

論文96-33B-4-21

학습 기능을 갖는 자동 규칙 생성 퍼지 신경망

(Fuzzy Neural Network with Rule Generation Based on Back-Propagation Algorithm)

鄭 載 卿 *, 李 東 潤 *, 鄭 起 旭 *, 金 元 燦 *

(Jaekyung Chong, Dongyun Lee, Keewook Jung, and Wonchan Kim)

요 약

본 논문에서는 퍼지 모델링을 위한 새로운 fuzzy neural network을 제안한다. Fuzzy neural network은 4개의 layer로 구성되며, 각 layer는 if-then 퍼지 추론의 각 연산 단계에 해당된다. Fuzzy neural network의 학습 과정이라고 할 수 있는 파라미터 조정에는 back-propagation 알고리즘을 바탕으로 heuristic 방법이 사용되었다. 제안하는 network의 타당성 및 유용성을 증명하기 위하여 비선형 시스템의 모델링 결과를 이전의 연구와 비교하였다.

Abstract

This paper presents a new fuzzy neural network for fuzzy modeling. The fuzzy neural network is composed of 4 layers and the nodes of each layer represent the each step of the if-then fuzzy inference. A heuristic based on the back-propagation algorithm is proposed to adjust the parameters of the fuzzy neural network. We prove the feasibility of the network using the experiments on modeling a nonlinear mathematical system and the comparison with previous research.

I. 서 론

퍼지 모델링 (Fuzzy Modeling)은 대상 시스템을 언어적으로 표현하는 방법의 하나로, 특히 복잡한 비선형 시스템이나 시스템의 기능을 정량적으로 정의하기 힘든 경우에 알맞은 모델링 방법이다. 그러나, 대상 시스템에 대한 전문적인 지식이나 경험 없이 퍼지 규칙 (fuzzy rules)이나 퍼지 추론 시스템(fuzzy inference system)의 멤버십(membership)함수를 결정하는 것은 쉬운 일이 아니다. 이러한 어려움을 극복하는 방법 중의 하나로, 퍼지 추론 시스템과 신경망(neural network, 이하 NN)을 합성하는 방법이 널리 연구되고 있다^{[1]-[15]} 각 기술간의 장점을 이용하고자 하는

이런 형태의 합성에 대한 연구는 크게 두 가지 접근 방법으로 분류될 수 있다. 첫째로, 일반적인 NN의 구조를 퍼지화(fuzzification)하는 방법과 둘째로 퍼지 모델을 만들에 있어서 NN을 하나의 도구로 사용하는 방법이 있다. 본 논문에서는 후자의 접근 방법에 근거하는 새로운 퍼지 신경망(Fuzzy Neural Network, 이하 FNN)을 제안한다.

퍼지 추론의 각 연산 과정은 FNN의 해당 노드 (node)형태로 구현되었다. 따라서 본 논문이 제안하는 FNN은 언어로 표현된 퍼지 규칙을 적용하기에 편리하며, 또 만들어진 FNN으로부터 퍼지 규칙을 이해하기에도 쉬운 형태를 갖게 된다. 본 FNN의 학습 과정은 기본적으로 back-propagation(이하 BP) 알고리즘^[17]에 기초를 두고 있다. 학습 과정을 통하여 FNN의 퍼지 멤버십 함수를 구성하는 파라미터들이 조정되고 결정된다. 본 논문이 제안하는 FNN은 목표로 하는 시스템 또는 함수를 모델링하는 데 있어서 기존의 경험이 나 지식을 사용할 수도 있고, 또한 선형적 지식 없이

* 正會員, 서울大學校 電氣工學部 集積 回路研究室 (Integrated Circuits Design Lab., School of Elec. Eng., Seoul National Univ.)

接受日字: 1995年9月7日, 수정완료일: 1996年3月21日

원하는 모델을 찾아 나갈 수도 있다.

이하에 언급되는 FNN은 특별한 언급이 없는 한, 모두 본 논문이 제안하는 FNN을 의미하는 것으로 약속한다. 다음 장에서는 FNN의 형성과 각 노드 함수에 대해서 설명하고, 제3장에서는 FNN의 학습 과정을 설명한다. 제4장에서는 비선형 함수를 모델링하는 실험 결과를, 다른 연구와 비교 설명함으로써 본 FNN의 유용성을 검토하기로 한다.

II. Fuzzy Neural Network의 구성

1. Network 설계

FNN은 일종의 adaptive multilayer network이다. 각 계층(layer)는 고유의 노드를 갖고 있다. FNN의 기본 구성 개념은 다음과 같다.

- 1) Network은 퍼지 추론 시스템으로써 동작하여야 한다.
- 2) Network은 목표 시스템의 입출력 학습 패턴을 이용하는 학습 기능을 갖추어야 한다.
- 3) Network은 하드웨어로 구현하기 용이하여야 한다.

본 장에서의 설명을 위하여 그림 1에 간단한 FNN의 예를 나타내었다. 이 경우, FNN은 두개의 입력 x_1 과 x_2 , 그리고 하나의 출력 y_0 에 대하여 두개의 규칙을 갖는 형태로 구성되었다. 이 구성은 다음과 같은 if-then fuzzy rule로 표현할 수 있다.

Rule 1 : IF x_1 is mf_{11} AND x_2 is mf_{12} , THEN y_0 is fw_1

Rule 2: IF x_1 is mf_{21} AND x_2 is mf_{22} , THEN y_0 is fw_2 .

FNN 첫 번째 계층의 파라미터들은 학습을 통하여 조정되고 결정된다. FNN의 4가지 계층 중에서 학습 가능한 파라미터를 갖는 계층으로는 앞서 말한 첫 번째 계층과 함께 세 번째 계층인 OMF가 있다. 그러나, IMF 노드와 OMF 노드의 파라미터들을 동시에 고려하게 된다면, 그 학습 과정이 지나치게 복잡해지기 때문에 실용적인 학습 시간을 달성하기 힘들다. 때문에 본 FNN에서는 퍼지 추론에서 보다 영향력을 갖는 입력 멤버십 함수의 형태를 결정하는데 학습 과정의 중점을 두도록 하였다. 다만, 출력 멤버십 함수의 형태를

보다 간단한 형태로 바꿀 수 있다면, 이 두 가지 계층의 파라미터를 동시에 고려하는 학습 과정을 고안할 수 있을 것이다.

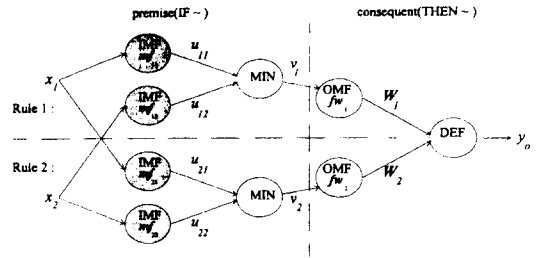


그림 1. Fuzzy Neural Network(FNN)의 간단한 예. 두개의 입력과 출력으로 두개의 규칙을 구성하며, 조건부와 결론부로 구분되는 형태를 갖고 있다. mf 's 과 fw 들은 각각 첫 번째와 세 번째 계층의 노드 함수에 사용되는 파라미터이다

Fig. 1. The proposed Fuzzy Neural Network(FNN). This example has two inputs, one output and two rules. The premise and consequent of the if-then fuzzy rule are related to the FNN. mf 's and fw 's are the parameters of the nodes in the first and the third layer respectively.

2. Node 함수

FNN의 첫 번째 계층의 노드는 IMF(Input Membership Function)라고 부른다. IMF 노드는 각 if then rule내의 하나의 입력에 대하여 하나씩 존재하는데, 그 역할은 각 입력에 대해 해당 if-then rule에서의 멤버십값을 계산하는 일이다. FNN에서 p번째 규칙, q번째 입력에 대한 IMF 노드 함수는 다음과 같다.

$$u_{pq} = IMF(x_q; mf_{pq}), \tag{1}$$

여기에서 x_q 는 입력, mf_{pq} 는 파라미터 집합, 그리고 u_{pq} 는 IMF 노드의 출력이다.

퍼지 멤버십 함수는 여러 가지 형태를 갖을 수 있다. 실제적인 응용의 경우에는, 대부분 삼각형이나 사다리꼴 등의 모양을 사용한다¹¹⁾. 그 이유는 함수를 표현하기 용이하고 또 함수로부터 멤버십 값을 계산하기도 간단하기 때문이다. 반면 이론적인 연구의 경우에는 가우스 함수나 S자형의 함수가 많이 사용되는데, 그 이유는 이러한 함수들이 수학적 분석에 보다 알맞기 때

문이라고 할 수 있다¹²⁾. 본 논문은 보다 실용적인 방법에 그 중점을 둔 바, 사다리꼴 또는 piecewise linear type을 멤버십 함수로 사용한다. 그림 2에 p번째 규칙의 q번째 입력에 대한 piecewise linear type 멤버십 함수를 8개의 파라미터($\theta_{pq1}, \dots, \theta_{pq8}$)로 나타낸 예를 보였다.

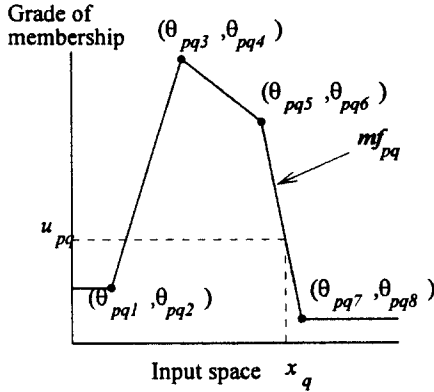


그림 2. 퍼지 멤버십 함수에 대한 piecewise linear 표현 방법. $\theta_{pq1}, \dots, \theta_{pq8}$ 은 이 퍼지 멤버십 함수를 나타내는 파라미터로 학습 과정을 통하여 수정되고 결정된다

Fig. 2. The piecewise linear representation of fuzzy membership function. $\theta_{pq1}, \dots, \theta_{pq8}$ are parameters to represent this fuzzy membership function and to be modified in learning process.

FNN의 두 번째 계층에는 MIN 노드가 있다. MIN 노드는 if-then rule의 조건부에서의 AND 연산을 수행하는 역할을 한다. MIN 노드 함수를 p번째 규칙에 대하여 나타내면 다음과 같다.

$$v_p = \min_{u_\pi} (u_{p1}, \dots, u_{pM}) \quad \text{when } u_\pi \leq u_p \text{ for all } i, j \in [1, M], \quad (2)$$

이식에서 M 은 MIN 노드의 입력 수이고, u_{pk} ($k \in [1, M]$)는 IMF 노드의 출력 (다시 말해서 MIN 노드의 입력), 그리고 v_p 는 MIN 노드의 출력이다.

많은 경우, 조건부의 AND 연산은 식 (2)와 같이 minimum 연산으로 구현된다. 사용될 수 있는 다른 연산자로는 softmin 연산자¹³⁾와 "include in" 연산자¹³⁾가 있는데, 이 연산자들은 극한에서 minimum 연산과 거의 같은 결과를 만들어 내는 한편, 모든 점에서 미분가능하다는 특징을 갖고 있다. 바로 미분가능하다

는 점이 학습 과정에서 중요하게 작용한다. 이 문제에 대해서는 제3장에서 보다 자세히 설명하기로 한다.

FNN의 세 번째 계층의 노드들은 OMF(Output Membership Function)라고 부른다. If-then fuzzy rule 결론부(consequence)의 출력 퍼지 멤버십 함수는 조건부에서의 연산 결과에 따라 전체 추론에 기여하는 바가 달라지게 된다. 그림 3에 나타낸 바와 같이 OMF 노드는 CUT 연산을 통하여 출력 퍼지 멤버십 함수를 제한하게 된다.

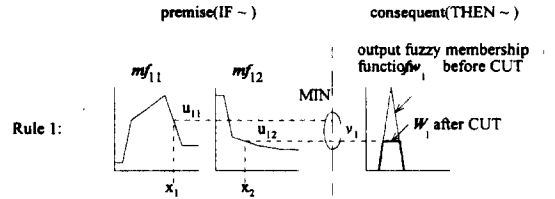


그림 3. OMF 노드에서 CUT 연산이 이루어지는 과정. v_1 값이 출력 퍼지 멤버십 함수 fw_1 을 cut하여, 식(3) OMF의 출력인 W_1 이 된다

Fig. 3. CUT operation done in OMF. v_1 cuts an output fuzzy membership function fw_1 , which becomes the output of OMF, W_1 , as in (3).

식으로 나타낸 p번째 규칙에 대한 OMF 노드 함수는 다음과 같다.

$$W_p = \text{OMF}(v_p; fw_p) \quad (3)$$

이 식에서 v_p 는 MIN의 출력, fw_p 는 출력 퍼지 멤버십 함수, W_p 는 OMF 노드의 출력이다. 주의할 점은 W_p 가 scalar값이 아니라는 것이다.

FNN 네 번째 계층의 노드는 DEF라고 부르는데, 이 이름은 비퍼지화(defuzzification)에서 유래한다. 대부분의 경우, 퍼지 추론 결과로 퍼지 값이 아닌 하나의 확정 값(scalar)을 얻게 되는데, 이 값은 각 규칙의 연산 결과 W_p 들을 maximum 연산으로 총합하고 이를 비퍼지 화하여 구해지는 값이다. 바로 이 두 가지 연산, maximum 연산과 비퍼지화 연산이 DEF 노드에서 행하여 진다.

그림 4에 본 논문에서 사용된 비퍼지화 방법인 COG(Center Of Gravity)방법을 나타내었다.그림 4에서 $y \in [0, y_b]$ 를 출력 공간(output space)에 대한 좌표로, z 를 멤버십 값에 대한 좌표로 사용하였다.

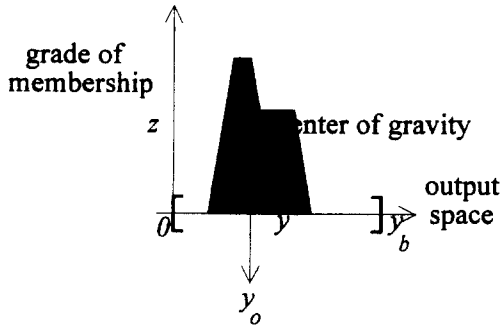


그림 4. 비퍼지화에 사용된 COG 방법. $y \in [0, y_b]$ 이 출력 변수 좌표로, z 가 멤버십 값의 좌표로 사용되었고, y_0 는 COG한 결과이다

Fig. 4. The COG method in defuzzification. $y \in [0, y_b]$ is the coordinate of output space, z is the grade of membership and y_0 is COG output.

COG 방법은 다음과 같은 적분이나 누적 형태의 식으로 나타낼 수 있다.

$$\begin{aligned}
 y_0 &= \text{COG}(\text{final result of fuzzy inference}) \\
 &= \frac{\int_0^{y_b} y \cdot z \, dy}{\int_0^{y_b} z \, dy} \quad (4) \\
 &\approx \frac{\sum_{i=1}^N i\delta y \cdot z(i\delta y) \cdot \delta y}{\sum_{i=1}^N z(i\delta y) \cdot \delta y}
 \end{aligned}$$

여기에서,

z : grade of membership

y : coordinate of output space

$$N = \frac{y_b}{\delta y}$$

계산상의 편의를 위하여, $\delta y = 1$ 과 $z(i\delta y) = z_i$ 을 가정하면, y_0 는 다음과 같다.

$$y_0 = \frac{\sum_{i=1}^N i \cdot z_i}{\sum_{i=1}^N z_i} = \text{COG}(z_1, \dots, z_N) \quad (5)$$

그러므로 COG는 N z_i 의 함수이다. 이 경우, 식 (3)의 W_p 는 N 개의 scalar값에 대한 집합 $\{w_{p1}, \dots, w_{pN}\}$ 으로 표현할 수 있다. 이로부터 w_{pi} 는 다음과 같다.

$$w_{pi} = \text{OMF}_i(v_p; fw_{pi}), i \in [1, N], \quad (6)$$

여기에서 v_p 는 MIN의 출력이고 fw_{pi} 는 $y = i\delta y$ 에서의 출력 퍼지. 멤버십 함수 fw_p 에 대한 scalar값이다. 주의할 점은 식(6)에서 OMF 대신에 OMF_i 가 사용된 것인데, 이것은 W_p 의 i 번째 값을 출력으로 얻게 됨을 강조하기 위함이다.

결과적으로 DEF 노드는 N 개의 MAX 종속 노드와 하나의 COG 종속 노드로 나눌 수 있으며, 이를 그림으로 나타내면 그림 5와 같다. MAX 종속 노드의 노드 함수는 $y = i\delta y$ 에서 다음과 같다.

$$\begin{aligned}
 z_i &= \text{MAX}(w_{i1}, \dots, w_{iL}) \\
 &= w_{ki} \quad \text{when } w_{ki} \geq w_{jn} \quad \text{for all } k, n \in [1, L]. \quad (7)
 \end{aligned}$$

이 식에서 L 은 규칙의 개수이고 $w_{jn}(j \in [1, L])$ 은 $y = i\delta y$ 에서의 W_j 값, 그리고 z_i 는 MAX 종속 노드의 출력이다.

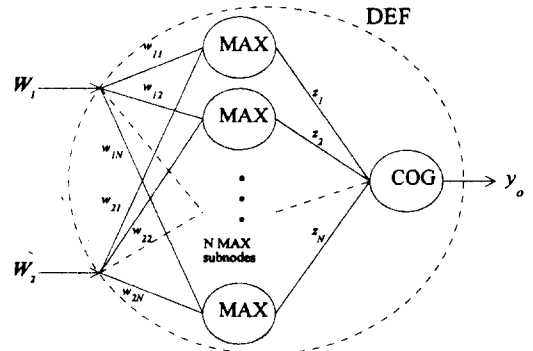


그림 5. DEF 노드의 내부 구성. DEF 노드는 N 개의 MAX 종속 노드와 하나의 COG 종속 노드로 나눌 수 있다. W_1 와 W_2 는 scalar 값이 아니고 y_0 는 scalar값이다.

Fig. 5. The decomposition of the DEF node. It is decomposed into N MAX subnodes and one COG subnode. In this figure, W_1 and W_2 are non-scalar values and y_0 is a scalar value.

COG를 사용하는 본 방법이 FNN의 구성에 있어서 필수적인 것은 아니므로 이외의 다른 비퍼지화 방법을 사용할 수도 있다. 실제로 FNN의 구현에 있어서 본 논문에서 사용한 DEF는 다소 복잡한 연산을 요구하고 있다. 분석을 위해서 보편적으로 사용되는 COG방법을 이용했을 뿐, 보다 간단한 방법으로도 유사한 결과를 얻을 수 있다^{[2][4]}. 그러나, 예전의 연구^[18]를 통하여, 우리는 COG 방법에 근거하는 퍼지 추론이, 그 복

잡성에도 불구하고 분석적인 면에서 하나의 지침이 된다고 고려되는 바, 본 논문에서는 이 방법을 사용하였다. 본 FNN의 구현에 있어서는 구성의 복잡성과 연산의 정확성 사이에서 타협 점을 찾아야 하는데, 그 이유는 N을 크게 할 수록 식(4)에서의 근사값이 보다 정확해지기는 하지만, 그만큼 MAX 중속 노드의 개수가 증가하여 결과적으로 연산의 복잡도를 증가시키기 때문이다.

III. FNN의 학습 과정

퍼지 모델을 만드는 초기 과정으로, 모델에 대한 규칙 개수, 입력 개수 그리고 초기 규칙의 파라미터 - 식(1)의 *mf*, 식(3)의 *fw* - 를 결정해야 한다. 이 초기화 과정에는 대상 시스템에 대한 사용자의 정량적 지식이나 경험, 또는 수치 데이터가 사용된다. 학습 과정에서는 대상 시스템으로부터 얻어지는 학습 패턴을 이용하여 *mf* 값을 조정해 나가게 된다. 본 논문에서는 FNN의 파라미터 조정을 위하여 BP(Back Propagation) 알고리즘을 사용한다. 모든 영역에서 FNN의 노드 함수들이 모두 미분가능하다고 가정하면(실제로는 많은 부분이 그렇지 않지만) 전체 시스템의 cost 함수는 다음과 같다.

$$E = \frac{1}{2} (y_0 - t)^2, \tag{8}$$

이 식에서 y_0 는 FNN의 최종 출력이며, t 는 대상 시스템의 목표 값이다.

식(1)에서의 θ_{pqr} , 다시 말해서 p 번째 규칙, q 번째 입력에 대한 r 번째 파라미터의 조정은 다음과 같은 식에 기반을 둔다.

$$\Delta\theta_{pqr} = -\eta \frac{\partial E}{\partial \theta_{pqr}}, \tag{9}$$

이 식에서 η 는 학습 지수(learning coefficient)이다. 식(9)를 chain rule을 이용하여 전개하면,

$$\begin{aligned} \Delta\theta_{pqr} &= -\eta \frac{\partial E}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \\ &= -\eta \frac{\partial E}{\partial v_p} \frac{\partial MIN}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \\ &= -\eta \left(\sum_{i=1}^N \frac{\partial E}{\partial w_{\pi}} \frac{\partial OMF_i}{\partial v_p} \right) \frac{\partial MIN}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \\ &= -\eta \left(\sum_{i=1}^N \frac{\partial E}{\partial z_i} \frac{\partial MAX}{\partial w_{\pi}} \frac{\partial OMF_i}{\partial v_p} \right) \frac{\partial MIN}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \end{aligned}$$

$$\begin{aligned} &= -\eta \left(\sum_{i=1}^N \frac{\partial E}{\partial y_0} \frac{\partial COG}{\partial z_i} \frac{\partial MAX}{\partial w_{\pi}} \frac{\partial OMF_i}{\partial v_p} \right) \frac{\partial MIN}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \\ &= -\eta (t - y_0) \left(\sum_{i=1}^N \frac{\partial COG}{\partial z_i} \frac{\partial MAX}{\partial w_{\pi}} \frac{\partial OMF_i}{\partial v_p} \right) \frac{\partial MIN}{\partial u_{pq}} \frac{\partial IMF}{\partial \theta_{pqr}} \end{aligned} \tag{10}$$

이 식에서 OMF 노드 함수는 식(3)이 아닌 식(6)이 사용되었으며, DEF 노드는 식(7)과 식(5)를 이용하여 풀어서 적용하였다. 결과적으로 얻어진 식(10)은 전체 FNN에 대한 BP의 일반식이 된다. 앞서 2장에서 언급한 바와 같이 학습에 사용된 파라미터는 IMF 노드의 변수로 제한되어 있다. 현재 OMF 노드의 단순화와 이 노드의 파라미터도 학습에 이용하는 방식에 대한 연구가 계속되고 있다.

식(10)에서 모든 노드 함수는 미분가능한 것으로 가정하였으나, 사실 MIN, MAX, IMF 노드 함수에는 그 미분에 있어 불연속이 존재한다. *Softmin* 연산자^[3]는 모든 영역에서 미분가능하지만, 실제 사용하기에는 복잡한 면이 있기 때문에 본 논문에서는 "include-in" 연산자^[13]와 비슷한 *fuzzymin* 연산자를 사용하였다. 이에 대한 미분형은 다음과 같다.

$$\frac{\partial}{\partial u_{pk}} MIN(u_{p1}, \dots, u_{pM}) = MAX(A + v_p \frac{-u_{pk}}{A}, 0), k \in [1, M]. \tag{11}$$

이 식에서 v_p 는 식(2)에 언급된 바와 같이 MIN 노드의 출력이다. 그리고 A 는 fuzziness를 조절하는 파라미터이다. 이 연산자의 미분 특성은 그림 6에 나타낸 바와 같은데, v_p 가 최소값이므로 u_{pk} 가 v_p 보다 큰 영역에서만 의미를 가지며, 이 영역에서의 미분값은 불연속 점이 없다.

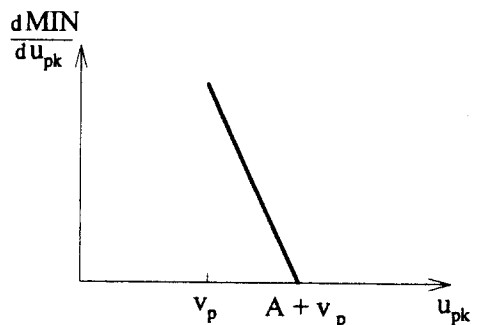


그림 6. fuzzymin 연산자의 미분 특성
Fig. 6. Derivative of fuzzymin operator.

OMF 노드의 cut연산은 minimum 연산의 연장이

므로 식 (11)로 미분식을 대신할 수 있으며, MAX 노드도 유사한 방식을 사용하여 처리된다.

그러나, IMF 노드 함수의 미분에서 나타나는 불연속성은 이러한 형태의 함수를 멤버십 함수로 사용하는 것 피하기 힘든 문제이다. 그런데 불연속성은 학습 과정에서 최적 값을 중심으로 진동(oscillation)하는 원인이 된다. 본 논문에서는 학습이 진행됨에 따라 진동 폭을 줄여 나가서 학습의 결과가 최적 값에 수렴하도록 하는 heuristic을 그림 7의 흐름도와 같이 제안하였다.

따라서 본 논문에서는 piecewise linear IMF 노드 함수를 그대로 사용하는 대신, 그림 7에 흐름도로 나타낸 바와 같은 heuristic 방법을 통하여 학습 과정 중 야기될 수 있는 수렴 문제를 극복하도록 하였다.

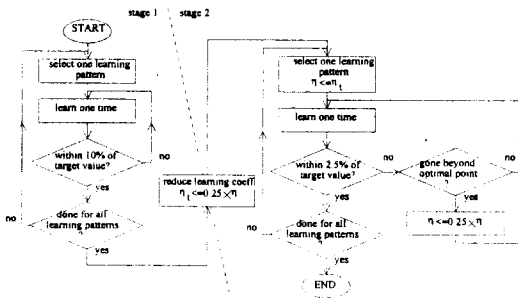


그림 7. FNN의 학습 과정에 사용된 heuristic 방법. 파라미터의 조정이 두개의 단계를 거쳐 이루어진다

Fig. 7. The proposed heuristic used in FNN learning. Rough parameter adjustment is done in stage 1 and fine tuning is done in stage 2.

그림 7의 stage 1에서는 FNN의 출력이 목표치의 10%에 도달할 때까지 파라미터들을 조정한다. 이후 학습 지수 η 를 1/4로 줄인다. Stage 2에서는 FNN의 출력이 목표치의 2.5%안에 들도록 파라미터들을 다시 세조정한다. 식(10)의 학습 규칙이 수렴 문제를 발생하는 것으로 판단되면, 다시 η 값을 줄인다. 만약 본 방식에 보다 많은 수의 stage를 포함시킨다면 학습 시간은 길어지겠지만, 보다 정확한 결과값을 얻을 수 있을 것이다.

IV. 실험 결과

본 논문이 제안하는 FNN을 다른 연구¹¹⁾(이하

Sugeno 모델)와 비교함으로써 그 타당성을 검토하기로 한다. Sugeno 모델은 다음과 같은 비선형 시스템을 예로 들어 모델링 방법을 설명하고 있다.

$$y = (1 + x_1^2 + x_2^{15})^2 \tag{12}$$

Sugeno 모델에서는, 소위 RC(regularity criterion)을 사용하여 입력의 가능한 조합 중 학습을 위한 입력 집합을 골라내고, S(c) criterion을 사용하여 규칙의 개수와 입력 공간을 분할하였다. 그 이후, 얻어진 입력 집합을 근사하여 사다리꼴형의 퍼지 집합을 만든 뒤, 이를 파라미터의 초기값으로 삼았다.

학습 행위에 따른 결과 비교를 위해 본 논문이 제안하는 FNN에도 같은 규칙과 데이터를 사용하였다. Sugeno 모델은 cost 함수로 mean square error criterion PI(Performance Index)를 사용하여, 새로운 입력 변수 집합을 고르는데 사용하였다. 학습 과정을 거친 결과, Sugeno 모델은 PI값이 0.318에서 0.079로 줄어드는 결과를 보였다. 본 FNN의 경우 BP 알고리즘과 heuristic 처리 과정을 통하여 PI값이 0.0292 이하로 줄어드는 결과를 보였다. 이 학습 방법은 대상 시스템에 대한 전처리 분석을 통하여 FNN의 초기화를 수행하므로 일종의 supervised learning이라고 말할 수 있다.

Sugeno의 모델은 규칙 개수와 입력 집합을 얻기 위해, 전체 알고리즘을 반복하게 된다. 만약 입력과 출력에 대한 학습 패턴으로부터 직접 퍼지 규칙을 구할 수 있다면, 학습과 연산에 소요되는 시간과 노력을 절약할 수 있을 것이다. 이러한 방식을 본 논문에서는 직접 규칙 생성(direct rule generation)이라 부르기로 한다. 이 방법은 일종의 unsupervised learning이라고 할 수 있다. 직접 규칙 생성법의 특징은 초기화 과정이 간단하다는 것으로, 앞서의 Sugeno 모델에서와 같은 입출력 데이터에 대한 전처리 작업이 필요하지 않다는 것이다. 다시 말해서, 모델링하고자 하는 대상 시스템을 일종의 블랙 박스로 보고 가능한 모든 출력 공간에 대하여 필요한 수만큼의 규칙 수를 정한 뒤, 입출력 학습 패턴을 이용하여 직접 학습 과정에 들어가게 된다. 따라서 매 출력에 대하여 가능한 출력 공간을 모두 고려하기 위해서 보다 많은 수의 규칙이 필요하다는 단점이 있는 반면, 전처리 작업을 필요로 하지 않기 때문에 전체 학습 과정을 단순화시킬 수 있다는 장점을 갖는다. 실제 학습 과정은 3장에 설명한 바와 동일하다.

직접 규칙 생성의 유용성을 알아보기 위하여, 식(12)의 수치 모델을 다시 대상으로 하였다. 먼저 식(12)을 이용하여, 여러 입력 (x_1, x_2) 에 대한 출력 y 값들을 구하여, 학습 패턴을 만들었다. 이 패턴을 바탕으로 직접적인 규칙 생성과 조정을 수행한다. 이 방식에서는 규칙 개수를 미리 정해 놓지 않기 때문에 규칙 수가 약간 증가하는 점은 감수해야 한다. 그림 8에서 보이는 바와 같이 if then 퍼지 규칙의 결론부에 대해 7 fuzzy word {NB, NM, NS, ZE, PS, PM, PB}를 사용하여, 각 fuzzy word에 대하여 하나의 규칙을 할당, 결과적으로 전체 규칙 수는 7개가 된다. 직접 규칙 생성에서는 일체의 전처리가 없기 때문에, IMF 노드의 mf 에 대한 초기 파라미터는 그림 9의 (c)와 같이 단순한 형태로 잡아 두었다.

NB NM NS ZE PS PM PB

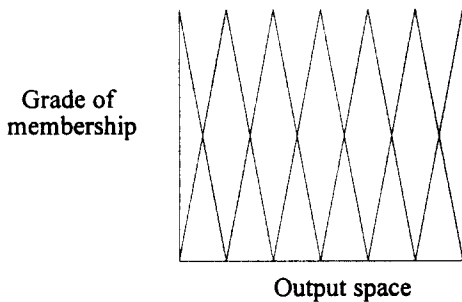


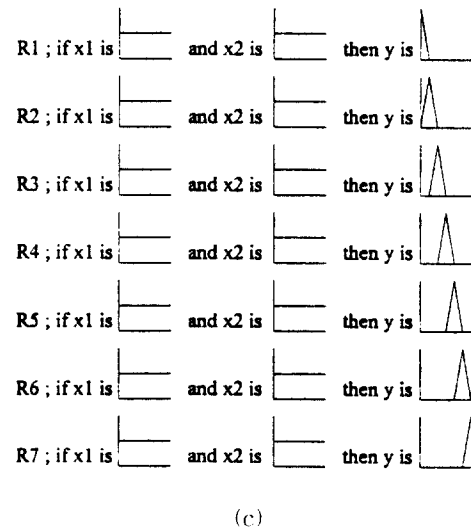
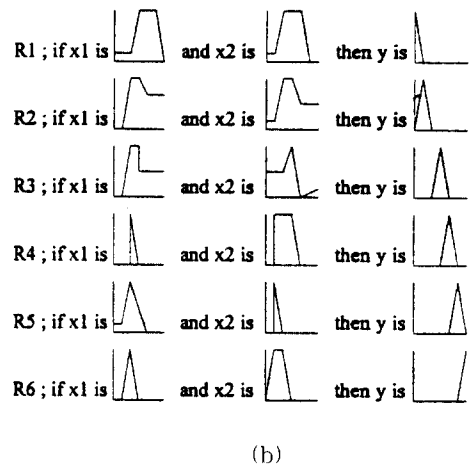
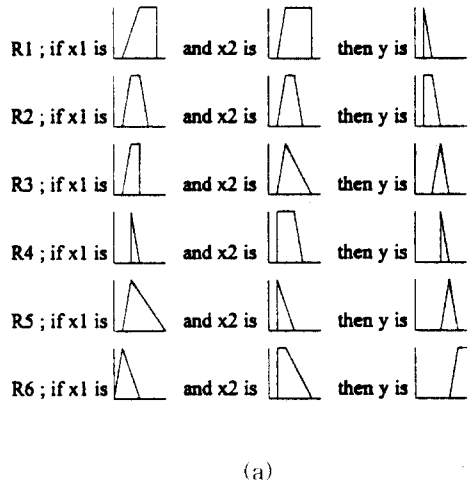
그림 8. 직접 규칙 생성에 사용된 7 fuzzy words {NB, NM, NS, ZE, PS, PM, PB}

Fig. 8. 7 fuzzy words {NB, NM, NS, ZE, PS, PM, PB} used in direct rule generation.

이러한 초기값으로 학습을 수행한 결과, PI값이 0.0111이 됨을 알 수 있었다. 보다 복잡한 시스템에 대하여 이러한 방식을 적용한다면 출력 공간을 7 fuzzy word 이상으로 세분해야 될 것이다. 이상의 결과를 그림 9, 10와 표 1에 정리하여 나타내었다.

그림 9는 학습 결과 얻어진 퍼지 규칙을 그림으로 나타낸 것이다. 그림 9의 (a)는 비교의 대상이 된 Sugeno 모델의 학습 결과이고 (b)는 Sugeno 모델과 같은 초기화를 거친 FNN의 학습 결과이다. 또한 그림 9의 (c)는 앞서 언급한 바와 같이 직접 규칙 생성 방식을 위한 초기 상태를 나타낸 것이며, (d)는 직접 규칙 생성 방식에 따른 학습 결과 규칙을 나타낸 것이다. 그림 9에서 알 수 있는 바와 같이, 멤버십 함수의 모양을

참조하면 만들어진 FNN을 언어적으로 표현할 수 있다.



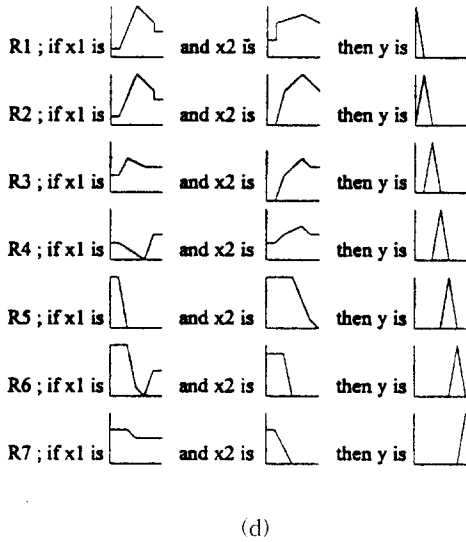


그림 9. 학습 결과인 규칙 형태. (a) Sugeno 모델, (b) Sugeno 모델과 같은 초기화를 거친 FNN의 학습결과 (c) 직접 규칙 생성의 초기 규칙 형태 (d) 직접 규칙 생성의 학습 결과

Fig. 9. Fuzzy rules after learning. (a) Sugeno's model, (b) supervised model, (c) initial fuzzy rules for direct rule generation (unsupervised) and (d) the result fuzzy rules of direct rule generation.

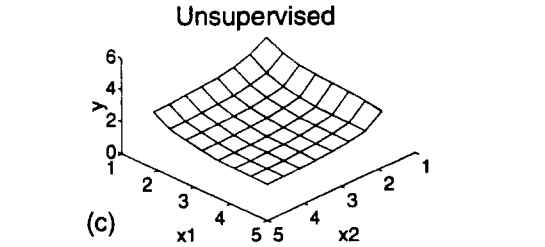
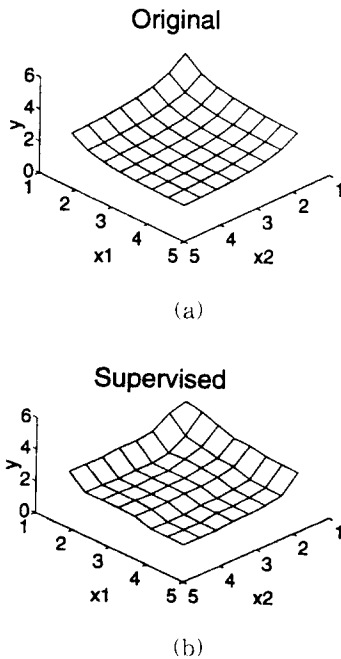


그림 10. 모델링 결과 비교. (a) 대상 시스템 본래의 입출력 관계, (b) Sugeno모델과 같은 초기 조건을 사용하여 모델링한 경우 (c) 직접 규칙 생성 방법을 사용한 모델링의 경우

Fig. 10. Comparison of model surfaces (a) original numerical system surface, (b) model surface learned with the same initial condition as in Sugeno's case and (c) model surface learned in direct rule generation.

그림 10은 대상 시스템을 3차원 표면 그래프로 나타내고 본 FNN의 학습 결과 두 가지(Sugeno 모델과 같은 초기치를 사용하는 경우와 직접 규칙 생성을 적용한 경우)의 표면 그래프를 비교한 것이다. 표 1에 이 결과값을 비교 정리하였다.

표 1. 학습 결과에 대한 비교표

Table 1. Comparison to the network of previous research.

	Sugeno's	Supervised	Unsupervised
Target System	$y = (1 + x_1^2 + x_2^6)^2$		
PI	0.079	0.0292	0.0111
No. of Rules	6	6	7
Initial condition	Rules and MFs	Rules and MFs	None

본 FNN은 비디오 카메라의 자동 초점 조절 장치 (auto focusing system)에 응용되었다¹⁹⁾. 비디오 카메라의 자동 초점 조절에는 여러 가지 변수가 작용하는데 그 인과 관계가 다소 복잡하고 비선형적이며, 또한 사용자의 주관적인 요소가 많기 때문에 FNN이 적용되기에는 적절한 분야이다. 사용된 입력 변수로는 초점값이라고 불리는 휘도 신호의 고주파 성분을 화면을 분할하여 누적한 값과 분할 화면당 휘도 신호의 합,

줌(zoom) 정도등이 사용되었고, 이 때의 출력값은 초점 조절용 모터의 회전 속도와 회전 방향, 그리고 움직임 정보 등이 된다. 이런 방식으로 구현된 자동 초점 조절 장치는 기존의 장치에 비하여 보다 다양한 화면 상황에서 적절한 초점 조절이 가능해 진다.

자동 초점 조절 장치를 대상으로 하는 경우에는 본 FNN의 학습 과정까지 함께 하드웨어로 구현하는 것은 무의미하기 때문에 실제로 구현된 하드웨어는 학습 기능이 제거되어 설계되었다. 이 때, 학습 과정은 별도의 시스템을 사용하여 off-line에서 수행되었고 그 결과를 구현된 하드웨어에 적재하여 퍼지 추론 과정은 실시간에서 이루어질 수 있도록 하였다. 다만 추론 중에 발생하는 각 규칙의 결과를 참조할 수 있도록 하여, 그 결과를 학습 과정에 반영할 가능성을 마련해 두는 것으로 그 대안을 삼았다. 화상 신호의 특징을 고려하여 하드웨어는 6b 해상도를 갖도록 하였으며, 그 구조는 순차적 파이프 라인 처리 방식을 사용하여 설계의 복잡도와 크기를 줄일 수 있도록 하였다.

V. 결 론

본 논문을 통하여, 퍼지 추론 과정과 신경망의 학습 기능을 결합하는 새로운 형태의 FNN을 제안하였다. If then 퍼지 추론의 각 과정이 FNN의 노드로 구현되었고, 학습 과정에서는 FNN의 파라미터들이 BP 알고리즘과 제안하는 heuristic 처리 과정을 기반으로 하여 조정되고 결정된다. 비선형 수치 시스템에 대한 모델링 실험을 통하여 본 FNN의 타당성을 나타내었다.

학습 과정에 사용된 학습식과 heuristic 과정에 대해서는 개선의 여지가 있다고 여겨지는 바 추가의 연구를 진행 중에 있다. 본 논문에 제안된 FNN은 하드웨어로 구현되어 비디오 카메라의 자동 초점 조절 장치에 적용되었으며, 또한 추후의 여러 제어 시스템에 적용될 예정이다.

참 고 문 헌

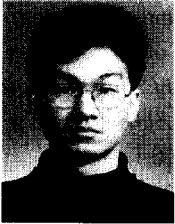
[1] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, pp. 7-31, Feb. 1993.
 [2] J. R. Jang, "Self-learning fuzzy controllers

based on temporal back propagation," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 714-723, Sept. 1992.

- [3] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 724-740, Sept. 1992.
 [4] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 801-806, Sept. 1992.
 [5] T. Takagi and M. sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 1, pp. 116-132, 1985.
 [6] H. Takagi, "Fusion technology of fuzzy theory and neural networks -- survey and future directions --," in *Proc. Int. Conf. Fuzzy Logic and Neural Networks, IIZUKA'90*, vol. 1, pp. 13-26, July 1990.
 [7] W. Pedrycz, "Fuzzy sets and neuro computations as foundations of pattern recognition," in *Proc. Int. Conf. Fuzzy Logic and Neural Networks, IIZUKA'90*, Tutorials, pp. 89-120, July 1990.
 [8] T. Yamakawa, "Pattern recognition hardware system employing a fuzzy neuron," in *Proc. Int. Conf. Fuzzy Logic and Neural Networks, IIZUKA'90*, vol. 2, pp. 943-948, July 1990.
 [9] P. Simpson, "Fuzzy min-max neural networks-Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776-786, Sept. 1992.
 [10] P. Simpson, "Fuzzy min max neural networks-Part 2: Clustering," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 32-45, 1993.
 [11] H. Ishibuchi, R. Fujioka, and H. Tanaka, "Neural networks that learn from fuzzy if then rules," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 85-97, 1993.
 [12] K. Uehara and M. Fujise, "Multistage fuzzy inference formulated as linguistic-truth-value propagation and its learning algo-

- rithm based on back-propagating error information," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 205-221, 1993.
- [13] W. Pedrycz and A. F. Rocha, "Fuzzy-set based models of neurons and knowledge-based networks," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 254-266, 1993.
- [14] C. -T. Lin and C. S. G. Lee, "Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems," *IEEE Trans. Fuzzy Systems*, vol. 2, no. 1, pp. 46-63, Feb. 1994.
- [15] H. K. Kwan and Y. Cai, "A fuzzy neural network and its application to pattern recognition," *IEEE Trans. Fuzzy Systems*, vol. 2, no. 3, pp. 185-193, Aug. 1994.
- [16] J. C. Bezdek, Guest Editorial of *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 641-642, Sept. 1992.
- [17] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
- [18] S.Jung, S.Moon, J.Chong, and W.Kim, "A study on the efficient defuzzifier," *Proceedings of KITE Fall Conf. '91*, vol. 14, no. 2, pp. 682-685, 1991.
- [19] Y. Lee, S. -I. Jang, K. Chung, D. Lee, W. Kim, and C. -W. Lee, "A fuzzy-control processor for automatic focusing," *IEEE Trans. Consumer Electronics*, vol. 40, no. 2, pp. 138-144, May 1994.

 저 자 소 개



鄭載卿(正會員)

1965년 9월 2일생. 1988년 2월 서울대학교 전자공학과 졸업. 1990년 2월 동 대학원 석사 학위 취득. 1990년 3월 ~ 현재 동 대학원 박사과정. 주관심 분야는 퍼지 추론기와 신경망 회로 구현

李東潤(正會員)

1968년 11월 11일생. 1991년 2월 서울대학교 전자공학과 졸업. 1993년 2월 동 대학원 석사 학위 취득. 1993년 3월 ~ 현재 동 대학원 박사과정. 주관심 분야는 데이터 변환기



鄭起旭(正會員)

1969년 7월 15일생. 1992년 2월 서울대학교 전자공학과 졸업. 1994년 2월 동 대학원 석사 학위 취득. 1994년 3월 ~ 현재 동 대학원 박사과정. 주관심 분야는 RF 통신

金元燦(正會員) 현재 서울 대학교 전자공학과 교수