Trí Tuệ Nhân Tạo

GVGD: Th.S Nguyễn Võ Ngọc Thạch

Khoa Cơ Khí Công Nghệ

ĐH Nông Lâm Tp.HCM

Email: nvnthach@hcmuaf.edu.vn

MỤC LỤC

Chương 1 Logic mờ và ứng dụng ….. 5

1.1 Tập hợp mờ

1.1.1 Tập hợp rõ

1.1.2 Tập hợp mờ

1.1.3 Các phép toán trên tập hợp mờ

1.1.4 Biến ngôn ngữ

1.2 Quan hệ mờ

1.2.1 Quan hệ rõ

1.2.2 Quan hệ mờ

1.2.3 Hình chiếu

1.2.4 Kết hợp các quan hệ rõ

1.2.5 Kết hợp các quan hệ mờ

1.3 Suy diễn mờ

1.3.1 Mệnh đề mờ

1.3.2 Diễn dịch luật IF … THEN …

1.3.3 Logic cổ điển

1.3.4 Logic mờ

1.4 Hệ thống xử lý mờ

1.4.1 Khối mờ hóa

1.4.2 Khối suy diễn mờ

1.4.3 Khối giải mờ

1.4.4 Tính phi tuyến của hệ thống xử lý mờ

1.4.5 Hệ thống xử lý mờ loại 2

1.5 Điều khiển mờ

1.6 Nhận dạng hàm thành viên và luật hợp thành

1.6.1 Nhận dạng luật hợp thành

1.6.2 Nhận dạng hàm thành viên

1.6.3 Nhận dạng hệ thống động dùng mô hình mờ

1.6.4 Mô phỏng và dự báo

1.6.5 Cân bằng kênh thông tin

1.7 Phân nhóm mờ

1.7.1 Phân nhóm rõ

1.7.2 Phân nhóm mờ

1.8 Hướng dẫn sử dụng phần mềm Matlab/Fuzzy toolbox

Bài tập

Chương 2 Mạng neuron nhân tạo …….. 43

2.1 Khái niệm cơ bản

2.1.1 Mạng neuron nhân tạo

2.1.2 Neuron

2.1.3 Kết nối

2.1.4 Huấn luyện mạng neuron

2.2 Mạng perceptron một lớp

2.2.1 Đơn vị tuyến tính (linear unit, LU)

2.2.2 Đơn vị phân loại tuyến tính (linear graded unit, LGU)

2.2.3 Đơn vị ngưỡng tuyến tính (linear threshold unit, LTU)

2.2.4 Ứng dụng trong phân nhóm

2.2.5 Ứng dụng trong nhận dạng hệ thống động

2.2.6 Ứng dụng trong điều khiển

2.2.7 Ứng dụng trong nhận dạng ký tự

2.3 Mạng truyền thẳng nhiều lớp

2.3.1 Tổng quan

2.3.2 Giải thuật huấn luyện lan truyền ngược (back propagation)

2.3.3 Các thông số của luật học lan truyền ngược

2.4 Mạng hồi quy

2.5 Mạng Hopfield

2.5.1 Mạng Hopfield rời rạc

2.5.2 Mạng Hopfield liên tục

2.5.3 Bộ nhớ kết hợp (associative memory)

2.5.4 Hopfield recurrent associative memory

2.5.5 Bộ nhớ kết hợp hai chiều (bidirectional associative memory BAM)

2.6 Các giải thuật học không giám sát

2.6.1 Luật học Hebb

2.6.2 Luật học cạnh tranh
2.6.3 Differential Hebbian learning rule
2.6.4 Differential competitive learning rule
2.7 Self-organizing feature maps
2.8 Hướng dẫn sử dụng phần mềm Matlab/Neural networks toolbox
2.8.1 Ví dụ
2.8.2 Neural Network Toolbox
2.8.3 Các hàm tiêu biểu
Bài tập

Chương 3 : Hệ thống neuron - mờ và mờ - neuron …..77
3.1 Hệ thống mờ trên cơ sở mạng neuron
3.2 Mạng neuron trên cơ sở logic mờ
3.3 Các ứng dụng tiêu biểu
Bài tập

Chương 4 : Giải thuật di truyền ….. 85
4.1 Giải thuật di truyền
4.2 Các thông số của giải thuật di truyền
4.3 Các ứng dụng tiêu biểu
Bài tập

Bài tập lớn 91

Tài liệu tham khảo 93

Chương 1 LOGIC MỜ VÀ ỨNG DỤNG

Logic mờ (fuzzy logic) là một công cụ dùng để mô hình hóa các quyết định của con người. Ví dụ: người lái xe quan sát chướng ngại vật, đánh giá tình trạng của đường (tốt hay xấu, rộng hay hẹp, thẳng hay cong, tối hay sáng ...) để ra các quyết định điều khiển xe (gas, thắng, tay lái ...). Để có thể tiến hành mô hình hóa, cần có các hiểu biết về cách thức quyết định của con người.

1.1 TẬP HỢP MỜ (fuzzy set)

1.1.1 Tập hợp rõ

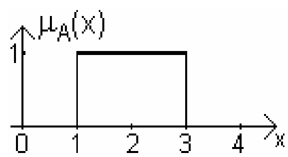
a) Hàm thành viên (membership function)

Hàm thành viên của tập rõ chỉ có thể có một trong hai giá trị

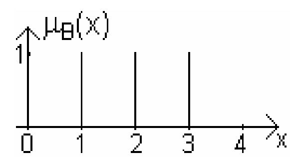
$\mu_A(x) = 1$ có nghĩa $x \in A$.

$\mu_A(x) = 0$ có nghĩa $x \notin A$.

Ví dụ: Tập hợp A = [1, 3] có hàm thành viên $\mu_A(x)$ (hình 1.1). Tập hợp B = {1, 2, 3} có hàm thành viên $\mu_B(x)$ (hình 1.2).



Hình 1.1



Hình 1.2

b) Các phép toán cơ bản

Các phép toán của các tập hợp rõ (hội, giao, hiệu, bù, tích) có thể được định nghĩa thông qua các hàm thành viên như sau

Hội: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} = \min\{1, \mu_A(x) + \mu_B(x)\}$

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

Hiệu: $\mu_{A \setminus B}(x) = \mu_A(x)[1 - \mu_B(x)]$

Bù: $\mu_{\overline{A}}(x) = 1 - \mu_A(x)$

Tích: $\mu_{A \times B}(x,y) = \mu_A(x)\mu_B(y)$

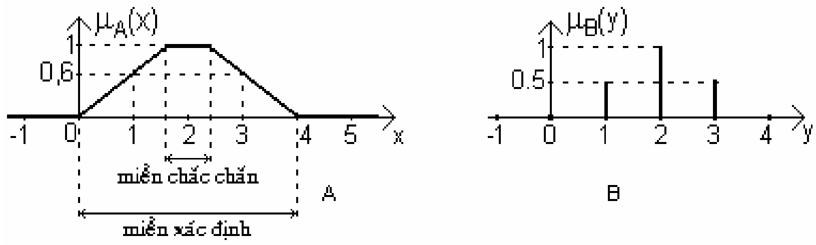
1.1.2 Tập hợp mờ

Tập mờ A trên tập cơ sở X được xác định bởi hàm thành viên $\mu_A(x)$ với $0 \le \mu_A(x) \le 1$.

$\mu_A(x) = 1$ có nghĩa $x \in A$ với độ chắc chắn 100%.

$\mu_A(x) = 0$ có nghĩa $x \notin A$.

Độ cao $H_A = \sup_{x \in X} \{\mu_A(x)\}$



Hình 1.3: Tập mờ xác định trên cơ sở R (hình A) và Z (hình B)

1.1.3 Các phép toán trên tập hợp mờ

a) Hội (union)

Hội của hai tập mờ cùng cơ sở

Xét hai tập mờ A và B với các hàm thành viên $\mu_A(x)$ và $\mu_B(x)$. Hội của A và B là tập mờ, ký hiệu AUB, xác định bởi hàm thành viên

$$\mu_{AUB}(x) = s(\mu_A(x),\mu_B(x))$$

trong đó s(a,b) là ánh xạ $[0,1]x[0,1]\rightarrow [0,1]$ thoả các điều kiện sau

1) $s(1,1) = 1$, $s(0,a) = s(a,0) = a$
2) $s(a,b) = s(b,a)$
3) nếu $a \le a'$ và $b \le b'$ thì $s(a,b) \le s(a',b')$
4) $s(s(a,b),c) = s(a,s(b,c))$

s(a,b) được gọi là s-norm.

Một số s-norm thông dụng

Luật SUM : $\mu_{AUB}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}$

Luật MAX : $\mu_{AUB}(x) = \max\{\mu_A(x), \mu_B(x)\}$

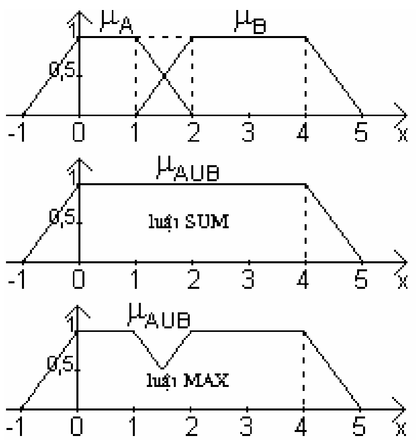
Luật EINSTEIN : $\mu_{AUB}(x) = [\mu_A(x) + \mu_B(x)]/[1+ \mu_A(x)\mu_B(x)]$

Luật khác : $\mu_{AUB}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$

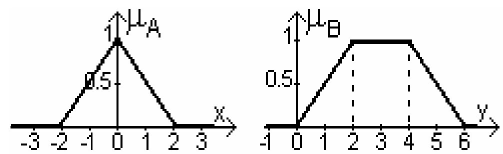
Ví dụ : Hội hai tập mờ A và B dùng luật SUM và luật MAX (hình 1.4 và bảng 1.1)

Bảng 1.1

x	-1	-0,5	0	1	1,5	2	4	4,5	5
$\mu_A(x)$	0	0,5	1	1	0,5	0	0	0	0
$\mu_B(x)$	0	0	0	0	0,5	1	1	0,5	0
$\mu_{AUB}(x)$ (luật SUM)	0	0,5	1	1	1	1	1	0,5	0
$\mu_{AUB}(x)$ (luật MAX)	0	0,5	1	1	0,5	1	1	0,5	0



Hình 1.4 : Hội của hai tập mờ



Hình 1.5

<u>Hội của hai tập mờ khác cơ sở</u>

A: tập mờ trên cơ sở X với hàm thành viên $\mu_A(x)$

B: tập mờ trên cơ sở Y với hàm thành viên $\mu_B(y)$

<u>A</u>: tập mờ mở rộng của A là tập mờ trên cơ sở XxY với hàm thành viên $\mu_{\underline{A}}(x,y) = \mu_A(x), \forall y$

<u>B</u>: tập mờ mở rộng của B là tập mờ trên cơ sở XxY với hàm thành viên $\mu_{\underline{B}}(x,y) = \mu_B(y), \forall x$

Hội của 2 tập mờ khác cơ sở A và B là hội của hai tập mờ mở rộng <u>A</u> và <u>B</u> với hàm thành viên

$$\mu_{AUB}(x,y) = \mu_{\underline{A}U\underline{B}}(x,y)$$

Ví dụ: Cho hai tập mờ A và B (hình 1.5). Các tập mờ mở rộng <u>A</u> và <u>B</u> được xác định bởi các bảng 1.2 và 1.3. Bảng 1.4 cho hội của A và B dùng luật SUM. Bảng 1.5 cho hội của A và B dùng luật MAX.

Bảng 1.2 : $\mu_{\underline{A}}(x,y) = \mu_A(x), \forall y$

		y						
$\mu_{\underline{A}}(x,y)$		0	1	2	3	4	5	6
x	-2	0	0	0	0	0	0	0
	-1	0,5	0,5	0,5	0,5	0,5	0,5	0,5
	0	1	1	1	1	1	1	1
	1	0,5	0,5	0,5	0,5	0,5	0,5	0,5
	2	0	0	0	0	0	0	0

Bảng 1.3 : $\mu_{\underline{B}}(x,y) = \mu_B(y), \forall x$

		y						
$\mu_{\underline{B}}(x,y)$		0	1	2	3	4	5	6
x	-2	0	0,5	1	1	1	0,5	0
	-1	0	0,5	1	1	1	0,5	0
	0	0	0,5	1	1	1	0,5	0
	1	0	0,5	1	1	1	0,5	0
	2	0	0,5	1	1	1	0,5	0

Bảng 1.4 : $\mu_{AUB}(x,y) = \mu_{\underline{A}UB}(x,y)$ dùng luật SUM

		y						
$\mu_{AUB}(x,y)$		0	1	2	3	4	5	6
	-2	0	0,5	1	1	1	0,5	0
	-1	0,5	1	1	1	1	1	0,5
x	0	1	1	1	1	1	1	1
	1	0,5	1	1	1	1	1	0,5
	2	0	0,5	1	1	1	0,5	0

Bảng 1.5 : $\mu_{AUB}(x,y) = \mu_{\underline{AUB}}(x,y)$ dùng luật MAX

		y						
$\mu_{AUB}(x,y)$		0	1	2	3	4	5	6
	-2	0	0,5	1	1	1	0,5	0
	-1	0,5	0,5	1	1	1	0,5	0,5
x	0	1	1	1	1	1	1	1
	1	0,5	0,5	1	1	1	0,5	0,5
	2	0	0,5	1	1	1	0,5	0

b) Giao (intersection)

<u>Giao của hai tập mờ cùng cơ sở</u>

Xét hai tập mờ A và B với các hàm thành viên $\mu_A(x)$ và $\mu_B(x)$. Giao của A và B là tập mờ, ký hiệu $A \cap B$, xác định bởi hàm thành viên

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

Trong đó t(a,b) là ánh xạ $[0,1]x[0,1] \rightarrow [0,1]$ thoả các điều kiện sau

1) $t(0,0) = 0$, $t(1,a) = t(a,1) = a$
2) $t(a,b) = t(b,a)$
3) nếu $a \le a'$ và $b \le b'$ thì $t(a,b) \le t(a',b')$
4) $t(t(a,b),c) = t(a,t(b,c))$

t(a,b) được gọi là t-norm.

Một số t-norm thông dụng

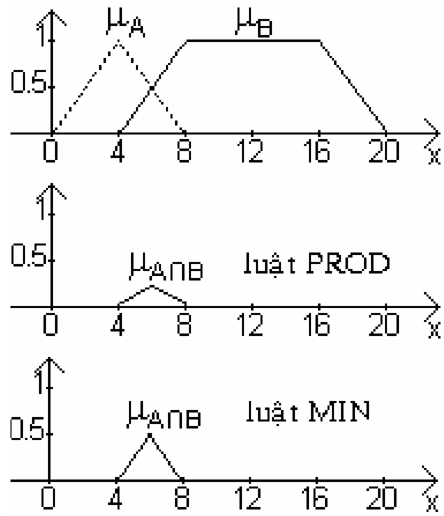
Luật PROD : $\mu_{A\cap B}(x) = \mu_A(x)\mu_B(x)$

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

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

Luật khác : $\mu_{A\cap B}(x) = \mu_A(x)\mu_B(x)/[2 - \mu_A(x) - \mu_B(x) + \mu_A(x)\mu_B(x)]$

Ví dụ: Giao của hai tập mờ A và B dùng luật PROD và luật MIN (hình 1.6 và bảng 1.6)



Hình 1.6: Giao của hai tập mờ

Bảng 1.6

x	0	2	4	5	6	7	8	16	18	20
$\mu_A(x)$	0	0,5	1	0,75	0,5	0,25	0	0	0	0
$\mu_B(x)$	0	0	0	0,25	0,5	0,75	1	1	0,5	0
$\mu_{A\cap B}(x)$ (luật PROD)	0	0	0	0,1875	0,25	0,1875	0	0	0	0
$\mu_{A\cap B}(x)$ (luật MIN)	0	0	0	0,25	0,5	0,25	0	0	0	0

<u>Giao của hai tập mờ khác cơ sở</u>

A : tập mờ trên cơ sở X với hàm thành viên $\mu_A(x)$

B : tập mờ trên cơ sở Y với hàm thành viên $\mu_B(y)$

Giao của 2 tập mờ khác cơ sở A và B là giao của hai tập mờ mở rộng $\underline{A}$ và $\underline{B}$ với hàm thành viên

$$\mu_{A\cap B}(x,y) = \mu_{\underline{A}\cap\underline{B}}(x,y)$$

<u>Ví dụ</u> : Giao hai tập mờ A và B (hình 1.5, bảng 1.2 và 1.3) dùng luật PROD (bảng 1.7) và luật MIN (bảng 1.8)

Bảng 1.7 : $\mu_{A\cap B}(x,y) = \mu_{\underline{A}\cap\underline{B}}(x,y)$ dùng luật PROD

		y						
$\mu_{A\cap B}(x,y)$		0	1	2	3	4	5	6
x	-2	0	0	0	0	0	0	0
	-1	0	0,25	0,5	0,5	0,5	0,25	0
	0	0	0,5	1	1	1	0,5	0
	1	0	0,25	0,5	0,5	0,5	0,25	0
	2	0	0	0	0	0	0	0

Bảng 1.8 : $\mu_{A\cap B}(x,y) = \mu_{\underline{A\cap B}}(x,y)$ dùng luật MIN

	$\mu_{A\cap B}(x,y)$	y						
		0	1	2	3	4	5	6
x	-2	0	0	0	0	0	0	0
	-1	0	0,5	0,5	0,5	0,5	0,5	0
	0	0	0,5	1	1	1	0,5	0
	1	0	0,5	0,5	0,5	0,5	0,5	0
	2	0	0	0	0	0	0	0

c) Bù (complement)

Xét tập mờ A với hàm thành viên $\mu_A(x)$. Bù của A là tập mờ, ký hiệu $\overline{A}$, xác định bởi hàm thành viên

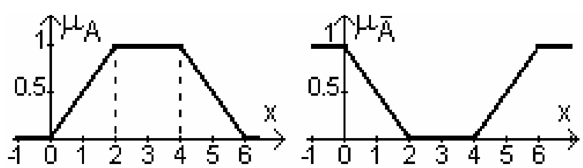
$$\mu_{\overline{A}}(x) = c(\mu_A(x))$$

Trong đó c(a) là ánh xạ $[0,1] \rightarrow [0,1]$ thoả các điều kiện sau

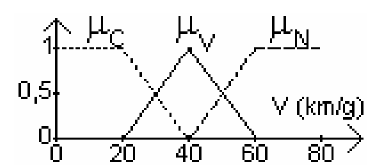
1) $c(0) = 1$, $c(1) = 0$
2) nếu $a < b$ thì $c(a) \geq c(b)$

Định nghĩa thông dụng

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$$



Hình 1.7 : bù của tập mờ



Hình 1.8 : các tập mờ của tốc độ xe

1.1.4 Biến ngôn ngữ

a) Biến ngôn ngữ

Ví dụ : tốc độ xe V được mô tả bởi 3 tập mờ (hình 1.8)

- Chậm với hàm thành viên $\mu_C(v)$
- Vừa với hàm thành viên $\mu_V(v)$
- Nhanh với hàm thành viên $\mu_N(v)$

b) Gia tử (hedge)

Các trạng từ nhấn mạnh : rất, hơi, có vẽ, . . ., dùng để định nghĩa các tập mờ mới từ các tập mờ có sẵn.

<u>rất</u> $\mu_{\text{rất nhanh}}(v) = [\mu_{\text{nhanh}}(v)]^2$

<u>rất rất</u> $\mu_{\text{rất rất nhanh}}(v) = [\mu_{\text{nhanh}}(v)]^n$ với $n > 2$

<u>thực sự là</u> $\mu_{\text{thực sự nhanh}}(v) = \begin{cases} 2[\mu_{\text{nhanh}}(v)]^2 & \text{nếu } 0 \le \mu_{\text{nhanh}}(v) \le 0{,}5 \\ 1 - 2[1 - \mu_{\text{nhanh}}(v)]^2 & \text{nếu } 0{,}5 \le \mu_{\text{nhanh}}(v) \le 1 \end{cases}$

<u>hơi</u> $\mu_{\text{hơi nhanh}}(v) = \sqrt{\mu_{\text{nhanh}}(v)}$

Ví dụ: tập mờ 'nóng' và các tập mờ 'rất nóng', 'rất rất nóng', 'thực sự nóng', và 'hơi nóng' với các hàm thành viên cho ở bảng 1.9

Bảng 1.9

nhiệt độ T	0^o	10^o	20^o	30^o	40^o	50^o	60^o	70^o	80^o	90^o
$\mu_{\text{nóng}}(T)$	0	0	0,1	0,2	0,3	0,4	0,5	0,6	0,8	1
$\mu_{\text{rất nóng}}(T)$	0	0	0,01	0,04	0,09	0,16	0,25	0,36	0,64	1
$\mu_{\text{rất rất nóng}}(T)$ (n = 3)	0	0	0,001	0,008	0,027	0,064	0,125	0,216	0, 512	1
$\mu_{\text{thực sự nóng}}(T)$	0	0	0,02	0,08	0,18	0,32	0,5	0,68	0,92	1
$\mu_{\text{hơi nóng}}(T)$	0	0	0,316	0,447	0,548	0,632	0,707	0,775	0,894	1

1.2 QUAN HỆ MỜ

1.2.1 Quan hệ rõ

Cho 2 tập rõ X và Y. Quan hệ giữa các tập hợp X và Y là tập hợp con Q của tập hợp tích XxY : Q⊂ XxY.

Ví dụ: X = {1, 2, 3} , Y = {2, 4}. Tập hợp tích

XxY = {(1,2), (1,4), (2,2), (2,4), (3,2), (3,4)}

Quan hệ Q(X,Y) : 'phần tử của X < phần tử của Y' được xác định bởi

Q = {(1,2), (1,4), (2,4), (3,4)}

Hàm thành viên của tập Q được cho ở bảng 1.10

Bảng 1.10 : $\mu_Q(x,y)$

		y	
$\mu_Q(x,y)$		2	4
x	1	1	1
	2	0	1
	3	0	1

Ví dụ: X = { Hải Phòng, Biên Hòa, Vũng Tàu }, Y = {Hà Nội, Cần Thơ}.

Tập hợp tích

XxY = {(Hải Phòng, Hà Nội), (Hải Phòng, Cần Thơ), (Biên Hòa, Hà Nội),
(Biên Hòa, Cần Thơ), (Vũng Tàu, Hà Nội), (Vũng Tàu, Cần Thơ)}

Quan hệ Q(X,Y) : 'thành phố x xa thành phố y' được xác định bởi

Q = {(Hải Phòng, Cần Thơ), (Biên Hòa, Hà Nội), (Vũng Tàu, Hà Nội)}

Hàm thành viên của tập Q được cho ở bảng 1.11

Bảng 1.11 : $\mu_Q(x,y)$

	$\mu_Q(x,y)$	y: Hà Nội	y: Cần Thơ
x	Hải Phòng	0	1
x	Biên Hòa	1	0
x	Vũng Tàu	1	0

1.2.2 Quan hệ mờ

Ví dụ: X = { Hải Phòng, Biên Hòa, Vũng Tàu }, Y = {Hà Nội, Cần Thơ}.

Quan hệ mờ Q(X,Y) : 'thành phố x xa thành phố y' được xác định bởi hàm thành viên $\mu_Q(x,y)$ cho ở bảng 1.12

Bảng 1.12 : $\mu_Q(x,y)$

	$\mu_Q(x,y)$	y: Hà Nội	y: Cần Thơ
x	Hải Phòng	0.1	1.0
x	Biên Hòa	0.9	0.2
x	Vũng Tàu	0.8	0.3

Ví dụ: X và Y là các tập hợp số (tập rõ).

Quan hệ x xấp xỉ y có thể được đặc trưng bởi hàm thành viên $\mu_{XX}(x,y) = e^{-(x-y)^2}$.

Quan hệ x rất lớn hơn y có thể được đặc trưng bởi hàm thành viên $\mu_{ML}(x,y) = \dfrac{1}{1+e^{-(x-y)}}$

1.2.3 Hình chiếu

Cho hai tập hợp rõ X, Y và quan hệ Q (tập mờ trên cơ sở XxY) xác định bởi hàm thành viên $\mu_Q(x,y)$. Hình chiếu của Q trên X là tập mờ Q_x trên cơ sở X xác định bởi hàm thành viên

$$\mu_{Qx}(x) = \max_{y\in Y} \mu_Q(x,y)$$

Hình chiếu của Q trên Y là tập mờ Q_y trên cơ sở Y xác định bởi hàm thành viên

$$\mu_{Qy}(y) = \max_{x\in X} \mu_Q(x,y)$$

1.2.4 Kết hợp các quan hệ rõ

P : quan hệ rõ trên XxY

Q : quan hệ rõ trên YxZ

Quan hệ kết hợp PoQ là tập hợp con của XxZ sao cho

$$(x,z) \in PoQ \quad \Leftrightarrow \quad \exists y \in Y \text{ sao cho } (x,y) \in P \text{ và } (y,z) \in Q$$

Hàm thành viên của PoQ được xác định bởi

$$\mu_{PoQ}(x,z) = \max_{y \in Y} (\mu_P(x,y)\mu_Q(y,z)) = \max_{y \in Y} \min(\mu_P(x,y),\mu_Q(y,z))$$

1.2.5 Kết hợp các quan hệ mờ

P : quan hệ mờ trên cơ sở XxY

Q : quan hệ mờ trên cơ sở YxZ

Quan hệ mờ kết hợp PoQ (trên cơ sở XxZ) được xác định bởi hàm thành viên

$$\mu_{PoQ}(x,z) = \max_{y \in Y} t(\mu_P(x,y),\mu_Q(y,z))$$

với t() là t chuẩn. Sử dụng t chuẩn là luật MIN, ta có phép kết hợp MAX-MIN

$$\mu_{PoQ}(x,z) = \max_{y \in Y} \min(\mu_P(x,y),\mu_Q(y,z))$$

Sử dụng t chuẩn là luật PROD, ta có phép kết hợp MAX-PROD

$$\mu_{PoQ}(x,z) = \max_{y \in Y} (\mu_P(x,y)\mu_Q(y,z))$$

1.3 SUY DIỄN MỜ

1.3.1 Mệnh đề mờ

<u>Mệnh đề đơn</u>

Mệnh đề **X is A** được đặc trưng bởi tập mờ A với hàm thành viên $\mu_A(x)$.

<u>Mệnh đề kép</u>

Mệnh đề **X is A and Y is B** được đặc trưng bởi tập mờ $A \cap B$ với hàm thành viên $\mu_{A \cap B}(x,y)$.

Mệnh đề **X is A or Y is B** được đặc trưng bởi tập mờ $A \cup B$ với hàm thành viên $\mu_{A \cup B}(x,y)$.

Mệnh đề **X is not A** được đặc trưng bởi tập mờ $\overline{A}$ với hàm thành viên $\mu_{\overline{A}}(x)$.

Mệnh đề **(X is not A and Y is B) or z is C** được đặc trưng bởi tập mờ $(\overline{A} \cap B) \cup C$ với hàm thành viên $\mu_{(\overline{A} \cap B) \cup C}(x,y,z)$.

1.3.2 Diễn dịch luật IF … THEN …

Cho các mệnh đề p và q. Phép diễn dịch truyền thống

$$\text{IF } p \text{ THEN } q \tag{1.3.1}$$

$(p \Rightarrow q)$ có bảng chân trị ở bảng 1.13. Ta thấy $p \Rightarrow q$ tương đương với

$\overline{p}$ or q (1.3.2)

hoặc (p and q) or $\overline{p}$ (1.3.3)

Bảng 1.13 : bảng chân trị của phép suy diễn $p \Rightarrow q$

(T: true, F: false)

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Có nhiều phương pháp diễn dịch mờ luật $p \Rightarrow q$.

a) Các phương pháp dựa vào diễn dịch cổ điển (non local)

Các phương pháp nầy được gọi là toàn cục theo nghĩa

$p \Rightarrow q$ hàm nghĩa not $p \Rightarrow$ not q

Phương pháp Dienes-Rescher : dựa vào (1.3.2) với luật MAX

$$\mu_{p\Rightarrow q}(x,y) = \max\{1-\mu_p(x), \mu_q(y)\} \quad (1.3.4)$$

Phương pháp Lukasiewicz : dựa vào (1.3.2) với luật SUM

$$\mu_{p\Rightarrow q}(x,y) = \min\{1,1-\mu_p(x)+\mu_q(y)\} \quad (1.3.5)$$

Phương pháp Zadeh : dựa vào (1.3.3) với luật MAX (hội) và MIN (giao)

$$\mu_{p\Rightarrow q}(x,y) = \max\{\min\{\mu_p(x),\mu_q(y)\},1-\{\mu_p(x)\} \quad (1.3.6)$$

b) Các phương pháp của Mamdani (thông dụng nhất trong điều khiển mờ)

Các phương pháp nầy được gọi là cục bộ (local) theo nghĩa $p \Rightarrow q$ không hàm nghĩa not $p \Rightarrow$ not q. Mamdani diễn dịch luật IF … THEN … như là giao của 2 tập mờ

$$\mu_{p\Rightarrow q}(x,y) = \mu_{p\cap q}(x,y) \quad (1.3.7)$$

Luật MIN

$$\mu_{p\Rightarrow q}(x,y) = \min\{\mu_p(x), \mu_q(y)\} \quad (1.3.8)$$

Luật PROD

$$\mu_{p\Rightarrow q}(x,y) = \mu_p(x)\mu_q(y) \quad (1.3.9)$$

1.3.3 Logic cổ điển

a) Bảng chân trị của các phép logic cơ bản

Bảng 1.14 : bảng chân trị của các phép logic cơ bản

p	q	p Λ q	p V q	$p \rightarrow q$	$p \leftrightarrow q$	$\overline{p}$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

b) Modus ponens : $(p \land (p \rightarrow q)) \rightarrow q$ (1.3.10)

GT1 : X is A

GT2 : IF X is A THEN Y is B

KL : Y is B

c) Modus tollens : $(\overline{q} \land (p \rightarrow q)) \rightarrow \overline{p}$ (1.3.11)

GT1 : Y is not B

GT2 : IF X is A THEN Y is B

KL : X is not A

d) Tam đoạn luận (Hypothetical Syllogism)**:** $((p \rightarrow q) \land (q \rightarrow r)) \rightarrow (p \rightarrow r)$ (1.3.12)

GT1 : IF X is A THEN Y is B

GT2 : IF Y is B THEN Z is C

KL : IF X is A THEN Z is C

1.3.4 Logic mờ

a) Quy tắc kết hợp suy diễn

Cho

A' : tập mờ trên cơ sở X với hàm thành viên $\mu_{A'}(x)$.

Q : quan hệ mờ trên cơ sở XxY với hàm thành viên $\mu_Q(x,y)$.

Mục tiêu : xác định tập mờ B' (hình 1.7).

Gọi $\underline{A}$' là tập mờ mở rộng của A trên cơ sở XxY

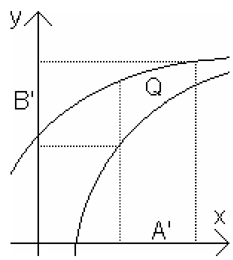
$$\mu_{\underline{A}'}(x,y) = \mu_{A'}(x),\ \forall y$$

Giao giữa $\underline{A}$' và Q

$$\mu_{\underline{A}'\cap Q}(x,y) = t\{\mu_{\underline{A}'}(x,y),\ \mu_Q(x,y)\}$$

Chiếu tập mờ $\underline{A}'\cap Q$ lên tập Y ta được tập mờ B'

$$\mu_{B'}(y) = \sup_x t\{\mu_{\underline{A}'}(x,y),\ \mu_Q(x,y)\}$$



Hình 1.7

b) Modus ponens tổng quát

GT1 : X is A'

GT2 : IF X is A THEN Y is B

KL : Y is B'

A' càng gần A thì B' càng gần B

$$\mu_{B'}(y) = \sup_{x\in X} t\{\mu_{A'}(x),\ \mu_{A\to B}(x,y)\} \tag{1.3.13}$$

c) Modus tollens tổng quát

GT1 : Y is B'

GT2 : IF X is A THEN Y is B

KL : X is A'

B' càng khác B thì A' càng khác A

$$\mu_{A'}(x) = \sup_{y\in Y} t\{\mu_{B'}(y),\ \mu_{A\to B}(x,y)\} \tag{1.3.14}$$

d) Tam đoạn luận tổng quát (hypothetical syllogism)

GT1 : IF X is A THEN Y is B

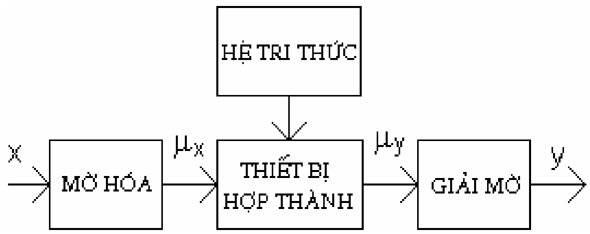
GT2 : IF Y is B' THEN Z is C

KL : IF X is A THEN Z is C'

B càng gần B' thì C' càng gần C

$$\mu_{A\to C'}(x,z) = \sup_{y\in Y} t\{\mu_{A\to B}(x,y),\ \mu_{B'\to C}(y,z)\} \tag{1.3.15}$$

1.4 HỆ THỐNG XỬ LÝ MỜ



Hình 1.9: Sơ đồ khối của hệ thống xử lý mờ

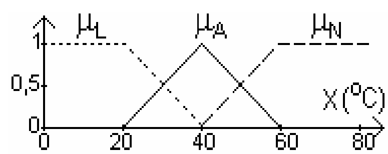
x : tín hiệu vào (rõ), y : tín hiệu ra (rõ)

1.4.1 Khối mờ hóa (fuzzifier)

Ví dụ : Tín hiệu vào là nhiệt độ x với các tập mờ lạnh (L), ấm (A) và nóng (N) định nghĩa ở hình 1.10. Tín hiệu ra của khối mờ hóa là vectơ

$$\mu_X(x) = [\mu_L(x), \mu_A(x), \mu_N(x)]^T$$

Khi x = 35, ta có $\mu_X(35) = [0.25, 0.75, 0]^T$



Hình 1.10

1.4.2 Suy diễn mờ (fuzzy inference engine)

a) Mệnh đề hợp thành (phép suy diễn, implication)

IF A THEN B

đặc trưng bởi tập mờ với hàm thành viên $\mu_{A\Rightarrow B}(x,y)$. trong đó A là mệnh đề điều kiện và B là mệnh đề kết luận

b) Hàm thành viên của mệnh đề hợp thành (theo Mamdani)

Xét A là tập mờ trên cơ sở X với hàm thành viên $\mu_A(x)$, và B là tập mờ trên cơ sở Y với hàm thành viên $\mu_B(y)$. Mệnh đề hợp thành

IF A THEN B

có hàm thành viên xác định bởi

$$\mu_{A\Rightarrow B}(x,y) = \mu_{A\cap B}(x,y)$$

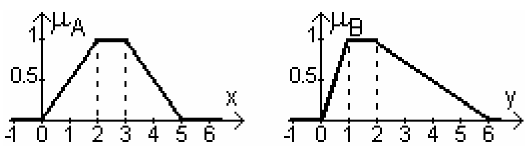
Diễn dịch phép suy diễn dùng luật MIN ta có

$$\mu_{A\Rightarrow B}(x,y) = \min\{\mu_A(x), \mu_B(y)\}$$

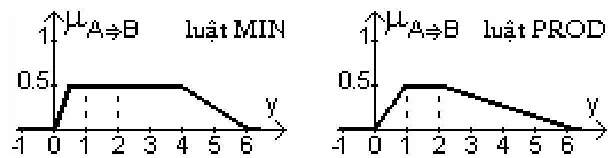
Diễn dịch phép suy diễn dùng luật PROD ta có

$$\mu_{A\Rightarrow B}(x,y) = \mu_A(x)\mu_B(y)$$

Ví dụ : Cho 2 tập mờ A và B ở hình 1.11. Mệnh đề hợp thành IF A THEN B có hàm thành viên cho ở bảng 1.10 (luật MIN) và 1.11 (luật PROD).



Hình 1.11: Tập mờ A và B



Hình 1.12 : Tập mờ $\mu_{A\Rightarrow B}(y)$ khi x = 1

Bảng 1.10 : $\mu_{A\Rightarrow B}(x,y)$ dùng luật MIN

$\mu_{A\Rightarrow B}(x,y)$		y						
		0	1	2	3	4	5	6
x	0	0	0	0	0	0	0	0
	1	0	0,5	0,5	0,5	0,5	0,25	0
	2	0	1	1	0,75	0,5	0,25	0
	3	0	1	1	0,75	0,5	0,25	0
	4	0	0,5	0,5	0,5	0,5	0,25	0
	5	0	0	0	0	0	0	0

Bảng 1.11 : $\mu_{A\Rightarrow B}(x,y)$ dùng luật PROD

$\mu_{A\Rightarrow B}(x,y)$		y						
		0	1	2	3	4	5	6
x	0	0	0	0	0	0	0	0
	1	0	0,5	0,5	0,375	0,25	0,125	0
	2	0	1	1	0,75	0,5	0,25	0
	3	0	1	1	0,75	0,5	0,25	0
	4	0	0,5	0,5	0,375	0,25	0,125	0
	5	0	0	0	0	0	0	0

c) Hàm thành viên của mệnh đề hợp thành nhiều điều kiện

Xét A là tập mờ trên cơ sở X với hàm thành viên $\mu_A(x)$, B là tập mờ trên cơ sở Y với hàm thành viên $\mu_B(y)$ và C là tập mờ trên cơ sở Z với hàm thành viên $\mu_C(z)$. Mệnh đề hợp thành

IF A AND B THEN C

(ký hiệu A Λ B ⇒ C, với A Λ B là mệnh đề điều kiện, C là mệnh đề kết luận) có hàm thành viên xác định bởi

$$\mu_{A\wedge B\Rightarrow C}(x,y,z) = \mu_{(A\cap B)\cap C}(x,y,z)$$

Dùng luật MIN ta có

$$\mu_{A\wedge B\Rightarrow C}(x,y,z) = \min\{h, \mu_C(z)\}$$

Dùng luật PROD ta có

$$\mu_{A\wedge B\Rightarrow C}(x,y,z) = h\mu_C(z)$$

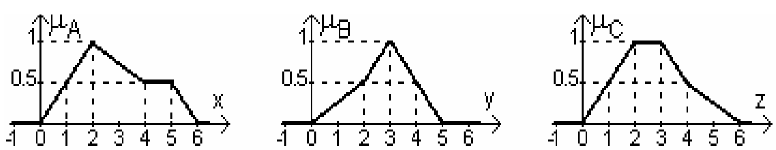
$$\text{với } h = \begin{cases} \min\{\mu_A(x), \mu_B(y)\}\ , & \text{luật MIN} \\ \mu_A(x)\mu_B(y)\ , & \text{luật PROD} \end{cases}$$

Ví dụ : Cho 3 tập mờ A, B và C ở hình 1.13. Mệnh đề hợp thành IF A AND B THEN C có hàm thành viên cho ở bảng 1.12 dùng luật MIN-MIN và luật PROD-PROD.

Bảng 1.12 : Hàm thành viên của mệnh đề hợp thành

IF A AND B THEN C với x = 3 và y = 2

z	0	1	2	3	4	5	6
$\mu_{A\wedge B\Rightarrow C}(z)$ luật MIN MIN	0	0,5	0,5	0,5	0,5	0,25	0
$\mu_{A\wedge B\Rightarrow C}(z)$ luật PROD PROD	0	0,1875	0,375	0,375	0,1875	0,09375	0



Hình 1.13

d) Kết hợp các mệnh đề hợp thành (aggregation)

Kết quả của phép kết hợp các mệnh đề hợp thành

R_1 : IF A_1 THEN B_1

R_2 : IF A_2 THEN B_2

:

R_n : IF A_n THEN B_n

là tập mờ xác định bởi hội của các tập mờ của các mệnh đề hợp thành $\mu_R = \mu_{R1\cup R2\cup\ldots\cup Rn}$.

Phụ thuộc vào cách thức diễn dịch phép suy diễn (implication) IF… THEN và phép kết hợp (aggregation) các mệnh đề hợp thành, ta có các phương pháp thông dụng sau

MAX-MIN : kết hợp dùng luật MAX và suy diễn dùng luật MIN

MAX-PROD : kết hợp dùng luật MAX và suy diễn dùng luật PROD

SUM-MIN : kết hợp dùng luật SUM và suy diễn dùng luật MIN

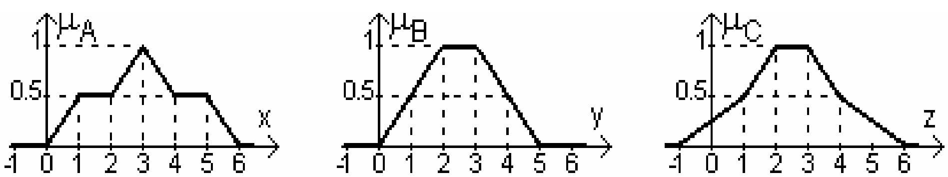
SUM-PROD : kết hợp dùng luật SUM và suy diễn dùng luật PROD

Ví dụ : Cho 3 tập mờ A, B và C ở hình 1.14. Hàm thành viên của các mệnh đề hợp thành

IF A THEN C

IF B THEN C

được cho ở bảng 1.13



Hình 1.14

Bảng 1.13 : Hàm thành viên $\mu_{(A \Rightarrow C) \cup (B \Rightarrow C)}(z)$của các mệnh đề hợp thành

IF A THEN C và IF B THEN C với x = 2 và y = 4

z	-1	0	1	2	3	4	5	6
MAX-MIN	0	0,25	0,5	0,5	0,5	0,5	0,25	0
SUM-MIN	0	0,5	1	1	1	1	0,5	0
MAX-PROD	0	0,125	0,25	0,5	0,5	0,25	0,125	0
SUM-PROD	0	0,25	0,5	1	1	0,5	0,25	0

1.4.3 Khối giải mờ (defuzzifier)

a) Phương pháp cực đại

$\bar{x} = \max_x \{\mu_A(x)\}$

Trường hợp có nhiều giá trị của x làm $\mu_A(x)$ đạt cực đại, người ta có thể lấy cận trái, cận phải hoặc điểm trung bình

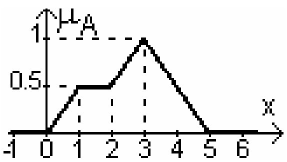
Ví dụ:

Ở hình 1.15, $\bar{x} = 3$

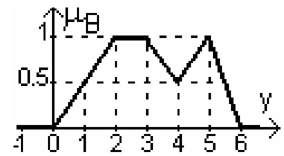
Ở hình 1.16, $\bar{y} = 2$ (cận trái)

$\bar{y} = 5$ (cận phải)

$\bar{y} = 3{,}5$ (trung bình)



Hình 1.15



Hình 1.16

b) Phương pháp điểm trọng tâm (center of gravity, centroid)

Giá trị giải mờ là hoành độ điểm trọng tâm của đồ thị μ(x)

$$\bar{x} = \frac{\int x\mu(x)dx}{\int \mu(x)dx}$$

c) Phương pháp trung bình (center of average)

Để đơn giản hóa tính toán, ta có thể rời rạc hoá μ(x). Phương pháp điểm trọng tâm trở thành phương pháp trung bình

$$\bar{x} = \frac{\sum_k x_k \mu(x_k)}{\sum_k \mu(x_k)}$$

1.4.4 Tính phi tuyến của hệ thống xử lý mờ

Ví dụ : Các biến x và y được mờ hóa như hình 1.17 ($-2 \le y \le 2$). Các luật hợp thành được xác định bởi

IF X IS S THEN Y IS P

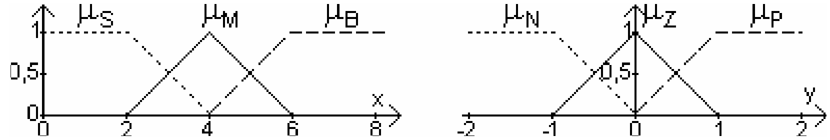
IF X IS M THEN Y IS Z

IF X IS B THEN Y IS N

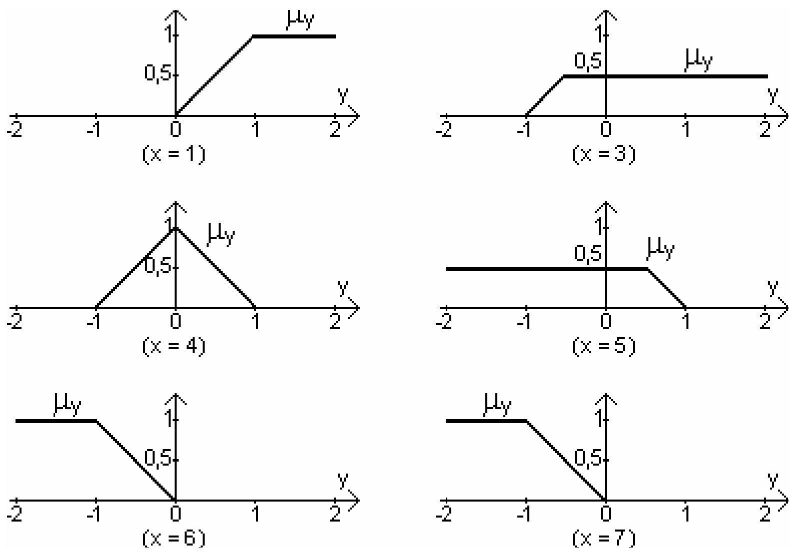
Suy diễn mờ theo phương pháp MAX-MIN. Giải mờ dùng phương pháp điểm trọng tâm (rời rạc hóa tại các giá trị nguyên của y). Ta được kết quả ở hình 1.18 và bảng 1.14.

Bảng 1.14 : Quan hệ phi tuyến giữa $\bar{y}$ và x

x	0	1	2	3	4	5	6	7	8
$\bar{y}$	1,5	1,5	1,5	1,0	0,0	-1,0	-1,5	-1,5	-1,5



Hình 1.17



Hình 1.18

1.4.5 Hệ thống xử lý mờ loại 2

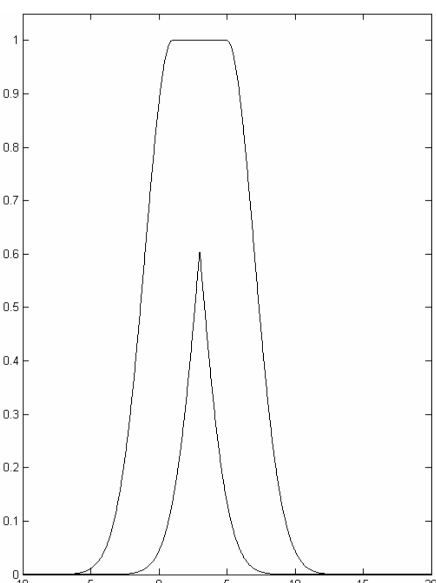
1) Tập mờ loại 2

Tập mờ loại 2 được đặc trưng bởi hàm thành viên với giá trị mờ. Ví dụ hình 1.19 trình bày tập mờ loại 2 với hàm thành viên có giá trị nằm trong khoảng xác định bởi

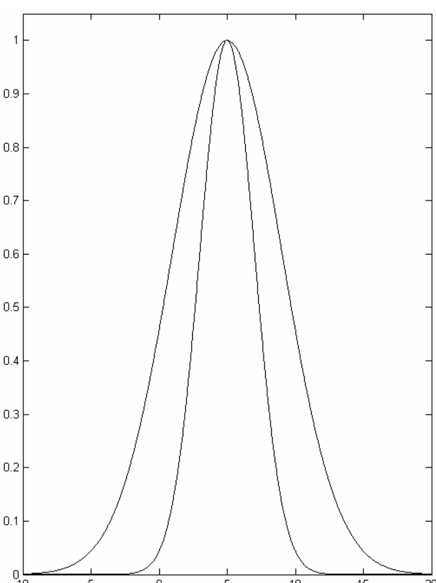
$$\mu_1(x) = \exp\left\{-\frac{(x-1)^2}{8}\right\} \quad \text{và} \quad \mu_2(x) = \exp\left\{-\frac{(x-5)^2}{8}\right\} \tag{1.4.1}$$

Hình 1.20 trình bày tập mờ loại 2 với hàm thành viên có giá trị nằm trong khoảng xác định bởi

$$\mu_1(x) = \exp\left\{-\frac{(x-5)^2}{8}\right\} \quad \text{và} \quad \mu_2(x) = \exp\left\{-\frac{(x-5)^2}{32}\right\} \tag{1.4.2}$$



Hình 1.19



Hình 1.20

2) Mệnh đề hợp thành

Mệnh đề hợp thành

(X is A) and (Y is B)

được đặc trưng bởi 1 tập mờ loại 2 với hàm thành viên xác định bởi (AND dùng luật PROD)

$$\left[\underline{\mu}_A(x)\,\underline{\mu}_B(y),\quad \overline{\mu}_A(x)\,\overline{\mu}_B(y)\right] \tag{1.4.3}$$

3) Giải mờ

Trước khi giải mờ, các tập mờ loại 2 được chuyển sang loại 1. Các giá trị y_r và y_l được xác định dùng phương pháp tâm của tập hợp (center of set) như sau :

$$y_r = \frac{\sum_{n=1}^{R} \underline{\mu}^n w_r^n + \sum_{h=R+1}^{N} \overline{\mu}^h w_r^h}{\sum_{n=1}^{R} \underline{\mu}^n + \sum_{h=R+1}^{N} \overline{\mu}^h} \tag{1.4.4}$$

$$y_l = \frac{\sum_{n=1}^{L} \overline{\mu}^n w_l^n + \sum_{h=L+1}^{N} \underline{\mu}^h w_l^h}{\sum_{n=1}^{L} \overline{\mu}^n + \sum_{h=L+1}^{N} \underline{\mu}^h} \tag{1.4.5}$$

với $1 \le R \le N\text{-}1$ và $1 \le L \le N\text{-}1$.

Định nghĩa

$$\phi_r^T = \frac{1}{\sum_{n=1}^{R} \underline{\mu}^n + \sum_{h=R+1}^{N} \overline{\mu}^h}\left[\underline{\mu}^1\underline{\mu}^2\underline{\mu}^3..\underline{\mu}^R\,\overline{\mu}^{R+1}\,\overline{\mu}^{R+2}...\overline{\mu}^N\right]$$

$$\phi_l^T = \frac{1}{\sum_{n=1}^{L} \overline{\mu}^n + \sum_{h=L+1}^{N} \underline{\mu}^h}\left[\overline{\mu}^1\,\overline{\mu}^2...\overline{\mu}^L\,\underline{\mu}^{L+1}\,\underline{\mu}^{L+2}...\underline{\mu}^N\right]$$

$$w_r = \left[w_r^1 w_r^2 w_r^3..w_r^N\right]^T \qquad w_l = \left[w_l^1 w_l^2 w_l^3..w_l^N\right]^T$$

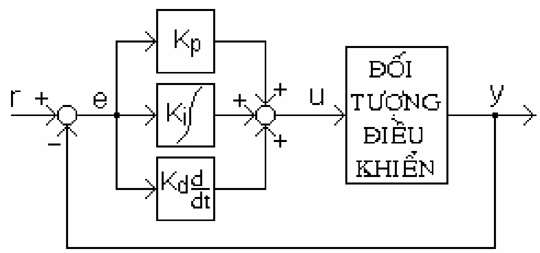
$$\Rightarrow \quad y_l(k) = \phi_l^T(k) w_l(k) \qquad y_r(k) = \phi_r^T(k) w_r(k)$$

Tín hiệu ra (giải mờ) được xác định như sau

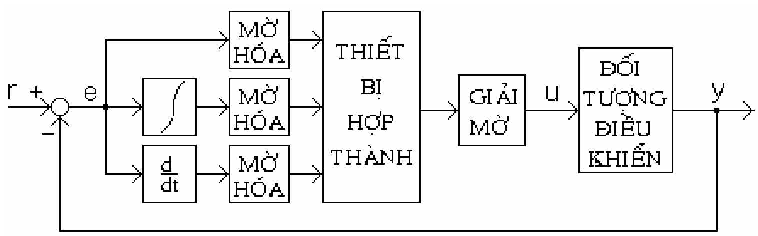
$$y(k) = \frac{1}{2}[y_l(k) + y_r(k)] \tag{1.4.6}$$

1.5 ĐIỀU KHIỂN MỜ

Phương pháp Mamdani. Phương pháp Sugeno.



Hình 1.28 : Điều khiển PID truyền thống



Hình 1.29 : Điều khiển PID mờ

1.6 NHẬN DẠNG HÀM THÀNH VIÊN VÀ LUẬT HỢP THÀNH

Các tập mờ của tín hiệu vào x được chọn bao phủ toàn miền xác định của x

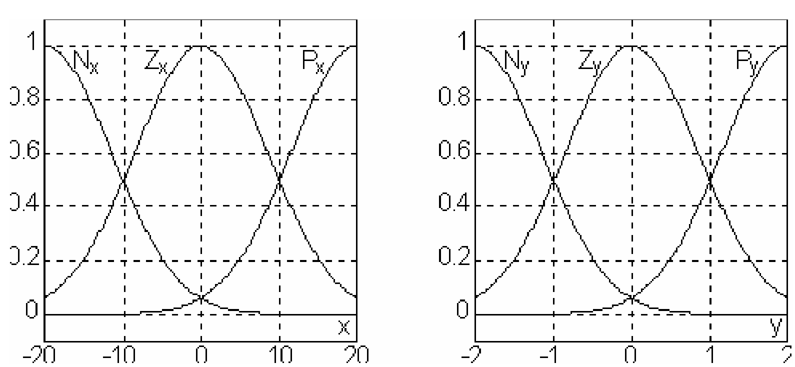
$$\forall x \in [x_{min}, x_{max}], \exists Z : \mu_Z(x) \neq 0.$$

Hình 1.19

1.6.1 Nhận dạng luật hợp thành

Mục tiêu : Cho trước các tập mờ của tín hiệu vào x và ra y xác định các luật hợp thành từ các dữ liệu đo được x(k) và y(k), k = 1, 2, … N.

Ví dụ : Bảng 1.15 cho các dữ liệu đo. Các tập mờ của các tín hiệu x và y được định nghĩa ở hình 1.20.



Hình 1.20

Bảng 1.15

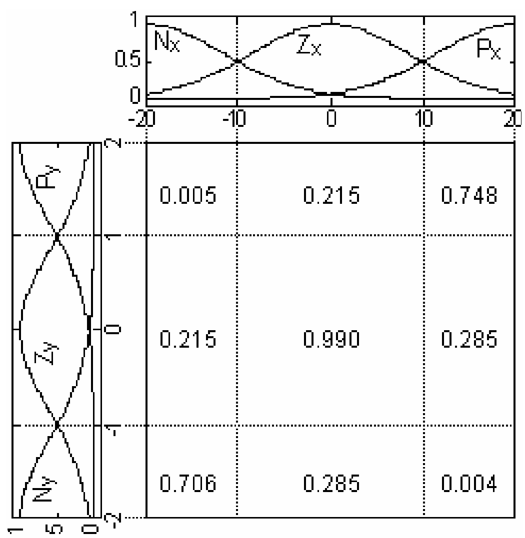
x	-15	-10	-5	0	5	10	15
y	-1.5	-1.1	-0.5	0.1	0.4	0.9	1.6

Từ các tập mờ của x và y, ta xác định được giá trị của các hàm thành viên của dữ liệu ở bảng 1.16.

Bảng 1.16

x	-15	-10	-5	0	5	10	15
$\begin{bmatrix}\mu_{Nx}(x)\\ \mu_{Zx}(x)\\ \mu_{Px}(x)\end{bmatrix}$	$\begin{bmatrix}0.84\\0.21\\0.00\end{bmatrix}$	$\begin{bmatrix}0.50\\0.50\\0.00\end{bmatrix}$	$\begin{bmatrix}0.21\\0.84\\0.01\end{bmatrix}$	$\begin{bmatrix}0.06\\1.00\\0.06\end{bmatrix}$	$\begin{bmatrix}0.01\\0.84\\0.21\end{bmatrix}$	$\begin{bmatrix}0.00\\0.50\\0.50\end{bmatrix}$	$\begin{bmatrix}0.00\\0.21\\0.84\end{bmatrix}$
y	-1.5	-1.1	-0.5	0.1	0.4	0.9	1.6
$\begin{bmatrix}\mu_{Ny}(y)\\ \mu_{Zy}(y)\\ \mu_{Py}(y)\end{bmatrix}$	$\begin{bmatrix}0.84\\0.21\\0.00\end{bmatrix}$	$\begin{bmatrix}0.57\\0.43\\0.00\end{bmatrix}$	$\begin{bmatrix}0.21\\0.84\\0.01\end{bmatrix}$	$\begin{bmatrix}0.05\\0.99\\0.08\end{bmatrix}$	$\begin{bmatrix}0.02\\0.89\\0.17\end{bmatrix}$	$\begin{bmatrix}0.00\\0.57\\0.43\end{bmatrix}$	$\begin{bmatrix}0.00\\0.17\\0.89\end{bmatrix}$

Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội. Kết quả được cho ở hình 1.21. Giá trị trong mỗi ô là giá trị lớn nhất (MAX) của tích (PROD) của các hàm thành viên tương ứng của x và y. Ví dụ ô tương ứng với N_X và N_Y chứa giá trị $\max\{\mu_{Nx}(x)\mu_{Ny}(y)\} = 0.706$.



Hình 1.21 : Giá trị lớn nhất của tích của các hàm thành viên

Luật hợp thành được xác định tương ứng với ô có giá trị lớn nhất của mỗi cột ở hình 1.21. Giá trị trên mỗi ô có thể được diễn dịch là độ tin cậy của luật hợp thành. Ta có các luật hợp thành sau

IF X IS N_X THEN Y IS N_Y

IF X IS Z_X THEN Y IS Z_Y

IF X IS P_X THEN Y IS P_Y

với độ tin cậy lần lượt là 0.706, 0.99 và 0.748. Việc chọn các tập mờ ảnh hưởng trực tiếp đến giá trị của độ tin cậy. Nếu độ tin cậy có giá trị bé, ta cần phải chọn lại các tập mờ.

1.6.2 Nhận dạng hàm thành viên

Mục tiêu: Cho trước các tập mờ của tín hiệu vào x và các luật hợp thành, xác định các tập mờ của tín hiệu ra y từ các dữ liệu đo được x(k) và y(k), k = 1, 2, … N.

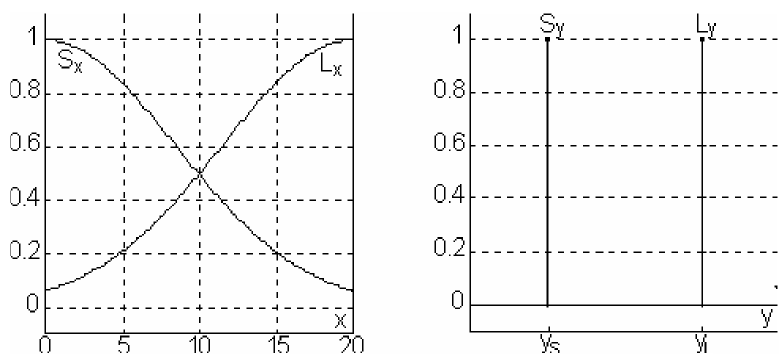
Giới hạn: các tập mờ của tín hiệu ra y có dạng đơn trị (singleton).

Ví dụ: Các dữ liệu đo được cho ở bảng 1.17. Các tập mờ của các tín hiệu x và y được định nghĩa ở hình 1.22. Các luật hợp thành được cho bởi

IF X IS S_X THEN Y IS L_Y

IF X IS L_X THEN Y IS S_Y

Bảng 1.18 xác định giá trị của các hàm thành viên $\mu_{Sx}(x)$ và $\mu_{Lx}(x)$ từ các tập mờ của x. Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội, và giải mờ dùng phương pháp trung bình, ta xác định được giá trị giải mờ $\overline{y}$ ở bảng 1.18.



Hình 1.22 : Các tập mờ của x và y

Bảng 1.17

x	0	5	15	20
y	4	3	2	1

Bảng 1.18

x	0	5	15	20
$\begin{bmatrix}\mu_{Sx}(x)\\ \mu_{Lx}(x)\end{bmatrix}$	$\begin{bmatrix}1.00\\ 0.06\end{bmatrix}$	$\begin{bmatrix}0.84\\ 0.21\end{bmatrix}$	$\begin{bmatrix}0.21\\ 0.84\end{bmatrix}$	$\begin{bmatrix}0.06\\ 1.00\end{bmatrix}$
$\overline{y}$	$\frac{y_L + 0.06y_S}{1.06}$	$\frac{0.84y_L + 0.21y_S}{1.05}$	$\frac{0.21y_L + 0.84y_S}{1.05}$	$\frac{0.06y_L + y_S}{1.06}$
y	4	3	2	1

Các giá trị y_L và y_S có thể được xác định bằng cách cực tiểu hoá hàm mục tiêu sau (phương pháp bình phương tối thiểu, least squares)

$$J = \sum_k [\bar{y}_k - y_k]^2$$

$$= [\frac{y_L + 0.06y_S}{1.06} - 4]^2 + [\frac{0.84y_L + 0.21y_S}{1.05} - 3]^2 + [\frac{0.21y_L + 0.84y_S}{1.05} - 2]^2 + [\frac{0.06y_L + y_S}{1.06} - 1]^2$$

J đạt cực tiểu tại

$$\frac{\partial J}{\partial y_L} = 0 \Rightarrow 1.57y_L + 0.42y_S - 6.63 = 0$$

$$\frac{\partial J}{\partial y_S} = 0 \Rightarrow 0.42y_L + 1.57y_S - 3.37 = 0$$

Ta được $y_L = 3.9$ và $y_S = 1.1$.

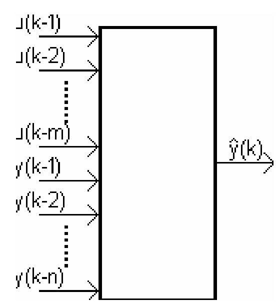
1.6.3 Nhận dạng hệ thống động dùng mô hình mờ (nonlinear dynamic system identification)

Đối tượng

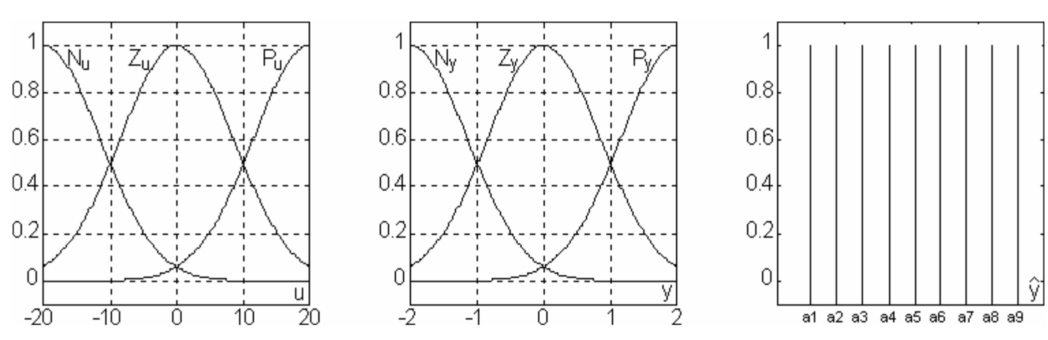
$$y(k) = f(y(k\text{-}1), y(k\text{-}2), \dots, y(k\text{-}n), u(k\text{-}1), u(k\text{-}2), \dots, u(k\text{-}m)) \qquad (1.5.1)$$

Mô hình

$$y(k) = f(y(k\text{-}1), y(k\text{-}2), \dots, y(k\text{-}n), u(k\text{-}1), u(k\text{-}2), \dots, u(k\text{-}m)) \qquad (1.5.2)$$



Hình 1.23 : Mô hình mờ



Hình 1.24 : Các tập mờ của tín hiệu vào và ra

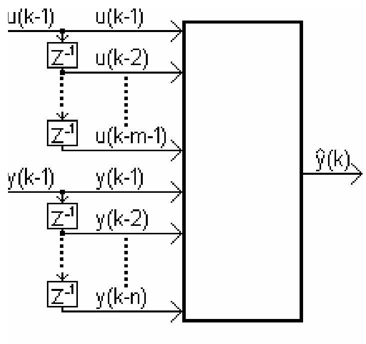
Luật hợp thành

IF [u(k-1) IS N_u] AND [u(k-2) IS Z_u] … AND [u(k-m) IS P_u] AND

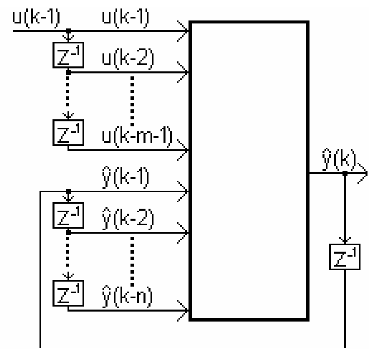
[y(k-1) IS N_y] AND [y(k-2) IS Z_y] … AND [y(k-n) IS P_y] THEN y(k) IS a_1

Nhận dạng : ước lượng giá trị của a_1, a_2, … , a_p từ các dữ liệu đo u(k) và y(k)

1.6.4 Mô phỏng và dự báo

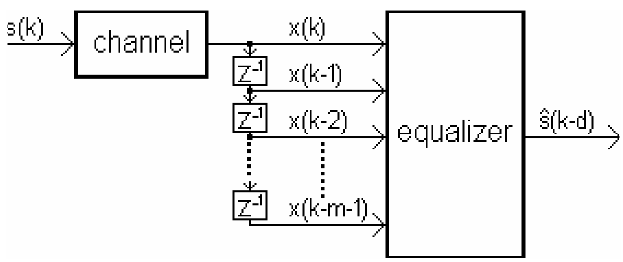


Hình 1.25 : Dự báo



Hình 1.26 : Mô phỏng

1.6.5 Cân bằng kênh thông tin



Hình 1.27 : Cân bằng kênh thông tin

1.7 PHÂN NHÓM MỜ

1.7.1 Phân nhóm rõ (hard c-partition) [7]

Xét tập X = {x_1, x_2, … x_n}. Gọi P(X) là tập hợp của tất cả các tập hợp con của X (power set). Phân nhóm tập X thành c nhóm (hard c-partition)

$A_i \in P(X)$, i = 1, 2, … c

sao cho

$$\bigcup_{i=1}^{c} A_i = X \qquad (1.8.1)$$

$$A_i \cap A_j = \varnothing \ \forall i \neq j \qquad (1.8.2)$$

Mỗi tập con A_i được gọi là 1 nhóm (cluster).

Định nghĩa: Ma trận thành phần U là ma trận cxn

$$U_{ij} = \begin{cases} 1 & , x_j \in A_i \\ 0 & , x_j \notin A_i \end{cases} \tag{1.8.3}$$

Vectơ điểm trọng tâm V là vectơ 1xc (tổng quát x_j là vectơ px1 $\rightarrow$ V là ma trận pxc)

$$V_i = \frac{\sum_{j=1}^{n} U_{ij} x_j}{\sum_{j=1}^{n} U_{ij}} \tag{1.8.4}$$

Ma trận khoảng cách D là ma trận cxn,

$$D_{ij} = \|x_j - V_i\| \tag{1.8.5}$$

Hàm mục tiêu

$$J(U,V) = \sum_{i=1}^{c} \sum_{j=1}^{n} U_{ij} \| x_j - V_i \|^2 = \sum_{i=1}^{c} \sum_{x_j \in A_i} \| x_j - V_i \|^2 = \sum_{i=1}^{c} \sum_{x_j \in A_i} D_{ij}^2 \tag{1.8.6}$$

Giải thuật phân nhóm rõ (ISODATA)

Bước 1: Khởi động trị U(0)

k = 0, 1, 2, …

Bước 2: Tính V(k) theo (1.8.4)

Bước 3: Cập nhật U(k)

$$U_{ij}(k+1) = \begin{cases} 1 & \| x_j - V_i(k) \| = \min_m \| x_j - V_m(k) \| \\ 0 & \text{otherwise} \end{cases} \tag{1.8.7}$$

Bước 4: Nếu $\| U(k+1) - U(k) \| < \varepsilon$ (chọn trước) : dừng.

Ngược lại lặp lại từ bước 2

Ví dụ: Phân nhóm tập hợp X = {1, 2, 4, 7, 9} thành 2 nhóm.

k = 0 : Ma trận thành phần $U(0) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

Vectơ điểm trọng tâm $V(0) = \begin{bmatrix} 1.50 \\ 6.67 \end{bmatrix}$

Ma trận khoảng cách $D(0) = \begin{bmatrix} 0.50 & 0.50 & 2.50 & 5.50 & 7.50 \\ 5.67 & 4.67 & 2.67 & 0.33 & 2.33 \end{bmatrix}$

Vì $D_{13}(0) < D_{23}(0)$ nên phần tử "4" được xếp vào nhóm 1

k = 1 : Ma trận thành phần $U(1) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

Vectơ điểm trọng tâm $V(1) = \begin{bmatrix} 2.33 \\ 8.00 \end{bmatrix}$

Ma trận khoảng cách $D(1) = \begin{bmatrix} 1.33 & 0.33 & 1.67 & 4.67 & 6.67 \\ 7.00 & 6.00 & 4.00 & 1.00 & 1.00 \end{bmatrix}$

Tất cả các phần tử của mỗi nhóm đều gần điểm trọng tâm của nhóm hơn điểm trọng tâm của nhóm khác. Việc phân nhóm kết thúc

1.7.2 Phân nhóm mờ (fuzzy c-partition) [7]

Ma trận thành phần U là ma trận c x n, $0 \le U_{ij} \le 1$

Vectơ điểm trọng tâm V là vectơ c x 1

$$V_i = \frac{\sum_{j=1}^{n} U_{ij}^m x_j}{\sum_{j=1}^{n} U_{ij}^m}, \quad 1 \le i \le c, \quad 1 < m < \infty \qquad (1.8.8)$$

Hàm mục tiêu : phân nhóm X sao cho hàm mục tiêu $J_m(U,V)$ cực tiểu

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} U_{ij}^m \| x_j - V_i \|^2 \qquad (1.8.9)$$

[7] chứng minh được rằng $J_m(U,V)$ cực tiểu với ma trận thành phần U

$$U_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\| x_j - V_i \|}{\| x_j - V_k \|}\right)^{\frac{2}{m-1}}}, \quad 1 \le i \le c, \quad 1 \le j \le n \qquad (1.8.10)$$

<u>Giải thuật phân nhóm mờ</u>

Bước 1: Khởi động trị U(0)

k = 1, 2, …

Bước 2: Tính V(k) theo (1.8.8)

Bước 3: Cập nhật U(k) theo (1.8.10)

Bước 4: Nếu $\| U(k) - U(k\text{-}1) \| < \varepsilon$ (chọn trước) : dừng. Ngược lại lặp lại từ bước 2

Ví dụ: Phân nhóm tập hợp X = {1, 2, 4, 7, 9} thành 2 nhóm.

Chọn m = 2

k = 0 : Ma trận thành phần $U(0) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$

Vectơ $V(0) = \begin{bmatrix} 1.5000 \\ 6.6667 \end{bmatrix}$

k = 1 : Ma trận thành phần : $U(1) = \begin{bmatrix} 0.9923 & 0.9887 & 0.5322 & 0.0037 & 0.0882 \\ 0.0077 & 0.0113 & 0.4678 & 0.9963 & 0.9118 \end{bmatrix}$

Vectơ $V(1) = \begin{bmatrix} 1.8387 \\ 7.4920 \end{bmatrix}$

Do U(1) – U(0) và V(1) – V(0) lớn nên tiếp tục vòng lặp

k = 2 : Ma trận thành phần : $U(2) = \begin{bmatrix} 0.9836 & 0.9991 & 0.7230 & 0.0090 & 0.0425 \\ 0.0164 & 0.0009 & 0.2270 & 0.9910 & 0.9575 \end{bmatrix}$

Vectơ $V(2) = \begin{bmatrix} 2.0366 \\ 7.8107 \end{bmatrix}$

Do U(2) – U(1) và V(2) – V(1) lớn nên tiếp tục vòng lặp

k = 3 : Ma trận thành phần : $U(3) = \begin{bmatrix} 0.9774 & 1.0000 & 0.7902 & 0.0260 & 0.0283 \\ 0.0226 & 0.0000 & 0.2098 & 0.9740 & 0.9717 \end{bmatrix}$

Vectơ $V(3) = \begin{bmatrix} 2.1173 \\ 7.9049 \end{bmatrix}$

Do U(3) – U(2) và V(3) – V(2) lớn nên tiếp tục vòng lặp

k = 4 : Ma trận thành phần : $U(4) = \begin{bmatrix} 0.9745 & 0.9996 & 0.8114 & 0.0332 & 0.0247 \\ 0.0255 & 0.0004 & 0.1886 & 0.9668 & 0.9753 \end{bmatrix}$

Vectơ $V(4) = \begin{bmatrix} 2.1444 \\ 7.9322 \end{bmatrix}$

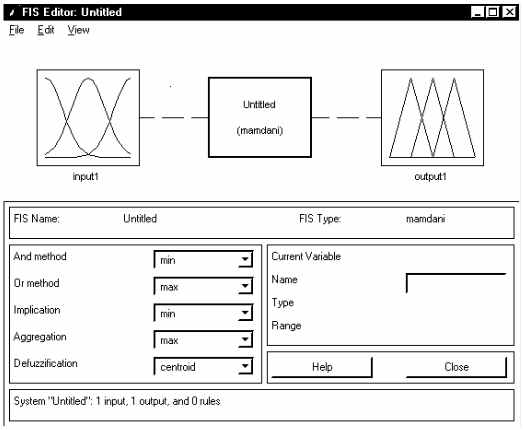
Do U(4) – U(3) và V(4) – V(3) bé nên dừng vòng lặp

Chương trình phân nhóm mờ

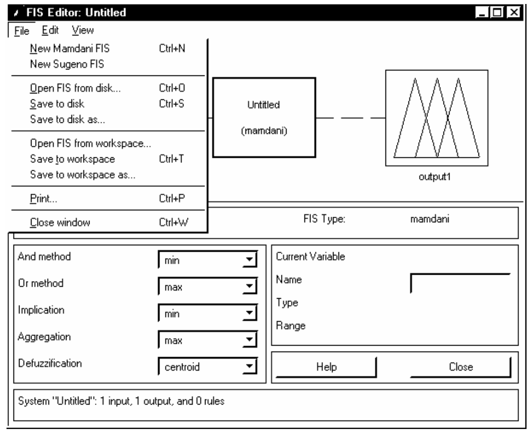
```
clear all
X = [1, 2, 4, 7, 9];
U = [1, 1, 0, 0, 0;
     0, 0, 1, 1, 1];
V = [3/2; 20/3];
n = 5;
m = 2;
c = 2;
%   tính U
mm  = 2/(m-1);
for i = 1 : c,
    for j = 1 : n,
        U(i,j) = 0;
        for k = 1 : c, U(i,j) = U(i,j) + (abs(X(j)-V(i))/abs(X(j)-V(k)))^mm; end
        U(i,j) = 1/U(i,j);
    end
end
%   tính V
for i = 1 : c
    XX1 = 0; XX2 = 0;
    for j = 1 : n,
        XX1 = XX1 + X(j)*(U(i,j)^m);
        XX2 = XX2 + U(i,j)^m;
    end
    V(i) = XX1/XX2;
end
X
U
V
```

1.8 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM MATLAB/FUZZY TOOLBOX

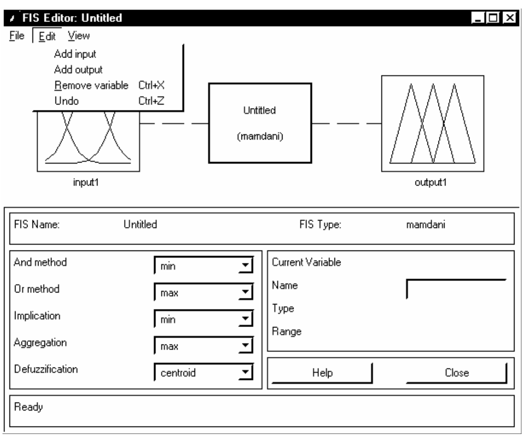
>> fuzzy



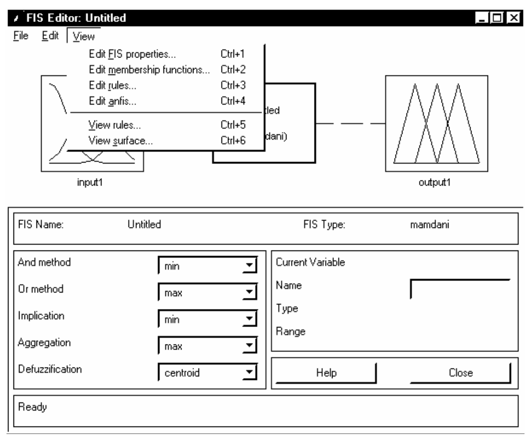
Menu **File**



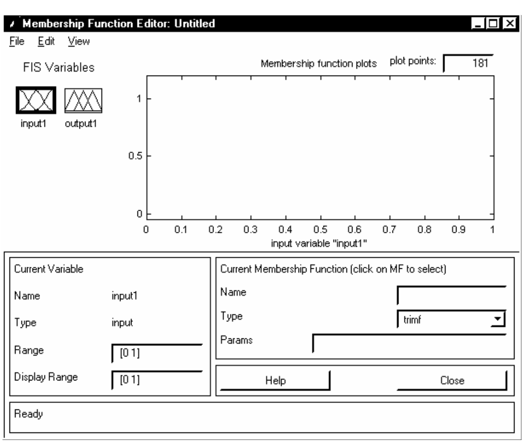
Menu **Edit**



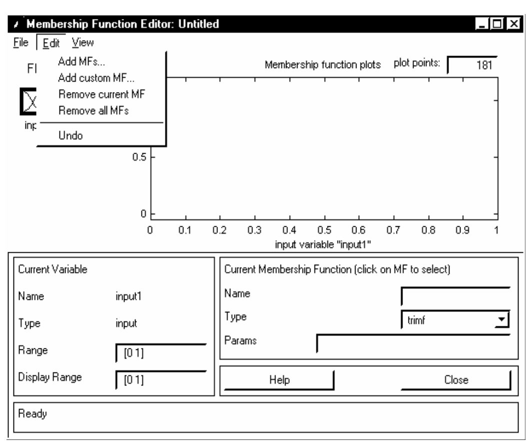
Menu **View**



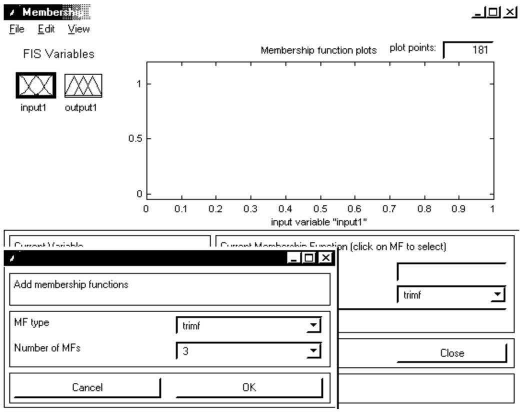
View/Edit membership functions



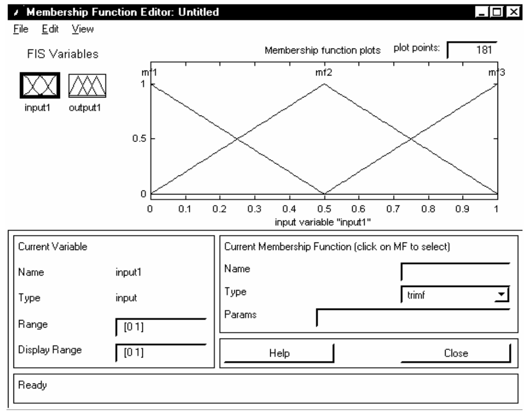
Menu **Edit**



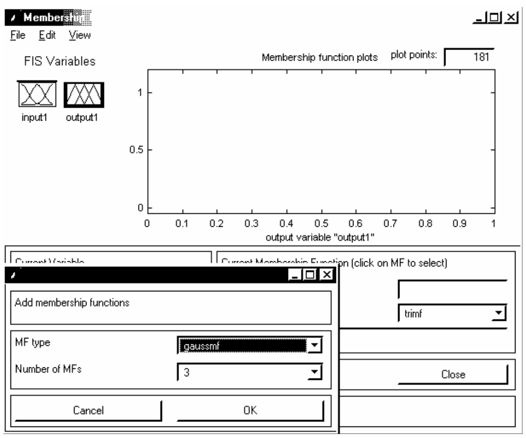
<u>E</u>dit/Add MFs

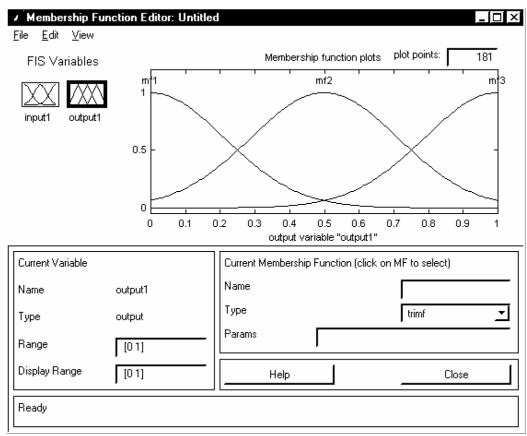


Chọn số và loại tập mờ input

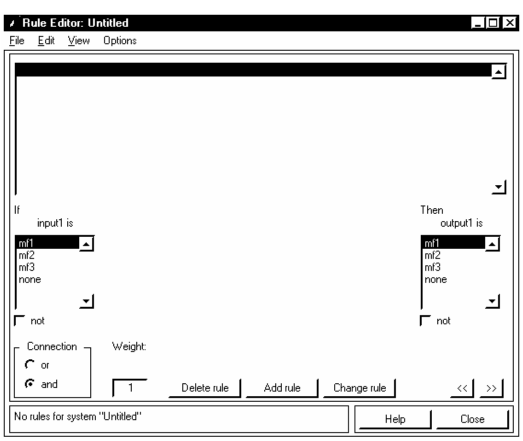


Các tập mờ output (tương tự input)

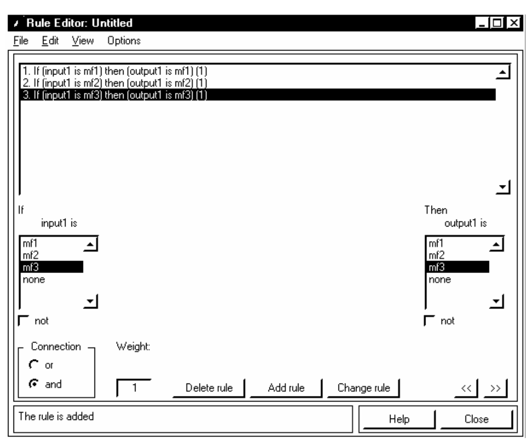




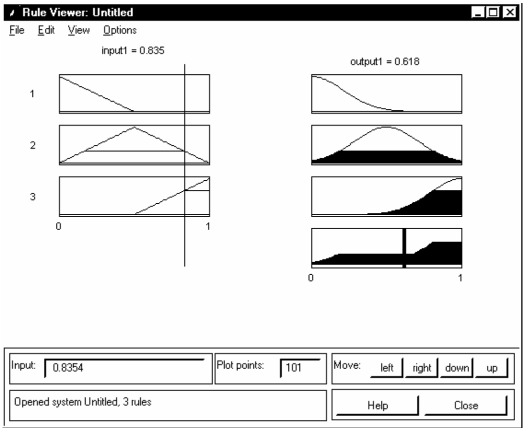
Sau khi chọn **<u>V</u>iew/Edit <u>r</u>ules**



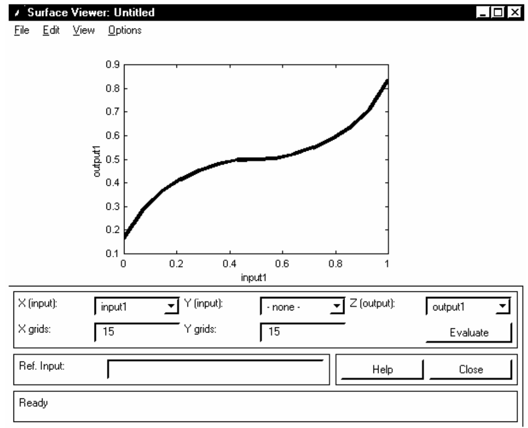
Add rule



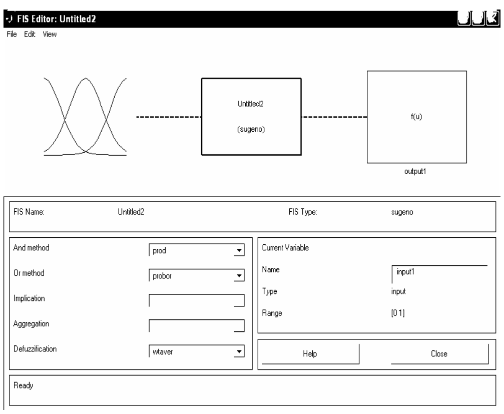
View / View rules



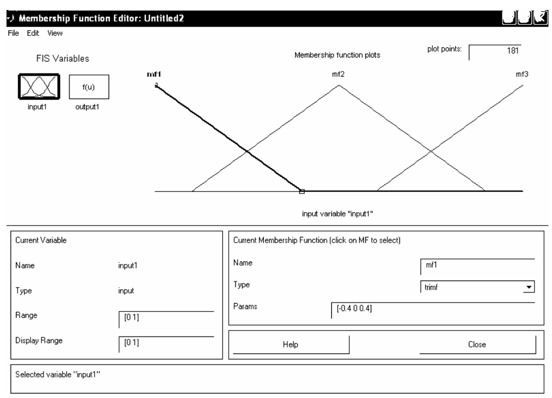
View / View surface



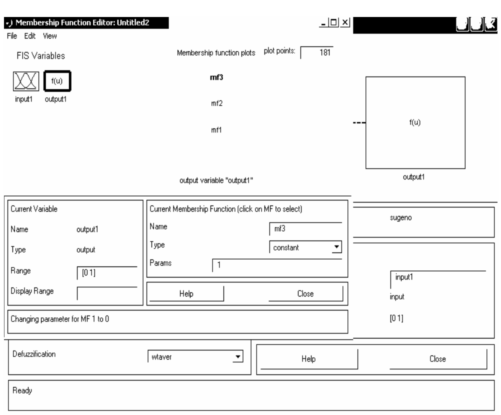
File / New Fis / Sugeno



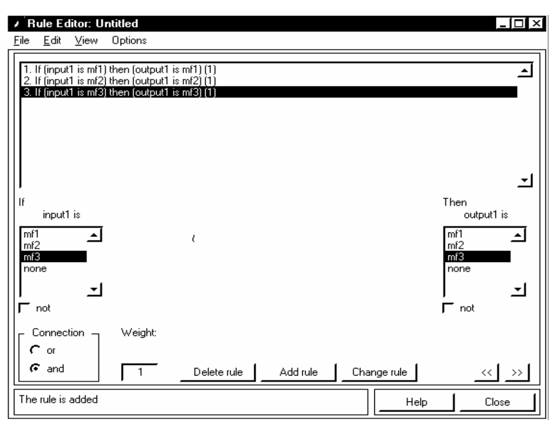
Edit / Membership Function



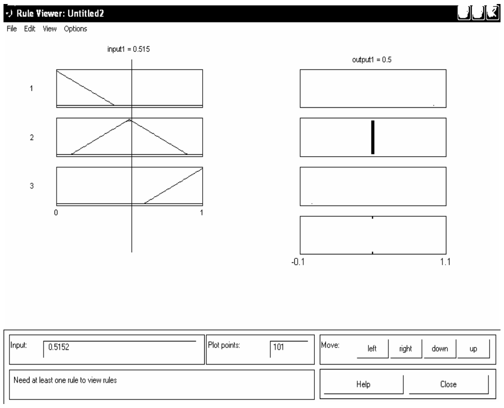
Edit / Membership Function



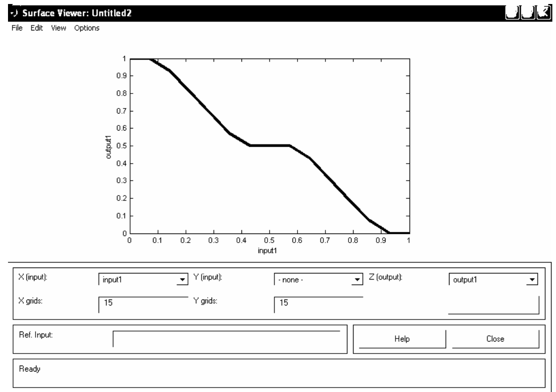
View/Edit rules



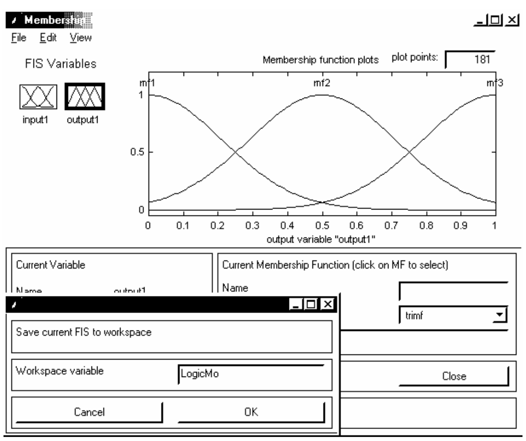
View / View rules



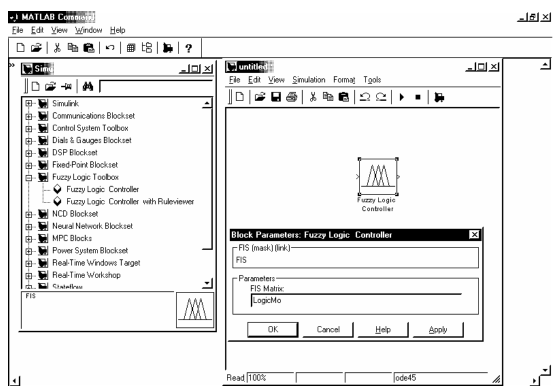
View / View surface



File/Save to workspace as



Fuzzy Logic Controller



Trên **Simulink Library Browser**, chọn **Fuzzy Logic Toolbox/Fuzzy Logic Controller** hoặc **Fuzzy Logic Toolbox/Fuzzy Logic Controller with Ruleviewer** và đặt vào cửa sổ làm việc của SIMULINK. Click biểu tượng và đặt tên FIS Matrix trùng với tên đã đặt cho bộ xử lý mờ.

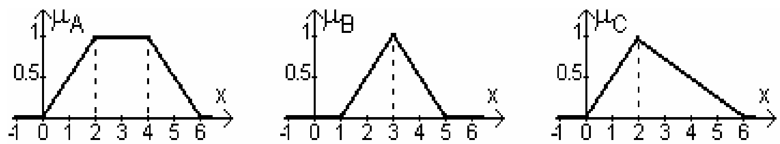
Bài tập

1.1 Tìm các hàm thành viên $\mu_A(x)$, $\mu_B(x)$, $\mu_{\overline{A}}(x)$, $\mu_{\overline{B}}(x)$, $\mu_{A\cap B}(x)$, $\mu_{AUB}(x)$, $\mu_{A\backslash B}(x)$, $\mu_{AxB}(x,y)$ của các tập rõ sau

a) A = [1, 5], B = [3, 9]

b) A = {1, 2, 3}, B = {2, 4}

1.2 Cho các tập mờ cùng cơ sở A, B, C với các hàm thành viên hình P1.1



Hình P1.1

Tìm hàm thành viên của các tập mờ sau

a) $\overline{A}$, $\overline{B}$, $\overline{C}$

b) A∩B, A∩C, B∩C, A∩B∩C theo luật PROD và luật MIN

c) AUB, AUC, BUC, AUBUC theo luật SUM và luật MAX

d) rất C, rất rất C (chọn n = 3), thực sự C, hơi C

1.3 Cho các tập mờ cùng cơ sở A, B, C với các hàm thành viên ở bảng P1.1

Bảng P1.1

x	-3	-2	-1	0	1	2	3
$\mu_A(x)$	0	0,25	0,5	0,75	1	0,5	0
$\mu_B(x)$	0	0,5	1	0,75	0,5	0,25	0
$\mu_C(x)$	0	0,5	1	1	1	0,5	0

Tìm hàm thành viên của các tập mờ sau

a) $\overline{A}$, $\overline{B}$, $\overline{C}$

b) A∩B, A∩C, B∩C, A∩B∩C theo luật PROD và luật MIN

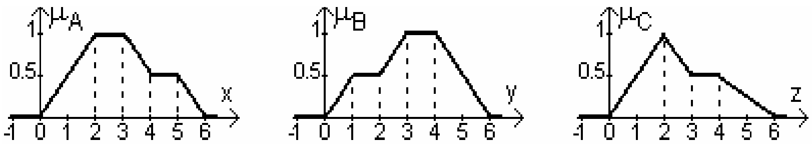
c) AUB, AUC, BUC, AUBUC theo luật SUM và luật MAX

d) rất A, rất rất A (chọn n = 3), thực sự A, hơi A

1.4 Cho các tập mờ khác cơ sở A, B, C với các hàm thành viên hình P1.2. Tìm hàm thành viên của các tập mờ sau

a) A∩B, A∩C, B∩C, A∩B∩C theo luật PROD và luật MIN

b) AUB, AUC, BUC, AUBUC theo luật SUM và luật MAX



Hình P1.2

1.5 Cho các tập mờ khác cơ sở A và B với các hàm thành viên ở bảng P1.2 và P1.3

Bảng P1.2

x	-3	-2	-1	0	1	2	3
$\mu_A(x)$	0	0,25	0,5	0,75	1	0,5	0

Bảng P1.3

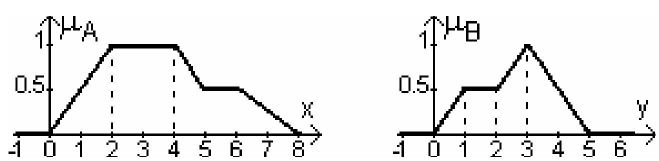
y	-3	-2	-1	0	1	2	3
$\mu_B(y)$	0	0,5	1	0,75	0,5	0,25	0

Tìm hàm thành viên của các tập mờ sau

a) $A \cap B$ theo luật PROD và luật MIN

b) AUB theo luật SUM và luật MAX

1.6 Cho các tập mờ A và B với các hàm thành viên hình P1.3. Tìm hàm thành viên của các mệnh đề hợp thành $\mu_{A \Rightarrow B}(x,y)$ và $\mu_{B \Rightarrow A}(y,x)$ dùng luật MIN và luật PROD.



Hình P1.3

1.7 Cho các tập mờ A và B với các hàm thành viên ở bảng P1.4 và P1.5. Tìm hàm thành viên của các mệnh đề hợp thành $\mu_{A \Rightarrow B}(x,y)$ và $\mu_{B \Rightarrow A}(y,x)$ dùng luật MIN và luật PROD.

Bảng P1.4

x	-3	-2	-1	0	1	2	3	4
$\mu_A(x)$	0	0,5	1	1	0.75	0,5	0,5	0

Bảng P1.5

y	0	2	3	4	5	6	7	8
$\mu_B(y)$	0	0,5	0,5	1	0,5	0,5	0,25	0

1.8 Cho các biến x và y với các tập mờ ở hình P1.3. Xác định giá trị giải mờ của x và y

a) Dùng phương pháp điểm cực đại (cận trái, cận phải, trung bình).

b) Dùng phương pháp điểm trọng tâm.

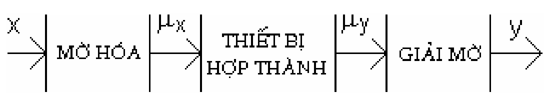
1.9 Cho hệ thống xử lý mờ hình P1.4 với các tập mờ của x và y định nghĩa như hình P1.5. Các luật hợp thành được xác định bởi

IF X IS N_X THEN Y IS L_Y

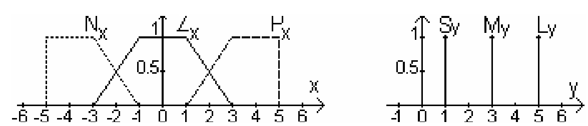
IF X IS Z_X THEN Y IS M_Y

IF X IS P_X THEN Y IS S_Y

Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội, và giải mờ dùng phương pháp điểm trọng tâm. Xác định tập mờ và giá trị giải mờ của y với các giá trị khác nhau của x: x =-4, x =-2, x = 0, x = 2, x = 4.



Hình P1.4



Hình P1.5

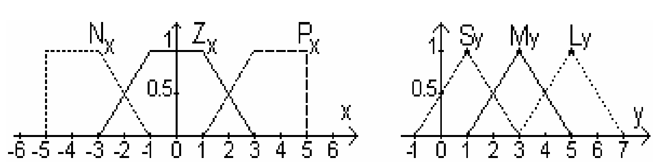
1.10 Cho hệ thống xử lý mờ hình P1.4 với các tập mờ của x và y định nghĩa như hình P1.6. Các luật hợp thành được xác định bởi

IF X IS N_X THEN Y IS L_Y

IF X IS Z_X THEN Y IS M_Y

IF X IS P_X THEN Y IS S_Y

Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội, và giải mờ dùng phương pháp điểm trọng tâm. Xác định tập mờ và giá trị giải mờ của y với các giá trị khác nhau của x: x =-4, x =-2, x = 0, x = 2, x = 4.



Hình P1.6

1.11 Cho hệ thống xử lý mờ hình P1.4 với các tập mờ của x và y định nghĩa như hình P1.6. Các luật hợp thành được xác định bởi

IF X IS N_X THEN Y IS L_Y

IF X IS Z_X THEN Y IS M_Y

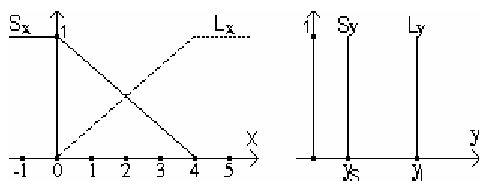
IF X IS P_X THEN Y IS S_Y

Sử dụng luật MIN để diễn dịch phép giao, luật SUM để diễn dịch phép hội, và giải mờ dùng phương pháp điểm trọng tâm. Xác định tập mờ và giá trị giải mờ của y với các giá trị khác nhau của x: x =-4, x =-2, x = 0, x = 2, x = 4.

1.12 Cho hệ xử lý mờ hình P1.4. Các tập mờ của x và y được định nghĩa ở hình P1.7. Các luật hợp thành được xác định bởi

IF X IS S_X THEN Y IS L_Y

IF X IS L_X THEN Y IS S_Y



Hình P1.7

Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội, và giải mờ dùng phương pháp trung bình. Xác định giá trị của y_S và y_L theo các dữ liệu đo ở bảng P1.6.

Bảng P1.6

x	0	1	3	4
y	6	5	3	2

1.13 Cho hệ xử lý mờ hình P1.4. Các tập mờ của x và y được định nghĩa ở hình P1.6. Dữ liệu đo được cho ở bảng P1.7. Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội. Xác định các luật hợp thành và độ tin cậy tương ứng.

Bảng P1.7

x	-4	-2	0	2	4
y	6	4	3	2	0

1.14 Cho hệ thống xử lý mờ với các tập mờ của x, z và y định nghĩa như hình P1.8. Các luật hợp thành được xác định bởi

IF X IS S_X AND Z IS S_Z THEN Y IS L_Y

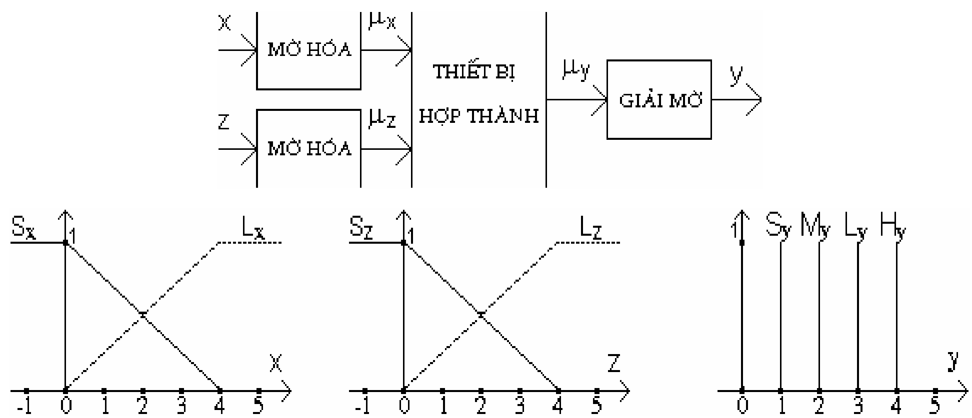
IF [X IS S_X AND Z IS L_Z] OR [X IS L_X AND Z IS S_Z] THEN Y IS M_Y

IF X IS L_X AND Z IS L_Z THEN Y IS S_Y

Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật MAX để diễn dịch phép hội, và giải mờ dùng phương pháp trung bình. Xác định tập mờ và giá trị giải mờ của y với các giá trị của x và z ở bảng P1.8

Bảng P1.8

x	1	1	1	2	2	2	3	3	3
z	1	2	3	1	2	3	1	2	3



Hình P1.8

1.15 Phân nhóm tập họp X dùng phương pháp rõ và phương pháp mờ (với m = 2)

a) $X = \{1, 2, 5, 8, 12\}$, c = 2

b) $X = \{1, 2, 5, 8, 12\}$, c = 3

c) $X = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 \\ 0 & 1 & 2 & 2 & 2 & 0 & 1 & 2 \end{bmatrix}$, c = 2

1.16 Dùng phương pháp phân nhóm mờ, viết m file phân chia tập dữ liệu X thành hai và ba nhóm. Minh họa bằng đồ thị.

a) $X = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 2 & 2 & 1 & 4 & 2 & 3 & 3 \\ 0 & 1 & 2 & 2 & 2 & 0 & 1 & 2 & 0 & 1 & 0 & 0 & 3 & 4 & 3 \end{bmatrix}$

b) X = [-5 -4 -3 -2 -1 0 1 2 3 4 5]

1.17 Dùng Simulink mô phỏng hệ thống rời rạc H(z) (dùng khối *discrete transfer function* với sample time = 0.1) với tín hiệu vào là ồn trắng (dùng khối *band limited white noise* với sample time = 1). Lưu các tín hiệu vào và ra của hệ thống vào tập tin (dùng khối *to file* với sample time = 0.1). Viết m file để đọc dữ liệu trên trên tập tin và nhận dạng mô hình mờ của hệ thống. So sánh tín hiệu ra của hệ thống với tín hiệu ra mô phỏng và dự báo. Nhận xét.

a) $H(z) = \dfrac{1}{z - 0.5}$ b) $H(z) = \dfrac{1}{z - 0.9}$

1.18 Xây dựng bộ cân bằng mờ cho kênh thông tin với hàm truyền đạt $H(z) = \dfrac{0.4}{z - 0.5}$.

a) dùng bộ cân bằng ngang bậc 5.

b) dùng bộ cân bằng ngang bậc 10.

1.19 Dùng Simulink mô phỏng hệ thống điều khiển mờ Mamdani cho đối tượng có hàm truyền đạt $H(s) = \dfrac{1}{20s + 1}$ (chọn chu kỳ lấy mẫu = 1s).

1.20 Dùng Simulink mô phỏng hệ thống điều khiển mờ Sugeno cho đối tượng có hàm truyền đạt $H(s) = \dfrac{1}{20s + 1}$ (chọn chu kỳ lấy mẫu = 1s).

Chương 2 MẠNG NEURON NHÂN TẠO

2.1 KHÁI NIỆM CƠ BẢN

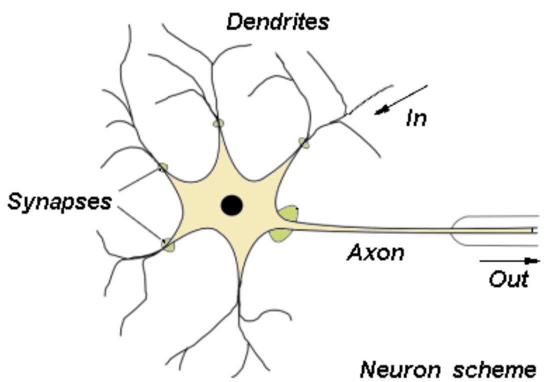
2.1.1 Mạng neuron nhân tạo (artificial neural networks, ANN)

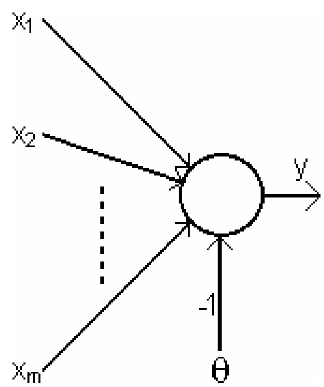
Mạng neuron nhân tạo mô phỏng hoạt động của não người để giải quyết các bài toán kỹ thuật. Bộ não người có khoảng 10^{10} neuron. Các neuron nầy được kết nối với nhau thành mạng. Việc xử lý thông tin được thực hiện nhờ vào sự lan truyền của tín hiệu từ neuron nầy sang neuron khác thông qua các sợi trục thần kinh (axon). Mạng neuron nhân tạo được đặc trưng bởi ba yếu tố

- phần tử xử lý (neuron)
- cấu trúc và ghép nối của các phần tử xử lý
- phương pháp huấn luyện mạng

2.1.2 Neuron (phần tử xử lý: processing element PE) hình 2.1

Hình 2.1 trình bày 1 neuron (a) và 1 neuron nhân tạo (b).





Hình 2.1 : A) Neuron B) Neuron nhân tạo

Mỗi neuron nhân tạo được đặc trưng bởi quan hệ giữa các tín hiệu vào $x_1, x_2, \dots, x_m$ và tín hiệu ra y

$$net = f(x_1, x_2, \dots, x_m) - \theta$$

$$y = a(net)$$

Trong đó θ: ngưỡng tác động của neuron, a(.): hàm tác động (activation function), f(.): hàm tích hợp (integration function).

a) <u>Hàm tích hợp</u>: kết hợp các thông tin nhận được ở tín hiệu vào. Ta có một số hàm tích hợp thông dụng như sau

- Hàm tích hợp tuyến tính

$$net = f(x) - \theta = \sum_{j=1}^{m} w_j x_j - \theta = w^T x \qquad (2.1.1)$$

với $x = [x_1, x_2, \dots, x_m, -1]^T$ và $w = [w_1, w_2, \dots, w_m, \theta]^T$: vectơ các trọng số

- Hàm tích hợp dạng toàn phương (quaratic function)

$$net = f(x) - \theta = \sum_{j=1}^{m} w_j x_j^2 - \theta \qquad (2.1.2)$$

b) <u>Hàm tác động</u> (hàm kích hoạt, hàm truyền, activation function, transfer function): tạo tín hiệu ra y = a(net). Ta có một số hàm tác động thông dụng như sau

- Hàm tác động tuyến tính

$$a(net) = net \qquad (2.1.3)$$

- Bước nhảy đơn vị (step function)

$$a(net) = \begin{cases} 1 & \text{nếu} \ \ net > 0 \\ 0 & \text{nếu} \ \ net \le 0 \end{cases} \qquad (2.1.4)$$

- Hàm dấu (signum function)

$$a(net) = \begin{cases} 1 & \text{nếu} \ \ net \ge 0 \\ -1 & \text{nếu} \ \ net < 0 \end{cases} \qquad (2.1.5)$$

- Hàm dốc (ramp function)

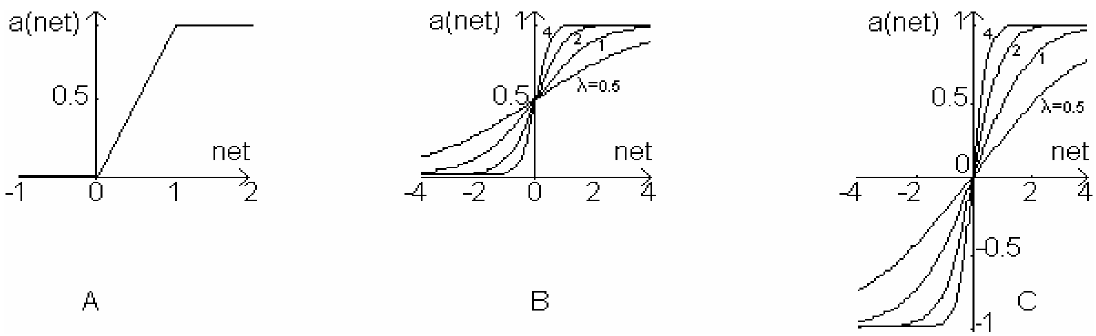
$$a(net) = \begin{cases} 0 & \text{nếu } net < 0 \\ net & \text{nếu } 0 \le net \le 1 \\ 1 & \text{nếu } net > 1 \end{cases} \qquad (2.1.6)$$

- Hàm sigmoid đơn cực (unipolar sigmoid function, logsig)

$$a(net) = \frac{1}{1 + e^{-\lambda net}} \qquad (2.1.7)$$

- Hàm sigmoid lưỡng cực (bipolar sigmoid function, tansig)

$$a(net) = \frac{2}{1 + e^{-\lambda net}} - 1 \qquad (2.1.8)$$

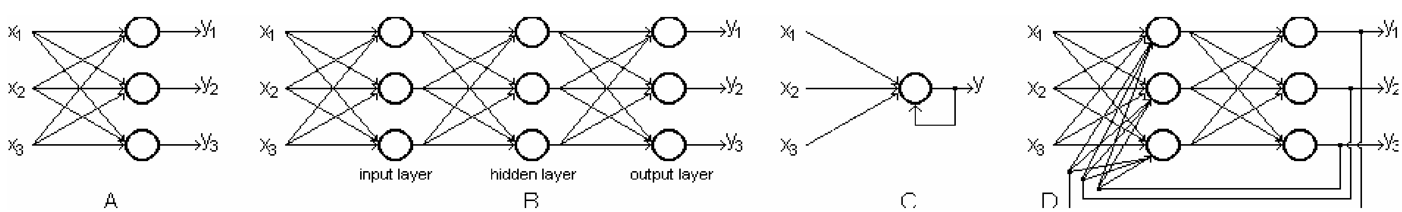


Hình 2.2 : A) hàm dốc B) Hàm sigmoid đơn cực C) Hàm sigmoid lưỡng cực

Trong (2.1.7) và (2.1.8), λ là hằng số dương xác định độ dốc của hàm tác động ở lân cận điểm net = 0. Khi $\lambda \to \infty$ hàm sigmoid đơn cực tiến đến bước nhảy đơn vị, hàm sigmoid lưỡng cực tiến đến hàm dấu.

2.1.3 Kết nối

Hình 2.3 trình bày một số kết nối các neuron thành mạng



Hình 2.3 : Các cấu trúc tiêu biểu của mạng neuron

A) mạng truyền thẳng 1 lớp (single layer feedforward network)

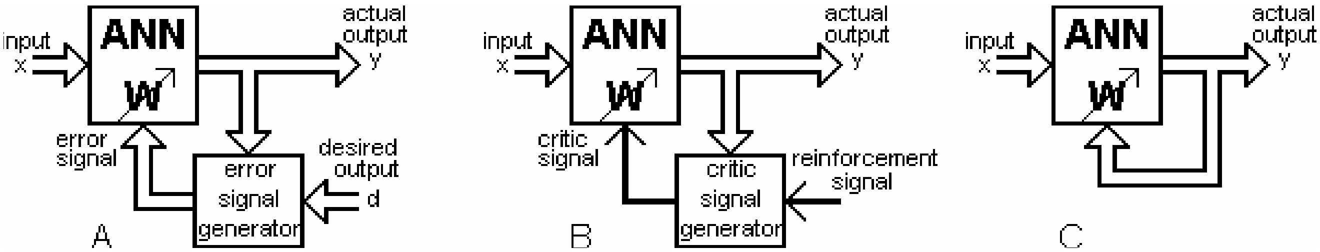
B) mạng truyền thẳng nhiều lớp (multi layer feedforward network)

C) nút đơn hồi tiếp (single node with feedback to itself)

D) mạng hồi quy nhiều lớp (multi layer recurrent network)

2.1.4 **Huấn luyện mạng** (training, learning)

Mục đích của huấn luyện mạng là xác định các bộ trọng số của các neuron trong mạng từ các dữ liệu. Nếu quá trình huấn luyện có sử dụng tín hiệu ra mong muốn d, ta nói các neuron học có giám sát. Nếu quá trình huấn luyện không sử dụng tín hiệu ra mong muốn, ta nói các neuron học không có giám sát. Nếu quá trình huấn luyện không sử dụng tín hiệu ra mong muốn nhưng có sử dụng 1 tín hiệu đánh giá chất lượng của mạng, ta nói quá trình học của mạng là quá trình học tăng cường (hình 2.4).



Hình 2.4 : Các phương pháp học (learning)

A) học có giám sát (supervised learning)

B) học tăng cường (reinforcement learning)

C) học không có giám sát (unsupervised learning, self organizing)

Luật học các trọng số thường có dạng sau

$$w(k+1) = w(k) + \eta r(k)x(k) \tag{2.1.9}$$

trong đó $\eta > 0$: hằng số học (learning constant), $r = f_r(w, x, d)$: tín hiệu học.

Với các tín hiệu học $f_r(w, x, d)$ khác nhau, ta có các luật học (learning rule) khác nhau. Ví dụ luật học của Hebb được xác định bởi

$$r(k) = y(k) \tag{2.1.10}$$

Ta có

$$w(k+1) = w(k) + \eta y(k)x(k) \tag{2.1.11}$$

Ta thấy luật học của Hebb không sử dụng tín hiệu ra mong muốn d(k): phương pháp học của Hebb là phương pháp học không có giám sát.

2.2 MẠNG PERCEPTRON MỘT LỚP (single-layer perceptron networks)

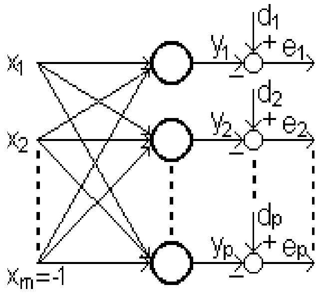
Hình 2.6 trình bày mạng perceptron một lớp với m-1 tín hiệu vào x_j và p tín hiệu ra y_i ($x_m = -1$ tương ứng với ngưỡng tác động w_{im} của neuron).

$$y_i = a\left(\sum_{j=1}^{m} w_{ij}x_j\right) = a(w_i^T x) \tag{2.2.1}$$

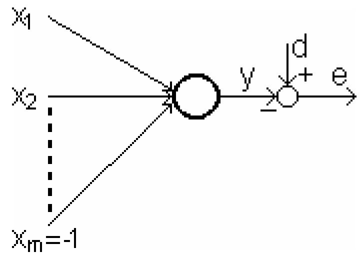
Do các neuron được huấn luyện một cách độc lập với nhau, xét trường hợp p = 1

$$y = a\left(\sum_{j=1}^{m} w_j x_j\right) = a(w^T x)$$

Mục tiêu : huấn luyện mạng dùng N mẫu dữ liệu {x(k), d(k), k = 1, 2, … N} sao cho y(k) = d(k), k = 1, 2, … N.



Hình 2.6 : Mạng perceptron một lớp



Hình 2.7

2.2.1 Đơn vị tuyến tính (linear unit)

Đơn vị tuyến tính (cònđược gọi là phần tử tuyến tính thích nghi: adaptive linear element, ADALINE) là 1 neuron với hàm tích hợp và hàm tác động tuyến tính a(net) = net

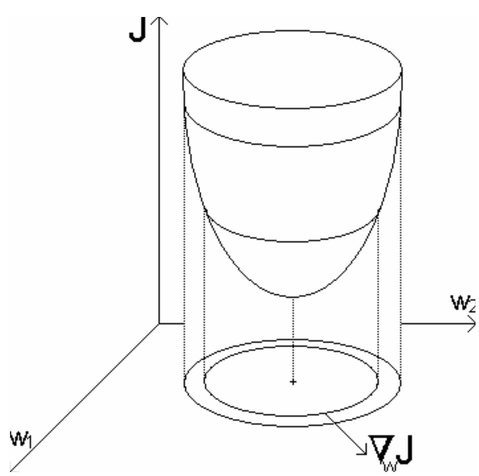
$$y = \sum_{j=1}^{m} w_j x_j = w^T x \qquad (2.2.2)$$

Hàm mục tiêu : Xác định vectơ thông số w sao cho

$$J(w) = \frac{1}{2}(d - y)^2 \qquad (2.2.3)$$

cực tiểu, với d là tín hiệu ra mong muốn.

Điều kiện cần và đủ để tồn tại nghiệm w là các mẫu dữ liệu x độc lập tuyến tính.



Hình 2.8

Sử dụng giải thuật steepest descent, ta có

$$w(k+1) = w(k) - \eta \nabla_w J(w) \qquad (2.2.4)$$

với $\eta > 0$ là hằng số học, $\nabla_w J(w)$ là vectơ gradient của J(w)

$$\nabla_w J(w) = \left[\frac{\partial J}{\partial w}\right]^T = \frac{\partial J}{\partial y}\frac{\partial y}{\partial net}\left[\frac{\partial net}{\partial w}\right]^T = -(d-y)\frac{\partial a}{\partial net}x = -(d-y)x \qquad (2.2.5)$$

$$\Rightarrow \quad w(k+1) = w(k) + \eta(d - y)x \qquad (2.2.6)$$

(luật học đệ quy Widrow-Hoff, còn gọi là luật học adaline hoặc least mean square LMS). So sánh với (2.1.9), tín hiệu học của luật học Widrow-Hoff là $r(k) = d(k) - y(k)$.

2.2.2 Đơn vị phân loại tuyến tính (linear graded unit, LGU)

LGU là 1 neuron với hàm tích hợp tuyến tính và hàm tác động là hàm sigmoid. Ta có

$$J(w) = \frac{1}{2}(d - y)^2 = \frac{1}{2}(d - a(net))^2 = \frac{1}{2}(d - a(w^Tx))^2 = \frac{1}{2}\left(d - a\left(\sum_{j=1}^{m} w_j x_j\right)\right)^2 \qquad (2.2.7)$$

Ta có

$$\nabla_w J(w) = \left[\frac{\partial J}{\partial w}\right]^T = \frac{\partial J}{\partial y}\frac{\partial y}{\partial net}\frac{\partial net}{\partial w} = -(d - y)\frac{\partial a(net)}{\partial net}x \qquad (2.2.8)$$

$$\Rightarrow \quad w(k+1) = w(k) + \eta(d - y)\frac{\partial a(net)}{\partial net}x \qquad (2.2.9)$$

(luật học delta, delta learning rule). Tín hiệu học được xác định bởi

$$r = (d - y)\frac{\partial a(net)}{\partial net} \qquad (2.2.10)$$

Ví dụ : Cho 1 LGU với 3 tín hiệu vào, hàm tác động $a(net) = \dfrac{2}{1+e^{-net}} - 1$.

Các mẫu dữ liệu huấn luyện

$x(1) = [\,1, -2\ \ , 0\ \ , -1]^T$, $\qquad d(1) = -1$

$x(2) = [\,0,\ 1.5, -0.5, -1]^T$, $\qquad d(2) = -1$

$x(3) = [-1,\ 1\ \ , 0.5\ , -1]^T$, $\qquad d(3) = 1$

Chọn trị khởi động $w(1) = [1, -1, 0, 0.5]^T$ và $\eta = 0.1$. Áp dụng luật học delta (2.2.9), ta có

$$\frac{\partial a(net)}{\partial net} = \frac{1}{2}[1-a^2(net)] = \frac{1}{2}[1-y^2]$$

$$\Delta w(k) = 0.5\eta[d(k) - y(k)][1 - y^2(k)]x(k)$$

k = 1 : $x(1) = [1, -2, 0, -1]^T$, $d(1) = -1$, $w(1) = [1, -1, 0, 0.5]^T$

$net(1) = w(1)^T x(1) = 2.5$

$$y(1) = \frac{2}{1+e^{-2.5}} - 1 = 0.8483$$

$$w(2) = w(1) + 0.5\eta(d(1) - y(1))(1 - y^2(1))x(1) = [0.974, -0.948, 0, 0.526]^T$$

k = 2 : $x(2) = [0, 1.5, -0.5, -1]^T$, $d(2) = -1$, $w(2) = [0.974, -0.948, 0, 0.526]^T$

$$net(2) = w(2)^T x(2) = -1.948$$

$$y(2) = \frac{2}{1+e^{-1.948}} - 1 = -0.75$$

$$w(3) = w(2) + 0.5\eta(d(2) - y(2))(1 - y^2(2))x(2) = [0.974, -0.956, 0.003, 0.531]^T$$

k = 3 : $x(3) = [-1, 1, 0.5, -1]^T$, $d(3) = 1$, $w(3) = [0.974, -0.956, 0.003, 0.531]^T$

$$net(3) = w(3)^T x(3) = -2.46$$

$$y(3) = \frac{2}{1+e^{-2.46}} - 1 = -0.843$$

$$w(4) = w(3) + 0.5\eta(d(3) - y(3))(1 - y^2(3))x(3) = [0.947, -0.930, 0.016, 0.505]^T$$

Kết thúc 1 epoch. Quá trình huấn luyện được lặp lại với x(4) = x(1) …

<u>Ghi chú</u> : Chương trình matlab của ví dụ

```
x(:,1) = [ 1,-2  , 0  ,-1  ]',        d(1) = -1
x(:,2) = [ 0, 1.5,-0.5,-1  ]',        d(2) = -1
x(:,3) = [-1, 1  , 0.5,-1  ]',        d(3) =  1
w(:,1) = [ 1,-1  , 0  , 0.5]',        eta= 0.1
for k = 1 : 3,
    net(k) = w(:,k)'*x(:,k)
    y(k) = (2/(1+exp(-net(k)))) -1
    dw(:,k) = eta*(d(k)-y(k))*((1-y(k)^2)/2)*x(:,k)
    w(:,k+1) = w(:,k) + dw(:,k)
end
```

2.2.3 Đơn vị ngưỡng tuyến tính (Linear threshold unit, LTU)

LTU là 1 neuron với hàm tích hợp tuyến tính và hàm tác động là hàm dấu. Vì a() là hàm dấu nên d(k) và y(k) chỉ có thể nhận các giá trị 1 hoặc –1

$$y(k) = \text{sign}\left(\sum_{j=1}^{m} w_j x_j(k)\right) = \text{sign}(w^T x) \qquad (2.2.11)$$

Điều kiện cần và đủ để tồn tại nghiệm w_i là các mẫu dữ liệu $\{x(k), d(k), k = 1, 2, \dots p\}$ phân chia một cách tuyến tính (linearly separable). Nghĩa là tồn tại một siêu phẳng

$$\sum_{j=1}^{m} w_j x_j(k) = 0 \qquad (2.2.12)$$

trong không gian dữ liệu sao cho các dữ liệu thuộc hai nhóm khác nhau nằm ở hai phía khác nhau đối với siêu phẳng (các dữ liệu cùng nhóm nằm cùng một phía).

Luật học perceptron (perceptron learning rule) :

$$\Delta w(k) = w(k+1) - w(k) = \eta r(k)x(k)$$

Tín hiệu học

$$r(k) = d(k) - y(k) \qquad (2.2.13)$$

Thay vào (2.1.9), ta có

$$\Delta w(k) = \eta[d(k) - y(k)]x(k) = \begin{cases} 2\eta d(k)x(k) & \text{nếu } y(k) \neq d(k) \\ 0 & \text{nếu } y(k) = d(k) \end{cases} \qquad (2.2.14)$$

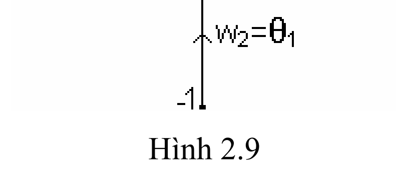
Định lý: Giả thiết vectơ thông số chính xác tồn tại (xác định bởi siêu phẳng (2.2.12) sao cho các dữ liệu cùng nhóm nằm cùng một phía đối với siêu phẳng). Với luật học perceptron (2.2.14), vectơ thông số w hội tụ đến vectơ thông số chính xác sau một số bước học hữu hạn.

Ví dụ : Huấn luyện LTU hình 2.9 theo các dữ liệu cho ở bảng 2.1.

Bảng 2.1

x_o	0,5	-1	2	-2
d	1	-1	1	-1

(phân loại $x_o(k)$ thành 2 nhóm).



Hình 2.9

Định nghĩa : $x(k) = \begin{bmatrix} x_o(k) \\ -1 \end{bmatrix}$: $x(1) = \begin{bmatrix} 0.5 \\ -1 \end{bmatrix}$, $x(2) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $x(3) = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, $x(4) = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$

(thành phần thứ 2 của x(k) ứng với ngưỡng của neuron)

Chọn $\eta = 0.5$ và khởi động trị vectơ thông số $w(1) = \begin{bmatrix} -2 \\ 1.5 \end{bmatrix}$, ta được

$k = 1$: $x(1) = \begin{bmatrix} 0.5 \\ -1 \end{bmatrix}$, $d(1) = 1$

$$y(1) = \text{sign}(w(1)^T x(1)) = \text{sign}([-2 \;\; 1.5] \begin{bmatrix} 0.5 \\ -1 \end{bmatrix}) = -1$$

$$w(2) = w(1) + \eta(d(1) - y(1))x(1) = \begin{bmatrix} -2 \\ 1.5 \end{bmatrix} + 0.5(1 - (-1))\begin{bmatrix} 0.5 \\ -1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ 0.5 \end{bmatrix}$$

$k = 2$: $x(2) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $d(2) = 1$

$$y(2) = \text{sign}(w(2)^T x(2)) = \text{sign}([-1.5 \;\; 0.5] \begin{bmatrix} -1 \\ -1 \end{bmatrix}) = 1$$

$$w(3) = w(2) + \eta(d(2) - y(2))x(2) = \begin{bmatrix} -1.5 \\ 0.5 \end{bmatrix} + 0.5(-1 - (1))\begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 1.5 \end{bmatrix}$$

$k = 3$: $x(3) = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, $d(3) = 1$

$$y(3) = \text{sign}(w(3)^T x(3)) = \text{sign}([-0.5 \;\; 1.5]\begin{bmatrix} 2 \\ -1 \end{bmatrix}) = -1$$

$$w(4) = w(3) + \eta(d(3) - y(3))x(3) = \begin{bmatrix} -0.5 \\ 1.5 \end{bmatrix} + 0.5(1 - (-1))\begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}$$

$k = 4$: $x(4) = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$, $d(4) = -1$

$$y(4) = \text{sign}(w(4)^T x(4)) = \text{sign}([1.5 \;\; 0.5]\begin{bmatrix} -2 \\ -1 \end{bmatrix}) = -1$$

$$w(5) = w(4) + \eta(d(4) - y(4))x(4) = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} + 0.5(-1 - (-1))\begin{bmatrix} -2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} = w(4)$$

$k = 5$: $x(5) = x(1) = \begin{bmatrix} 0.5 \\ -1 \end{bmatrix}$, $d(5) = d(1) = 1$

$$y(5) = \text{sign}(w(5)^T x(5)) = \text{sign}([1.5\ \ 0.5]\begin{bmatrix}0.5\\-1\end{bmatrix}) = 1$$

$$w(6) = w(5) + \eta(d(5) - y(5))x(5) = \begin{bmatrix}1.5\\0.5\end{bmatrix} + 0.5(1-(1))\begin{bmatrix}0.5\\-1\end{bmatrix} = \begin{bmatrix}1.5\\0.5\end{bmatrix} = w(5)$$

……………

2.2.4 Ứng dụng trong phân nhóm

Hàm tác động của lớp ra thường là hàm sigmoid hoặc hardlim

2.2.5 Ứng dụng trong nhận dạng hệ thống động

Đối tượng

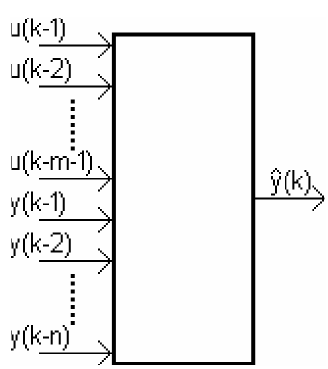
$$y(k) = f(y(k-1), y(k-2), \ldots, y(k-n), u(k-1), u(k-2), \ldots, u(k-m)) \qquad (2.2.15)$$

Mô hình

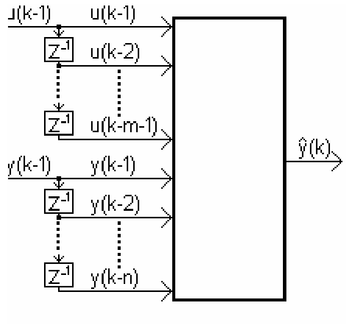
$$y(k) = f(y(k-1), y(k-2), \ldots, y(k-n), u(k-1), u(k-2), \ldots, u(k-m)) \qquad (2.2.16)$$

Nhận dạng : huấn luyện mạng từ các dữ liệu đo u(k) và y(k)

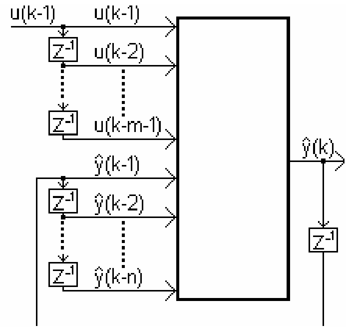
Ứng dụng : - dự báo
- mô phỏng
- cân bằng kênh thông tin
- điều khiển



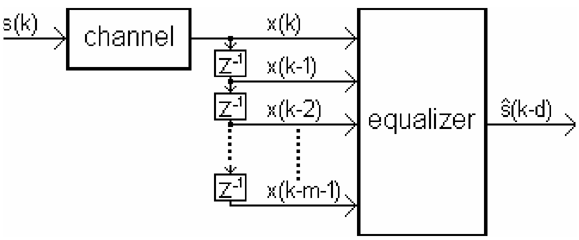
Hình 2.10 : Mô hình dùng mạng neuron



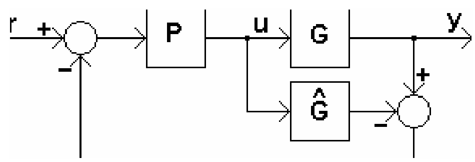
Hình 2.11 : Dự báo



Hình 2.12 : Mô phỏng



Hình 2.13: Cân bằng kênh thông tin



Hình 2.14 : Điều khiển mô hình nội

2.2.6 Ứng dụng trong điều khiển

Điều khiển mô hình nội, điều khiển trượt …

2.2.7 Ứng dụng trong nhận dạng ký tự (OCR : optical character recognition)

2.3 MẠNG TRUYỀN THẲNG NHIỀU LỚP

2.3.1 Tổng quan

Mọi quan hệ phi tuyến đều có thể được xấp xỉ với độ chính xác tùy ý bởi một mạng truyền thẳng nhiều lớp với số nút ẩn đủ lớn, hàm tích hợp tuyến tính hoặc đa thức, và hàm tác động squashing .

Hàm $a(f) : R \rightarrow [0,1]$ (hoặc $R \rightarrow [-1,1]$) được gọi là hàm squashing nếu, $a(f)$ không giảm khi f tăng, $a(-\infty) = 0$ (hoặc -1), và $a(+\infty) = 1$. Các hàm định nghĩa bởi (2.1.4), (2.1.5), (2.1.6), (2.1.7), (2.1.8) đều là các hàm squashing.

Với mạng perceptron nhiều lớp (có lớp ẩn), các điều kiện

- các mẫu dữ liệu phân chia một cách tuyến tính (LTU)
- các mẫu dữ liệu độc lập tuyến tính (LGU)

là không cần thiết.

Ví dụ : Mạng perceptron 3 lớp (hình 2.15) thực hiện phép logic sau

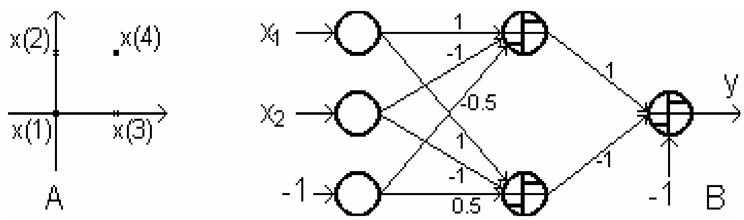
$x(1) = [0, 0]^T$, $d(1) = 1$

$x(2) = [0, 1]^T$, $d(2) = -1$

$x(3) = [1, 0]^T$, $d(3) = -1$

$x(4) = [1, 1]^T$, $d(4) = 1$

với các dữ liệu không phân chia tuyến tính



Hình 2.15 : A) không gian dữ liệu, B) Mạng perceptron 3 lớp

2.3.2 Giải thuật huấn luyện lan truyền ngược (back propagation)

Dữ liệu huấn luyện : $\{x(k), d(k), k = 1, 2, \ldots N\}$

<u>Trường hợp p = 1</u>

Lớp ẩn (hidden layer)

$$net_q = \sum_{j=1}^{m} v_{qj} x_j = v_q^T x \tag{2.3.1}$$

$$z_q = a(net_q) = a(v_q^T x) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{2.3.2}$$

Lớp ra (output layer)

$$net = \sum_{q=1}^{r} w_q z_q = \sum_{q=1}^{r} w_q a(v_q^T x) = \sum_{q=1}^{r} w_q a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{2.3.3}$$

$$y = a(net) = a\left(\sum_{q=1}^{r} w_q z_q\right) = a\left(\sum_{q=1}^{r} w_q a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) = a\left(\sum_{q=1}^{r} w_q a(v_q^T x)\right) \tag{2.3.4}$$

Hàm mục tiêu

$$J(w) = \frac{1}{2}[d - y]^2 = \frac{1}{2}\left[d - a\left(\sum_{q=1}^{r} w_q z_q\right)\right]^2 = \frac{1}{2}\left[d - a\left(\sum_{q=1}^{r} w_q a(v_q^T x)\right)\right]^2 \tag{2.3.5}$$

Luật học đệ quy (gọi là luật học delta tổng quát hóa, generalized delta learning rule) được xây dựng dùng phương pháp steepest-descent.

- Với các nhánh nối giữa các nút ẩn z và nút ra y

$$\Delta w = -\eta \left[\frac{\partial J}{\partial w} \right]^T = -\eta \frac{\partial J}{\partial y} \frac{\partial y}{\partial net} \left[\frac{\partial net}{\partial w} \right]^T = \eta [d - y] \frac{\partial a(net)}{\partial net} z = \eta \delta_o z \qquad (2.3.6)$$

$$\delta_o = [d - y] \frac{\partial a(net)}{\partial net} \qquad (2.3.7)$$

- Với các nhánh nối giữa các nút vào x và nút ẩn z_q

$$\Delta v_q = -\eta \left[\frac{\partial J}{\partial v_q} \right]^T = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial net} \frac{\partial net}{\partial z_q} \frac{\partial z_q}{\partial net_q} \left[\frac{\partial net_q}{\partial v_q} \right]^T$$

$$= \eta [d - y] \frac{\partial a(net)}{\partial net} w_q \frac{\partial a(net_q)}{\partial net_q} x = \eta \delta_q x \qquad (2.3.8)$$

$$\delta_q = [d - y] \frac{\partial a(net)}{\partial net} w_q \frac{\partial a(net_q)}{\partial net_q} = \delta_o w_q \frac{\partial a(net_q)}{\partial net_q} \qquad (2.3.9)$$

Ví dụ : Trường hợp các hàm tác động a() là hàm sigmoid lưỡng cực (2.1.8), ta có

$$\frac{\partial a(net)}{\partial net} = \frac{1}{2} [1 - a(net)^2] = \frac{1}{2} [1 - y^2]$$

$$\frac{\partial a(net_q)}{\partial net_q} = \frac{1}{2} [1 - a(net_q)^2] = \frac{1}{2} [1 - z_q^2]$$

(2.3.7) và (2.3.8) trở thành

$$\delta_o = \frac{1}{2} (1 - y^2)(d - y) \qquad (2.3.10)$$

$$\delta_q = \frac{1}{2} (1 - z_q^2) \delta_o w_q \qquad (2.3.11)$$

$$\Rightarrow \quad \Delta w = \frac{1}{2} \eta (1 - y^2)(d - y) z \qquad (2.3.12)$$

$$\Rightarrow \quad \Delta v_q = \frac{1}{2} \eta (1 - z_q^2) \delta_o w_q x \qquad (2.3.13)$$

<u>Trường hợp tổng quát p > 1</u>

Lớp ẩn (hidden layer)

$$net_q = v_q^T x = \sum_{j=1}^{m} v_{qj} x_j \qquad (2.3.14)$$

$$z_q = a(net_q) = a(v_q^T x) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \qquad (2.3.15)$$

Lớp ra (output layer)

$$net_i = w_i^T z = \sum_{q=1}^{r} w_{iq} z_q = \sum_{q=1}^{r} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \qquad (2.3.16)$$

$$y_i = a(net_i) = a\left(\sum_{q=1}^{r} w_{iq} z_q\right) = a\left(\sum_{q=1}^{r} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \qquad (2.3.17)$$

Hàm mục tiêu

$$J(w) = \frac{1}{2}\sum_{i=1}^{p}(d_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{p}(d_i - a(net_i))^2 = \frac{1}{2}\sum_{i=1}^{p}\left(d_i - a\left(\sum_{q=1}^{r} w_{iq} z_q\right)\right)^2 \qquad (2.3.18)$$

Luật học đệ quy (gọi là luật học delta tổng quát hóa, generalized delta learning rule) được xây dựng dùng phương pháp steepest-descent.

- Với các nhánh nối giữa nút ẩn và nút ra

$$\Delta w_{iq} = -\eta \frac{\partial J}{\partial w_{iq}} = -\eta \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial net_i}\frac{\partial net_i}{\partial w_{iq}} = \eta(d_i - y_i)\frac{\partial a(net_i)}{\partial net_i} z_q = \eta \delta_{oi} z_q \qquad (2.3.19)$$

$$\delta_{oi} = -\frac{\partial J}{\partial net_i} = -\frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial net_i} = (d_i - y_i)\frac{\partial a(net_i)}{\partial net_i} \qquad (2.3.20)$$

- Với các nhánh nối giữa nút vào và nút ẩn

$$\Delta v_{qj} = -\eta \frac{\partial J}{\partial v_{qj}} = -\eta \sum_{i=1}^{p} \frac{\partial J}{\partial y_i}\frac{\partial y_i}{\partial net_i}\frac{\partial net_i}{\partial z_q}\frac{\partial z_q}{\partial net_q}\frac{\partial net_q}{\partial v_{qj}}$$

$$\Delta v_{qj} = \eta \sum_{i=1}^{p} [d_i - y_i]\frac{\partial a(net_i)}{\partial net_i} w_{iq} \frac{\partial a(net_q)}{\partial net_q} x_j = \eta \delta_{hq} x_j \qquad (2.3.21)$$

$$\delta_{hq} = -\frac{\partial J}{\partial net_q} = -\frac{\partial J}{\partial z_q}\frac{\partial z_q}{\partial net_q} = \frac{\partial a(net_q)}{\partial net_q}\sum_{i=1}^{p} [d_i - y_i]\frac{\partial a(net_i)}{\partial net_i} w_{iq}$$

$$\delta_{hq} = \frac{\partial a(net_q)}{\partial net_q} \sum_{i=1}^{p} \delta_{oi} w_{iq} \tag{2.3.22}$$

Ví dụ : trường hợp hàm tác động a() là hàm sigmoid lưỡng cực (2.1.8), ta có

$$\frac{\partial a(net)}{\partial net} = \frac{1}{2}\left(1 - a^2(net)\right)$$

(2.3.20) và (2.3.21) trở thành

$$\delta_{oi} = \frac{1}{2}(1 - y_i^2)(d_i - y_i) \tag{2.3.23}$$

$$\delta_{hq} = \frac{1}{2}(1 - z_q^2)\sum_{i=1}^{p} \delta_{oi} w_{iq} \tag{2.3.24}$$

$$\Rightarrow \quad \Delta w_{iq} = \frac{1}{2}\eta(1 - y_i^2)(d_i - y_i)z_q \tag{2.3.25}$$

$$\Rightarrow \quad \Delta v_{qj} = \frac{1}{2}\eta(1 - z_q^2)\, x_j \sum_{i=1}^{p} \delta_{oi} w_{iq} \tag{2.3.26}$$

Luật học lan truyền ngược (back propagation learning rule)

Bước 0 : Khởi động trị

Chọn hằng số học η, ngưỡng dừng E_{max}

Các trọng số có trị số nhỏ và ngẫu nhiên

$J = 0, k = 1$

Bước 1 : Lan truyền thuận dữ liệu

Ap dụng mẫu dữ liệu thứ k.

Xác định tín hiệu tại các nút trong mạng

Xác định hàm mục tiêu J

Xác định tín hiệu học tại các nút ra theo (2.3.20)

$$\delta_{oi} = (d_i - y_i)\frac{\partial a(net_i)}{\partial net_i}$$

Bước 2 : Lan truyền ngược sai lệch

Cập nhật các vectơ trọng số theo (2.3.19), (2.3.21)

$$\Delta w_{iq} = \eta \delta_{oi} z_q$$

$\Delta v_{qj} \quad = \eta\delta_{hq}x_j$

Xác định các tín hiệu học theo (2.3.22)

$$\delta_{hq} \quad = \frac{\partial a(net_q)}{\partial net_q}\sum_{i=1}^{p}\delta_{oi}w_{iq}$$

Bước 3 : Nếu k < p : k = k + 1, nhảy đến bước 1

Nếu k = p : (kết thúc 1 epoch)

Nếu $J > J_{max}$: k = 1, nhảy đến bước 1

Nếu $J \le J_{max}$: Kết thúc

Ví dụ : Xây dựng giải thuật học cho mạng lan truyền ngược hình 2.16 với hàm tác động là hàm sigmoid đơn cực

$$y = a(net) = \frac{1}{1+e^{-net}} \Rightarrow \frac{\partial a(net)}{\partial net} = y(1-y)$$

Từ (2.3.7)

$\delta_9 = y_9(1-y_9)(d - y_9)$

Từ (2.3.9)

$\delta_6 = y_6(1-y_6)w_{96}\delta_9$

$\delta_7 = y_7(1-y_7)w_{97}\delta_9$

$\delta_3 = y_3(1-y_3)[w_{63}\delta_6+w_{73}\delta_7]$

$\delta_4 = y_4(1-y_4)[w_{64}\delta_6+w_{74}\delta_7]$

Từ (2.3.6)

$\Delta w_{96} = \eta\delta_9 y_6$ $\quad \Delta w_{97} = \eta\delta_9 y_7$ $\quad \Delta w_{98} = \eta\delta_9 y_8 = -\eta\delta_9$

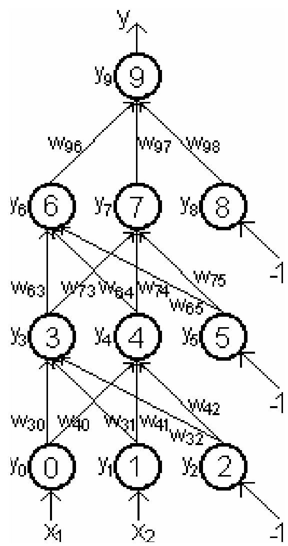
Từ (2.3.8)

$\Delta v_{63} = \eta\delta_6 y_3$ $\quad \Delta v_{64} = \eta\delta_6 y_4$ $\quad \Delta v_{65} = \eta\delta_6 y_5 = -\eta\delta_6$

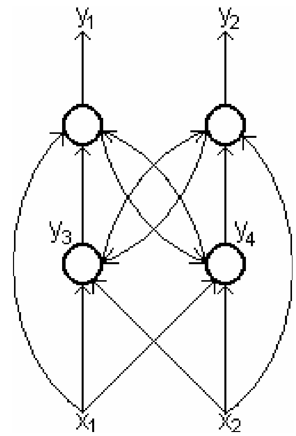
$\Delta v_{73} = \eta\delta_7 y_3$ $\quad \Delta v_{74} = \eta\delta_7 y_4$ $\quad \Delta v_{75} = \eta\delta_7 y_5 = -\eta\delta_7$

$\Delta v_{30} = \eta\delta_3 y_0$ $\quad \Delta v_{31} = \eta\delta_3 y_1$ $\quad \Delta v_{32} = \eta\delta_3 y_2 = -\eta\delta_3$

$\Delta v_{40} = \eta\delta_4 y_0$ $\quad \Delta v_{41} = \eta\delta_4 y_1$ $\quad \Delta v_{42} = \eta\delta_4 y_2 = -\eta\delta_4$



Hình 2.16



Hình 2.17

2.3.3 Các thông số của luật học lan truyền ngược

Việc huấn luyện mạng chịu tác động bởi các yếu tố sau

a) **Trị khởi động của các trọng số**: Các trọng số nên khởi động với các giá trị bé và ngẫu nhiên. Các giá trị lớn của vectơ trọng số có thể làm các hàm tác động bảo hòa sâu khi bắt đầu học và quá trình học kéo dài.

b) **Hằng số học**: η lớn $\rightarrow$ hội tụ nhanh nhưng có thể gây vọt lố. Có thể chọn η giảm dần. Nếu hàm mục tiêu giảm chậm $\rightarrow$ tăng η. Nếu hàm mục tiêu dao động $\rightarrow$ giảm η

c) **Hàm mục tiêu**: Có thể dùng hàm khả vi bất kỳ.

d) **Luật học**

e) **Dữ liệu huấn luyện**: Dữ liệu huấn luyện phải bao quát các khả năng có thể xảy ra.

f) **Số lớp ẩn và số neuron ở mỗi lớp ẩn**: Thử sai trên nguyên tắc ưu tiên cho mạng đơn giản nhất đáp ứng được yêu cầu sử dụng.

2.4 MẠNG HỒI QUY (recurrent neural networks, feedback networks)

Có thể có hồi tiếp giữa các neuron hoặc tự hồi tiếp ở mỗi neuron. Trong mạng hồi quy có thể xảy ra quá trình tự dao động (limit cycles). Mạng hồi quy thích hợp với các ứng dụng dự báo các tín hiệu ngẫu nhiên (time series prediction).

2.4.1 Feedback Back-propagation network

Tác động tại nút i thỏa phương trình vi phân cấp 1

$$\tau \dot{y}_i + y_i = a(net_i) = a(\sum_j w_{ij} y_j + x_i) \qquad (2.4.1)$$

τ : relaxation time scale. Điểm cân bằng (fixed point) $\dot{y}_i = 0$

$$\Rightarrow \quad y_i = a(net_i) = a(\sum_j w_{ij} y_j + x_i) \tag{2.4.2}$$

với $net_i = \sum_j w_{ij} y_j + x_i$: net input to node i

Hàm mục tiêu

$$J = \frac{1}{2} \sum_{k=1}^{n} J_k = \frac{1}{2} \sum_{k=1}^{n} (d_k - y_k)^2 \tag{2.4.3}$$

với n là số output. Giải thuật steepest-descent

$$\Delta w_{pq}^{T} = -\eta \frac{\partial J}{\partial w_{pq}} = -\eta \sum_{k=1}^{n} \frac{\partial J_k}{\partial y_k} \frac{\partial y_k}{\partial w_{pq}} = \eta \sum_{k=1}^{n} (d_k - y_k) \frac{\partial y_k}{\partial w_{pq}} \tag{2.4.4}$$

$$\text{với} \quad \frac{\partial y_k}{\partial w_{pq}} = \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial w_{pq}} = a'(net_k)[\delta_{kp} y_q + \sum_j w_{kj} \frac{\partial y_j}{\partial w_{pq}}] \tag{2.4.5}$$

($\delta_{kp} = 1$ nếu k = p và $\delta_{kp} = 0$ nếu $k \neq p$)

$$\Rightarrow \quad \frac{\partial y_k}{\partial w_{pq}} - a'(net_k) \sum_j w_{kj} \frac{\partial y_j}{\partial w_{pq}} = \sum_j \left(\delta_{kj} - a'(net_k) w_{kj}\right) \frac{\partial y_j}{\partial w_{pq}} = \delta_{kp} a'(net_k) y_q$$

$$\Rightarrow \quad \sum_j L_{kj} \frac{\partial y_j}{\partial w_{pq}} = \delta_{kp} a'(net_k) y_q, \qquad k = 1 \ldots n \tag{2.4.6}$$

$$\text{với} \quad L_{kj} = \delta_{kj} - a'(net_k) w_{kj} \tag{2.4.7}$$

$$\Rightarrow \quad \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial w_{pq}} \\ \frac{\partial y_2}{\partial w_{pq}} \\ \vdots \\ \frac{\partial y_n}{\partial w_{pq}} \end{bmatrix} = [\delta_{kp} a'(net_k) y_q]$$

(vectơ cột $[\delta_{kp} a'(net_k) y_q]$ chỉ có 1 phần tử khác 0 ở hàng thứ k = p : $a'(net_p) y_q$)

$$\Rightarrow \quad L \left[\frac{\partial y}{\partial w_{pq}} \right]^T = [\delta_{kp} a'(net_k) y_q]$$

$$\Rightarrow \quad \left[\frac{\partial y}{\partial w_{pq}} \right]^T = L^{-1} [\delta_{kp} a'(net_k) y_q]$$

$$\Rightarrow \quad \frac{\partial y_k}{\partial w_{pq}} = \sum_j [L^{-1}]_{kj}\,[\delta_{jp} a'(net_j) y_q] = [L^{-1}]_{kp} a'(net_p) y_q \tag{2.4.8}$$

$$\Rightarrow \quad \Delta w_{pq} = \eta \sum_{k=1}^{n} (d_k - y_k) \frac{\partial y_k}{\partial w_{pq}} = \eta a'(net_p) \sum_{k=1}^{n} (d_k - y_k)\,[L^{-1}]_{kp} y_q$$

$$\Rightarrow \quad \Delta w_{pq} = \eta \delta_p y_q \tag{2.4.9}$$

với $\delta_p = a'(net_p) \sum_{k=1}^{n} e_k \,[L^{-1}]_{kp}$ (2.4.10)

$$e_k = d_k - y_k \tag{2.4.11}$$

Để tránh nghịch đảo ma trận, đặt

$$z_p = \sum_{k=1}^{n} e_k\,[L^{-1}]_{kp} \;\Rightarrow\; \sum_{p=1}^{n} L_{pi} z_p = e_p \tag{2.4.12}$$

$$\delta_p = a'(net_p) z_p \tag{2.4.13}$$

Từ (2.4.7)

$$z_i - \sum_{p=1}^{n} a'(net_p)\, w_{pi} z_p = e_i \tag{2.4.14}$$

(2.4.14) chính là phương trình của fixed point của phương trình vi phân có cùng dạng với (2.4.1)

$$\tau \dot{z}_i + z_i = \sum_{p=1}^{n} a'(net_p)\, w_{pi} z_p + e_i \tag{2.4.15}$$

Do đó giá trị của z_p có thể được xác định bởi error propagation network.

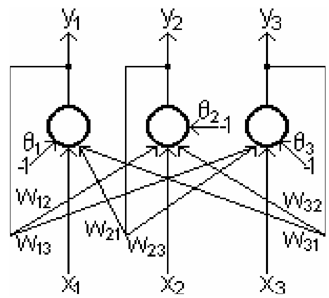
2.5 MẠNG HOPFIELD

2.5.1 Mạng Hopfield rời rạc

Mạng Hopfield là mạng hồi tiếp 1 lớp (single-layer feedback network) với các đặc tính sau

- Mỗi nút có một tín hiệu vào từ bên ngoài (external input) x_i và một ngưỡng θ_i. Tín hiệu vào chỉ tác động 1 lần khi khởi động.
- Không có tự hồi tiếp : $w_{ii} = 0, \forall i$.
- Đối xứng : $w_{ij} = w_{ji}, \forall i, j$.
- Cập nhật không đồng bộ (asynchronous) : mỗi thời điểm chỉ cập nhật 1 nút

$$y_i(k+1) = \mathrm{sign}\Big(\sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} y_j(k) + x_i - \theta_i \Big) \tag{2.5.1}$$



Hình 2.18: Mạng Hopfield 3 neuron

Hình 2.18 trình bày mạng Hopfield với 3 neuron.

Ví dụ: Mạng Hopfield với $w_{12} = w_{21} = -1$, $w_{11} = w_{22} = 0$, $\theta_1 = \theta_2 = 0$ (hình 2.19). Luật cập nhật (bất đồng bộ) :

$$y_1(k+1) = \text{sign}(w_{12}y_2(k) + x_1(k))$$

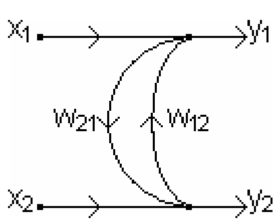
$$y_2(k+2) = \text{sign}(w_{21}y_1(k+1) + x_2(k))$$

Với : $x_1(0) = x_2(0) = 0$. Khởi động trị : $y_1(0) = y_2(0) = -1$

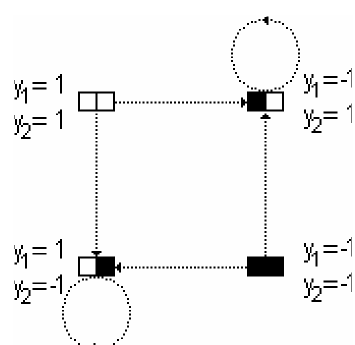
$\Rightarrow \quad y_1(1) = \text{sign}(w_{12}y_2(0) + x_1(0)) = 1$

$\quad\quad y_2(2) = \text{sign}(w_{21}y_1(1) + x_2(0)) = -1$

Hình 2.20 trình bày sơ đồ chuyển trạng thái của mạng với các thứ tự cập nhật khác nhau. Ta thấy mạng có 2 điểm cân bằng bổ phụ nhau $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ và $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$.



Hình 2.19 : Mạng Hopfield



Hình 2.20 : Sơ đồ chuyển trạng thái

Ghi chú

- Tại điểm cân bằng, hàm năng lượng

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}y_iy_j - \sum_{i=1}^{n} x_iy_i + \sum_{i=1}^{n}\theta_iy_i$$

  đạt giá trị cực tiểu.

- Với cập nhật đồng bộ ta có

$$\begin{bmatrix} y_1(1) \\ y_2(1) \end{bmatrix} = \begin{bmatrix} \text{sign}(w_{12}y_2(0) + x_1(0)) \\ \text{sign}(w_{21}y_1(0) + x_2(0)) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

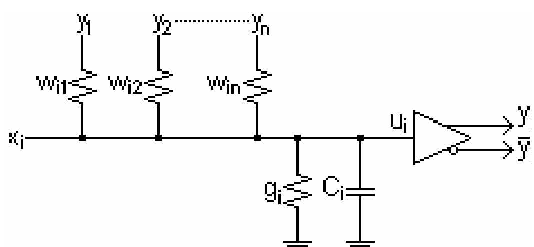
$$\begin{bmatrix} y_1(2) \\ y_2(2) \end{bmatrix} = \begin{bmatrix} \text{sign}(w_{12}y_2(1) + x_1(1)) \\ \text{sign}(w_{21}y_1(1) + x_2(1)) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

2.5.2 Mạng Hopfield liên tục

Định luật Kirchhoff 1

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij}\left(y_j - u_i\right) + x_i - g_i u_i - C_i \frac{du_i}{dt} = 0$$

Quan hệ vào ra của khuếch đại thuật toán có thể xem như hàm tác động của neuron.



Hình 2.21: Thực hiện một nút của mạng Hopfield liên tục dùng khuếch đại thuật toán

(y_j, u_i : điện thế, x_i : cường độ dòng điện, w_{ij}, g_i : điện dẫn, C_i : điện dung)

2.5.3 Mạng kết hợp (associative network)

1) Các loại mạng kết hợp : Ta phân biệt 3 loại mạng kết hợp

- Mạng liên kết chéo (hetero-associative networks)

$\Phi(x^i) = y^i$, $\forall i = 1, 2, \ldots , N$ (2.5.2)

Nếu tín hiệu vào 'gần' x^i thì tín hiệu ra là y^i.

Ứng dụng tiêu biểu : lưu trữ ảnh.

Luật học Hebbian : để liên kết các vectơ $x^1, x^2, \ldots, x^N$ với các vectơ $y^1, y^2, \ldots, y^N$ có thể định nghĩa bộ liên kết tuyến tính (linear associator) $\Phi(x)$ như sau

$\Phi(x) = Wx$ (2.5.3)

với

$$W = YX^{+} \tag{2.5.4}$$

$$Y = [y^1,\ y^2,\ \dots,\ y^N],\quad X = [x^1,\ x^2,\ \dots,\ x^N],\quad X^{+} = (X^TX)^{-1}X^T$$

Nếu x^i là các vectơ trực chuẩn, $X^TX = I$

$$W = YX^T = \sum_{i=1}^{N} y^i \left(x^i\right)^T \tag{2.5.5}$$

W được gọi là ma trận tương quan chéo (cross-correlation matrix).

- Mạng tự liên kết (auto-associative networks)

$$\Phi(x^i) = x^i,\ \ \forall i = 1, 2, \dots, N \tag{2.5.6}$$

Nếu tín hiệu vào ‘gần‘ x^i thì tín hiệu ra là x^i.

Ứng dụng tiêu biểu : chỉnh sửa các vectơ input bị nhiễu.

- Pattern recognition networks

$$\Phi(x^i) = i,\ \ \forall i = 1, 2, \dots, N \tag{2.5.7}$$

Ứng dụng tiêu biểu : nhận dạng, phân loại.

2) Độ đo khoảng cách giữa 2 vectơ:

- Khỏang cách Euclide của vectơ (mx1) x và x’

$$\sqrt{\sum_{k=1}^{m}(x_k - x'_k)^2} \tag{2.5.8}$$

- Khỏang cách Hamming : số phần tử tương ứng khác nhau của hai vectơ (mx1) x và x’

$$HD(x,x') = \begin{cases} \sum_{k=1}^{m} |x_k - x'_k|, & x_k, x'_k \in \{0,1\} \\ \frac{1}{2}\sum_{k=1}^{m} |x_k - x'_k|, & x_k, x'_k \in \{-1,1\} \end{cases} \tag{2.5.9}$$

2.5.4 Bộ nhớ kết hợp đệ quy của Hopfield

Luật cập nhật (2.5.1) được gọi là luật phục hồi dữ liệu (data retrieval rule)

$$y_i(k+1) = \mathrm{sign}\Big(\sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ij}y_j(k) + x_i - \theta_i\Big)$$

và được thực hiện một cách bất đồng bộ và ngẫu nhiên.

Giải thuật lưu trữ (storage algorithm) :

Nếu $x_i \in \{-1,1\}$ thì $W = \sum_{i=1}^{N} x^i \left(x^i\right)^T - NI$ (2.5.10)

Nếu $x_i \in \{0,1\}$ thì $w_{pq} = \sum_{i=1}^{N} (2x_p^i - 1)(2x_q^i - 1)$ nếu $p \neq q$ và $w_{pp} = 0$ (2.5.11)

Ví dụ: dùng bộ nhớ Hopfield để lưu trữ N = 2 vectơ $x^1 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$ và $x^2 = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$

Ma trận trọng số

$$W = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} [1, 1, -1] + \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} [-1, 1, 1] - 2I = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

2.5.5 Bộ nhớ kết hợp hai chiều (bidirectional associative memory BAM)

Bộ nhớ kết hợp hai chiều

$y = a(Wx)$ (2.5.12)

$x = a(W^T y)$ (2.5.13)

Luật cập nhật có thể đồng bộ hoặc bất đồng bộ.

$y^{(1)} = a(Wx^{(0)})$ first forward pass

$x^{(2)} = a(W^T y^{(1)})$ first backward pass

$y^{(3)} = a(Wx^{(2)})$ second forward pass

$x^{(4)} = a(W^T y^{(3)})$ second backward pass

……………………………

Giải thuật lưu trữ (storage algorithm, outer-product learning rule) :

Nếu $x_i \in \{-1,1\}$ thì $W = \sum_{i=1}^{N} y^i \left(x^i\right)^T$ (2.5.14)

Nếu $x_i \in \{0,1\}$ thì $w_{pq} = \sum_{i=1}^{N}(2x_p^i - 1)(2x_q^i - 1)$ nếu $p \neq q$ và $w_{pp} = 0$ (2.5.15)

2.6 CÁC GIẢI THUẬT HỌC KHÔNG GIÁM SÁT (unsupervised learning rules)

Đặc điểm của các giải thuật học không giám sát là không có tín hiệu ra mong muốn. Mạng phải tự khám phá các đặc trưng của dữ liệu nhập và diễn dịch thành tín hiệu ra của mạng. Các mạng nầy được gọi là mạng tự tổ chức (self-organizing networks).

Xét một mạng 1 lớp với m tín hiệu vào x, n tín hiệu ra y, các trọng số từ tín hiệu vào thứ j đến tín hiệu ra thứ i là w_{ij}. Gọi s() là một hàm đơn điệu, không giảm và bị chặn.

2.6.1 Luật học Hebbian

$$\dot{w}_{ij} = -w_{ij} + s_i(y_i)s_j(x_j) \qquad (2.6.1)$$

Luật học (2.6.1) dùng để đo độ giống nhau (familiarity) hoặc chiếu vào các thành phần chính của dữ liệu nhập (projecting onto the principal components of the input data). Luật học Hebbian có các biến thể sau

a) Luật học của Sanger

$$\dot{w}_{ij} = \eta y_i\left(x_j - \sum_{k=1}^{i} y_k w_{kj}\right) \qquad (2.6.2)$$

Luật học (2.6.2) dùng trong mạng thành phần chính (principal component network), để trích n thành phần chính đầu tiên của dữ liệu nhập (the first n principal components of the input data) trong đó

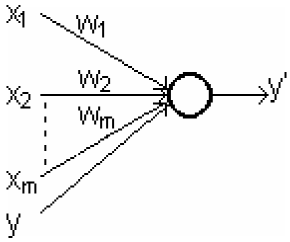
$$y_i = \sum_{j=1}^{m} w_{ij}x_j \qquad (2.6.3)$$

là tín hiệu ra của một mạng ADALINE. Người ta chứng minh được rằng w_i hội tụ đến vectơ riêng thứ i chuẩn hoá của ma trận tương quan $C = E(xx^T)$ tương ứng với trị riêng lớn thứ i (the i-th normalized eigenvector of the correlation matrix $C = E(xx^T)$ belonging to the i-th largest eigenvalue).

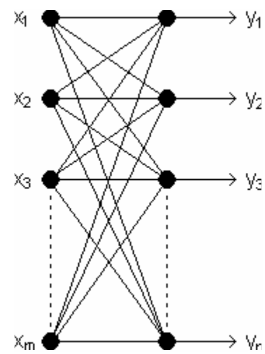
b) Luật học của Grossberg

$$\dot{w}_i = \eta(yx_i - w_i)u(x_i) \qquad (2.6.4)$$

với $0 < \eta < 1$, u() là bước nhảy đơn vị. Ta thấy các thông số w_i chỉ thay đổi khi $x_i > 0$. Luật học được sử dụng trong cấu trúc instar.



Hình 2.22



Hình 2.23 : Mạng Kohonen (winner-take-all network)

2.6.2 Luật học cạnh tranh (competitive learning rule) được mô tả như sau

$$\dot{w}_{ij} = s_i(y_i)\left(s_j(x_j) - w_{ij}\right) \quad (2.6.5)$$

với

$$s_i(y_i) = \frac{1}{1+e^{-cy_i}}, \ \ c > 0 \quad (2.6.6)$$

Luật học nầy điều chế tín hiệu sai lệch $[s_j(x_j) - w_{ij}]$ với tín hiệu cạnh tranh $s_i(y_i)$. Với c lớn, tín hiệu $s_i(y_i)$ có thể xấp xỉ chỉ số thắng - thua nhị phân. Tại mỗi thời điểm, ta nói neuron thứ i thắng cuộc nếu $s_i(y_i) = 1$ và thua cuộc nếu $s_i(y_i) = 0$ (learn if win). Các nút ra tranh nhau tác động và thường được gọi là winner-take-all nodes. Luật học cạnh tranh có các dạng cụ thể sau

a) Luật học cạnh tranh tuyến tính (linear competitive learning rule): Nếu ta chọn

$$s_j(x_j) = x_j \quad (2.6.7)$$

luật học (2.6.5) trở thành luật học cạnh tranh tuyến tính

$$\dot{w}_{ij} = s_i(y_i)\left(x_j - w_{ij}\right) \quad (2.6.8)$$

b) Luật học Kohonen (winner-take-all learning rule) : Mạng Kohonen (Hình 2.23)

$$y_i = a(\,w_i^T x\,) \quad (2.6.9)$$

Luật học Kohonen (winner-take-all) là một dạng đặc biệt của (2.6.8). Nó dựa vào việc phân dữ liệu nhập thành các nhóm chứa các phần tử tương tự nhau. Luật học gồm 2 bước

- Similarity matching

$$\| x - \hat{w}_i^{(k)} \| = \min_{1 \le j \le n} \| x - \hat{w}_j^{(k)} \| \quad (2.6.10)$$

- Updating (if neuron i wins)

$$\hat{w}_i^{(k+1)} = \hat{w}_i^{(k)} + \alpha^{(k)}[x - \hat{w}_i^{(k)}] \quad (2.6.11)$$

$$\hat{w}_j^{(k+1)} = \hat{w}_j^{(k)} \text{ for } j \neq i \tag{2.6.12}$$

với $\alpha^{(k)}$ là hằng số học ở bước thứ k, $\hat{w}_i = \dfrac{w_i}{\| w_i \|}$ là vectơ chuẩn hoá của w_i.

Ghi chú

- Hàm tác động không liên quan đến quá trình học.
- Quá trình học sẽ dừng khi bộ trọng số thay đổi không đáng kể.
- $\| x - \hat{w}_j^{(k)} \|^2 = \|x\|^2 + \| \hat{w}_j^{(k)} \|^2 - 2 x^T \hat{w}_j^{(k)} = \|x\|^2 + 1 - 2net_j$
- $\| x - \hat{w}_i^{(k)} \|^2$ min $\Leftrightarrow$ net_i max : neuron i thắng cuộc nếu tín hiệu vào x tương quan nhiều nhất với $\hat{w}_i^{(k)}$, hay nếu x 'song song' với $\hat{w}_i^{(k)}$ hơn với các vectơ khác $\hat{w}_j^{(k)}$. Kết quả của (2.6.11) làm quay vectơ trọng số của neuron thắng cuộc $\hat{w}_i^{(k)}$ về phía vectơ vào.

2.6.3 Luật học Hebbian vi sai (differential Hebbian)

$$\dot{w}_{ij} = -w_{ij} + s_i(y_i)s_j(x_j) + \dot{s}_i(y_i)\dot{s}_j(x_j) \tag{2.6.13}$$

với

$$\dot{s}_i(y_i) = \frac{ds_i}{dt} = \frac{\partial s_i}{\partial y_i}\dot{y}_i \tag{2.6.14}$$

Luật học nầy có các biến thể sau

a) Luật học Hebbian vi sai đơn giản nhất

$$\dot{w}_{ij} = -w_{ij} + \dot{s}_i(y_i)\dot{s}_j(x_j) \tag{2.6.15}$$

b) Luật học Kosko-Klopf

$$\dot{w}_{ij} = -aw_{ij} + [by_i x_j - cw_{ij}]u(\dot{y}_i)u(\dot{x}_j) \tag{2.6.16}$$

với u() : bước nhảy đơn vị, a, b, c là các hằng số dương với a << c.

2.6.4 Luật học cạnh tranh vi sai (differential competitive) kết hợp luật học cạnh tranh và luật học Hebbian vi sai

$$\dot{w}_{ij} = \dot{s}_i(y_i)\left(s_j(x_j) - w_{ij}\right) \tag{2.6.17}$$

(chỉ học khi có sự thay đổi). Các biến thể của luật học cạnh tranh vi sai

a) Luật học cạnh tranh vi sai tuyến tính (linear differential competitive learning rule)

$$\dot{w}_{ij} = \dot{s}_i(y_i)\left(x_j - w_{ij}\right) \tag{2.6.18}$$

Luật học nầy thường dùng trong nhận dạng (pattern recognition) và ước lượng hàm xác suất (probability function).

b) Luật học cạnh tranh vi sai rời rạc (discrete differential competitive learning rule)

$$w_{ij}(k+1) = w_{ij}(k) + \Delta s_i(y_i(k))\left(x_j(k) - w_{ij}(k)\right) \tag{2.6.19}$$

Luật học nầy thường dùng trong ước lượng điểm trọng tâm (centroid estimation) và nhận dạng âm vị (phoneme recognition).

2.7 SELF-ORGANIZING FEATURE MAPS

Feature mapping chuyển đổi các mẫu với số chiều bất kì (trong không gian mẫu) thành tín hiệu ra của các mảng neuron 1 hoặc 2 chiều (trong không gian đặc trưng). Bên cạnh việc giảm số chiều, một mục tiêu chính của feature mapping là bảo toàn mối quan hệ lân cận trong không gian mẫu (hai mẫu lân cận nhau trong không gian mẫu có các tín hiệu ra gần nhau).

Luật học của Kohonen feature map như sau

$$\text{similarity matching : } \| x - \hat{w}_i^{(k)} \| = \min_{1 \le j \le n} \| x - \hat{w}_j^{(k)} \| \tag{2.6.20}$$

(neuron i wins)

$$\text{updating : } \quad \hat{w}_{ij}^{(k+1)} = \begin{cases} \hat{w}_{ij}^{(k)} + \alpha^{(k)}[x_k - \hat{w}_i^{(k)}] & \text{for } i \in N_{i*}^{(k)} \\ \hat{w}_{ij}^{(k)} & \text{otherwise} \end{cases} \tag{2.6.21}$$

trong đó $N_{i*}^{(k)}$ là tập lân cận của winner node i ở thời điểm k. Với luật học nầy, các nút lân cận nhau được cập nhật tương tự nhau và do đó sẽ có đáp ứng tương tự nhau với các mẫu gần nhau.

2.8 HƯỚNG DẪN SỬ DỤNG NEURAL NETWORKS TOOLBOX

2.8.1 Ví dụ

```
load mang.mat XX
TT    = XX(   1,:);        %    time
UU    = XX(2:4,:);         %    input
YY    = XX(   5,:);        %    output
subplot(211), plot(TT, UU(1,:),'b'), hold on, plot(TT, UU(2,:),'b'), plot(TT, UU(3,:),'b')
subplot(312), plot(Time,YY,'b')
%     NET
PR    = [-1, 1; -0.5 0.5; -5 2];          %    min – max values of 3 inputs
```

```
MyNet = newff(PR, [4,1], {'tansig' 'purelin'});
MyNet.trainParam.epochs = 400;
MyNet.trainParam.goal = 0.00001;
MyNet.adaptParam.passes = 10;
MyNet = train(MyNet,UU,YY);
Tsamp = 0.001;                          % sampling time
gensim(MyNet,Tsamp)
```

2.8.2 Neural Network Toolbox Version 2.0.1 (R11) 01-Jul-1998

a) Learning functions

learncon - Conscience bias learning function.
learngd - Gradient descent weight/bias learning function.
learngdm - Gradient descent w/momentum weight/bias learning function.
learnh - Hebb weight learning function.
learnhd - Hebb with decay weight learning function.
learnis - Instar weight learning function.
learnk - Kohonen weight learning function.
learnlv1 - LVQ1 weight learning function.
learnlv2 - LVQ2 weight learning function.
learnos - Outstar weight learning function.
learnp - Perceptron weight/bias learning function.
learnpn - Normalized perceptron weight/bias learning function.
learnsom - Self-organizing map weight learning function.
learnwh - Widrow-Hoff weight/bias learning rule.

b) <u>New networks</u>

network - Create a custom neural network.
newc - Create a competitive layer.
newcf - Create a cascade-forward backpropagation network.
newelm - Create an Elman backpropagation network.
newff - Create a feed-forward backpropagation network.
newfftd - Create a feed-forward input-delay backprop network.

newgrnn - Design a generalized regression neural network.

newhop - Create a Hopfield recurrent network.

newlin - Create a linear layer.

newlind - Design a linear layer.

newlvq - Create a learning vector quantization network.

newp - Create a perceptron.

newpnn - Design a probabilistic neural network.

newrb - Design a radial basis network.

newrbe - Design an exact radial basis network.

newsom - Create a self-organizing map.

c) <u>Simulink support</u>

gensim - Generate a SIMULINK block to simulate a neural network.

d) <u>Training functions</u>

trainbfg - BFGS quasi-Newton backpropagation.

trainbr - Bayesian regularization.

traincgb - Powell-Beale conjugate gradient backpropagation.

traincgf - Fletcher-Powell conjugate gradient backpropagation.

traincgp - Polak-Ribiere conjugate gradient backpropagation.

traingd - Gradient descent backpropagation.

traingdm - Gradient descent with momentum backpropagation.

traingda - Gradient descent with adaptive lr backpropagation.

traingdx - Gradient descent w/momentum & adaptive lr backpropagation.

trainlm - Levenberg-Marquardt backpropagation.

trainoss - One step secant backpropagation.

trainrp - Resilient backpropagation (Rprop).

trainscg - Scaled conjugate gradient backpropagation.

trainwb - By-weight-and-bias network training function.

trainwb1 - By-weight-&-bias 1-vector-at-a-time training function.

e) <u>Transfer functions</u>

compet - Competitive transfer function.

hardlim - Hard limit transfer function.

hardlims - Symmetric hard limit transfer function.

logsig - Log sigmoid transfer function. % logsig(n) = 1 / (1 + e^{-n})

poslin - Positive linear transfer function.

purelin - Linear transfer function.

radbas - Radial basis transfer function.

satlin - Saturating linear transfer function.

satlins - Symmetric saturating linear transfer function.

softmax - Soft max transfer function.

tansig - Hyperbolic tangent sigmoid transfer function. % tansig(n) = 2/(1+e^{-2n}) -1

tribas - Triangular basis transfer function.

f) <u>Using networks</u>

sim - Simulate a neural network.

init - Initialize a neural network. % MyNet = init(MyNet)

adapt - Allow a neural network to adapt.

train - Train a neural network.

disp - Display a neural network's properties.

display - Display the name and properties of a neural network variable.

Bài tập

2.1 Xác định perceptron thực hiện phép toán logic sau

a) NOT(x), với $x \in \{0, 1\}$

b) OR(x_1, x_2), với x_1 và $x_2 \in \{0, 1\}$

c) AND(x_1, x_2), với x_1 và $x_2 \in \{0, 1\}$

2.2 Giải thích tại sao không thể thực hiện phép XOR với 1 perceptron.

2.3 Cho 1 perceptron với 1 neuron LGU, 3 tín hiệu vào, hàm tác động là hàm sigmoid đơn cực

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

Huấn luyện mạng neuron theo luật học delta (2.2.13) với các mẫu dữ liệu huấn luyện

$x(1) = [\ 1, -2\ \ ,\ 0\ \ ,\ -1]^T$, $\quad d(1) = 0$

$x(2) = [\ 0,\ 1.5, -0.5, -1]^T$, $\quad d(2) = 0$

$x(3) = [-1,\ 1\ \ , 0.5\ , -1]^T$, $\quad d(3) = 1$

trị khởi động $w(1) = [1, -1, 0, 0.5]^T$, và hằng số học $\eta = 0.1$. Xác định các giá trị của vectơ thông số w(k) trong epoch huấn luyện đầu tiên.

2.4 Xây dựng và vẽ mạng perceptron LTU để phân loại các nhóm dữ liệu sau

a) Nhóm 1: $\{[3, 2]^T, [1, -2]^T, [-2, 0]^T\}$, $\quad d = 1$

Nhóm 2: $\{[2, 1]^T, [0, -1]^T, [-1, 2]^T$, $\quad d = -1$

b) Nhóm 1: $\{[0, 0]^T, [1, 0]^T, [0, 1]^T\}$, $\quad d = 1$

Nhóm 2: $\{[0.5, 0.5]^T, [0.3, 0.3]^T\}$, $\quad d = -1$

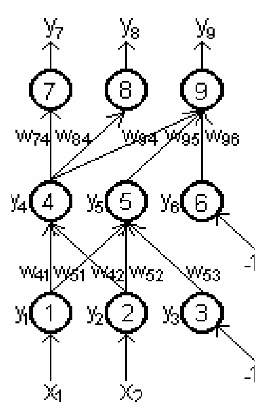
c) Nhóm 1: $\{[0, 0]^T, [1, 0]^T, [0, 1]^T\}$, $\quad d = 1$

Nhóm 2: $\{[0.5, 0]^T, [0, 0.5]^T\}$, $\quad d = -1$

2.5 Xác định luật cập nhật các trọng số của mạng 3 lớp lan truyền ngược hình P2.1 với hằng số học $\eta = 0.2$ trong các trường hợp sau

a) Hàm tác động của tất cả các nút là hàm sigmoid đơn cực (2.1.7) với $\lambda = 2$.

b) Hàm tác động của tất cả các nút là hàm sigmoid lưỡng cực (2.1.8) với $\lambda = 2$.



Hình P2.1

2.6 Huấn luyện LTU theo các dữ liệu cho ở bảng P2.1.

Bảng P2.1

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

2.7 Xác định giải thuật học có giám sát mạng truyền thẳng hai lớp (1 lớp ẩn, 1 lớp ra) với hai tín hiệu vào, một tín hiệu ra, lớp ẩn có 3 neuron, lớp ra có 1 neuron. Các neuron có hàm tích hợp tuyến tính

a) các neuron có hàm tác động tuyến tính

b) các neuron có hàm tác động tansig $a(net) = \dfrac{2}{1+e^{-net}} - 1$

c) các neuron ở lớp ẩn có hàm tác động tansig $a(net) = \dfrac{2}{1+e^{-net}} - 1$, neuron ở lớp ra có hàm tác động tuyến tính

Biết hàm mục tiêu $J = \dfrac{1}{2}(d-y)^2$.

2.8 Xác định giải thuật học có giám sát mạng truyền thẳng ba lớp (2 lớp ẩn, 1 lớp ra) với hai tín hiệu vào, một tín hiệu ra, mỗi lớp ẩn có 3 neuron, lớp ra có 1 neuron. Các neuron có hàm tích hợp tuyến tính

a) các neuron có hàm tác động tuyến tính

b) các neuron có hàm tác động tansig $a(net) = \dfrac{2}{1+e^{-net}} - 1$

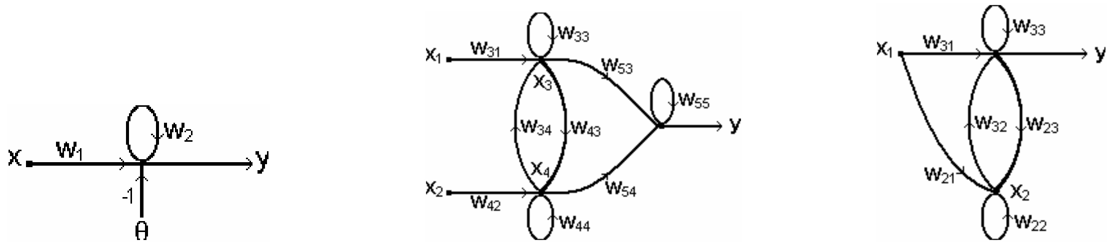
c) các neuron ở các lớp ẩn có hàm tác động tansig $a(net) = \dfrac{2}{1+e^{-net}} - 1$, neuron ở lớp ra có hàm tác động tuyến tính

2.9 Cho mạng neuron hình P2.2. Biết các neuron có hàm tích hợp tuyến tính và hàm tác động $a(net) = \dfrac{2}{1+e^{-net}} - 1$.

a) Viết phương trình xác định tín hiệu y.

b) Gọi d là tín hiệu ra mong muốn. Xác định giải thuật huấn luyện với hàm mục tiêu

$$J = \frac{1}{2}(d-y)^2.$$



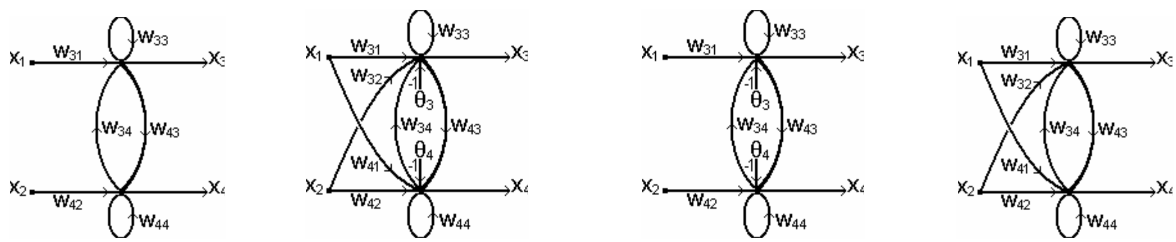
Hình P2.2

2.10 Cho mạng hình P2.3. Các neuron có hàm tích hợp và hàm tác động tuyến tính.

a) Viết phương trình xác định các tín hiệu của các neuron.

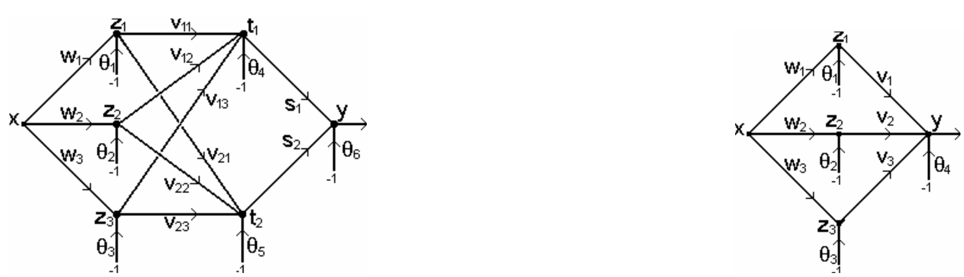
b) Gọi d là tín hiệu ra mong muốn. Xác định giải thuật huấn luyện với hàm mục tiêu

$$J = \frac{1}{2}(d_3 - x_3)^2 + \frac{1}{2}(d_4 - x_4)^2$$



Hình P2.3

2.11 Cho mạng truyền thẳng hình P2.4. Các neuron có hàm tích hợp tuyến tính và hàm tác động a(net) = $\frac{1}{1+e^{-net}}$. Xác định luật học lan truyền ngược dùng phương pháp steepest descent với hàm mục tiêu J = $\frac{1}{2}(d-y)^2$.



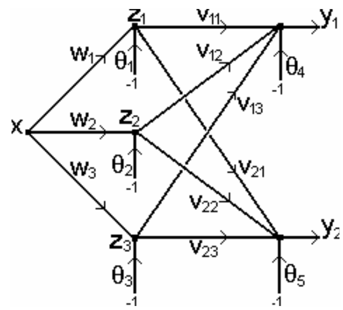
Hình P2.4

2.12 Cho mạng hình P2.5. Các neuron có hàm tích hợp và hàm tác động tuyến tính.

a. Viết phương trình xác định các tín hiệu của các neuron.

b. Gọi d_1 và d_2 là các tín hiệu ra mong muốn. Xác định giải thuật huấn luyện với hàm mục tiêu

$$J = \frac{1}{2}(d_1-y_1)^2 + \frac{1}{2}(d_2-y_2)^2$$



hình P2.5

2.13 Xác định mạng Hopfield (với hàm tác động signum) dùng để lưu trữ các vectơ sau $\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$.

Vẽ giản đồ trạng thái của mạng khi tín hiệu vào bằng $\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$.

2.14 Huấn luyện mạng truyền thẳng để phân nhóm các phần tử của tập $X \subset R^2$ thành hai nhóm. (Tự tạo tập dữ liệu gồm 200 phần tử trong R^2)

2.15 Dùng Simulink mô phỏng hệ thống rời rạc $H(z) = \dfrac{1}{z - 0.9}$ (dùng khối *discrete transfer function* với sample time = 0.01) với tín hiệu vào là ồn trắng (dùng khối *band limited white noise* với sample time = 0.1). Lưu các tín hiệu vào và ra của hệ thống vào tập tin (dùng khối *to file* với sample time = 0.01). Viết m file để đọc dữ liệu trên trên tập tin và huấn luyện mạng truyền thẳng. So sánh tín hiệu ra của hệ thống với tín hiệu ra mô phỏng và dự báo. Nhận xét.

2.16 Xây dựng bộ cân bằng dùng mạng neuron cho kênh thông tin với hàm truyền đạt $H(z) = \dfrac{0.1}{z - 0.9}$.

Chương 3 HỆ THỐNG NEURON - MỜ VÀ MỜ - NEURON

3.1 MỜ-NEURON (neural network - based fuzzy systems)

3.1.1 Ứng dụng mạng neuron trong hệ thống xử lý mờ

- Dùng mạng neuron thực hiện các hàm thành viên của các tập mờ.

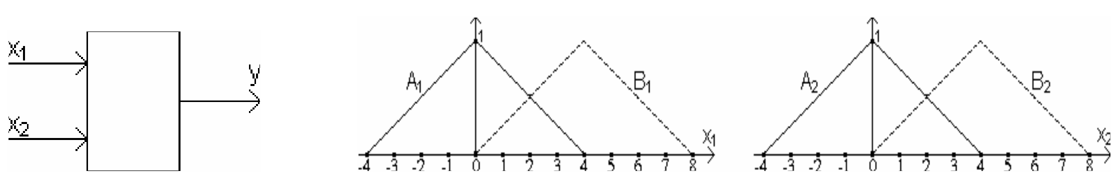
  Ví dụ : Huấn luyện perceptron với hàm tác động $a(net) = e^{-\left(\frac{net - m}{\sigma}\right)^2}$ để thực hiện hàm thành viên của tập mờ.
- Dùng mạng neuron thực hiện các phép toán AND, OR, NOT … trên các tập mờ.
- Dùng mạng neuron thực hiện các luật hợp thành (fuzzy logic inference).
- Fuzzy aggregation networks
- Neural network - driven fuzzy reasoning
- Neural network - based fuzzy modeling

3.1.2 Mạng ANFIS (adaptive network based fuzzy inference system)

Ví dụ 3.1: Cho hệ thống xử lý mờ hình 3.1 với 2 tín hiệu vào x_1, x_2, 1 tín hiệu ra y và các luật hợp thành như sau

R_1 : IF x_1 IS A_1 AND x_2 IS A_2 THEN $y = f_a = a_o + a_1x_1 + a_2x_2$

R_2 : IF x_1 IS B_1 AND x_2 IS B_2 THEN $y = f_b = b_o + b_1x_1 + b_2x_2$



Hình 3.1

Sử dụng luật PROD diễn dịch phép giao và phép suy diễn, luật SUM diễn dịch phép hội, và giải mờ dùng phương pháp trung bình, hàm thành viên của các luật hợp thành được xác định như sau

$$\mu_{R1}(x_1, x_2, y) = \mu_{A1\cap A2}(x_1, x_2).\mu_{fa}(y) = \mu_{A1}(x_1)\ \mu_{A2}(x_2).\mu_{fa}(y) = \mu_1.\mu_{fa}(y)$$

$$\mu_{R2}(x_1, x_2, y) = \mu_{B1\cap B2}(x_1, x_2).\mu_{fb}(y) = \mu_{B1}(x_1)\ \mu_{B2}(x_2).\mu_{fb}(y) = \mu_2.\mu_{fb}(y)$$

và giá trị giải mờ

$$y = \frac{\mu_1 f_a + \mu_2 f_b}{\mu_1 + \mu_2} = \frac{\mu_1}{\mu_1 + \mu_2}(\ a_o + a_1x_1 + a_2x_2) + \frac{\mu_2}{\mu_1 + \mu_2}(\ b_o + b_1x_1 + b_2x_2) = \Phi^T\theta$$

với

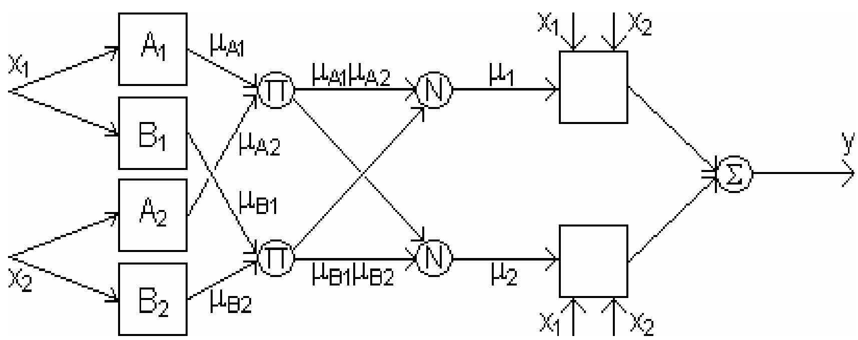
$\mu_1 = \mu_{A1}(x_1)\ \mu_{A2}(x_2)$

$\mu_2 = \mu_{B1}(x_1)\ \mu_{B2}(x_2)$

$$\Phi = \left[\frac{\mu_1 x_1}{\mu_1+\mu_2}, \frac{\mu_1 x_2}{\mu_1+\mu_2}, \frac{\mu_1}{\mu_1+\mu_2}, \frac{\mu_2 x_1}{\mu_1+\mu_2}, \frac{\mu_2 x_2}{\mu_1+\mu_2}, \frac{\mu_2}{\mu_1+\mu_2}\right]^T$$

$\theta = [a_1, a_2, a_o, b_1, b_2, b_o]^T$

Ta có thể biểu diễn hệ thống xử lý mờ nầy dưới dạng mạng, gọi là mạng ANFIS, như hình 3.2.



Hình 3.2 : Mạng ANFIS

Mạng ANFIS gồm 5 lớp

- Lớp 1: xác định giá trị của các hàm thành viên của tín hiệu vào.
- Lớp 2: xác định giá trị của các hàm thành viên của các luật hợp thành.
- Lớp 3: chuẩn hóa.
- Lớp 4: xác định giá trị của hàm thành viên (dạng vạch) của tập mờ của tín hiệu ra.
- Lớp 5: giải mờ (dùng phương pháp trung bình).

Để huấn luyện mạng ta có thể sử dụng hàm mục tiêu sau

$$J = \frac{1}{2}(d - y)^2$$

trong đó d là tín hiệu ra mong muốn (học có giám sát). Ta có giải thuật huấn luyện sau

$$\theta(k+1) = \theta(k) - \eta\nabla J = \theta(k) + \eta(d - y)\Phi$$

Ví dụ 3.2: Cho hệ thống xử lý mờ hình 3.1 với 2 tín hiệu vào x_1, x_2, 1 tín hiệu ra y và các luật hợp thành như sau

R_1 : IF x_1 IS A_1 OR x_2 IS A_2 THEN $y = f_a = a_o + a_1x_1 + a_2x_2$

R_2 : IF x_1 IS B_1 AND x_2 IS B_2 THEN $y = f_b = b_o + b_1x_1 + b_2x_2$

Sử dụng luật PROD diễn dịch phép giao và phép suy diễn, luật SUM diễn dịch phép hội, và giải mờ dùng phương pháp trung bình, hàm thành viên của các luật hợp thành được xác định như sau

$$\mu_{R1}(x_1, x_2, y) = \mu_{A1\cup A2}(x_1, x_2).\mu_{fa}(y) = \min\{1, \mu_{A1}(x_1)+\mu_{A2}(x_2)\}.\mu_{fa}(y) = \mu_1.\mu_{fa}(y)$$

$$\mu_{R2}(x_1, x_2, y) = \mu_{B1\cap B2}(x_1, x_2).\mu_{fb}(y) = \mu_{B1}(x_1)\ \mu_{B2}(x_2).\mu_{fb}(y) = \mu_2.\mu_{fb}(y)$$

và giá trị giải mờ

$$y = \frac{\mu_1 f_a + \mu_2 f_b}{\mu_1 + \mu_2} = \frac{\mu_1}{\mu_1 + \mu_2}(\ a_o + a_1x_1 + a_2x_2) + \frac{\mu_2}{\mu_1 + \mu_2}(\ b_o + b_1x_1 + b_2x_2) = \Phi^T\theta$$

với

$$\mu_1 = \min\{1, \mu_{A1}(x_1)+\mu_{A2}(x_2)\}$$

$$\mu_2 = \mu_{B1}(x_1)\ \mu_{B2}(x_2)$$

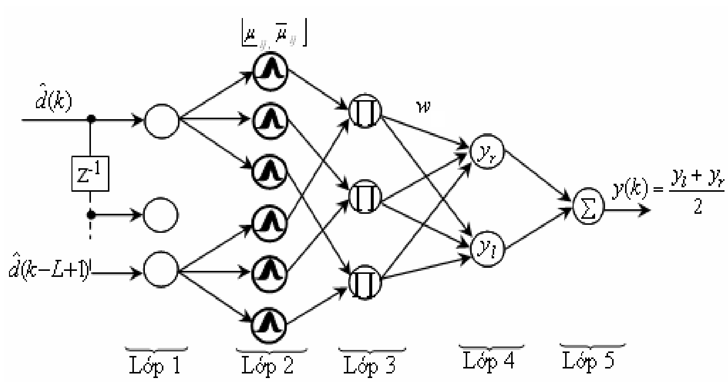
$$\Phi = \left[\frac{\mu_1 x_1}{\mu_1 + \mu_2}, \frac{\mu_1 x_2}{\mu_1 + \mu_2}, \frac{\mu_1}{\mu_1 + \mu_2}, \frac{\mu_2 x_1}{\mu_1 + \mu_2}, \frac{\mu_2 x_2}{\mu_1 + \mu_2}, \frac{\mu_2}{\mu_1 + \mu_2}\right]^T$$

$$\theta = [a_1, a_2, a_o, b_1, b_2, b_o]^T$$

Mạng ANFIS có dạng như hình 3.2. Giải thuật huấn luyện mạng tương tự như ví dụ 3.1.

$$\theta(k+1) = \theta(k) + \eta(d - y)\Phi$$

3.1.3 Mạng xử lý mờ loại 2



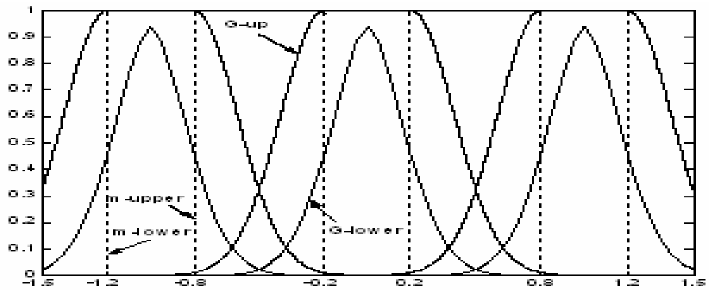
Hình 3.3: Hệ thống xử lý mờ loại 2.

Tín hiệu vào thứ i có j giá trị ngôn ngữ (tập mờ) loại 2 dạng khoảng $\left[\underline{\mu}_{ij}, \overline{\mu}_{ij}\right]$.

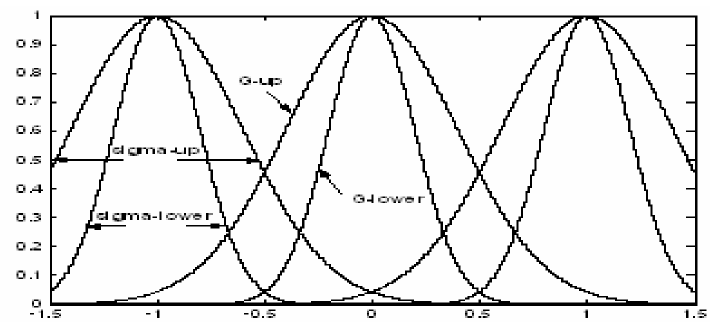
Lớp 1: Các nút ở lớp này có chức năng truyền giá trị ngõ vào đến các nút của lớp kế tiếp.

Lớp 2: Các nút ở lớp này biểu diễn các hàm thành viên dạng Gaussian với trung bình *m* (mean) và variance σ .

$$\mu_{ij}(\hat{d}_i(k) = \exp\left\{-\frac{(\hat{d}_i(k)-m_{ij})^2}{2.\sigma_{ij}^{\;2}}\right\}$$



Hình 3.4.Các tập mờ với m (mean) bất định



Hình 3.5. Các tập mờ với σ bất định

Trong đó

- Nếu σ_{ij} cố định thì m_{ij} bất định: $m_{ij} \in [m_{ij}^{lower} m_{ij}^{upper}]$ (hình 1.31).

- Nếu m_{ij} cố định thì σ_{ij} bất định: $\sigma_{ij} \in [\sigma_{ij}^{lower} \sigma_{ij}^{upper}]$ (hình 1.32).

Lớp 3 : Lớp này thực hiện các luật hợp thành. Các phép giao của tập khoảng trong mệnh đề điều kiện được suy diễn dùng phép PRO(nhân). Các hàm thành viên mệnh đề điều kiện là các hàm thành viên loại 2 được biểu diễn bởi khoảng $[\mu_{ij}^{lower}(\hat{d}_i), \mu_{ij}^{upper}(\hat{d}_i)]$. Mệnh đề kết luận thứ n là tập mờ loại 2 xác định bởi hàm thành viên :

$$\tilde{\mu}^{n} = \left[\prod \underline{\mu}_{ij}(d_i(k)),\ \prod \overline{\mu_{ij}}(d_i(k))\right]$$

với n = 1,2,3..N (N : số luật hợp thành).

Lớp 4 : Các nút ở lớp này thực hiện việc giảm loại từ loại 2 sang loại 1. Các giá trị y_r và y_1 được xác định dùng phương pháp tâm của tập hợp (center - of - set) như sau :

$$y_r = \frac{\sum_{n=1}^{R} \underline{\mu}^n w_r^n + \sum_{h=R+1}^{N} \overline{\mu}^h w_r^h}{\sum_{n=1}^{R} \underline{\mu}^n + \sum_{h=R+1}^{N} \overline{\mu}^h}$$

$$y_l = \frac{\sum_{n=1}^{L} \overline{\mu}^n w_l^n + \sum_{h=L+1}^{N} \underline{\mu}^h w_l^h}{\sum_{n=1}^{L} \overline{\mu}^n + \sum_{h=L+1}^{N} \underline{\mu}^h}$$

với R $(1 \le R \le N-1)$ và L $(1 \le L \le N-1)$.

Định nghĩa

$$\phi_r^T = \frac{1}{\sum_{n=1}^{R} \underline{\mu}^n + \sum_{h=R+1}^{N} \overline{\mu}^h} \left[\underline{\mu}^1 \underline{\mu}^2 \underline{\mu}^3 .. \underline{\mu}^R \overline{\mu}^{R+1} \overline{\mu}^{R+2} ... \overline{\mu}^N \right]$$

$$\phi_l^T = \frac{1}{\sum_{n=1}^{L} \overline{\mu}^n + \sum_{h=L+1}^{N} \underline{\mu}^h} \left[\overline{\mu}^1 \overline{\mu}^2 ... \overline{\mu}^L \underline{\mu}^{L+1} \underline{\mu}^{L+2} ... \underline{\mu}^N \right]$$

$$w_r = \left[w_r^1 w_r^2 w_r^3 .. w_r^N \right]^T \qquad w_l = \left[w_l^1 w_l^2 w_l^3 .. w_l^N \right]^T$$

$$y_l(k) = \phi_l^T(k) w_l(k);$$

$$y_r(k) = \phi_r^T(k) w_r(k)$$

Lớp 5: Lớp này là lớp ra có chức năng giải mờ. Tín hiệu ra được tính như sau:

$$y(k) = \frac{1}{2}[y_l(k) + y_r(k)]$$

Giải thuật huấn luyện: Định nghĩa hàm mục tiêu

$$J(k) = \frac{1}{2} e^2(k)$$

Quá trình huấn luyện dùng giải thuật steepest descent:

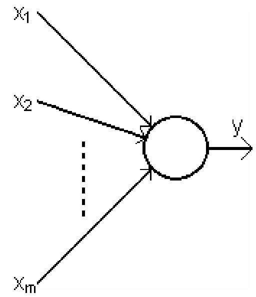
$$w_l(k+1) = w_l(k) - \eta \left[\frac{\partial J(k)}{\partial w_l(k)} \right]^T$$

$$w_r(k+1) = w_r(k) - \eta \left[\frac{\partial J(k)}{\partial w_r(k)} \right]^T$$

Với $\eta > 0$ là hằng số học.

3.2 NEURON-MỜ

3.2.1 Ứng dụng logic mờ trong mạng neuron



Hình 3.3 : Fuzzy neuron $y = \mu(x_1, x_2, \ldots, x_m)$

3.2.2 Fuzzy classification with the back propagation network

3.2.3 Fuzzy associative memories

3.2.4 Fuzzy ART models

3.2.5 Neural networks with fuzzy training

3.3 CÁC ỨNG DỤNG TIÊU BIỂU

3.3.1 Pattern regcognition

3.3.2 Clustering

3.3.3 Image processing

3.3.4 Speech recognition

3.3.5 System diagnosis

Bài tập

3.1 Cho mạng ANFIS với 2 tín hiệu vào x_1, x_2 và 1 tín hiệu ra y. Các tập mờ của các tín hiệu vào được cho ở hình P3.1. Các luật hợp thành xác định bởi

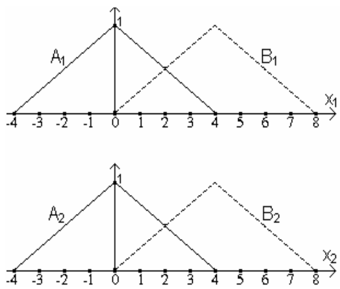
R_1 : IF x_1 IS A_1 AND x_2 IS A_2 THEN y = a

R_2 : IF x_1 IS A_1 AND x_2 IS B_2 THEN y = b

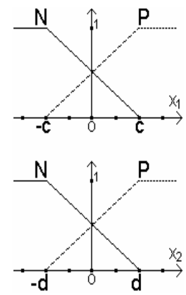
R_3 : IF x_1 IS B_1 AND x_2 IS A_2 THEN y = c

R_4 : IF x_1 IS B_1 AND x_2 IS B_2 THEN y = d

Xác định giải thuật huấn luyện các thông số a, b, c, d dùng phương pháp steepest descent với hàm mục tiêu $J = 0.5(r-y)^2$ với r là tín hiệu ra mong muốn của mạng.



Hình P3.1



Hình P3.2

3.2 Cho mạng ANFIS với 2 tín hiệu vào x_1, x_2 và 1 tín hiệu ra y. Các tập mờ của các tín hiệu vào được cho ở hình P3.2. Các luật hợp thành xác định bởi

R_1 : IF x_1 IS N AND x_2 IS N THEN $y = \alpha$

R_2 : IF x_1 IS N AND x_2 IS P THEN $y = \beta$

R_3 : IF x_1 IS P AND x_2 IS N THEN $y = \gamma$

R_4 : IF x_1 IS P AND x_2 IS P THEN $y = \lambda$

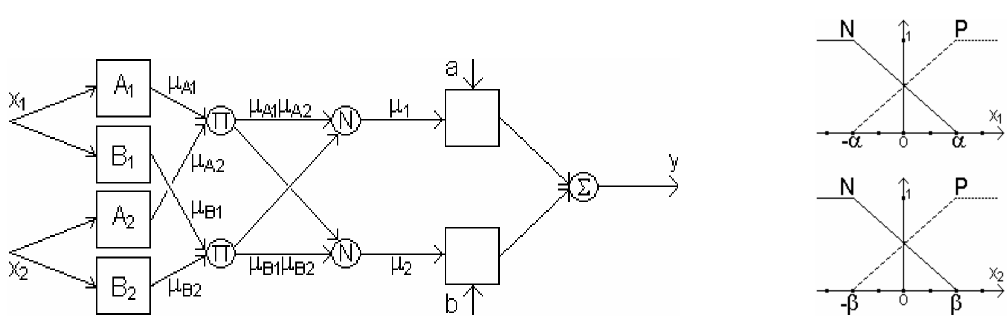
Xác định giải thuật huấn luyện các thông số c, d dùng phương pháp steepest descent với hàm mục tiêu $J = 0.5(r-y)^2$ với r là tín hiệu ra mong muốn của mạng. Giả thiết $|x_1| < c$ và $|x_2| < d$.

3.3 Cho mạng ANFIS với 2 tín hiệu vào x_1, x_2 và 1 tín hiệu ra y. Các tập mờ của các tín hiệu vào được cho ở hình P3.3. Sử dụng luật PROD để diễn dịch phép giao và phép suy diễn, luật SUM để diễn dịch phép hội và giải mờ dùng phương pháp trung bình. Các luật hợp thành xác định bởi

R_1 : IF x_1 IS A_1 AND x_2 IS A_2 THEN y = a

R_2 : IF x_1 IS B_1 AND x_2 IS B_2 THEN y = b

Xác định giải thuật huấn luyện các thông số a và b dùng phương pháp steepest descent với hàm mục tiêu $J = 0,5(r-y)^2$ và hằng số học η với r là tín hiệu ra mong muốn của mạng. Biết $|x_1| < \alpha$, $|x_2| < \beta$.



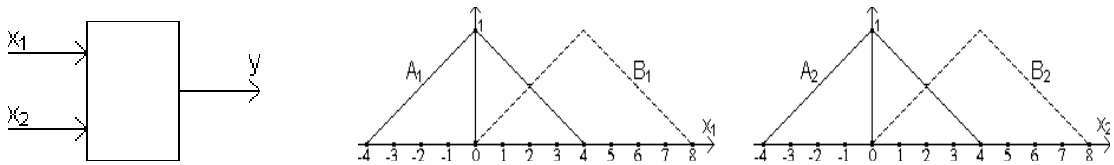
Hình P3.3

3.4 Cho hệ thống xử lý mờ hình P3.4 với 2 tín hiệu vào x_1, x_2, 1 tín hiệu ra y và các luật hợp thành

R_1 : IF x_1 IS A_1 AND x_2 IS A_2 THEN $y = a_o$

R_2 : IF (x_1 IS A_1 AND x_2 IS B_2) OR (x_1 IS B_1 AND x_2 IS A_2) THEN y = a_1

R_1 : IF x_1 IS B_1 AND x_2 IS B_2 THEN y = a_2



Hình P3.4

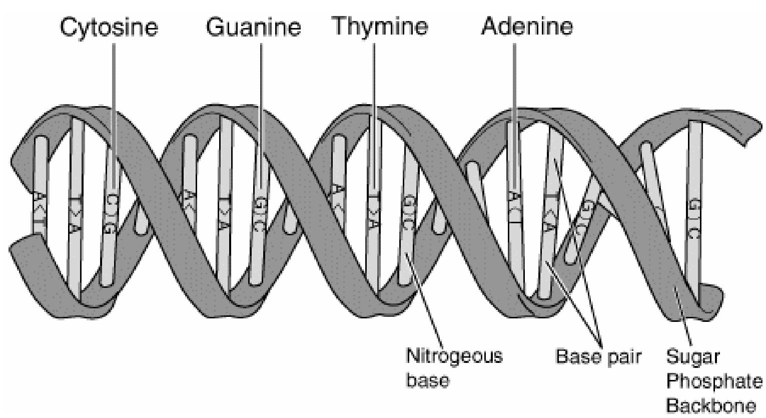
Sử dụng luật PROD diễn dịch phép giao và phép suy diễn, luật SUM diễn dịch phép hội, và giải mờ dùng phương pháp trung bình. Vẽ mạng ANFIS biểu diễn hệ thống và xây dựng giải thuật huấn luyện các thông số a_o, a_1, a_2 của mạng.

Chương 4 GIẢI THUẬT DI TRUYỀN

Giải thuật di truyền (genetic algorithms, GA) mô phỏng quá trình tiến hóa trong tự nhiên. Hai ứng dụng chính của giải thuật di truyền

- tối ưu hóa : tìm cực trị của hàm số, xử lý ảnh, nhận dạng, điều khiển
- học máy (machine learning)

Giải thuật di truyền là một công cụ quan trọng trong huấn luyện thông số và cấu trúc của mạng neuron. Các thông số và cấu trúc cần huấn luyện được mã hóa bằng gen hoặc nhiễm sắc thể. Giải thuật di truyền được sử dụng như công cụ tìm kiếm để xác định thông số và cấu trúc tối ưu. Giải thuật di truyền cũng có thể được sử dụng để xác định các hàm thành viên và luật hợp thành của hệ thống xử lý mờ.



4.1 GIẢI THUẬT DI TRUYỀN (GA)

4.1.1 Ví dụ 1 [4]: Tìm giá trị lớn nhất của hàm số $f(x) = 2x - \frac{x^2}{16}$ trên khoảng [0, 31].

Rời rạc hóa khoảng [0, 31] bởi 32 số nguyên từ 0 đến 31 và mã hóa các số nguyên nầy bởi một số nhị phân 5 bit (số bit n = 5), gọi là nhiễm sắc thể (chromosome). Tập tổng các nhiễm sắc thể G^5 xác định bởi

$$G^5 = \{ 00000,\ 00001,\ 00010,\ 00011,\ 00100,\ 00101,\ 00110,\ 00111,\ 01000,\ 01001,\ 01010,\ 01011,\ 01100,\ 01101,\ 01110,\ 01111,\ 10000,\ 10001,\ 10010,\ 10011,\ 10100,\ 10101,\ 10110,\ 10111,\ 11000,\ 11001,\ 11010,\ 11011,\ 11100,\ 11101,\ 11110,\ 11111 \}$$

mỗi số nhị phân 5 bit của G^5 được gọi là một nhiễm sắc thể, mỗi bit được gọi là một gen. Chọn kích thước quần thể m = 4. Quần thể ban đầu $p^{(1)}$ được chọn một cách ngẫu nhiên từ G^5

$$p^{(1)} = \{ 00010,\ 01001,\ 10011,\ 11000 \}$$

Đánh giá các nhiễm sắc thể trong quần thể dùng hàm f(x). Loại bỏ các nhiễm sắc thể 'yếu' (có f(x) bé và bổ sung bằng các nhiễm sắc thể 'khoẻ' (có f(x) lớn. Cụ thể như cho ở bảng 4.1

Bảng 4.1

giá trị nhị phân $x \in p^{(1)}$	giá trị thập phân $x \in p^{(1)}$	$f(x) = 2x - \frac{x^2}{16}$	$e(x) = \frac{f(x)}{\frac{1}{m}\sum_{x_i \in p^{(1)}} f(x_i)}$	int{e(x)}	n_x
00010	2	3.75	0.34	0	0
01001	9	12.94	1.17	1	1
10011	19	15.44	1.40	1	2*
11000	24	12.00	1.10	1	1

Ghi chú: n_x = int{e(x)}: số lượng bản sao trong quần thể mới. Để kích thước quần thể mới m = 4, nhiễm sắc thể 10011 được chọn với $n_x = 2$ bản sao do e(10011) = 1.40 = max. Ta được

$p^{(1)}{}_n = \{01001,\ 10011,\ 10011,\ 11000\}$

Vì $p^{(1)}{}_n \neq p^{(1)}$ nên ta sẽ tìm quần thể thế hệ thứ hai $p^{(2)}$ qua lai tạo (cross over) các nhiễm sắc thể trong quần thể $p^{(1)}{}_n$.

Bảng 4.2

$p^{(1)}{}_n$	nhiễm sắc thể phối ngẫu (chọn ngẫu nhiên trong $p^{(1)}{}_n$) g_m	Vị trí lai ghép i (chọn ngẫu nhiên $1 \le i \le n = 5$)	$p^{(2)}$
01001	100**11**	3	**010**11
10011	010**01**	3	**100**01
10011	1**1000**	1	**1**1000
11000	1**0011**	1	**1**0011

Sau khi lai ghép có thể thực hiện các đột biến (với xác suất bé) trên quần thể mới.

Đánh giá các nhiễm sắc thể trong quần thể mới dùng hàm f(x). Loại bỏ các nhiễm sắc thể 'yếu' (có f(x) bé và bổ sung bằng các nhiễm sắc thể 'khoẻ' (có f(x) lớn. Cụ thể như cho ở bảng 4.3

Bảng 4.3

giá trị nhị phân $x \in p^{(2)}$	giá trị thập phân $x \in p^{(2)}$	$f(x) = 2x - \frac{x^2}{16}$	$e(x) = \frac{f(x)}{\frac{1}{m}\sum_{x_i \in p^{(1)}} f(x_i)}$	int{e(x)}	n_x
01011	11	14.44	1.00	1	1
10001	17	15.94	1.10	1	1
11000	24	12.00	0.83	1	1
10011	19	15.44	1.07	1	1

Quần thể mới

$p^{(2)}_n = \{01011,\ 10001,\ 11000,\ 10011\} = p^{(2)}$: dừng.

Nhiễm sắc thể có độ thích hợp cao nhất là 10001 tương ứng với x = 17.

Chú ý rằng f(x) đạt cực đại khi x = 16 : $f_{max} = f(16) = 16$ so với f(17) = 15.94.

4.1.2 Ví dụ 2: Tìm đường đi ngắn nhất qua các thành phố sau sao cho mỗi thành phố chỉ được đi qua 1 lần

1) London	3) Dunedin	5) Beijing	7) Tokyo
2) Venice	4) Singapore	6) Phoenix	8) Victoria

Do có 8 thành phố, ta có thể mã hóa dùng nhiễm sắc thể với 8 gen, mỗi gen tượng trưng bởi 1 số từ 1 đến 8 (tương ứng với thành phố trong danh sách), mỗi số chỉ xuất hiện 1 lần trong nhiễm sắc thể.

Lai tạo (cross over)

```
                *     *
Parent1   (3  5  7  2  1  6  4  8)
Parent2   (2  5  7  6  8  1  3  4)

Child     (5  8  7  2  1  6  3  4)
```

(loại bỏ các gen đã xuất hiện)

Đột biến (mutation) bằng cách hoán vị 2 gen

```
          *           *
Trước: (5  8  7  2  1  6  3  4)

Sau:   (5  8  6  2  1  7  3  4)
```

Fitness = nghịch đảo tổng khoảng cách của các thành phố liên tiếp nhau trong nhiễm sắc thể.

4.1.3 Các bước cơ bản của giải thuật di truyền như sau

1. Rời rạc hóa không gian tìm kiếm và mã hóa chúng dưới dạng các nhiễm sắc thể.

   Nhiễm sắc thể có thể là bất kỳ cấu trúc dữ liệu nào, ví dụ

   Dãy số nhị phân (0101 ... 1100)

   Dãy số thập phân (2 3 0 ... 9 6 3)

Dãy số thực (43.2 -33.1 ... 0.0 89.2)

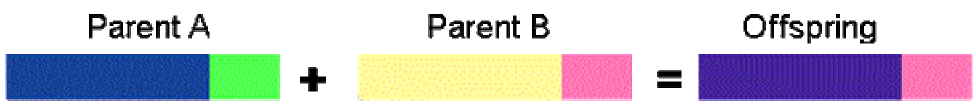
...

2. Tạo một cách ngẫu nhiên quần thể ban đầu $p^{(1)}$ gồm m nhiễm sắc thể
3. Xác định giá trị của hàm chất lượng f(x) cho m nhiễm sắc thể của quần thể.

   f(x) > 0 trên không gian tìm kiếm.
4. Loại bỏ các nhiễm sắc thể có f(x) thấp.
5. Tạo quần thể mới bằng cách thực hiện các bước sau

   5.1. Chọn lọc (selection): chọn hai nhiễm sắc thể cha mẹ từ quần thể (có thể chọn ngẫu nhiên hoặc chọn theo giá trị của hàm chất lượng f(x)).

   5.2. Lai tạo (cross over): Với 1 xác suất lai tạo chọn trước, lai tạo hai nhiễm sắc thể cha mẹ để tạo ra các nhiễm sắc thể con. Nếu không có lai tạo, các nhiễm sắc thể con trùng với các nhiễm sắc thể cha mẹ.



Lai tạo tại 1 điểm

Lai tạo tại 2 điểm

Lai tạo đều (uniform crossover)

Lai tạo số học

   5.3. Đột biến (mutation): Với một xác suất đột biến chọn trước, thực hiện đột biến tại các gen của các nhiễm sắc thể con. Đột biến giúp thoát khỏi cực trị địa phương.

5.4. Tạo quần thể mới từ các nhiễm sắc thể con.

6. Nếu điều kiện dừng không thỏa mản: Quay lại bước 3.

4.2 CÁC THÔNG SỐ CỦA GIẢI THUẬT DI TRUYỀN

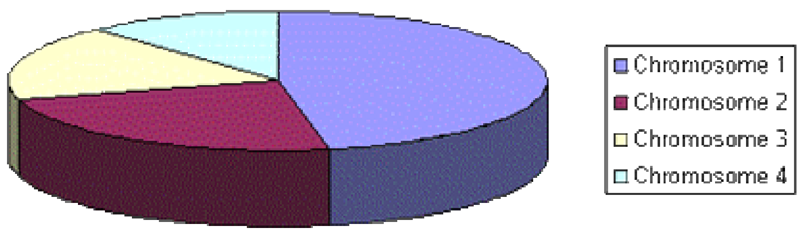
4.2.1 Xác suất lai tạo (cross over rate): Thường có giá trị lớn, khoảng 80%-95%.

4.2.2 Xác suất đột biến (mutation rate): Thường có giá trị bé, khoảng 0.5%-1%.

4.2.3 Kích thước quần thể (population size): Thường chọn trong khoảng 20-30.

4.2.4 Chọn lọc (selection): Có nhiều phương pháp chọn lọc nhiễm sắc thể từ quần thể để lai tạo (cross over). Phỏng theo thuyết tiến hoá của Darwin, các nhiễm sắc thể tốt nhất sẽ được chọn để tạo ra các nhiễm sắc thể con. Các phương pháp chọn lọc thông dụng như sau

1) Roulette wheel selection: Các nhiễm sắc thể cha và mẹ được chọn theo fitness. Các nhiễm sắc thể 'tốt' có nhiều cơ hội được chọn hơn. Ví dụ các nhiễm sắc thể 'khỏe' tương ứng với phần có diện tích lớn trên roulette wheel.



Giải thuật roulette wheel.

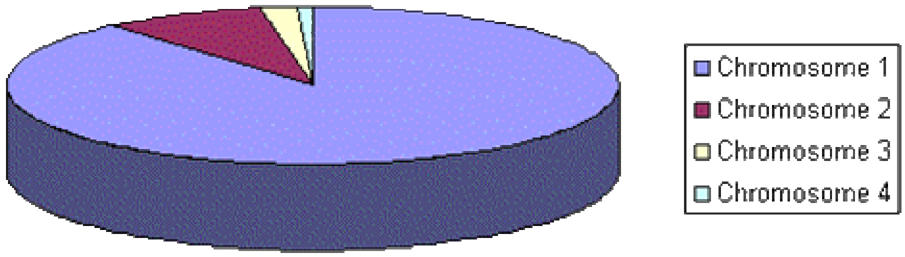
1. [Sum] Calculate sum of all chromosome fitnesses in population - sum S.

2. [Select] Generate random number from interval (0,S) - r.

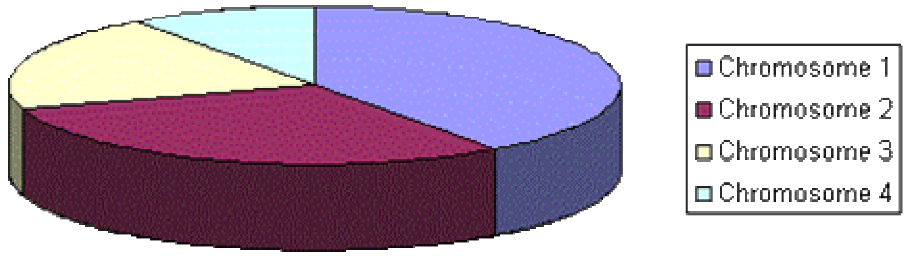
3. [Loop] Go through the population and sum fitnesses from 0 - sum s. When the sum s is greater then r, stop and return the chromosome where you are.

Of course, step 1 is performed only once for each population.

2) Rank selection: Nhược điểm của phương pháp roulette wheel là các nhiễm sắc thể 'yếu' có rất ít cơ hội được chọn. Để tránh nhược điểm trên, phương pháp rank selection xếp hạng các cá thể trong quần thể. Cá thể yếu nhất có fitness 1, cá thể kế tiếp có fitness 2 ... và cá thể mạnh nhất có fitness m (m là số nhiễm sắc thể trong quần thể). Ví dụ



Situation before ranking (graph of fitnesses)



Situation after ranking (graph of order numbers)

Phương pháp nầy có thể có tốc độ hội tụ chậm do ít có sư khác biệt giữa các nhiễm sắc thể khác nhau.

3) Steady state selection: Ý tưởng chính của phương pháp nầy là giử phần lớn các nhiễm sắc thể cho thế hệ kế tiếp. Cụ thể như sau: một số ít nhiễm sắc thể khỏe được chọn để lai tạo và thay thế các nhiễm sắc thể yếu nhất. Các nhiễm sắc thể khác được giử nguyên cho thế hệ kế tiếp.
4) Boltzman selection
5) Tournament selection

4.2.5 Phương pháp mã hóa (encoding).

4.2.6 Kiểu lai tạo và đột biến (crossover and mutation type).

4.3 CÁC ỨNG DỤNG TIÊU BIỂU

- Nonlinear dynamical systems - predicting, data analysis
- Designing neural networks, both architecture and weights
- Robot trajectory
- Evolving LISP programs (genetic programming)
- Strategy planning
- Finding shape of protein molecules
- TSP and sequence scheduling
- Functions for creating images

Bài tập

4.1 Mô tả giải thuật di truyền tìm giá trị lớn nhất của hàm số f(x) trên đoạn $a \le x \le b$.

4.2 Mô tả giải thuật di truyền để tìm giá trị lớn nhất của hàm số $y = f(x_1,x_2)$ trên miền xác định bởi $-a \le x_1 \le a$, $-b \le x_2 \le b$.

4.3 Mô tả giải thuật di truyền để tìm đường đi ngắn nhất nối 10 điểm, sao cho mỗi điểm chỉ đi qua 1 lần.

Bài tập lớn

1.1) Dùng phương pháp phân nhóm mờ, viết m file phân chia tập dữ liệu $X \subset R^2$ thành hai nhóm xác định bởi đường thẳng $x_2 = cx_1$. (Tự tạo tập dữ liệu gồm 200 phần tử trong R^2).

1.2) Học viên chọn 1 trong 3 câu a, b, c sau

a) Dùng Simulink mô phỏng hệ thống rời rạc $H(z) = \dfrac{(1-a)(1-b)}{(z-a)(z-b)}$ (dùng khối *discrete transfer function* với sample time = 0.1) với tín hiệu vào là ồn trắng (dùng khối *band limited white noise* với sample time = 1). Lưu các tín hiệu vào và ra của hệ thống vào tập tin (dùng khối *to file* với sample time = 0.1). Viết m file để đọc dữ liệu trên trên tập tin và nhận dạng mô hình mờ của hệ thống. So sánh tín hiệu ra của hệ thống với tín hiệu ra mô phỏng và dự báo. Nhận xét.

b) Xây dựng bộ cân bằng mờ cho kênh thông tin với hàm truyền đạt $H(z) = \dfrac{(1-a)(1-b)}{(z-a)(z-b)}$

c) Dùng Simulink mô phỏng hệ thống điều khiển mờ Mamdani cho đối tượng có hàm truyền đạt $H(z) = \dfrac{1}{(z-a)(z-b)}$ (chọn chu kỳ lấy mẫu = 0.05s).

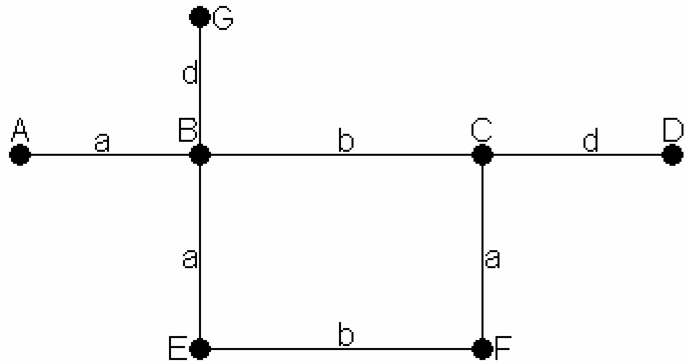
2.1) Huấn luyện mạng truyền thẳng để phân nhóm các phần tử của tập $X \subset R^2$ thành hai nhóm xác định bởi đường thẳng $x_2 = cx_1$. (Tự tạo tập dữ liệu gồm 200 phần tử trong R^2).

2.2) Học viên chọn 1 trong 2 câu a, b sau

a) Dùng Simulink mô phỏng hệ thống rời rạc $H(z) = \dfrac{(1-a)(1-b)}{(z-a)(z-b)}$ (dùng khối *discrete transfer function* với sample time = 0.1) với tín hiệu vào là ồn trắng (dùng khối *band limited white noise* với sample time = 1). Lưu các tín hiệu vào và ra của hệ thống vào tập tin (dùng khối *to file* với sample time = 0.1). Viết m file để đọc dữ liệu trên trên tập tin và huấn luyện mạng truyền thẳng. So sánh tín hiệu ra của hệ thống với tín hiệu ra mô phỏng và dự báo. Nhận xét.

b) Xây dựng bộ cân bằng dùng mạng neuron cho kênh thông tin với hàm truyền đạt $H(z) = \dfrac{(1-a)(1-b)}{(z-a)(z-b)}$

3.1) Dùng giải thuật di truyền, tìm đường đi ngắn nhất nối các điểm ABCDEFG. Biết d = a.



tt	1	2	3	4	5	6	7	8	9	10
a	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75
b	0.99	0.97	0.96	0.93	0.91	0.89	0. 87	0.86	0.84	0.83
c	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

tt	11	12	13	14	15	16	17	18	19	20
a	0.80	0.82	0.85	0.88	0.90	0.92	0.94	0.95	0.96	0.98
b	0.81	0.79	0.77	0.68	0.64	0.62	0.58	0.53	0.48	0.43
c	1.2	1.5	1.8	2	2.2	2.5	2.8	3	3.5	4

tt	21	22	23	24	25	26	27	28	29	30
a	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75
b	0.81	0.79	0.77	0.68	0.64	0.62	0.58	0.53	0.48	0.43
c	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9

tt	31	32	33	34	35	36	37	38	39	40
a	0.80	0.82	0.85	0.88	0.90	0.92	0.94	0.95	0.96	0.98
b	0.99	0.97	0.96	0.93	0.91	0.89	0. 87	0.86	0.84	0.83
c	10	12	15	18	20	25	30	35	40	45

tt	41	42	43	44	45	46	47	48	49	50
a	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75
b	0.89	0. 87	0.86	0.84	0.83	0.99	0.97	0.96	0.93	0.91
c	-0.1	-0.2	-0.3	-0.4	-0.5	-0.6	-0.7	-0.8	-0.9	-1

tt	51	52	53	54	55	56	57	58	59	60
a	0.80	0.82	0.85	0.88	0.90	0.92	0.94	0.95	0.96	0.98
b	0.62	0.58	0.53	0.48	0.43	0.81	0.79	0.77	0.68	0.64
c	-1.2	-1.5	-1.8	-2	-2.2	-2.5	-2.8	-3	-3.5	-4

tt	61	62	63	64	65	66	67	68	69	70
a	0.92	0.94	0.95	0.96	0.80	0.82	0.85	0.88	0.90	0.80
b	0.99	0.97	0.96	0.93	0.91	0.89	0. 87	0.86	0.84	0.83
c	-4.5	-5	-5.5	-6	-6.5	-7	-7.5	-8	-8.5	-9

tt	71	72	73	74	75	76	77	78	79	80
a	0.55	0.60	0.65	0.70	0.75	0.30	0.35	0.40	0.45	0.50
b	0.81	0.79	0.77	0.68	0.64	0.62	0.58	0.53	0.48	0.43
c	-10	-12	-15	-18	-20	-25	-30	-35	-40	-45

tt	81	82	83	84	85	86	87	88	89	90
a	0.55	0.60	0.65	0.70	0.75	0.30	0.35	0.40	0.45	0.50
b	0.79	0.68	0.62	0.53	0.81	0.77	0.64	0.43	0.43	0.48
c	-15	-20	-35	-40	-12	-10	-45	-25	-18	-30

Tài liệu tham khảo

[1] B. Kosko (Ed). *Neural networks for signal processing*. Prentice Hall. 1992.

[2] C.T. Lin and C.S.G. Lee. *Neural fuzzy systems*. Prentice Hall. 1996.

[3] O. Omidvar and D.L. Elliott (Eds). *Neural systems for control.* Academic Press. 1997.

[4] N.N. Phong. *Tính toán mềm và ứng dụng.* NXB Khoa học và Kỹ thuật. 2008.

[5] R. Rojas. *Neural networks. A systematicIntroduction.* Springer. 1996.

[6] T.J. Ross. *Fuzzy logic with engineering applications*. McGraw-Hill, 1997.

[7] L.X. Wang. *A course in fuzzy systems and control*. Prentice Hall. 1997.

[8] J. Yen and R. Langari. *Fuzzy logic. Intelligence, Control, and Information.* Prentice Hall. 2000.