

확장된 퍼지 가중치를 갖는 퍼지 신경망 학습알고리즘^{*}

손영수* · 나영남** · 배상현***

A learning algorithm of fuzzy neural networks with extended fuzzy weights

Young-Su Son*, Young-Nam Na**, Sang-Hyun Bae***

ABSTRACT

In this paper, first we propose an architecture of fuzzy neural networks with triangular fuzzy weights. The proposed fuzzy neural network can handle fuzzy input vectors. In both cases, outputs from the fuzzy neural network are fuzzy vectors. The input-output relation of each unit of the fuzzy neural network is defined by the extension principle of Zadeh. Also we define a cost function for the level sets(i.e., α -cuts) of fuzzy outputs and fuzzy targets. Then we derive a learning algorithm from the cost function for adjusting three parameters of each triangular fuzzy weight. Finally, we illustrate our approach by computer simulation examples.

I. 서 론

최근 신경망을 퍼지화하기 위한 다양한 방법들이

제안되어 왔다. 직접적인 퍼지화를 위한 한가지 방법은 퍼지 숫자로 종래의 신경망 구조에서 실제 입력값과 목표값을 확장하는 것이다. Ishibuchi et al. [4]는 퍼지 입력벡터에 대한 다계층 피드포워

^{*} 이 논문은 한국과학재단 지정 지역협력 연구센터인 조선대학교 수송기계부품 공장 자동화 연구센터의 연구 경비 지원에 의해 연구 되었습.

* 한려산업대학교 전자계산학과

** 조선대학교 대학원 전산통계학과

*** 조선대학교 자연과학대학 전산통계학과

드(feedforward) 신경망의 구조를 제안했다. 그리고 그 구조는 [5, 7]에서 퍼지 if-then 규칙의 수단으로 응용되었다. 다계층 피드포워드 신경망의 결합 가중치(connection weights)는 Hayashi et al. [3]과 Ishibuchi et al. [6]에서 퍼지화 되었다. 또한 Ishibuchi et al.이 삼각의 퍼지 가중치에 대한 명료한 학습 알고리즘을 유지하는 동안 Hayashi et al.은 델타(delta) 규칙을 퍼지화했다. 본 논문에서는, 실제 입력벡터에 대한 다중입력과 단일출력 퍼지 신경망을 논의했던 이전의 작업을 퍼지입력벡터에 대한 다중입력과 다중출력 퍼지신경망으로 확장하고, 퍼지입력벡터에 대한 삼각의 퍼지 가중치를 갖는 퍼지 신경망의 구조를 제안한다. 각 장치의 입출력 관계는 Zadeh [9]의 확장원칙으로 정의된다. 퍼지 신경망의 출력은 퍼지 가중치와 퍼지 입력의 레벨셋에 대한 간격계산 [1]에 의해 산술적으로 계산된다. 또한 본 논문은 퍼지 출력과 퍼지 목표의 레벨셋에 대한 코스트 함수를 정의한다. 이때의 명료한 학습 알고리즘은 확장된 삼각의 퍼지 가중치의 세 매개값을 조정하기 위한 코스트 함수로부터 유추된다. 마지막으로, 본 논문에서는 산술적인 예를 들어 컴퓨터 시뮬레이션을 이용해 평가, 설명한다.

2. 퍼지 신경망 구조

2.1 퍼지 값의 조작

퍼지 신경망의 구조를 설명하기 전에, 확장원칙 [9]에 의해 정의된 퍼지값의 조작을 본 논문에서는, 제각기 소문자(e. g, a, b, c, ...)와 대문자(e. g,

A, B, C, ...)로 제값과 퍼지값을 표시한다. 본 논문의 확장된 퍼지값은 다계층 피드포워드 신경망의 입력벡터와 결합 가중치가 퍼지화 되었기 때문에, 퍼지값의 덧셈, 곱셈, 비선형 함수는 식(2.1)(2.2)(2.3)과 같이 정의한다.

$$\begin{aligned}\mu_{A+B}(z) &= \max\{\mu_A(x) \vee \mu_B(y) \mid z \\ &= x+y\} \quad (2.1)\end{aligned}$$

$$\begin{aligned}\mu_{AB}(z) &= \max\{\mu_A(x) \wedge \mu_B(y) \mid z \\ &= xy\}, \quad (2.2)\end{aligned}$$

$$\mu_{Net}(z) = \max\{\mu_{Net}(x) \mid z = f(x)\}, \quad (2.3)$$

A, [B, Net] 는 퍼지 값들이고, $\mu(\cdot)$ 는 각 퍼지값의 소속 함수를 나타낸다. 그리고 \wedge 는 최소 연산자이고 $f(x)=1/(1+\exp(-x))$ 는 퍼지 신경망의 숨겨진층과 출력층의 활성화함수이다. 이러한 연산자들은 그림 1과 2에서 설명된다.

위의 퍼지값들의 조작은 수치적으로 단계집합(i. e., α -cuts)에서 수행된다. 퍼지값 X의 h-단계 집합은 아래와 같이 정의된다.

$$\begin{aligned}[X]_h &= \{x \mid \mu_x(x) \geq h, x \in \mathbf{R}\} \\ &\text{for } 0 < h \leq 1, \quad (2.4)\end{aligned}$$

여기에서 $\mu_x(x)$ 는 X의 전체 함수이고 R은 모든 실제값의 집합이다. 퍼지값의 단계집합의 간격이 가까와 질때, $[X]_h$ 를 아래와 같이 정의한다.

$$[X]_h = [[X]_h^-, [X]_h^+] \quad (2.5)$$

여기에서 $]$ 과 $[$ 는 각각 h-단계 집합 $[X]_h$ 의 하한과 상한이다.

간격 계산 [1]로 부터, 위 퍼지값들의 조작은 아래와 같이 h-단계 집합을 위해 수정된 것이다.

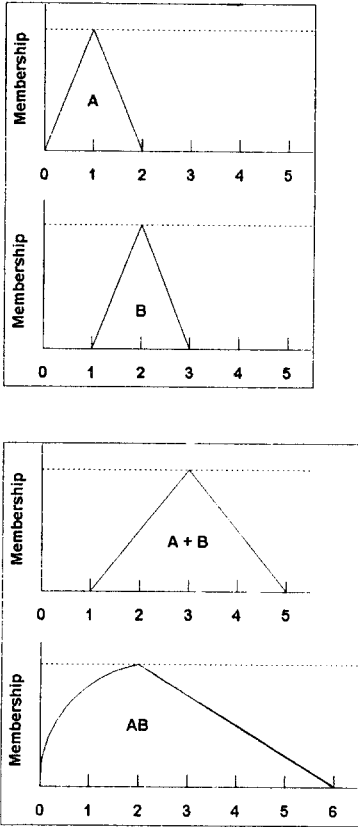


그림 1. 퍼지값 조작: 덧셈과 곱셈

$$\begin{aligned}
 [A]_h + [B]_h &= [[A]_h^l, [A]_h^r] \\
 &+ [[B]_h^l, [B]_h^r] \\
 &= [[A]_h^l + [B]_h^l, \\
 &[[A]_h^r, [B]_h^r] \quad (2.6)
 \end{aligned}$$

$$\begin{aligned}
 [A]_h \cdot [B]_h &= [[A]_h^l, [A]_h^r \cdot \\
 &[[B]_h^l, [B]_h^r] \\
 &= [\min\{[A]_h^l \cdot [B]_h^l, \\
 &[A]_h^l \cdot [B]_h^r, \\
 &[[A]_h^r \cdot [B]_h^l, A]_h^r \cdot [B]_h^r\}, \\
 &\max\{[A]_h^l \cdot [B]_h^r, \\
 &[[A]_h^r \cdot [B]_h^l, \\
 &[A]_h^r \cdot [B]_h^r\} \quad (2.7)
 \end{aligned}$$

$$f([Net]) = f([[Net]_h^l, [Net]_h^r]) \quad (2.8)$$

0 ≤ [X]_h ≤ [X]_h 일 경우 식(2.7)에서 다음과 같이 축약될 수 있다.

$$\begin{aligned}
 [A]_h \cdot [B]_h &= [\min\{[A]_h^l, [B]_h^l \cdot [A]_h^r, [B]_h^r\}, \\
 &\max\{[A]_h^r \cdot [B]_h^l, [A]_h^r \cdot [B]_h^r\}] \quad (2.9)
 \end{aligned}$$

2.2. 각 층별 입출력 관계

n_i: 입력층, n_h: 숨겨진층, n_o: 출력층을 갖는 세 계층의 피드포워드 신경망을 퍼지화 해보자. 입력 벡터, 목표 벡터, 결합 가중치와 바이어스(biases)는 퍼지화 되었다.(i.e., 확장된 퍼지값) 다음 장에서의 명확한 학습 규칙을 도출해 내기 위해서, 퍼지입력과 목표를 위한 퍼지값의 임의의 형식을 이용하는 동안 퍼지 가중치와 퍼지 바이어스를 삼각의 퍼지값내에서 제한시키게 된다. 퍼지화된 신경망의 각 층의 입출력 관계는 아래와 같이 쓰여질 수 있다.

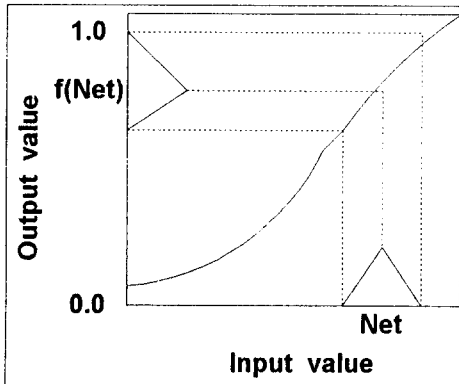


그림 2. 퍼지 활성화함수

입력층 :

$$O_{pi} = X_{pi}, i=1,2, \dots ,n_i. \quad (2.10)$$

숨겨진층 :

$$O_{pj} = f(\text{Net}_{pj}), j=1,2, \dots ,n_{11} \quad (2.11)$$

$$\text{Net}_{pj} = \sum W_{ji} \cdot O_{pi} + \theta_j \quad (2.12)$$

출력층 :

$$O_{pk} = f(\text{Net}_{pk}), k=1,2, \dots ,n_o \quad (2.13)$$

$$\text{Net}_{pk} = \sum W_{ki} \cdot O_{pj} + \theta_k \quad (2.14)$$

여기에서 X_{pi} 는 퍼지 입력, W_{ji} 와 W_{ki} 는 퍼지 가중치, θ_j 과 θ_k 는 퍼지 바이어스이다.(그림 3. 참조)

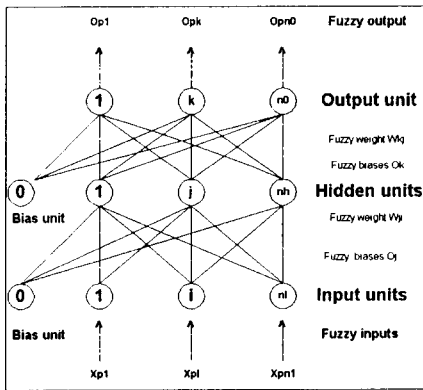


그림 3. 퍼지 신경망 구조

(2.10)-(2.14)에서의 입출력 관계는 Hayashi et al. [3]과 Ishibuchi et al. [6]에서와 같이 확장 원칙 [9]에 의해서 정의된다.

2.3 퍼지 출력의 계산

식(2.10)-(2.14)에서 각 층으로 부터의 퍼지

출력은 퍼지입력, 퍼지 가중치 그리고 퍼지 바이어스의 단계 집합으로 계산된다. 퍼지 신경망의 입출력 관계는 h-단계 집합으로 쓰여질 수 있다.

입력층 :

$$[O_{pi}]_h = [X_{pi}]_h \quad (2.15)$$

숨겨진층 :

$$[O_{pj}]_h = f(\text{Net}_{pj})_h, \quad (2.16)$$

$$[\text{Net}_{pj}]_h = \sum_{i=1}^{n_i} [W_{ji}]_h \cdot [O_{pi}]_h + [\theta_j]_h \quad (2.17)$$

출력층 :

$$[O_{pk}]_h = f(\text{Net}_{pk})_h, \quad (2.18)$$

$$[\text{Net}_{pk}]_h = \sum_{j=1}^{n_{11}} [W_{kj}]_h \cdot [O_{pj}]_h + [\theta_k]_h \quad (2.19)$$

식(2.15)-(2.19)로부터, O_{pk} 의 퍼지출력 h-단계 집합이 퍼지입력, 가중치, 바이어스로부터 계산됨을 알 수 있다. 식(2.6)-(2.9)로부터, 위의 관계는 X_{pi} 퍼지입력의 h-단계 집합이 음수가 아닐때 아래와 같이 수정된다.(즉, $0 \leq [X_{pi}]_h \leq [X_{pi}]_h$, 모든 i에 대해)

입력층 :

$$[O_{pi}]_h = [[O_{pi}]_h^L, [O_{pi}]_h^U] = [[X_{pi}]_h^L, [X_{pi}]_h^U] \quad (2.20)$$

숨겨진층 :

$$[O_{pj}]_h = [[O_{pj}]_h^L, [O_{pj}]_h^U] = [f([Net_{pj}]_h^L), f([Net_{pj}]_h^U)], \quad (2.21)$$

$$[Net_{pj}]_h^L = \sum_{i=1}^{n_i} [W_{ji}]_h^L \cdot [O_{pi}]_h^L$$

$$[W_{ji}]_h^L \geq 0$$

$$+ \sum_{i=1}^n [W_{ji}]_h^L \cdot [O_{Pi}]_h^U + [\Theta_j]_h^L, \quad [W_{ji}]_h^L < 0 \quad (2.22)$$

$$[Net_{pj}]_h^U = \sum_{i=1}^n [W_{ji}]_h^U \cdot [O_{Pi}]_h^U + [\Theta_j]_h^U, \quad [W_{ji}]_h^U \geq 0$$

$$+ \sum_{i=1}^n [W_{ji}]_h^U \cdot [O_{Pi}]_h^U + [\Theta_j]_h^U, \quad [W_{ji}]_h^U < 0 \quad (2.23)$$

출력층:

$$[O_{pk}]_h = [[O_{pk}]_h^L, [O_{pk}]_h^U]$$

$$= [f([Net_{pk}]_h^L), f([Net_{pk}]_h^U)] \quad (2.24)$$

$$[Net_{pk}]_h^L = \sum_{j=1}^{n_U} [W_{kj}]_h^L \cdot [O_{Pj}]_h^L + [\Theta_k]_h^L, \quad [W_{kj}]_h^L \geq 0$$

$$+ \sum_{j=1}^{n_U} [W_{kj}]_h^L \cdot [O_{Pj}]_h^U + [\Theta_k]_h^L, \quad [W_{kj}]_h^L < 0 \quad (2.25)$$

$$[Net_{pk}]_h^U = \sum_{j=1}^{n_U} [W_{kj}]_h^U \cdot [O_{Pj}]_h^U + [\Theta_k]_h^U, \quad [W_{kj}]_h^U \geq 0$$

$$+ \sum_{j=1}^{n_U} [W_{kj}]_h^U \cdot [O_{Pj}]_h^L + [\Theta_k]_h^U, \quad [W_{kj}]_h^U < 0 \quad (2.26)$$

3. 퍼지 신경망의 학습

3.1 코스트 함수

$T_p = (T_{p1}, T_{p2}, \dots, T_{pno})$ 을 퍼지입력 벡터 X_p 와 근사하게 될 n_0 차원 퍼지타겟이 되게 하자. 아래와 같이, k 번째 출력장치와 그에 상응하는 퍼지 타겟 T_{pk} 로부터 퍼지 출력 O_{pk} 의 h -단계 집합에 대한 코스트 함수를 정의한다.(그림 4 참조)

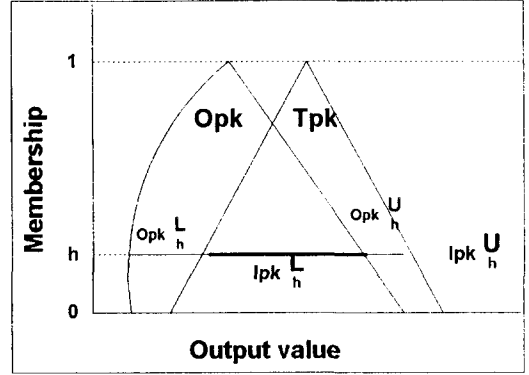


그림 4. 퍼지 출력 O_{pk} 와 퍼지 타겟 T_{pk}

$$e_{pkh} = e_{pkh}^L + e_{pkh}^U, \quad (3.1)$$

여기에서

$$e_{pkh}^L = h \cdot \frac{([T_{pk}]_h^L - [O_{pk}]_h^L)^2}{2}, \quad (3.2)$$

$$e_{pkh}^U = h \cdot \frac{([T_{pk}]_h^U - [O_{pk}]_h^U)^2}{2}. \quad (3.3)$$

코스트 함수 (3.1)에서, e_{pkh}^L 와 e_{pkh}^U 는 각각 h -단계 집합의 하한과 상한에 대한 스쿼어드 오류 (squared errors)로써 나타낼 수 있다. 코스트 함수의 정형화에서, 식(3.2)와 (3.3)에서 h 의 값에 의해 값을 갖게 된다. 퍼지 출력 벡터 O_p 와 퍼지 타겟 벡터 T_p 의 h -단계 집합에 대한 코스트 함수는 아래와 같이 정의된다.

$$e_{ph} = \sum_{k=1}^{n_U} e_{pkh}. \quad (3.4)$$

입출력 (X_p, T_p) 에 대한 코스트 함수는 아래 식으로서 얻어진다.

$$e_p = \sum_h e_{ph}. \quad (3.5)$$

3.2. 학습 알고리즘

본 논문의 마지막 부록에서 h-단계 집합에 대해 정의된 코스트 함수로부터 퍼지 신경망의 학습 알고리즘을 유추해 낸다. 코스트함수 e_m 를 감소하기 위해서 퍼지 가중치의 다른 단계 집합을 독립적으로 퍼지 가중치의 h-단계 집합을 수정한다면, 이 수정은 그림 5에서 보여진 것처럼 퍼지 가중치를 왜곡하게 된다.([3]참조) 그래서 독립적으로 퍼지 가중치의 h-단계 집합을 바꾸지 않는다. 본 논문에서는 가중치의 삼각형 모양이 그림 6에서처럼 파괴되지 않은 방법으로 각각의 퍼지 가중치의 값을 변경한다. 즉, 각 퍼지 가중치는 그것의 폭을 옮기고 바꿀수는 있지만 삼각형 모양을 파괴할수는 없도록 확장된 퍼지 가중치를 사용한다.

삼각의 퍼지 가중치가 그들의 세가지 매개값에 의해 특정화된 후에, 각 파라미터에 대한 변경 규칙을 얻어낸다. 삼각의 퍼지 가중치 W_{kj} , W_{ji} 와 삼각의 퍼지 바이어스 θ_k , θ_j 를 그들의 세가지 파라미터로 아래와 같이 표시한다.

$$\begin{aligned} W_{kj} &= (W_{kj}^L, W_{kj}^C, W_{kj}^U), \\ W_{ji} &= (W_{ji}^L, W_{ji}^C, W_{ji}^U), \end{aligned} \tag{3.6}$$

$$\begin{aligned} \theta_k &= (\theta_k^L, \theta_k^C, \theta_k^U), \\ \theta_j &= (\theta_j^L, \theta_j^C, \theta_j^U), \end{aligned} \tag{3.7}$$

여기에서 첨자 L, C, U는 삼각의 퍼지값의 하, 중, 상한을 표시한다.(그림7 참조)

퍼지 가중치 W_{kj} , W_{ji} 와 퍼지 바이어스 θ_k , θ_j 가 대칭적이라는 사실을 알수 있다. 즉,

$$\begin{aligned} W_{kj}^C &= \frac{W_{kj}^L + W_{kj}^U}{2}, \\ W_{ji}^C &= \frac{W_{ji}^L + W_{ji}^U}{2}, \end{aligned} \tag{3.8}$$

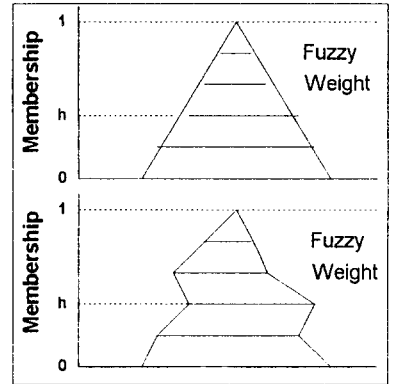


그림 5. h-단계 집합의 조정

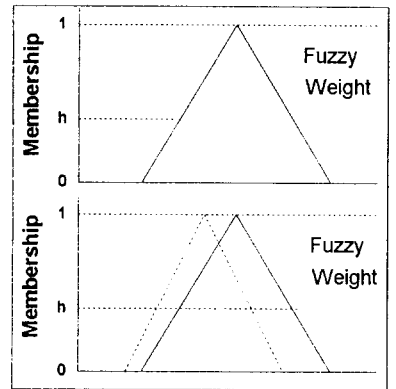


그림 6. 퍼지 가중치의 조정

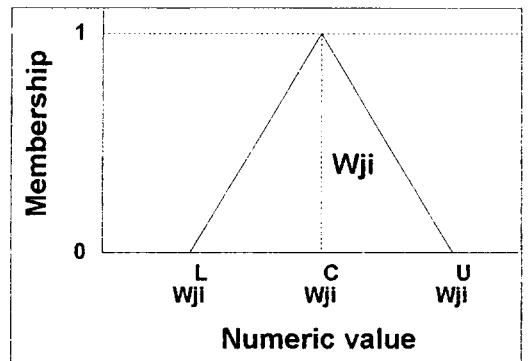


그림 7. 삼각의 퍼지 가중치 W_{ji}

$$\begin{aligned} \theta_k^C &= \frac{\theta_k^L + \theta_k^U}{2}, \\ \theta_j^C &= \frac{\theta_j^L + \theta_j^U}{2}. \end{aligned} \quad (3.9)$$

첫째, j번째 숨겨진 층과 k번째 출력 층 사이에서 퍼지 가중치 $W_{kj} = (w_{kj}^L, w_{kj}^U)$ 의 학습을 논의해 보자. Rumelhart et al. [8]의 간단한 방법에서 처럼, 아래와 같이 코스트 함수를 이용하는 각 파라미터에 대한 많은 조정을 나타낼 수 있다.

$$\begin{aligned} \Delta W_{kj}^L(t) &= -\eta \cdot \frac{\partial e_{ph}}{\partial w_{kj}^L} \\ &+ \alpha \cdot \Delta W_{kj}^L(t-1), \end{aligned} \quad (3.10)$$

$$\begin{aligned} \Delta W_{kj}^U(t) &= -\eta \cdot \frac{\partial e_{ph}}{\partial w_{kj}^U} \\ &+ \alpha \cdot \Delta W_{kj}^U(t-1), \end{aligned} \quad (3.11)$$

여기에서 η 는 학습상수(learning constant), α 는 운동량 상수(momentum constant), t 는 학습 시기를 나타낸다. 퍼지 가중치 W_{kj} 가 대칭이라고 생각한 후에, 퍼지 가중치 중심 w_{kj}^C 는 식(3.8)에 의해 특정화 된다. 식(3.10)과 (3.11)에서 각각 파생된 확실한 계산은 부록에서 다루게 된다.

식(3.10)과 (3.11)에서 파생된 것은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^L} &= \frac{\partial e_{ph}}{\partial [W_{kj}]_h^L} \cdot \frac{\partial [W_{kj}]_h^L}{\partial w_{kj}^L} \\ &+ \frac{\partial e_{ph}}{\partial [W_{kj}]_h^U} \cdot \frac{\partial [W_{kj}]_h^U}{\partial w_{kj}^L}. \end{aligned} \quad (3.12)$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^U} &= \frac{\partial e_{ph}}{\partial [W_{kj}]_h^L} \cdot \frac{\partial [W_{kj}]_h^L}{\partial w_{kj}^U} \\ &+ \frac{\partial e_{ph}}{\partial [W_{kj}]_h^U} \cdot \frac{\partial [W_{kj}]_h^U}{\partial w_{kj}^U}. \end{aligned} \quad (3.13)$$

W_{kj} 가 대칭적인 확장된 삼각의 퍼지값이기 때문에, 다음 관계는 그것의 h-단계 집합 $[W_{kj}]_h =$

$[[W_{kj}]_h^L, [W_{kj}]_h^U]$ 을 갖게 된다.

$$[W_{kj}]_h^L = w_{kj}^L \cdot \left(1 - \frac{h}{2}\right) + w_{kj}^U \cdot \frac{h}{2}. \quad (3.14)$$

$$[W_{kj}]_h^U = w_{kj}^L \cdot \frac{h}{2} + w_{kj}^U \cdot \left(1 - \frac{h}{2}\right). \quad (3.15)$$

그래서, 식(3.12)와 (3.13)은 다음과 같이 수정 될 수 있다.

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^L} &= \frac{\partial e_{ph}}{\partial [W_{kj}]_h^L} \cdot \left(1 - \frac{h}{2}\right) \\ &+ \frac{\partial e_{ph}}{\partial [W_{kj}]_h^U} \cdot \frac{h}{2}, \end{aligned} \quad (3.16)$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^U} &= \frac{\partial e_{ph}}{\partial [W_{kj}]_h^L} \cdot \frac{h}{2} \\ &+ \frac{\partial e_{ph}}{\partial [W_{kj}]_h^U} \cdot \left(1 - \frac{h}{2}\right). \end{aligned} \quad (3.17)$$

이런 관계는 h-단계 집합에 대한 에러신호는

$\frac{\partial e_{ph}}{\partial [W_{kj}]_h^L}$ 과 $\frac{\partial e_{ph}}{\partial [W_{kj}]_h^U}$ 이 퍼지 가중치 W_{kj} 을 w_{kj}^L 와 w_{kj}^U 로 바꾸기 위한 처음 단계를 전파시키기 위한 방법을 나타낸다. 퍼지 가중치 $W_{kj} = (w_{kj}^L, w_{kj}^U, w_{kj}^C)$ 는 아래 규칙에 의해 변경된다.

$$w_{kj}^L(t+1) = w_{kj}^L(t) + \Delta w_{kj}^L(t), \quad (3.18)$$

$$w_{kj}^U(t+1) = w_{kj}^U(t) + \Delta w_{kj}^U(t), \quad (3.19)$$

$$w_{kj}^C(t+1) = \frac{w_{kj}^L(t+1) + w_{kj}^U(t+1)}{2}. \quad (3.20)$$

식(3.18)-(3.20)에 의한 W_{kj} 의 조정후에, 하한은 상한보다 더 커진다. 이런 경우, 아래의 간단한 경험적방법(heuristics)을 이용한다.

$$w_{kj}^L(t+1) := \text{Min}\{w_{kj}^L(t+1), w_{kj}^U(t+1)\}, \quad (3.21)$$

$$w_{kj}^U(t+1) := \text{Max}\{w_{kj}^L(t+1), w_{kj}^U(t+1)\}. \quad (3.22)$$

퍼지 가중치 W_p 와 퍼지 바이어스 θ_k, θ_j 는 퍼지 가중치 W_k 와 같은 방법으로 바뀌게 된다(부록 참조). 퍼지 벡터의 m 개 입출력 $(X_m, T_p), p=1, 2, \dots, m$,이 훈련 데이터로 주어진다. h (즉, h_1, h_2, \dots, h_n)의 n 값은 퍼지 신경망의 학습에 사용된다. 이 경우, 퍼지 신경망의 학습 알고리즘은 아래와 같이 나타낼 수 있다.

학습 알고리즘(Learning algorithm)

단계 0: 퍼지 가중치와 퍼지 바이어스를 초기화한다.

단계 1: m 개의 입출력 퍼지벡터의 초기화와 입출력 벡터에 대한 소속함수 값을 결정한다.

단계 2: $h = h_1, h_2, \dots, h_n$ 에 대해 단계 3를 반복한다.

단계 3: $p = 1, 2, \dots, m$ 에 대해 아래의 프로시저를 반복한다.

1) Forward calculation: 퍼지 입력벡터 X_p 에 따른 퍼지 출력 벡터 O_p 의 h -단계 집합을 계산.

2) Back-propagation: 퍼지 출력 벡터 O_p 와 퍼지 타겟 벡터 T_p 의 h 단계 집합에 대한 코스트 함수 e_m 에 사용된 퍼지 가중치와 바이어스를 조정해서 층과 층사이의 가중치와 바이어스의 변화량을 계산.

단계 4: 단계3에 의해 학습상태가 만족되지 않으면 단계2로 분기하고, 만족되면 학습을 마치게 된다.

본 논문의 컴퓨터 모의실험에서는, 학습이 완료될 때 까지 학습알고리즘을 10000회 반복시키는

반복횟수를 이용했다.

4. 컴퓨터 모의실험

이 장에서는, 다양한 산술 예를 통한 컴퓨터 모의실험에 의해 제안된 학습 알고리즘을 입증하였다. 이 장의 모든 컴퓨터 모의실험에서, 학습 알고리즘을 다음 상술된 바와 같이 이용했다.

- (1) 숨겨진 장치의 수 : 6개
- (2) h 의 값 : $h = 0.2, 0.4, 0.6, 0.8, 1.0$.
- (3) 정지 상태 : 학습 알고리즘의 10000 반복
- (4) 학습 상수 : $\eta = 0.5$
- (5) 운동량 상수 : $\alpha = 0.9$
- (6) 퍼지 가중치와 바이어스의 초기값 : 폐구간의 임의의 실제 값 $[-1, 1]$.

4.1. 모의실험 1

실제값이 퍼지값의 특별한 경우로서 비취질수 있기에, 제안되었던 퍼지 신경망은 퍼지 입력과 같이 실제 입력을 다룰 수 있다. 이 실험에서, 실제값을 퍼지값으로 만든 비선형 퍼지 함수에 근접시키기 위해 제안된 방법을 사용한다.

이 실험에서의 연습용 자료는 실제 입력 x_p 값과 퍼지 목표 T_p 값의 쌍으로 구성된다. 각 쌍은 그림 8에서 처럼 입출력 공간내에 표현된다. 그림 8의 삼각형은 삼각의 퍼지 목표 T_p 의 소속함수로 나타난다. 그림 9의 실제 입력과 퍼지 출력의 6개 쌍들은 훈련용 데이터로 주어진다. 이러한 훈련용 데이터를 이용해서, 제안된 학습 알고리즘을 통해 하나의 입력장치, 6개의 숨겨진 장치 그리고 하나의 출력 장치를 갖는 퍼지 신경망을 훈련했다. 실제

입력 x_p 는 아래의 소속함수를 갖는 퍼지 값으로서 다루어진다.

$$\mu_{x_p}(x) = \begin{cases} 1 & \text{if } x = x_p, \\ 0 & \text{if } x \neq x_p. \end{cases} \quad (4.1)$$

훈련된 퍼지 신경망의 퍼지 출력은 실제 입력: $x=0.0, 0.1, \dots, 1.0$ 에 대한 결과는 그림 10에서 나타난다. 그림 9와 그림 10을 비교해 보면, 적합한 퍼지 훈련용 데이터로 새로운 퍼지 입력에 대한 좋은 일반화를 찾을수 있다는 것을 알 수 있다.

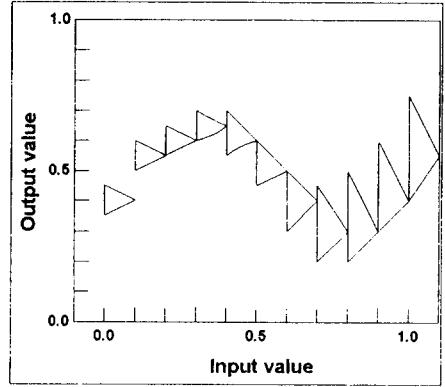


그림 10. 실험 1에서 훈련된 퍼지 신경망으로부터 퍼지 출력

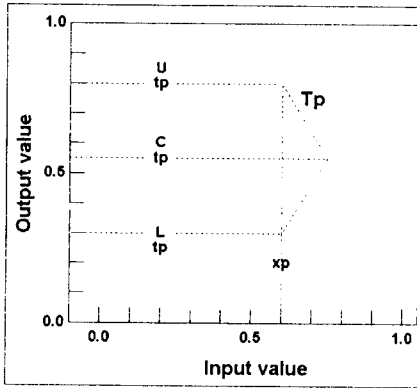


그림 8. 입출력 쌍 (x_p, T_p)의 표현

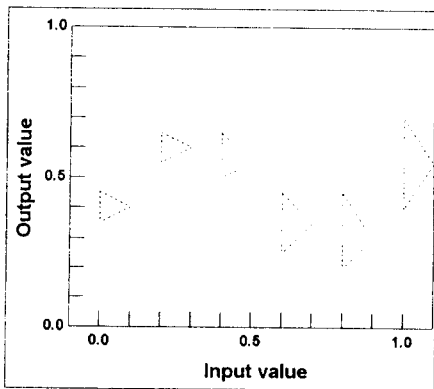


그림 9. 모의실험 2에서의 퍼지 훈련용 데이터

4.2. 모의실험 2

이 실험에서 퍼지 신경망을 통해 퍼지 if-then 규칙에 가까운 제안된 방법을 시도한다. 아래는 세 가지 퍼지 if-then 규칙이다.

If x is small then y is small.

If x is medium then y is medium.

If x is large then y is large.

“small”, “medium”, “large”과 같은 언어상의 전체 함수가 그림 9에 주어져 있다. 위의 퍼지 if-then 규칙으로부터, 아래의 훈련용 데이터를 얻을 수 있다.

$$\{(X_p, T_p)\} = \{(small, small), (medium, medium), (large, large)\} \quad (4.2)$$

여기서 비대칭 퍼지 목표가 연습용 데이터에 포함 된다는 것을 주시해야 한다. 그림 12에서 연습용 데이터를 이용해, 하나의 입력장치, 6개의 숨겨진 장치 그리고 하나의 출력 장치를 갖는 퍼지 신경망을 훈련했다. 훈련된 퍼지 신경망으로부터의 퍼지 출력은 훈련용 데이터에 대한 그림 13에 나타나

있다. 그림 12와 13을 비교해 보면, 적합한 훈련용 데이터를 관찰할 수 있다. 새로운 퍼지 입력 “medium small”과 “medium large”에 대한 퍼지 출력은 그림 14에 나타나 있다.(그림 11참조) 그림 12와 14를 비교해 보면, 새로운 퍼지 입력의 좋은 일반화를 발견할 수 있다. 새로운 퍼지 입력에 상응하는 퍼지 출력의 전체 함수는 그림 15에 나타난다. 그림 15의 두가지 퍼지 출력은 “medium small”과 “medium large”와 같이 언어적으로 해석될 수 있다.(그림 15과 11 비교) 따라서, 아래 2가지의 퍼지 if-then 규칙을 관찰할 수 있다.

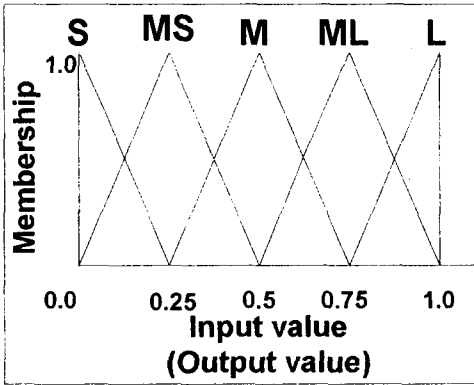


그림 11. 다섯가지의 언어적 값의 전체함수 (S:small; MS:medium small; M:medium; ML:medium large; L:large)

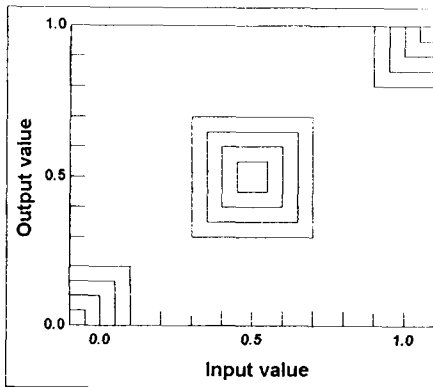


그림 12. 실험 2의 훈련용 퍼지 데이터

If x is medium small then y is medium small.
If x is medium large then y is medium large.

이 두가지 퍼지 if-then 규칙이 직관적 보간법인 세가지 퍼지 if-then 규칙과 일치함을 알수가 있다. 따라서 이 실험에서 훈련된 퍼지 신경망이 적합한 퍼지 if-then 규칙을 발견했음을 결론지을 수 있는 것이다.

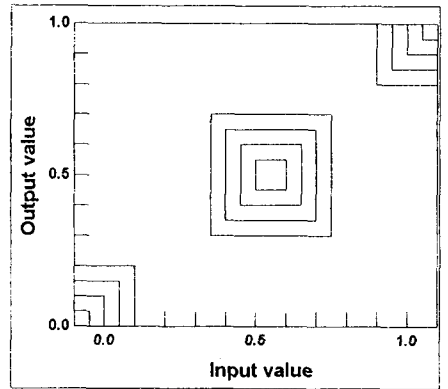


그림 13. 실험 2의 훈련용 데이터에 대한 훈련된 퍼지 신경망의 퍼지 출력

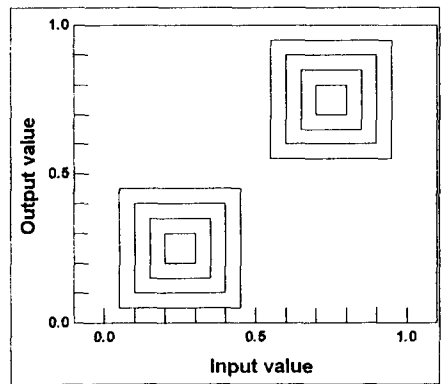


그림 14. 실험 2의 새로운 퍼지 입력에 대한 훈련된 퍼지 신경망의 퍼지 출력

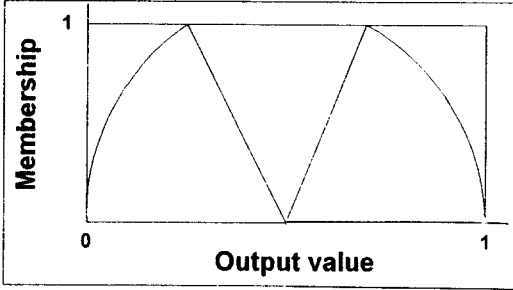


그림 15. 실험 2의 새로운 퍼지 입력에 따른 퍼지 출력의 소속 함수

5. 결 론

본 논문에서는, 확장 원칙에 의해 정의된 삼계층 퍼드포워드 퍼지 신경망의 입출력 관계의 대칭적 삼각 퍼지 가중치 학습 알고리즘을 얻어냈다. 얻어진 학습 알고리즘의 효과는 다양한 산술적 예제에 의한 컴퓨터 모의실험으로 설명되어졌다. 본 논문은 퍼지 입력, 퍼지 목표 그리고 퍼지 가중치를 갖는 퍼지 신경망의 학습 알고리즘을 도출하기 위한 시도중 하나이다. 대칭적 삼각 퍼지 가중치에 대한 유추된 학습 알고리즘이 이 논문의 산술적 예제에서 잘 수행되었고, 앞으로는 더욱 일반적인 퍼지 가중치의 확장을 이용한 학습알고리즘이 연구되어야한다.

부록 : 파생된 계산

식(3.10)과 (3.11)에서 파생된 $\frac{\partial e_{ph}}{\partial W_{kj}^L}$ 과 $\frac{\partial e_{ph}}{\partial W_{kj}^U}$ 은 (2.20)-(2.26)에서의 퍼지 신경망의 입출력 관

계로부터 계산될 수 있다. 아래와 같다.

(1) If $0 \leq [X_{kj}]_h^L \leq [X_{kj}]_h^U$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^L} &= \frac{\partial e_{ph}}{\partial [w_{kj}]_h^L} \cdot \frac{\partial [w_{kj}]_h^L}{\partial w_{kj}^L} \\ &+ \frac{\partial e_{ph}}{\partial [w_{kj}]_h^U} \cdot \frac{\partial [w_{kj}]_h^U}{\partial w_{kj}^L} \\ &= \frac{\partial e_{ph}}{\partial [w_{kj}]_h^L} \cdot \left(1 - \frac{h}{2}\right) \\ &+ \frac{\partial e_{ph}}{\partial [w_{kj}]_h^U} \cdot \frac{h}{2} \\ &= \frac{\partial e_{ph}}{\partial [Net_{pk}]_h^L} \cdot \frac{\partial [Net_{pk}]_h^L}{\partial [W_{kj}]_h^L} \cdot \left(1 - \frac{h}{2}\right) \\ &+ \frac{\partial e_{ph}}{\partial [Net_{pk}]_h^U} \cdot \frac{\partial [Net_{pk}]_h^U}{\partial [W_{kj}]_h^U} \cdot \frac{h}{2} \\ &= -\delta_{pkh}^L \cdot [O_{pj}]_h^L \cdot \left(1 - \frac{h}{2}\right) \\ &- \delta_{pkh}^U \cdot [O_{pj}]_h^U \cdot \frac{h}{2}, \end{aligned} \tag{A.1}$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^U} &= \frac{\partial e_{ph}}{\partial [w_{kj}]_h^L} \cdot \frac{\partial [w_{kj}]_h^L}{\partial w_{kj}^U} \\ &+ \frac{\partial e_{ph}}{\partial [w_{kj}]_h^U} \cdot \frac{\partial [w_{kj}]_h^U}{\partial w_{kj}^U} \\ &= \frac{\partial e_{ph}}{\partial [w_{kj}]_h^L} \cdot \frac{h}{2} \\ &+ \frac{\partial e_{ph}}{\partial [w_{kj}]_h^U} \cdot \left(\frac{1-h}{2}\right) \\ &= -\delta_{pkh}^L \cdot [O_{pj}]_h^L \cdot \frac{h}{2} \\ &- \delta_{pkh}^U \cdot [O_{pj}]_h^U \cdot \left(1 - \frac{h}{2}\right), \end{aligned} \tag{A.2}$$

여기에서,

$$\begin{aligned} \delta_{pkh}^L &= \frac{\partial e_{ph}}{\partial [Net_{pk}]_h^L} \\ &= h \cdot ([T_{pk}]_h^L - [O_{pk}]_h^L) \\ &\cdot [O_{pk}]_h^L \cdot (1 - [O_{pk}]_h^L), \end{aligned} \tag{A.3}$$

$$\begin{aligned} \delta_{pkh}^U &= \frac{\partial e_{ph}}{\partial [Net_{pk}]_h^U} \\ &= h \cdot ([T_{pk}]_h^U - [O_{pk}]_h^U) \\ &\cdot [O_{pk}]_h^U \cdot (1 - [O_{pk}]_h^U). \end{aligned} \tag{A.4}$$

(2) If $[X_{kj}]_h^L \leq [X_{kj}]_h^U < 0$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^L} &= -\delta_{pkh}^L \cdot [O_{pj}]_h^U \cdot \left(1 - \frac{h}{2}\right) \\ &\quad - \delta_{pkh}^U \cdot [O_{pj}]_h^L \cdot \frac{h}{2}, \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^U} &= -\delta_{pkh}^L \cdot [O_{pj}]_h^U \cdot \frac{h}{2} \\ &\quad - \delta_{pkh}^U \cdot [O_{pj}]_h^L \cdot \left(1 - \frac{h}{2}\right). \end{aligned} \quad (\text{A.6})$$

(3) If $[X_{kj}]_h^L < 0 \leq [X_{kj}]_h^U$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^L} &= -\delta_{pkh}^L \cdot [O_{pj}]_h^U \cdot \left(1 - \frac{h}{2}\right) \\ &\quad - \delta_{pkh}^U \cdot [O_{pj}]_h^L \cdot \frac{h}{2}, \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{kj}^U} &= -\delta_{pkh}^L \cdot [O_{pj}]_h^U \cdot \frac{h}{2} \\ &\quad - \delta_{pkh}^U \cdot [O_{pj}]_h^L \cdot \left(1 - \frac{h}{2}\right). \end{aligned} \quad (\text{A.8})$$

퍼지 바이어스 Θ_k 가 퍼지 가중치 $W_k(O_{pi}=1)$ 과 같이 비취질때, Θ_k 의 학습을 통해 파생된

$\frac{\partial e_{ph}}{\partial \theta_k^L}$ 과 $\frac{\partial e_{ph}}{\partial \theta_k^U}$ 은 $\frac{\partial e_{ph}}{\partial w_{kj}^L}$ 과 $\frac{\partial e_{ph}}{\partial w_{kj}^U}$ 과 같은 방법으로 얻어질 수 있다.

i 번째 입력장치와 j 번째 숨겨진 장치 사이의 퍼지 가중치 W_{ji} 의 학습을 통해 파생된 $\frac{\partial e_{ph}}{\partial w_{ji}^L}$ 과

$\frac{\partial e_{ph}}{\partial w_{ji}^U}$ 은 아래와 같이 계산될 수 있다.

(1) If $0 \leq [X_{ji}]_h^L \leq [X_{ji}]_h^U$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^L} &= -\delta_{ph}^L \cdot [O_{pi}]_h^L \cdot \left(1 - \frac{h}{2}\right) \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^U \cdot \frac{h}{2}, \end{aligned} \quad (\text{A.9})$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^U} &= -\delta_{ph}^L \cdot [O_{pi}]_h^L \cdot \frac{h}{2} \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^U \cdot \left(1 - \frac{h}{2}\right), \end{aligned} \quad (\text{A.10})$$

여기에서

$$\begin{aligned} \delta_{ph}^L &= [O_{pj}]_h^L \cdot (1 - [O_{pj}]_h^U) \\ &\quad \cdot \left[\sum_{k=1}^{n_0} \delta_{pkh}^L [W_{kj}]_h^L + \sum_{k=1}^{n_0} \delta_{pkh}^U [W_{kj}]_h^U \right]. \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} \delta_{ph}^U &= [O_{pj}]_h^U \cdot (1 - [O_{pj}]_h^L) \\ &\quad \cdot \left[\sum_{k=1}^{n_0} \delta_{pkh}^L [W_{kj}]_h^L + \sum_{k=1}^{n_0} \delta_{pkh}^U [W_{kj}]_h^U \right]. \end{aligned} \quad (\text{A.12})$$

(2) If $[X_{ji}]_h^L \leq [X_{ji}]_h^U < 0$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^L} &= -\delta_{ph}^L \cdot [O_{pi}]_h^U \cdot \left(1 - \frac{h}{2}\right) \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^L \cdot \frac{h}{2}, \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^U} &= -\delta_{ph}^L \cdot [O_{pi}]_h^U \cdot \frac{h}{2} \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^L \cdot \left(1 - \frac{h}{2}\right), \end{aligned} \quad (\text{A.14})$$

(3) If $[X_{ji}]_h^L < 0 \leq [X_{ji}]_h^U$ then

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^L} &= -\delta_{ph}^L \cdot [O_{pi}]_h^U \cdot \left(1 - \frac{h}{2}\right) \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^L \cdot \frac{h}{2}, \end{aligned} \quad (\text{A.15})$$

$$\begin{aligned} \frac{\partial e_{ph}}{\partial w_{ji}^U} &= -\delta_{ph}^L \cdot [O_{pi}]_h^U \cdot \frac{h}{2} \\ &\quad - \delta_{ph}^U \cdot [O_{pi}]_h^L \cdot \left(1 - \frac{h}{2}\right), \end{aligned} \quad (\text{A.16})$$

퍼지 바이어스 Θ_j 가 퍼지 가중치 $W_k(O_{pi}=1)$ 과 같이 비취질때, Θ_j 의 학습을 통해 파생된 $\frac{\partial e_{ph}}{\partial \theta_j^L}$ 과

$\frac{\partial e_{ph}}{\partial \theta_j^U}$ 은 같은 방법으로 얻어질 수 있다.

이 부록에서 유도된 것으로 부터, 본논문의 퍼지 신경망 학습 알고리즘이 표준적인 비퍼지 신경망에

대한 역전파(back-propagation) 알고리즘 [8]과 같은 방법으로 유도되었음을 알수 있다.

참 고 문 헌

1. G. Alefeld and J. Herzberger, Introduction to Interval Computations (Academic Press, New York, 1983).
2. J.J. Buckley and Y. Hayashi, Fuzzy neural networks: a survey, Fuzzy Sets and Systems 66 (1991) 1-13.
3. Y. Hayashi, J.J. Buckley and E. Czogala, Fuzzy neural network with fuzzy signals and weights, Internat. J. Intelligent Systems 8 (1993) 527-537.
4. H. Ishibuchi, R. Fujioka and H. Tanaka, An architecture of neural network for input vectors of fuzzy numbers., Proc FUZZ-IEEE '92 (San Diego, 8-12 March 1992) 643-650.
5. H. Ishibuchi, R. Fujioka and H. Tanaka, Neural networks that learn from fuzzy if-then rules, IEEE Trans. Fuzzy Systems, 1 (2) (1993) 85-97.
6. H. Ishibu, H. Okada and H. Tanaka, Fuzzy neural networks with fuzzy weights and fuzzy biases, Proc. ICNN '93 (San Francisco. 28 March-1 April 1993) 1650-1655.
7. H. Ishibuchi, H. Okada and H. Tanaka, Interpolation of fuzzy if-then rules by neural networks, Internat. J. Approx. Reason. 10(1) (1994) 3-27.
8. D.E. Rumelhart, J.L. McClelland and the PDP Research Group, Parallel Distributed Processing, Vol. 1(MIT Press, Cambridge, MA, 1986).
9. L.A. Zadeh, The concept of a linguistic variable and its application to approximate reason.ng: Parts 1-3, Inform. Sci. 8(1975) 199-249, 301-357; 9(1975) 43-80.