

# 신경회로망을 이용한 퍼지 제어규칙의 추정 및 퍼지 제어기의 구현

○전 용 성 박 상 배 이 균 경  
경북대학교 공과대학 전자공학과

## Identificaton of Fuzzy Rule and Implementation of Fuzzy Controller using Neural Network

Jeon, Yong-Sung Park, Sang-Bae Lee, Kyun-Kyung  
Dept. of Electronics Eng.  
Kyung Pook National University

### ABSTRACT

This paper proposes a modified fuzzy controller using a neural network. This controller can automatically identify expert's control rules and tune membership functions utilizing expert's control data. Identificaton capability of the fuzzy controller is examined using simple numerical data. The results show that the network in this paper can identify nonlinear systems more precisely than conventional fuzzy controller using neural network.

이러한 노력의 일환으로 Horikawa 등이 이미 신경회로망을 이용한 퍼지논리 제어기를 제시한 바 있다[1]. 그러나 Horikawa 등이 제시한 퍼지논리 제어기는 제어규칙의 후건부가 상수이므로, 전문가의 제어 데이터(data)를 이용한 모델 추정에 있어 큰 오차가 있다. 따라서 본 논문에서는 제어규칙의 후건부를 선형 방정식을 가지도록 신경회로망을 이용한 퍼지논리 제어기를 구성함으로써 후건부가 상수일 때 보다 오차를 좀 더 줄이도록 하였다. 결국, 본 논문에서 제시한 제어기는 전문가의 제어 데이터를 사용하여 제어규칙을 자동적으로 추정할 수 있으며 멤버십(membership) 함수를 조절할 수 있다.

### I. 서 론

퍼지논리 제어는 전문가에 의해 언어적으로 표현되는 규칙(rule)으로써 제어할 수 있다는 독특한 특징을 가지고 있다. 그러나 전문가는 흔히 이들 제어규칙을 시행착오를 통해서 배우게 되며, 학습된 규칙들이 때로는 부적절한 경우도 발생한다. 그리고 제어규칙들을 추정(identification)하는데 많은 시간을 필요로 하는 단점을 가지고 있다. 반면, 신경회로망은 전문가의 도움없이 학습을 통해 제어할 수 있다는 잇점은 있으나 신경회로망을 통해서 인간이 이해할 수 있는 제어규칙을 얻을 수 없다는 단점을 가지고 있다. 이와 같은 퍼지와 신경회로망의 장점만을 이용한 것이 신경회로망을 이용한 퍼지논리 제어기이며, 이를 뉴로퍼지 시스템이라 부른다[2]. 즉, 신경회로망이 학습을 할 수 있다는 점과 퍼지의 제어규칙이 인간이 이해하기가 쉽다는 점을 이용하여 퍼지의 알고리즘이 수행 될 수 있는 신경회로망을 구성하여 학습을 시킨 다음 이 신경회로망으로 부터 인간이 이해할 수 있는 퍼지의 제어규칙을 얻고자 하는 것이다.

### II. 본 론

#### 1. 신경회로망을 이용한 퍼지논리 제어기

신경회로망을 이용한 퍼지논리 제어기의 구조를 그림 1에 나타내었다. 이 회로망은 7개의 층(layer)으로 구성되며 회로망의 학습을 위해서는 역전달(back propagation) 알고리즘을 사용한다. 이 신경회로망은 2개의 입력과 1개의 출력을 가지며 각 입력에 대해 2개의 멤버십(membership) 함수를 가지는 퍼지논리 제어기이다. 이와같이 입력과 출력의 수 그리고 각 입력에 대한 멤버십 함수의 수단 정해지면, 이에 해당하는 신경회로망을 구성할 수 있다. 그런 다음 이 회로망을 입력에 대한 원하는 출력이 나오도록 학습시킨 후에 이들 입력과 출력을 실현시킬 수 있는 제어규칙 및 멤버십 함수의 모양은 회로망의 연결강도로 부터 얻을 수 있다.

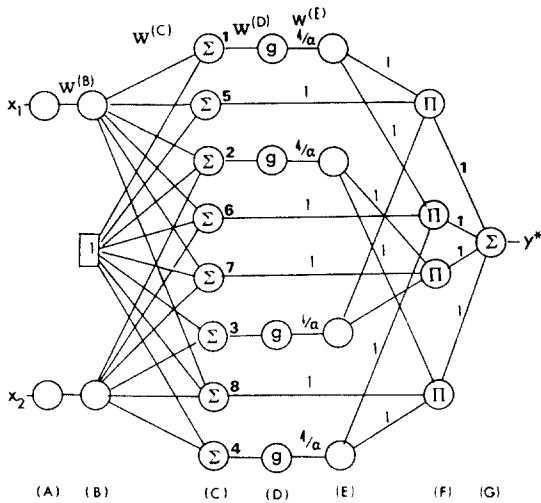


그림 1. 신경회로망을 이용한 퍼지논리 제어기  
Fig. 1. Fuzzy controller using neural network.

그림 1에서 각 뉴런의 출력단에 있는 'Σ'의 기호는 입력들의 합을 나타내며, 'Π'의 기호는 입력들의 곱을 나타낸다. 그리고 별다른 기호를 가지고 있지 않은 뉴런(neuron)은 그들의 입력을 다음 층에 전달할 뿐이다. (D)-층(layer)에 있는 뉴런의 출력  $g(x)$ 는 활성화(sigmoid) 함수인  $f(x)$ 를 미분한 식이며, 멤버십(membership) 함수로 사용된다.

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (1)$$

$$g(x) = f'(x) = \alpha \left( \frac{1}{1 + e^{-\alpha x}} \right) \cdot \left( 1 - \frac{1}{1 + e^{-\alpha x}} \right) \quad (2)$$

단,  $x$ 는 뉴런(neuron)의 입력이며,  $\alpha$ 는 기울기를 결정하는 계수이다.

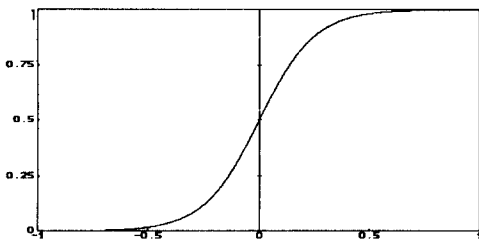


그림 2. 활성화 함수(f(x))  
Fig. 2. Sigmoid function(f(x)).

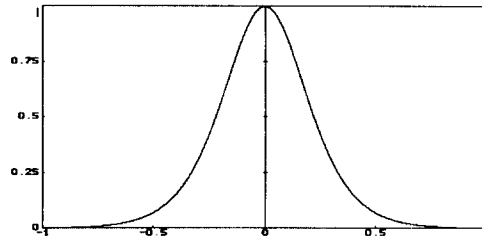


그림 3. 멤버십(membership) 함수(g(x))  
Fig. 3. Membership function(g(x)).

또한,  $W(B)$ ,  $W(C)$ ,  $W(D)$ ,  $W(E)$ , 그리고 1, -1은 연결강도라 부르며,  $W(C)$ ,  $W(D)$ 의 연결강도는 학습을 통해 수정된다. 반면  $W(D)$ 의 연결강도를  $4/\alpha$ 로 정한 이유는 멤버십 함수의 최대 크기를 1로 만들기 위해서이다.

그리고, 그림 1의 회로망에 의한 추론방법은 다음과 같다.

$R_k$  : 만약  $X_1$ 이  $A_{k1}$  이고  $X_2$ 가  $A_{k2}$  이면

$$y_k = a_k * X_1 + b_k * X_2 + c_k$$

$$y^* = \sum_{k=1}^4 (\mu_k \cdot y_k) ; \mu_k = A_{k1}(X_1) \cdot A_{k2}(X_2) \quad (3)$$

여기서  $R_k$ 는 k번째 제어규칙을 나타내며,  $A_{k1}$ ,  $A_{k2}$ 는 각각  $X_1$ ,  $X_2$ 에 대한 '크다', '작다', '높다', '낮다' 등의 언어적 표현을 나타낸다. 학습을 한 후에는  $W(D)$ 와  $W(C)$ 의 일부는 추정(identification) 오차가 최소가 되도록 이들 언어적 표현에 대한 멤버십 함수의 모양을 결정하게 된다. 즉  $W(D)$ 는 그림 3의 멤버십 함수의 기울기를 결정하며, (D)-층의 1번에서 4번까지의 뉴런과 연결된  $W(C)$ 는 멤버십 함수의 이동정도를 나타낸다. 그리고 (D)-층의 5번에서 8번까지의 뉴런과 연결된  $W(C)$ 는 제어규칙의 후건부의 계수인  $a_k$ ,  $b_k$ ,  $c_k$ 를 결정하게 된다. 연결강도  $W(B)$ 는 단지 다음 층(layer)의 입력이 -1에서 1사이가 되도록 규격화(scaling)의 역할만을 할 뿐이다.

## 2. 학습 알고리즘

연결강도  $W(C)$ 와  $W(D)$ 들은 제어규칙과 전건부의 멤버십 함수를 조절하기 위하여 아래와 같은 역전달(back propagation) 알고리즘을 사용하여 수정된다.

출력층(output layer)(G):

$$\delta_j^{(n)} = (t_j - O_j^{(n)})g'(i_j^{(n)}) \quad (4)$$

중간층(hidden layer)(C,D,F):

$$\delta_j^{(n)} = g'(i_j^{(n)}) \sum_k \delta_k^{(n+1)} W_{jk}^{(n+1)} \quad (5)$$

여기서,  $n$ 과  $j$ 는 각각  $n$ -층(layer)과  $j$ 번째 뉴런을 나타내며,  $i$ 와  $O$ 는 뉴런의 입력과 출력을 각각 나타낸다.  $t$ 는 전문가에 의해 주어지는 출력 데이터이다.  $g'$ 는 뉴런의 출력 함수의 미분을 나타내며  $W_{jk}^{(n+1)}$ 은  $(n)$ -층의  $j$ 번째 뉴런과  $(n+1)$ -층의  $k$ 번째 뉴런 사이의 연결강도를 나타낸다. 위의 1, 2식을 이용하여 그림 1에서의 (D), (F), (G)-층 및 (F)층과 직접 연결되지 않은 (C)-층의 1에서 4번까지의 뉴런에 대한 델타값을 구해보면 다음과 같다.

$$\delta^{(G)} = t - O^{(G)}$$

$$\delta_j^{(F)} = \delta^{(G)}$$

$$\delta_j^{(D)} = 4/\alpha \cdot \delta^{(E)}$$

$$\delta_j^{(C)} = g'(i_j^{(D)}) \cdot \delta_j^{(D)} \cdot W_j^{(D)}$$

$$\begin{aligned} &= \frac{\partial(O_j^{(D)})}{\partial(i_j^{(D)})} \cdot \delta_j^{(D)} \cdot W_j^{(D)} \\ &= \frac{\partial}{\partial(i_j^{(D)})} \left( \frac{\alpha e^{-a_j^{(D)}}}{(1+e^{-a_j^{(D)}})^2} \right) \cdot \delta_j^{(D)} \cdot W_j^{(D)} \\ &= \alpha \cdot O_j^{(D)} \cdot \left( 1 - \frac{2}{1+e^{-a_j^{(D)}}} \right) \cdot \delta_j^{(D)} \cdot W_j^{(D)} \end{aligned}$$

반면, (F)-층의 곱셈때문에 (E)-층과 (C)층의 5에서 8번까지의 뉴런에 대한 델타(delta)식은 다음과 같이 주어진다.

$$\delta_j^{(E)} = \sum_k \delta_k^{(F)} (\partial O_k^{(F)} / \partial O_j^{(E)}) \quad (6)$$

$$\delta_j^{(C)} = \sum_k \delta_k^{(F)} (\partial O_k^{(F)} / \partial O_j^{(C)}) \quad (7)$$

예를들어, (E)-층의 1번 뉴런과 (C)-층의 5번 뉴런의 델타값은 다음과 같다.

$$\delta_1^{(E)} = \delta^{(F)} (O_3^{(E)} O_5^{(C)} + O_4^{(E)} O_6^{(C)})$$

$$\delta_5^{(C)} = \delta^{(F)} (O_1^{(E)} O_3^{(E)})$$

그리고 연결강도는 다음과 같이 수정된다.

$$W_{ij}^{(n)(m+1)} = W_{ij}^{(n)(m+1)} + \eta \delta_j^{(n)} O_i^{(n-1)} \quad (8)$$

여기서,  $m$ 은 연결강도가  $m$ 번째 개선되는 것을 의미하며,  $\eta$ 는 학습율이라 한다.

끝으로,  $W^{(D)}$ 와 (C)층의 1번에서 4번까지의 뉴런(neuron)과 연결된  $W^{(C)}$ 는 그림 4와 같이 멤버십 모양이 정규화되도록 초기화시키고, 제어규칙의 후건부의 계수들을 결정하는 (C)층의 5번에서 8번까지의 뉴런과 연결된  $W^{(C)}$ 는 0.1로 초기화시킨다. 그리고,  $W^{(B)}$ 는 입력 데이터 중에서 최대치의 역수로 정해주고 나머지의 연결강도는 1로 정해준다.

### III. 시뮬레이션의 결과 및 고찰

비선형 시스템을 사용하여, 본 논문이 제시한 퍼지논리 제어기의 추정(identification) 능력을 평가해 보자. 비선형 시스템은 아래식과 같이 주어진다.

$$y = (1 + (X_1)^{1/2} + 1/(X_2))^2 \quad (9)$$

표 1에서는 첫 식으로 부터 얻은 25개의 입-출력 데이터를 보여주고 있다. 이들 데이터는 퍼지논리 제어기의 시스템 추정(identification)을 위해 사용된다. 그리고 추정된 시스템의 정확성을 평가하는 오차식은 다음과 같이 주어진다.

$$E = \frac{1}{N} \left( \sum_{i=1}^N \frac{|y_i - y^*|}{y_i} \right) \cdot 100(\%) \quad (10)$$

여기서  $N$ 은 데이터의 갯수이며,  $y_i$ 은 표 1에 있는 원하는 출력 값이고  $y_i^*$ 은 퍼지논리 제어기에 의해 추론된 값이다.

논문이 제시한 후건부가 선형 방정식을 가지는 제어기의 추정 능력을 알아보기 위한 시뮬레이션 방법은 다음과 같다.

1. 입력이 두개이고 출력이 1개이며 전건부에 3개의 멤버십 함수를 가지도록 그림 1과 같이 신경회로망을 구성한다.
2. 표 1에 있는 입-출력 데이터를 이용하여 신경회로망을 학습한다. 이때, 후건부의 계수들을 결정하는 연결강도의 학습율은 0.08로 정하고 전건부의 멤버십 모양을 결정하는 연결강도의 학습율은 0.0006으로 정하였다. 그런 다음 3000번의 학습을 수행한다. 학습후 모델의 정확성을 평가하기 위해 식(10)에 의한 오차를 구한다.

시뮬레이션 결과를 검토해보면, 표 2에서 알 수 있듯이 제어규칙의 후건부를 선형방정식으로 정하는 것이 보다 나은 모델 추정을 할 수 있음을 알 수 있다.

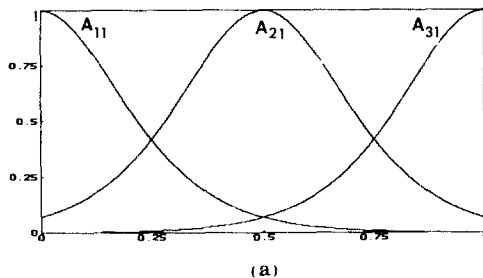
끝으로, 본 논문의 제어를 이용하였을 때, 학습전의 초기화된 멤버십 함수의 모양과 제어규칙을 그림 5에 나타내었고, 학습 후의 멤버십 함수의 모양과 제어규칙을 그림 6에 나타내었다.

표 1. 모델의 학습을 위한 입-출력 데이터  
Table 1. Input and output data for learning

	X <sub>1</sub>	X <sub>2</sub>	y	X <sub>1</sub>	X <sub>2</sub>	y
1	1	1	9.00	14	3	4
2	1	2	6.25	15	3	5
3	1	3	5.44	16	4	1
4	1	4	5.06	17	4	2
5	1	5	4.84	18	4	3
6	2	1	11.65	19	4	4
7	2	2	8.49	20	4	5
8	2	3	7.55	21	5	1
9	2	4	7.10	22	5	2
10	2	5	6.83	23	5	3
11	3	1	13.93	24	5	4
12	3	2	10.45	25	5	5
13	3	3	9.40			

표 2. 추정된 모델의 오차  
Table 2. Error for identified model

모 델	E(오차)
본 논문의 신경회로망을 이용한 퍼지제어기	0.04 (%)
Horikawa의 신경회로망을 이용한 퍼지제어기	0.17 (%)



RK : 만약 X<sub>1</sub>이 A<sub>k1</sub> 이고 X<sub>2</sub>가 A<sub>k2</sub> 이면  

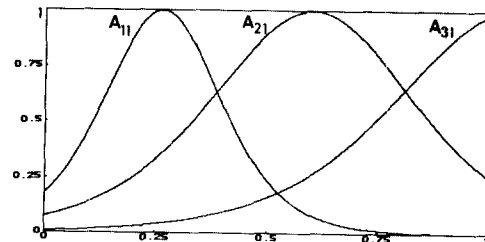
$$y_k = 0.1 * X_1 + 0.1 * X_2 + 0.1$$

(c)

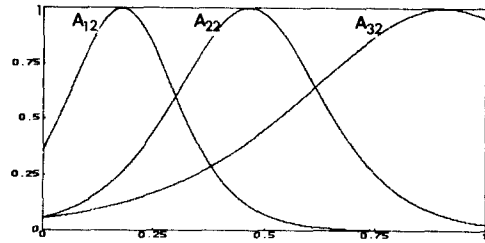
그림 5. 학습전의 멤버십 함수의 모양 및 제어규칙

- (a) x<sub>1</sub>에 대한 멤버십 함수
- (b) x<sub>2</sub>에 대한 멤버십 함수
- (c) 제어규칙

Fig. 5. Membership function and control rule before learning  
 (a) membership function for x<sub>1</sub>  
 (b) membership function for x<sub>2</sub>  
 (c) control rule



(a)



(b)

RK : 만약 X<sub>1</sub>이 A<sub>k1</sub>이고, X<sub>2</sub>가 A<sub>k2</sub>이면

$$y_1 = 1.20 * X_1 + 1.67 * X_2 + 6.02$$

$$y_2 = 1.20 * X_1 + 1.10 * X_2 + 2.07$$

$$y_3 = 1.75 * X_1 + 1.01 * X_2 + 2.48$$

$$y_4 = 3.74 * X_1 + 1.12 * X_2 + 5.82$$

$$y_5 = 1.81 * X_1 + 1.39 * X_2 + 2.53$$

$$y_6 = 1.74 * X_1 + 1.91 * X_2 + 3.02$$

$$y_7 = 6.00 * X_1 + 1.20 * X_2 + 6.08$$

$$y_8 = 3.08 * X_1 + 1.53 * X_2 + 3.28$$

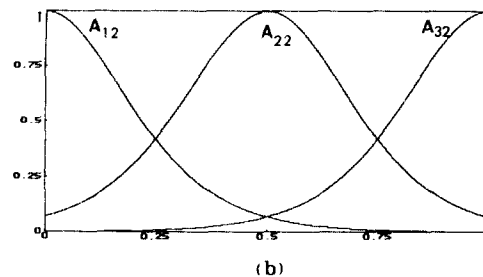
$$y_9 = 3.60 * X_1 + 3.50 * X_2 + 3.48$$

(c)

그림 6. 학습 후 멤버십 함수의 모양 및 제어규칙

- (a) x<sub>1</sub>에 대한 멤버십 함수
- (b) x<sub>2</sub>에 대한 멤버십 함수
- (c) 제어규칙

Fig. 6. Membership function and control rule after learning  
 (a) membership function for x<sub>1</sub>  
 (b) membership function for x<sub>2</sub>  
 (c) control rule



(b)

#### IV. 결 론

본 논문에서는 전문가의 제어 데이터를 이용하여 제어규칙의 추정 및 멤버십 함수의 모양을 조절할 수 있는 신경회로망을 이용한 퍼지논리 제어기를 제시하였다. 그리고 시뮬레이션을 통해 기존의 신경 회로망을 이용한 퍼지논리 제어기보다 본 논문이 제시한 신경회로망을 이용한 퍼지논리 제어기의 성능이 우수함을 알 수 있었다. 결국, 본 논문이 제시한 제어기를 이용하면 비선형 함수를 보다 정확히 추정할 수 있다.

#### 참 고 문 헌

- [1] S. Horikawa, T. Furuhashi, S. Okuma, and Y. Uchikawa, "A fuzzy controller using a neural network and its capability to learn expert's control rules," in *Proceedings of the International Conference on Fuzzy Logic & Neural Networks.*, Iizuka, Japan, pp. 103-106, 1990.
- [2] R. Masuoka, N. Watanabe, A. Kawamura, Y. Owada, and K. Asakawa, "Neurofuzzy system - Fuzzy inference using a structured neural network," in *Proceedings of the International Conference on Fuzzy Logic & Neural Networks.*, Iizuka, Japan, pp. 173-177, 1990.
- [3] T. J. Procky and E. H. Mamdani, "A linguistic self-organizing process controller," *Automat.*, vol. 15, no. 1, pp. 15-30, 1979.
- [4] T. Tagai and M. Sugano, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 1, pp. 116-132, 1985.
- [5] K. M. Passino, M. A. Sortori, and P. J. Antsaklis, "Neural computing for numeric-to-symbolic conversion in control systems," *IEEE Contr. Syst. Mag.*, pp. 44-51, Apr. 1989.