệ mờ và mạng nơ ron

Nguyễn Hoài Nam

Bộ môn Điều khiển tự động, Viện Điện, Trường Đại học Bách Khoa Hà Nội

*Email: nam.nguyenhoai@hust.edu.vn*
*Trang cá nhân: https://sites.google.com/view/n2c*

Ngày 6 tháng 5 năm 2021

# Overview

# Phương pháp huấn luyện mạng

- Thuật toán lan truyền ngược
- Phương pháp gradient
- Phương pháp chỉnh hướng học

# Thuật toán lan truyền ngược

- Mạng nhiều lớp: Xét mạng có $M$ lớp, $R$ đầu vào và $S^M$ đầu ra.
- Tập mẫu: $\Omega = \{\mathbf{p}_i; \mathbf{t}_i\}, i = 1 \div Q$, $\mathbf{p}_i$ - đầu vào mẫu và $\mathbf{t}_i$ - đầu ra mẫu (đích).
- Hàm mục tiêu:

$$J = \frac{1}{Q} \sum_{q=1}^{Q} J_q \to min, \tag{1}$$

trong đó $J_q = \mathbf{e}_q^T \mathbf{e}_q$, $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$, $\mathbf{a}_q^M$ - đầu ra tương ứng với đầu vào $\mathbf{p}_q$.

# Thuật toán lan truyền ngược

- Phương pháp gradient

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial J_q}{\partial w_{i,j}^m}, \tag{2}$$

$w_{i,j}^m$: trọng số của nơ ron thứ $i$ thuộc lớp $m$, liên kết với đầu ra của nơ-ron thứ $j$ thuộc lớp $m-1$; $k$: thể hiện giá trị hiện tại; $k+1$: thể hiện giá trị mới, $\alpha > 0$: tốc độ học.

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial J_q}{\partial b_i^m}, \tag{3}$$

$b_i^m$: ngưỡng của nơ-ron thứ $i$ thuộc lớp $m$.

# Quy tắc chuỗi

$$\frac{\partial J_q}{\partial w_{i,j}^m} = \frac{\partial J_q}{\partial n_i^m} \frac{\partial n_i^m}{\partial w_{i,j}^m} \tag{4}$$

$$n_i^m = b_i^m + \sum_{v=1}^{S^{m-1}} w_{i,v}^m a_v^{m-1}, \tag{5}$$

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1} \tag{6}$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1. \tag{7}$$

# Độ nhạy

Độ nhạy của nơ-ron thứ *i* thuộc lớp m:

$$s_i^m = \frac{\partial J_q}{\partial n_i^m} \tag{8}$$

Quá trình cập nhập các tham số của

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \tag{9}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \tag{10}$$

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m(\mathbf{a}^{m-1})^T \tag{11}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m, \tag{12}$$

$$\mathbf{s}^m = \frac{\partial J_q}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial J_q}{\partial n_1^m} & \frac{\partial J_q}{\partial n_2^m} & \cdots & \frac{\partial J_q}{\partial n_i^m} \end{bmatrix}^T, \tag{13}$$

# Độ nhạy

$$\frac{\partial {n_i}^{m+1}}{\partial {n_j}^{m}} = \frac{\partial \left( \sum_{l=1}^{S^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m}$$

$$= w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m}$$

$$= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m}$$

$$= w_{i,j}^{m+1} \dot{f}^m(n_j^m).$$

# Độ nhạy

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{S^m}^m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{bmatrix} = \mathbf{W}^{m+1}\dot{\mathbf{F}}^m(\mathbf{n}^m) \qquad (14)$$

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{S^m}^m) \end{bmatrix}. \qquad (15)$$

# Độ nhạy

Độ nhạy của lớp

$$\begin{aligned}\mathbf{s}^m &= \frac{\partial J_q}{\partial \mathbf{n}^m} \\ &= \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}\right)^T \frac{\partial J_q}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \frac{\partial J_q}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}.\end{aligned}$$

Từ công thức này ta có thể tính được độ nhạy của các lớp lần lượt từ lớp đầu ra ngược trở về lớp đầu vào như sau:

$$\mathbf{s}^M \to \mathbf{s}^{M-1} \to \ldots \to \mathbf{s}^2 \to \mathbf{s}^1. \tag{16}$$

# Độ nhạy

Độ nhạy noron thứ i thuộc lớp M

$$
\begin{aligned}
s_i^M &= \frac{\partial J_q}{\partial n_i^M} \\
&= \frac{\partial \left(\mathbf{t}_q - \mathbf{a}_q\right)^T \left(\mathbf{t}_q - \mathbf{a}_q\right)}{\partial n_i^M} \\
&= \frac{\partial \sum_{j=1}^{S^M} \left(t_j - a_j\right)^2}{\partial n_i^M} \\
&= -2(t_i - a_i)\frac{\partial a_i}{\partial n_i^M}.
\end{aligned} \tag{17}
$$

Vector độ nhạy lớp đầu ra

$$
\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t}_q - \mathbf{a}_q). \tag{18}
$$

Sai lệch giữa đầu ra mẫu và đầu ra tương ứng với đầu vào q

# Thuật toán lan truyền ngược

- Quá trình khởi tạo mạng: Chọn giá trị ban đầu cho các tham số.
- Quá trình lan truyền thuận

$$\mathbf{a}^0 = \mathbf{p}, \tag{19}$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}), \forall\, m = 0, 1, ..., M-1. \tag{20}$$

$$\mathbf{a} = \mathbf{a}^M. \tag{21}$$

- Quá trình lan truyền ngược độ nhạy

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}), \tag{22}$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T\mathbf{s}^{m+1}, \forall\, m = M-1, ..., 2, 1. \tag{23}$$

- Quá trình cập nhật các tham số của mạng

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha\mathbf{s}^m(\mathbf{a}^{m-1})^T \tag{24}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha\mathbf{s}^m \tag{25}$$

# Ví dụ

Xấp xỉ hàm số sau

$$y = 1 + cos(\frac{\pi}{4}x)$$

với $-2 \leq x \leq 2$.
Chọn mạng có cấu trúc gồm 2 lớp, lớp 1 có hàm truyền là *logsig* và lớp 2 có hàm truyền *purelin*.
Đầu vào mẫu thứ $i$:

$$x_i = -2 + 4i/(Q-1),\ \textit{với}\ i = 0, 1, 2..., Q-1. \tag{26}$$

Đầu ra mẫu tương ứng với $x_i$ là:

$$y_i = 1 + cos\left(\frac{\pi}{4}x_i\right). \tag{27}$$

# Ví dụ

Mẫu thứ nhất là: $x_1 = -2, y_1 = 1$.

- Khởi tạo mạng: $\mathbf{w}^1 = [1 \quad 1]^T$, $\mathbf{b}^1 = [2 \quad 2]^T$, $\mathbf{w}^2 = [1 \quad 1]$, $b^2 = -1$ và tốc độ học là $\alpha = 0.1$.
- Quá trình lan truyền thuận: $a^0 = -2$,

$$\begin{aligned}\mathbf{a}^1 &= \mathbf{f}^1(\mathbf{w}^1 a^0 + \mathbf{b}^1)\\ &= logsig\left(\begin{bmatrix}1\\1\end{bmatrix}(-2) + \begin{bmatrix}2\\2\end{bmatrix}\right)\\ &= logsig\left(\begin{bmatrix}0\\0\end{bmatrix}\right) = \begin{bmatrix}0,5\\0,5\end{bmatrix}\end{aligned} \tag{28}$$

$$\begin{aligned}a^2 &= f^2(\mathbf{w}^2\mathbf{a}^1 + b^2)\\ &= purelin([1 \quad 1]\begin{bmatrix}0,5\\0,5\end{bmatrix} - 1)\\ &= 0\end{aligned} \tag{29}$$

# Ví dụ

- Quá trình lan truyền ngược:
  $a = purelin(n) \rightarrow \frac{\partial a}{\partial n} = 1$, $a = logsig(n) \rightarrow \frac{\partial a}{\partial n} = a(1-a)$.
  Sai lệch là: $e = y_1 - a^2 = 1$.
  Độ nhạy của lớp 2:

$$\begin{aligned} s^2 &= -2\dot{f}^2(n^2)(t - a^2) \\ &= -2 \end{aligned} \tag{30}$$

Độ nhạy của lớp 1:

$$\begin{aligned} \mathbf{s}^1 &= \dot{\mathbf{F}}^2(\mathbf{n}^2)(\mathbf{w}^2)^T s^2 \\ &= \begin{bmatrix} 0,25 & 0 \\ 0 & 0,25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} (-2) = \begin{bmatrix} -0,5 \\ -0,5 \end{bmatrix} \end{aligned} \tag{31}$$

$$\dot{F}^1(n^1) = \begin{bmatrix} \partial a_1^1/\partial n_1^1 & 0 \\ 0 & \partial a_2^1/\partial n_2^1 \end{bmatrix}$$

$$= \begin{bmatrix} a_1^1(1-a_1^1) & 0 \\ 0 & a_2^1(1-a_2^1) \end{bmatrix}$$

- Quá trình cập nhật các tham số của mạng.

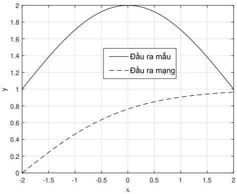
$$\begin{aligned}\mathbf{w}^1(1) =& \mathbf{w}^1(0) - \alpha \mathbf{s}^1 a^0 \\ =& \begin{bmatrix}1\\1\end{bmatrix} - 0.1\begin{bmatrix}-0,5\\-0,5\end{bmatrix}(-2) = \begin{bmatrix}0,9\\0,9\end{bmatrix}\end{aligned} \tag{32}$$

$$\begin{aligned}\mathbf{b}^1(1) =& \mathbf{b}^1(0) - \alpha \mathbf{s}^1 \\ =& \begin{bmatrix}2\\2\end{bmatrix} - 0.1\begin{bmatrix}-0,5\\-0,5\end{bmatrix} = \begin{bmatrix}2,05\\2,05\end{bmatrix}\end{aligned} \tag{33}$$

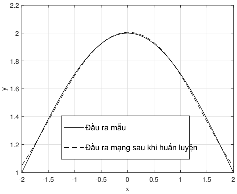
$$\begin{aligned}\mathbf{w}^2(1) =& \mathbf{w}^2(0) - \alpha s^2(\mathbf{a}^1)^T \\ =& \begin{bmatrix}1 & 1\end{bmatrix} - 0.1(-2)\begin{bmatrix}0,5 & 0,5\end{bmatrix} = \begin{bmatrix}1,1 & 1,1\end{bmatrix}\end{aligned} \tag{34}$$

$$\begin{aligned}b^2(1) =& b^2(0) - \alpha s^2 \\ =& -1 - 0.1(-2) \\ =& -0,8\end{aligned} \tag{35}$$

# Ví dụ




Hình 1: Đầu ra mạng với bộ tham số ban đầu.



# Phương pháp gradient ngẫu nhiên - Stochastic Gradient Descent (SGD)

- Cho hàm mục tiêu $F(\mathbf{x})$, trong đó $\mathbf{x}$ là véc tơ các tham số (trọng số, ngưỡng). Tìm nghiệm tối ưu $\mathbf{x}^*$ sao cho $F(\mathbf{x}^*) \rightarrow min$.
- Công thức lặp

Lặp lại nhiều lần sẽ tiến đến nghiệm tối ưu x*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F, \tag{36}$$

trong đó $\mathbf{x}_k$ là véc tơ tham số hiện tại, $\mathbf{x}_{k+1}$ là véc tơ tham số mới ở lần lặp thứ $k$, $\nabla F = \frac{\partial F}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_k}$ là véc tơ gradient, và $\alpha > 0$ là tốc độ học.

# Phương pháp SGD với hệ số chỉnh hướng học[1]

P2 Gradient ngẫu nhiên với hệ số chỉnh hướng học

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla F + \gamma(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{37}$$

trong đó $\gamma > 0$ là hệ số chỉnh hướng học.

$$\Delta x_k = x_{k+1} - x_k$$
$$\Delta x_{k-1} = x_k - x_{k-1}$$

$$\underline{\mathbf{x}_{k+1}} = \underline{\mathbf{x}_k} - \underline{\alpha\nabla F} + \gamma(\mathbf{x}_k - \mathbf{x}_{k-1}),$$

là hệ số chỉnh hướng học.

$$\Delta x_k = \gamma\,\Delta x_{k-1} - \alpha\nabla F$$

[1] [2] Murphy, K. P. ... e. The MIT Press, Cambridge, Massac...

# HỌC SÂU

Nguyễn Hoài Nam

Bộ môn Điều khiển tự động, Viện Điện, Trường Đại học Bách Khoa Hà Nội

*Email: nam.nguyenhoai@hust.edu.vn*
*Trang cá nhân: https://sites.google.com/view/n2c*

Ngày 27 tháng 5 năm 2021

# Overview

1. Các lớp của mạng sâu
2. Tạo mạng và huấn luyện mạng
3. Nhận dạng 10 chữ số viết tay

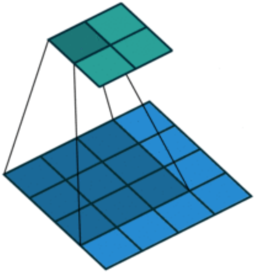
- Ảnh đầu vào.
  - Ảnh 2 chiều (2-D): $w \times h \times c$, trong đó $w$ là độ rộng, $h$ là độ dài và $c$ là số kênh của ảnh. $c = 1$-ảnh đen trắng, $c = 3$ - ảnh mầu.
  - Ảnh 3 chiều (3-D)$w \times h \times \times d \times c$, trong đó $w$ là độ rộng, $h$ là độ dài, $d$ là độ sâu và $c$ là số kênh của ảnh.
  - Ví dụ ảnh 2-D:
  moon = imread('moon.tif'); imshow(moon);

Hình 1: Ảnh 2-D, độ phân giải $537 \times 358$

```
>> moon(100:105,200:205)

ans =

  6×6 uint8 matrix

   207   209   209   207   214   215
   204   207   205   209   217   221
   209   213   209   214   217   224
   212   206   213   211   211   218
   215   216   212   209   216   215
   216   225   211   210   224   214
```

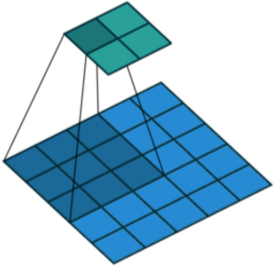
Hình 2: Một số điểm ảnh của hình mặt trăng

- Lớp tích chập
  - Đầu vào: $P_{m\times n}$
  - $W_{f\times f}$- ma trận trọng số của nơ-ron, còn được gọi là bộ lọc có kích thước là $f < min(m,n)$.
  - Đầu ra: $A = P * W$.



Hình 3: $Stride = 1$

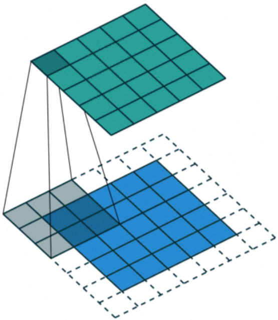
# Lớp tích chập



Hình 4: *Stride* $= 2$

# Lớp tích chập

Padding = 1. Có thể thay đổi kích thước đầu ra của lớp.



Hình 5: *Padding* $= [1\ 1]$

# Lớp chuẩn hóa

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (1)$$

$\mu_B$-giá trị trung bình, $\sigma_B^2$ - phương sai (mini batch, mỗi kênh)

$$y_i = \gamma \hat{x}_i + \beta \quad (2)$$

$\gamma, \beta$ - các tham số học

# Lớp ReLu

$$
\begin{aligned}
f(x) =& x, \text{nếu } x \geq 0 \\
=& 0, \text{nếu } x < 0
\end{aligned}
\tag{3}
$$

# Lớp poopling

- Làm giảm kích thước đầu vào.
- Chia đầu vào thành các vùng có dạng hình chữ nhật và lấy giá trị trung bình hoặc cực đại mỗi vùng.

- Hàm max
- Hàm average

# Lớp đầu ra

- Hàm softmax.

$$y_j(x) = \frac{e^{a_j(x)}}{\sum_{j=1}^{k} e^{a_j(x)}} \tag{4}$$

$a_j(x)$ - là đầu ra thứ $j$ của lớp liên kết đủ trước đó.

- Lớp đầu vào:
  $input\_layer = imageInputLayer(inputSize)$

$$inputSize = [h \quad w \quad c], \tag{5}$$

$h$ - độ cao, $w$ - độ rộng, $c$ - số kênh
$c = 1$ ảnh đen trắng
$c = 3$ ảnh màu.

- Lớp tích chập:
  $conv\_layer = convolution2dLayer(filterSize, numFilters)$

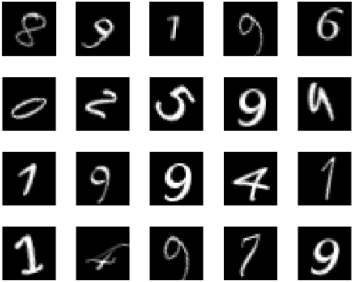
$$\begin{aligned} filterSize &= [h \quad w], \\ numFilters &= \text{ số filter} \end{aligned} \tag{6}$$

  $h$ - độ cao, $w$ - độ rộng, $c$ - số kênh
- $conv\_layer = convolution2dLayer(filterSize, numFilters,' P1', V1,' P2', V2, ...)$.
- 'Stride': $[u \quad v]$, $u$ - bước dọc, $v$ - bước ngang
- 'Padding': 'same', 'scalar', [a  b], [t  b  l  r].

# Tạo mạng và huấn luyện mạng

- batchNormalizationLayer
- reluLayer
- fullyConnectedLayer(outputSize)
- softmaxLayer
- classificationLayer

- Nhận dạng 10 chữ số viết tay $0 \div 9$.
- Tập mẫu gồm 10000 ảnh, mỗi chữ số có 1000 ảnh mẫu.



Hình 6: Một số ảnh chữ số viết tay

# References


- Deep Learning Toolbox User's Guide, Matlab.
- Martin Hagan and et al. Neural network design.